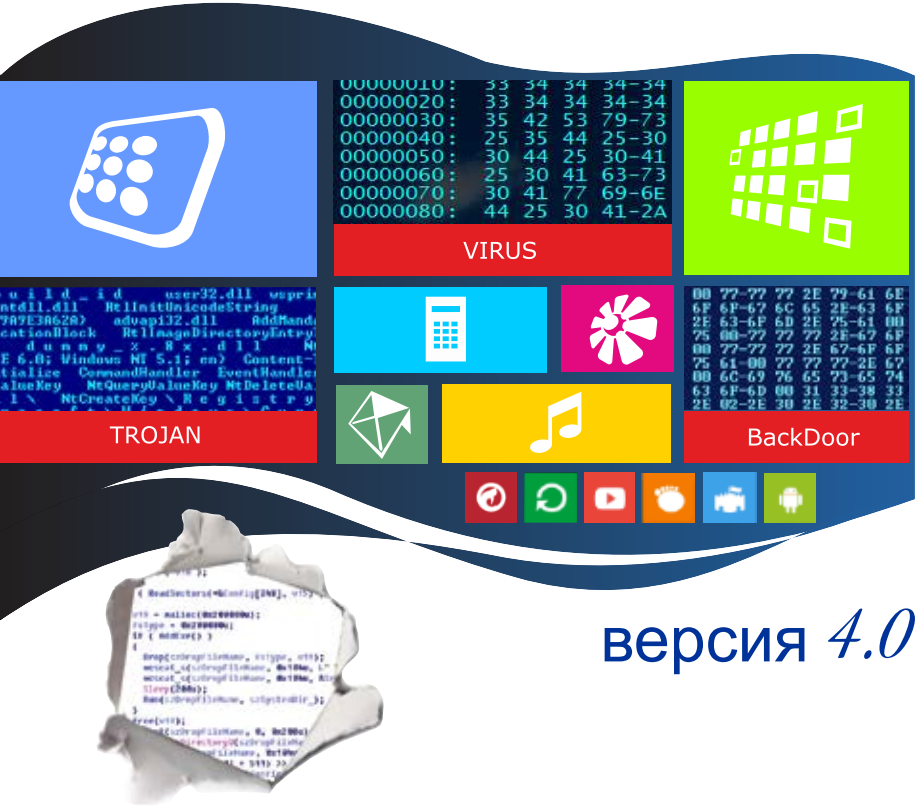
Валентин Холмогоров

PRO

Вирусы



версия *4.0*

ПРОСТО

СЕРИЯ
ПРОСТО

Валентин Холмогоров

PRO Вирусы

Версия 4.0

Автор идеи и научный редактор серии
С. Л. Деменок

НАУЧНО-ПОПУЛЯРНОЕ
ИЗДАТЕЛЬСТВО
«СТРАТА»
Санкт-Петербург, 2020

УДК 681.3.06(075)
ББК 32.973-01я2
Х72

Холмогоров В.
Х72 **PRO ВИРУСЫ.** Издание четвертое, переработанное и дополненное / Валентин Холмогоров. — СПб.: Страта, 2020. — 224 с., ил.

ISBN 978-5-907314-12-2

Время энтузиастов-одиночек, создававших компьютерные вирусы на заре информационной эпохи, давно прошло: в наши дни разработкой и распространением вредоносных программ занимаются хорошо организованные преступные группировки, имеющие жесткую иерархию и напоминающие по своей структуре настоящие мафиозные кланы. Объем этого подпольного рынка составляет сотни миллионов долларов.

Книга рассказывает об истории возникновения и развития технологий компьютерных вирусов, их разновидностях, внутренней архитектуре, способах распространения и принципах действия. Книга позволит читателям познакомиться с таинственным теневым миром киберпреступности, представители которого ежедневно осуществляют атаки на компьютеры простых пользователей по всему миру.

УДК 681.3.06(075)
ББК 32.973-01я2

ISBN 978-5-907314-12-2

Моей жене Галине
и детям — Анастасии и Даниилу

ПРЕДИСЛОВИЕ

Историю развития человеческой цивилизации с определенной степенью достоверности можно назвать историей борьбы за ресурсы. На протяжении многих эпох люди соперничали за пищу, золото, территории, нефть. В начале XXI века основным ресурсом для человечества стала информация.

Информация пронизывает всё современное общество, проникает во все без исключения сферы нашей жизни. Информационные потоки управляют движением самолетов и поездов, обеспечивают телефонную и спутниковую связь, являются движущей силой биржевой торговли и банковской сферы. Без непрерывных процессов передачи и обработки информации не загорится электрическая лампочка в квартире, не смогут пробить товар кассовые аппараты в супермаркете, замрут бензозаправочные станции, погаснут светофоры на улицах. Информация сегодня де-факто управляет миром. Вот почему сфера информационной безопасности является сейчас одной из наиболее актуальных и важных областей IT-индустрии. Она буквально балансирует на острие прогресса, скользит на гребне волны, всегда оставаясь на её вершине — ведь технологии в наши дни развиваются стремительно. А одним из наиболее значимых (и интересных) подразделов информационной безопасности является защита устройств от компьютерных угроз.

Еще двадцать лет назад компьютерные вирусы и троянские программы заявили о себе, как реальная и очень серьезная опасность, способная принести многомиллионные убытки как отдельным коммерческим компаниям, так и экономике государств в целом. По земному шару прокатилось несколько глобальных компьютерных эпидемий, а пользователи Интернета стонали под грудами рекламного спама, непрерывно сыпавшегося в их электронные почтовые ящики. Чуть позже

киберпреступники научились извлекать прибыль, шантажируя непосредственно самих владельцев персональных компьютеров: на свет появились троянцы-блокировщики, нарушавшие нормальную работу операционной системы, энкодеры, шифровавшие данные на дисках и требовавшие выкуп за их расшифровку, и, наконец, банковские троянцы, кравшие деньги непосредственно с электронных счетов ничего не подозревающей жертвы. И если потерю десятка личных фотографий из отпуска еще можно как-то пережить, то утрата бухгалтерской базы данных, реестра клиентов и контрагентов, договоров и прочей важной документации может стать для коммерческого предприятия настоящей катастрофой.

Десять лет назад были обнаружены первые троянцы для смартфонов, работавших под управлением операционной системы Google Android. Те, самые ранние экземпляры, еще не несли в себе значительной угрозы пользователям — они просто предлагали ему отправить платную СМС при установке новых приложений, и потому поначалу никто не воспринял их всерьез. Посетители многочисленных форумов в Интернете ехидно комментировали выпуск антивирусных программ для Android желанием разработчиков нагреть руки на несуществующей и якобы преувеличенной опасности.

Сегодня число известных вредоносных программ для ОС Windows исчисляется миллионами, для ОС Android — сотнями тысяч. Появились и активно распространяются вредоносные программы для так называемых «умных устройств», составляющих экосистему «интернета вещей» — телевизионных приставок, роутеров, сетевых хранилищ и дисковых накопителей, систем «умный дом», медиацентров.

В силу моей профессии ко мне часто обращаются знакомые с просьбами проконсультировать их по вопросам защиты информации и борьбы с вирусами. И я всякий раз сталкиваюсь с тем, что многие из них (даже те, кто является неплохим специалистом в других компьютерных областях) не слишком хорошо разбираются в данном предмете, не знакомы с некоторыми важными фактами, безоговорочно верят в домыслы и стереотипы, путаются в терминологии. Эта книга — попытка объединить под одной обложкой мой двадцатипятилетний опыт работы в сфере IT-технологий

и девятилетний — в области информационной безопасности и защиты информации. Изложенный здесь материал не претендует на энциклопедичность и техническую глубину, однако позволит получить базовые сведения о существующих на сегодняшний день угрозах, познакомит читателя с основными связанными с ними понятиями, расскажет об истории развития антивирусной индустрии, познакомит с наиболее опасными разновидностями вредоносных программ, принципами их деструктивной деятельности, путями распространения и методиками борьбы с ними. Иными словами, эта книга — начальное пособие для всех, кто интересуется теорией и практикой информационной безопасности и антивирусной защиты.

Предполагается, что читатели настоящего издания уже владеют базовыми знаниями о принципах работы современных персональных компьютеров и операционных систем, а также основными терминами, применяемыми в сфере IT. Если в тексте книги вам встретится незнакомое понятие, его значение можно уточнить в кратком глоссарии, который я привел в конце книги. Ну, а если у вас возникнут какие-либо вопросы, не освещенные на страницах этого издания, я буду рад видеть вас на моем персональном веб-сайте: http://holmogorov.ru или в Facebook: https://www.facebook.com/valentin.holmogorov.

Предисловие к четвертому изданию

С момента выхода предыдущего издания этой книги многое изменилось, впрочем, в мире информационных технологий что-то непрерывно меняется. Появились новые вредоносные программы, новые методы борьбы с ними.

В четвертом издании книги «PRO Вирусы» добавились разделы, рассказывающие о принципах работы мобильных банковских троянцев для Android и вредоносных программ для iOS. Отдельная глава посвящена троянцам, заражающим «умные» устройства, относящиеся к категории интернета вещей. Сегодня это — одна из самых актуальных и насущных угроз в мире информационной безопасности. На примере известного шифровальщина Troldesh показано, как работают современные энкодеры.

Об авторе

Валентин Холмогоров — в прошлом штатный сотрудник одной из ведущих российских антивирусных компаний, где проработал более 7 лет. Автор 38 книг, посвященных компьютерным технологиям, а также более 400 публикаций в различных периодических околокомпьютерных изданиях. Работал заместителем главного редактора журнала «Магия ПК», в течение пяти лет возглавлял IT-департамент в компании, занимавшейся организацией и комплексным техническим сопровождением протокольных мероприятий как российского, так и международного значения. Известен в Интернете как блогер и публицист. В настоящее время работает редактором легендарного журнала «Хакер» (http://xakep.ru), посвященного высоким технологиям и информационной безопасности.

ГЛАВА 1. ЗАКОУЛКИ ИСТОРИИ

Любое наблюдаемое в современном мире явление имеет свою предысторию, в той или иной степени обусловливающую его возникновение. И если сам момент появления первых вредоносных компьютерных программ установлен с более или менее высокой степенью достоверности, то по поводу идеи, подтолкнувшей вирусописателей к мысли о создании такого рода опасных приложений, до сих пор ведутся ожесточенные споры.

ПЕРВЫЕ ЛАСТОЧКИ

Общепринятое мнение гласит, что теоретические основы, послужившие фундаментом для разработки *самореплицирующихся* (то есть автоматически воспроизводящихся) компьютерных программ заложил еще в начале 50-х годов XX века американский математик венгерского происхождения Джон фон Нейман. В 1951 году фон Нейман на основе собственного цикла лекций создал научный труд под названием «Теория самовоспроизводящихся автоматов» (книга была опубликована уже после смерти автора, в 1966 году, издательством университета Иллинойса), в котором описал принципиальную возможность разработки так называемых клеточных автоматических устройств, способных к самовоспроизведению подобно клеткам живого организма. Именно этот принцип распространения и по сей день используют современные компьютерные вирусы и черви.

По поводу этимологии самого названия «вирус» применительно к самореплицирующимся компьютерным программам также ведутся ожесточенные споры. Считается, что впервые этот термин именно в таком контексте употребил американский астрофизик и писатель-фантаст Грегори Бенфорд в своем рассказе «Человек в шрамах», опубликованном в журнале Venture в мае 1970 года. А уже в 1975 году американский писатель-фантаст Джон Браннер выпустил роман «Оседлавший взрывную волну», в основу которого лег сюжет о появлении самораспространяющейся вредоносной программы. Книга рассказывала о компьютеризированном обществе, которым управляло с помощью глобальной электронной сети правительство диктаторов и тиранов. Программист, решивший спасти мир от диктатуры, написал программу, которую автор романа назвал «червем»; эта программа копировала себя с одного компьютера на другой, разрушая хранившуюся в них информацию. Чтобы остановить

«червя», правительство вынуждено было отключить сеть, лишившись таким образом власти. Роман быстро стал бестселлером, поистине культовой книгой в только зарождавшейся тогда среде компьютерных хакеров.

Однако одновременно с развитием теории программисты-энтузиасты проводили и первые практические опыты. Так, в 1971 году работавший в вычислительной лаборатории компании «Bolt, Beranek and Newman» американский программист Боб Томас занимался исследованием возможностей созданной им же самим подсистемы RSEXEC, позволявшей осуществлять удаленный запуск программ в операционной системе Tenex. Экспериментируя с системами передачи данных между различными вычислительными машинами, Томас написал программу, которую назвал «Ползуном» (The Creeper). «Ползун» самостоятельно копировал себя с одного компьютера на другой, перемещаясь таким образом по сети, и выводил на экран каждого терминала забавное сообщение: «*Я — Ползун… Если сможешь, поймай меня!*» (*I'm the Creeper… Catch me if you can!*). Эта небольшая программа не размножалась, а просто «ползала» с одного сетевого узла на другой: когда на удаленном компьютере запускалась новая копия Creeper, исходный экземпляр уничтожался. Фактически этот случай можно назвать первым в истории документально подтвержденным фактом успешной разработки автономно распространяющейся компьютерной программы, которую, впрочем, все же нельзя назвать полноценным «компьютерным вирусом», поскольку она не несла в себе никакого вредоносного функционала. К слову, история гласит, что, когда другому специалисту Bolt, Beranek and Newman, Рэю Томлинсону, надоело бороться с бесконечно отвлекающими его от работы «Ползунами», он написал другую программу, получившую наименование Reaper («Жнец»). Reaper в точности так же самостоятельно перемещался по сети, но с совершенно иной целью: программа вылавливала и безжалостно уничтожала всех «Ползунов», которые попадались ей на пути. Эта незамысловатая «игра в догонялки» продолжалась какое-то время, пока программисты лаборатории окончательно не утратили к ней интерес.

В 1974 году появилась первая в истории программа, которую по ряду формальных признаков можно назвать вредоносной, однако имен ее создателей история, увы, не сохранила. Нехитрое приложение с названием The Rabbit («Кролик»)

полностью соответствовало своему наименованию: ее предназначение вполне можно описать известной библейской формулой «плодитесь и размножайтесь». Программа автоматически создавала множество своих копий и увлеченно занималась этим до тех пор, пока не забивала всю доступную свободную память компьютера, что неизбежно вызывало его отказ. Именно это приложение стало своего рода основоположником целого семейства вредоносных программ, объединенных общей категорией: «логические бомбы».

А еще через год случился инцидент, вошедший в историю как первый случай самопроизвольного распространения программы, вышедшей из-под контроля своего создателя. Речь идет об игре The Animal, созданной программистом Джоном Уокером для компьютера UNIVAC 1108. Суть игры состояла в следующем: пользователь загадывал некое животное, а программа задавала ему наводящие вопросы, на которые он должен был отвечать «да» или «нет». Компьютер таким образом пытался угадать, что задумал человек. Однако код игры содержал досадную ошибку: при попытке пользователя добавить в базу приложения дополнительные вопросы, новая версия игры записывалась поверх старой, и кроме того с использованием специальной утилиты игра автоматически создавала свою копию в каждой директории, к которой пользователь имел доступ. Поскольку компьютеры UNIVAC были многопользовательскими, программа быстро проникала на другие компьютеры, использующие общие магнитные ленты в качестве носителя информации. Остановить бесконтрольное распространение игры The Animal (быстро получившей народное название Pervading Animal, «Всепроникающее животное») смогло только обновление операционной системы, в котором был изменен формат состояния файловых таблиц, используемый приложением для самокопирования.

Следующий аналогичный случай не заставил себя долго ждать. В 1980 году двое сотрудников компании Xerox, которая в те времена выпускала очень популярные персональные компьютеры Alto, имеющие возможность объединения в локальные сети, решили создать программу, которую, по аналогии с упоминавшимся в романе Браннера детищем программиста-бунтаря, назвали «Червем». Собственно, «Червь» Джона Хаппа и Джона Шока должен был нести положительную миссию:

по замыслу разработчиков, перемещаясь между подключенными к сети компьютерами, «Червь» был призван проверять операционную систему на наличие уязвимостей и по возможности устранять их, загружая с удаленного сервера соответствующие обновления. Однако на практике все получилось совсем не так, как задумали разработчики. Запустив вечером экспериментальную версию «Червя», Хапп и Шок отправились домой. Когда утром программисты вернулись на работу, они увидели, что все компьютеры, установленные в многоэтажном здании исследовательского центра Xerox, расположенного в калифорнийском городке Пало-Альто, благополучно зависли. В исходном коде «Червя» была допущена незначительная ошибка, благодаря которой программа начала бесконтрольно распространяться между различными узлами сети и блокировать их работу. Перезагрузка машин не помогала: часть входящих в сеть компьютеров была установлена в закрытых комнатах, к которым Хапп и Шок не имели доступа, и как только на перезагружаемой машине запускалась операционная система, «Червь» тут же копировал себя в ее память с другого компьютера, после чего система мгновенно выходила из строя. Отключив одну из машин от локальной сети, программисты вынуждены были экстренно создать другую программу, которая уничтожила бы взбесившегося «Червя». На полную ликвидацию последствий их совместного творчества ушло несколько дней.

Как бы то ни было, все эти случаи можно считать всего лишь прелюдией, своего рода подготовительным этапом перед целой эпохой, ознаменовавшейся появлением и распространением настоящих вредоносных программ и компьютерных вирусов.

ЭПОХА ВИРУСОВ

Что бы ни говорили о безопасности и защищенности данной системной платформы многочисленные поклонники компьютерной техники производства небезызвестной «яблочной» компании из города Купертино, первым в истории компьютерным вирусом, обнаруженным в «живой природе» (то есть за пределами компьютерной системы или вычислительной

лаборатории, где он был написан), стала вредоносная программа Elk Cloner для персональных компьютеров Apple II. Произошло это в 1981 году.

Собственно, выбор создателя Elk Cloner, 15-летнего школьника Ричарда Скренты, пал на компьютер производства Apple не случайно — на заре 80-х именно эти относительно недорогие и весьма «продвинутые» персоналки пользовались чрезвычайно высокой популярностью в США, занимая значительную долю розничного рынка. Скрента относился к категории молодых людей, которых сейчас принято называть жаргонным термином «гики» — он был не просто пользователем Apple II, а «продвинутым компьютерным гением», любившим покопаться в архитектуре операционной системы и «внутренностях» прикладных программ. Одноклассники часто брали у Ричарда дискеты с играми, которых у него имелось множество, однако он быстро заработал себе репутацию проказника и шутника, поскольку постоянно модифицировал одалживаемые друзьям программы, чтобы они время от времени выводили на экран компьютера различные забавные, а порой и неприличные или оскорбительные фразы. В конечном итоге приятели и вовсе перестали просить программы у Скренты, предпочитая заимствовать игры из более надежных источников. Именно тогда Ричард задумался о том, как он смог бы модифицировать программы дистанционно, физически не прикасаясь к дискете. Итогом его размышлений и стал вирус Elk Cloner.

Вирус распространялся вместе с компьютерной игрой, при каждом 50-м запуске которой отображал на экране стишок следующего содержания:

It will get on all your disks	*Он влезет на все ваши диски,*
It will infiltrate your chips	*Он проникнет в ваши чипы,*
Yes, it's Cloner!	*Да, это Cloner!*
It will stick to you like glue	*Он прилипнет к вам, словно клей,*
It will modify RAM too	*Модифицирует оперативную память,*
Send in the Cloner!	*Представляем Cloner!*

Если компьютер загружал операционную систему Apple DOS 3.3 с инфицированной дискеты, Elk Cloner копировался

в оперативную память, после чего дожидался, пока пользователь вставит в дисковод «чистую» системную дискету, и заражал ее, обеспечивая этим собственное распространение. Поскольку компьютеры под управлением Apple DOS 3.3 были чрезвычайно популярны в Северной Америке, Elk Cloner быстро сделался самым настоящим стихийным бедствием — вирус распространился настолько широко, что компания Apple вынуждена была даже выпустить специальную утилиту для его уничтожения, которая к тому же предотвращала повторное заражение системных дискет. Ну, а сам Ричард Скрента навсегда вписал свое имя в историю развития компьютерных технологий и информационной безопасности. Закончив Северо-Западный университет (Чикаго, Иллинойс), Скрента поработал программистом в таких известных компаниях, как Sun Microsystems, Netscape, America On-Line (AOL), а позже создал собственную фирму в Кремниевой долине, занимающуюся разработкой оригинального поискового движка, известного сейчас под наименованием Blekko.

В том же 1981 году было зафиксировано распространение самореплицирующейся программы под названием Virus 1,2,3 — и тоже в операционной системе Apple DOS 3.3 для компьютеров Apple II. Ну, а уже в 1986 году разразилась первая эпидемия среди IBM-совместимых компьютеров, массово заражавшихся вирусом-буткитом Brain.

Авторами этого творения стали 19-летний пакистанский программист Басит Фаруд Алви и его родной брат Амджат. Сами братья утверждали, что вирус был написан ими для защиты от нелицензионного копирования разработанного ими же медицинского программного обеспечения и предназначался только для компьютерных пиратов. После своего запуска Brain подменял собственной копией загрузочную запись на дискетах, отформатированных в файловой системе FAT (File Allocation Table), используемой, в частности, операционной системой MS-DOS. Оригинальная загрузочная запись перемещалась в другой сектор диска, помечавшийся как «плохой» (bad), а в качестве метки тома устанавливалось значение «©Brain». За год своего существования вирус заразил множество компьютеров во всем мире, в первую очередь на территории США и Великобритании.

Следующий, 1987 год, стал поистине урожайным на появление новых вредоносных программ. Так, в течение последующих

двенадцати месяцев были выявлены вирусы, известные под именами Vienna, Stoned, Ping Pong, Cascade. Наиболее масштабное распространение получила вредоносная программа Jerusalem, обнаруженная в Иерусалиме в октябре 1987 года и инфицировавшая компьютеры под управлением MS-DOS по всему миру.

Jerusalem был *резидентным вирусом*, использовавшим около 2 Кбайт оперативной памяти компьютера и заражавшим все запускающиеся в системе исполняемые файлы за исключением основного компонента ядра MS-DOS — файла *command.com*. Вирус обладал несколькими вредоносными функциями: благодаря возможности перехватывать используемые DOS системные прерывания, он мог значительно замедлять работу зараженной машины, спонтанно отключать рабочие станции от сети и препятствовать нормальному выводу документов на печать. Однако главная его опасность заключалась в том, что каждую пятницу, выпадавшую на 13-е число (кроме 1987 года), вирус уничтожал исполняемые файлы всех без исключения запускающихся на инфицированном компьютере программ, что, естественно, нарушало нормальную работу персоналок.

Примерно с этого момента различные инциденты, связанные с массовым распространением компьютерных вирусов и других вредоносных приложений, уже перестали кого-либо удивлять, и многочисленные журналисты, специализирующиеся на освещении событий в области «высоких технологий», попросту не обращали на них чересчур пристального внимания. За исключением одного случая, наделавшего по-настоящему много шума.

Роберт Моррис, старший сын Боба Морриса, одного из ведущих экспертов отдела Агентства Национальной Безопасности США по расследованию компьютерных преступлений, рос тихим и скромным мальчиком. Его единственной страстью были компьютеры. Уже в четырнадцатилетнем возрасте он переписал популярную у подростков компьютерную игру The Four Corners, добавив в нее множество новых функциональных возможностей. В 16 лет он стал настоящим экспертом по системе безопасности UNIX, обнаружив в «классическом» берклиевском коде этой платформы множество ошибок, которые не замедлил исправить. Однако он и сам не брезговал пользоваться обнаруженными «дырами» в защите, время от времени

подключаясь к удаленным электронным сетям в поисках интересующей его информации. Это увлечение привело к тому, что вскоре в компьютерном журнале Smitsonian появился материал, в котором Роберта называли одним из наиболее известных молодых хакеров в Америке. Именно Роберт Моррис является автором и разработчиком одной из наиболее известных реализаций протокола передачи данных UUCP — Unix-To-Unix CoPy. Обучаясь на четвертом курсе Гарвардского университета, Роберт уже читал лекции в Национальном Агенстве Безопасности США и исследовательских лабораториях военно-морского флота по безопасности операционной системы UNIX.

Полученные Робертом в ходе его самостоятельных разработок и изучения уже существующего опыта других программистов знания требовали практического применения. В качестве эксперимента Роберт решил написать программу, которая, используя обнаруженные им недоработки в созданном для UNIX протоколе FTP и программе sendmail, могла бы самостоятельно распространяться между объединенными в сеть компьютерами, но при этом умела бы эффективно «прятаться» в операционной системе и самостоятельно размножаться. Иными словами, «Червь» Морриса должен был объединять в себе все достоинства предыдущих попыток создания аналогичных программ. Поскольку эта разработка была всего лишь научным экспериментом, тестом на безопасность объединенных в сеть компьютерных систем, Роберт заложил в код «Червя» алгоритмы, сдерживающие его распространение. Никаких модулей, разрушающих файловую систему атакованных компьютеров, также задумано не было.

2 ноября 1988 года в 18.30 Роберт Моррис подключился к компьютерам лаборатории искусственного интеллекта MIT и запустил свою программу на исполнение. Когда спустя полчаса он снова попытался подключиться к сети, чтобы проверить ход эксперимента, удаленный компьютер не ответил: благодаря закравшейся в исходный код ошибке «Червь» начал бесконтрольно размножаться, блокируя нормальную работу вычислительных систем, и вскоре вырвался из локальной сети MIT на просторы ARPANET'а — глобальной компьютерной сети, являвшейся на тот момент предшественницей современного Интернета.

Программа Морриса-младшего стала настоящим бедствием для США: в течение нескольких дней функционирование ARPANET было парализовано практически полностью. По различным подсчетам, эпидемия поразила порядка 6000 компьютеров — около 10% всех работавших тогда в сети вычислительных машин, нанесенный «Червем» ущерб оценивался от скромной цифры в 150 000 до значительной суммы в 75 млн долларов США. Вскоре к делу подключилось ФБР, однако расследование длилось недолго: Моррис сам признался в содеянном, а пресса раздула скандал до неимоверного масштаба, прежде всего благодаря профессии его отца — ведущего эксперта АНБ США по борьбе с компьютерной преступностью.

Судебное дело Роберта Морриса было одним из первых дел по обвинению в совершении компьютерного преступления в США, до этого по данной статье под суд попал лишь известный во всем мире хакер Кевин Митник. Морриса признали виновным и приговорили к выплате штрафа в размере 10 000 долларов и 400 часам исправительных работ. Однако слава Роберта Морриса, получившего широкую известность благодаря созданному им «Червю», до сих пор не дает покоя сотням и тысячам вирусописателей на разных континентах. С тех пор многим амбициозным молодым людям, разбирающимся в программировании, рано или поздно приходит в голову идея попробовать свои силы в создании программы, которая, распространяясь по Сети, могла бы дестабилизировать работу удаленных компьютеров. Именно их стремление к дешевой славе принесло пользователям множество бессонных ночей, проведенных за восстановлением разрушенных систем.

НОВОЕ ВРЕМЯ

На заре 90-х годов развитие компьютерных технологий набрало поистине фантастическую скорость. Не отставали от мировых тенденций и вирусописатели. Еще в 1989 году появилась первая классическая *троянская* вредоносная программа, известная под именем AIDS. Этот троянец использовал механизм монетизации, который стал повсеместно распространенным только

спустя 17 лет, с возникновением вредоносных программ-шифровальщиков: AIDS делал недоступной всю хранящуюся на дисках компьютера информацию, после чего демонстрировал на экране требование о выкупе в размере 189 долларов. Однако в те далекие времена еще не существовало анонимных платежных систем вроде Bitcoin, которые позволили бы злоумышленникам получать от своих жертв деньги, оставаясь при этом в тени, поэтому автор троянца был арестован полицией при попытке обналичивания чека и осужден за вымогательство.

В 1990-м году получили распространение первые *стелс-вирусы*, такие как Frodo и Whale. Особенностью данной категории вредоносных программ является умение полностью или частично скрывать свое присутствие в зараженной операционной системе путем перехвата ряда системных функций и обращений ОС к инфицированным файловым объектам. Так, стелс-вирусы, способные заражать исполняемые файлы, обычно перехватывали обращения операционной системы на чтение файла, запись в файл или загрузку/отображение файла в память, с целью скрыть изменение размера этого файла после заражения. Кроме того, Whale, исполняемый модуль которого «весил» всего 9 килобайт, обладал довольно совершенными алгоритмами шифрования и антиотладки, весьма затруднявшими его детектирование и анализ.

Тогда же, в 1990 году, впервые были обнаружены *полиморфные вирусы*, первенцем среди которых принято считать вредоносную программу, известную под названием Chameleon. Полиморфные вирусы используют специальную технологию, формирующую код вредоносной программы прямо во время ее исполнения, то есть буквально «на ходу». При этом сам код, описывающий алгоритм формирования исполняемого файла вируса, тоже не остается постоянным и меняется от одного инфицированного объекта к другому. Благодаря тому что вирус в результате всех этих манипуляций непрерывно «мутирует», его опознание, выявление и обезвреживание средствами антивирусных программ были в то время чрезвычайно затруднены. Другим известным полиморфным вирусом, также распространявшимся в 90-м году, стала вредоносная программа под названием Tequila.

1992 год ознаменовался массовой эпидемией вируса Michelangelo, заразившего миллионы компьютеров по всему миру.

Michelangelo можно назвать одним из первых в истории загрузочных *вирусов*, поскольку он, инфицируя компьютеры, работающие под управлением MS-DOS, модифицировал основную загрузочную запись операционной системы (*Master Boot Record, MBR*). Свое название вирус получил за то, что активировал свои вредоносные функции один раз в год, 6 марта, в день рождения великого художника эпохи Возрождения. В этот день он заполнял первые 100 секторов жесткого диска зараженного компьютера нулями. Поскольку вирус заражал загрузочную запись, он внедрялся в память компьютера во время запуска операционной системы. Стоило пользователю вставить в дисковод инфицированной персоналки любую дискету, и при обращении к гибкому диску вирус немедленно заражал его, обеспечивая таким образом собственное распространение. Поскольку большую часть времени эта вредоносная программа находилась в состоянии «сна», активизируясь только раз в год, она успела широко распространиться по всему миру, прежде чем была впервые обнаружена.

Следующая крупная эпидемия, вызванная распространением загрузочного вируса, произошла уже в 1994 году, и ее виновником стала очень опасная вредоносная программа, получившая известность под именем OneHalf. OneHalf представлял собой *полиморфный загрузочный файловый вирус*. Инфицировав основную загрузочную запись (Master Boot Record, MBR), при запуске операционной системы он передавал управление своему основному исполняемому модулю, который при каждом запуске компьютера шифровал по две дорожки на жестком диске со случайным ключом. Загруженный в оперативную память резидентный компонент вируса играл роль своего рода драйвера, перехватывая все обращения операционной системы к диску и расшифровывая ранее подверженные шифрованию данные «на лету», вследствие чего операционная система не испытывала каких-либо проблем в процессе чтения-записи файлов и фактически «не замечала» присутствие вируса в системе. Однако при попытке удаления вредоносной программы доступ к хранящейся на зашифрованных участках жесткого диска информации автоматически прекращался. После успешного шифрования половины доступного объема диска в момент загрузки операционной системы вирус в некоторых случаях выводил на экран компьютера сообщение: «*Dis is one half. Press any key to continue…* ».

После нажатия пользователем любой клавиши на клавиатуре компьютера загрузка MS-DOS возобновлялась в штатном режиме. Вирус отличался присутствием в своей архитектуре полиморфных алгоритмов, обладал различными функциями антиотладки, препятствующими попыткам его анализа, и обладал способностью заражать исполняемые файлы.

Поскольку OneHalf шифровал диск от конца к началу, рано или поздно он зашифровывал весь его доступный объем, включая загрузочную область. В результате при следующем включении компьютера запуск операционной системы становился невозможным, и вся информация на диске безвозвратно терялась. В начале 1994 года OneHalf вызвал настоящую пандемию среди пользователей MS-DOS, он встречался на инфицированных компьютерах вплоть до второй половины 90-х.

1995 год оставил свой след в мировой истории запуском каталога «Yahoo!», первой стыковкой американского Шаттла с российской станцией «Мир», началом работ по восстановлению храма Христа Спасителя в Москве и выпуском операционной системы Microsoft Windows 95. Именно последний факт оказал наиболее серьезное влияние на сферу информационной безопасности, поскольку появление принципиально новой по своей архитектуре ОС открыл перед вирусописателями широчайшие горизонты для творчества.

В 1995 году появились первые в истории *макровирусы*, использовавшие в целях реализации своей вредоносной деятельности скриптовые языки, предназначенные для создания макросов в приложениях пакета прикладных программ Microsoft Office, и, в частности, Microsoft Word.

Развитие операционных систем семейства Windows 90-х повлекло появление и новых вредоносных программ, так или иначе использующих их ресурсы. Так, в 1998 году разразилась эпидемия вируса Melissa, предназначенного для массовой рассылки нежелательных рекламных почтовых сообщений — *спама*, а вскоре пользователей компьютеров постигла новая напасть — распространение вируса Win95.CIH, также известного под народным наименованием «Чернобыль».

Эта вредоносная программа была написана тайваньским студентом Чэнь Инхао. Она представляла собой *резидентный вирус*, способный работать только на компьютерах под управлением

операционной системы Windows 95/98. Вирус активизировался 26 апреля 1999 года (в годовщину аварии на Чернобыльской АЭС, из-за чего он и получил свое название) и уничтожил все данные на жестких дисках инфицированных компьютеров, а в некоторых случаях модифицировал содержимое микросхем FlashBIOS, приводя компьютеры в полностью неработоспособное состояние. Таким образом, Win95.CIH стал первым в истории компьютерным вирусом, способным взаимодействовать с компонентами аппаратной архитектуры ПК, такими как микросхемы BIOS.

Начиная с конца 90-х годов появление новых модификаций вредоносных программ стало приобретать лавинообразный характер, и к началу 2000-х число вирусов и троянцев для операционных систем семейства Microsoft Windows насчитывало уже сотни тысяч.

НАШИ ДНИ

Если вирусописатели 90-х были, в основной своей массе, энтузиастами-одиночками, создававшими вредоносное ПО либо для собственного развлечения, либо в целях самоутверждения, то в новом тысячелетии производство троянских программ и компьютерных вирусов встало на коммерческие рельсы. Злоумышленники научились извлекать из распространения своих творений непосредственную финансовую выгоду, и в наши дни объемы этого незаконного теневого рынка составляют, по разным подсчетам, многие сотни миллионов долларов.

В 2000 году серьезную опасность для пользователей представлял *червь* «I LOVE YOU», распространявшийся в виде вложения в сообщения электронной почты. Потенциальная жертва получала письмо с вложением, содержавшее строку «*ILoveYou*». При попытке открыть вложение на атакуемом компьютере запускался VBS-сценарий, который рассылал копию червя по всем адресам электронной почты из адресной книги Microsoft Outlook. В общей сложности от действия этой вредоносной программы пострадало более 3 млн пользователей по всему миру.

Другой почтовый червь, известный под наименованием MyDoom, атаковал компьютеры пользователей в 2004 году. Инфицировав компьютер, MyDoom блокировал пользователю доступ к сайтам антивирусных компаний и компании Microsoft. Червь предназначался для организации массовых атак на отказ в обслуживании (DDoS-атак) на различные сайты.

В 2006 году было зафиксировано появление первых массовых *ботнетов* — вредоносных сетей, созданных с использованием автономно действующих дистанционно управляемых программ (*ботов*). Одной из первых и наиболее крупных таких бот-сетей стал Rustock. Данный ботнет, предназначенный для рассылки рекламных почтовых сообщений — спама, насчитывал на начальном этапе более 150 тысяч зараженных ПК, а в момент пика своего роста — более 2 миллионов. Компьютеры, инфицированные Rustock, позволяли злоумышленникам отсылать более 25 тысяч рекламных писем в час, благодаря чему ботнет приносил своим создателям миллионные прибыли.

В 2007 году в России было отмечено появление первых *троянцев-винлокеров*, получивших в последующие годы чрезвычайно широкое распространение, быстро принявшее масштабы настоящего национального бедствия, а в 2010 году троянцы этого типа впервые вышли за рамки российских границ. Эта категория программ-вымогателей блокировала нормальную работу операционной системы Microsoft Windows, демонстрируя на экране компьютера изображение-баннер с требованием заплатить выкуп за разблокировку. Широкому распространению этого типа угроз способствовало также и то, что злоумышленники создали специальные программы-конструкторы для быстрого создания троянцев-винлокеров, благодаря чему производить таких троянцев в поистине промышленных масштабах получили возможность даже люди, абсолютно далекие от программирования.

Еще одна крупная эпидемия разразилась в 2008 году: ее причиной стал почтовый червь Conficker, заразивший более 12 миллионов компьютеров во всем мире. Этот червь распространялся, используя уязвимости в архитектуре ОС Microsoft Windows, и потому в течение довольно длительного времени борьба с ним была весьма затруднена — до тех пор, пока корпорация Microsoft не выпустила соответствующие обновления безопасности. По оценкам экспертов, червь нанес совокупный

ущерб пользователям в размере, превышающем 9 млрд долларов, а в устранении последствий эпидемии помимо Microsoft принимали участие крупнейшие мировые антивирусные компании. В том же 2008 году появились первые банковские троянцы, предназначенные для хищения денежных средств непосредственно с банковских счетов своих жертв, использующих системы дистанционного банковского обслуживания («банк — клиент»).

Примерно в 2008 году появились первые *троянцы-энкодеры* (шифровальщики). Эти вредоносные программы также традиционно относят к категории *троянцев-вымогателей*, однако они представляют более серьезную опасность для пользователей, поскольку шифруют хранящиеся на дисках компьютера файлы и требуют оплаты выкупа за их расшифровку. На момент написания этой книги специалистам по информационной безопасности известно несколько десятков тысяч модификаций троянцев-шифровальщиков, причем новые их разновидности появляются с завидной регулярностью. Так, эпидемия червя-шифровальщика WannaCry, начавшаяся в мае 2017 года, привела к заражению более 500 000 компьютеров в 200 государствах мира. Этим червем были заражены сети многих коммерческих и государственных учреждений. Из-за эпидемии в ряде британских госпиталей было отложено выполнение ранее назначенных медицинских процедур, обследований и срочных операций. Известный бывший сотрудник ЦРУ, американский диссидент Эдвард Сноуден утверждал, что уязвимость операционных систем семейства MS Windows, благодаря которой WannaCry распространялся по планете, была давно известна техническим специалистам АНБ. Однако они не посчитали нужным проинформировать об этом компанию Microsoft, а заявили об уязвимости только тогда, когда заражение компьютеров приобрело глобальный характер.

Уже спустя месяц, в июне 2017 года, случилось массовое распространение шифровальщика Petya, от которого пострадали в основном жители Украины (но досталось пользователям и в других странах мира). Этот червь использовал уязвимость в протоколе SMB, и заражал загрузочную запись, перехватывая управление на этапе запуска операционной системы, после чего шифровал файлы на жестких дисках инфицированной машины. От действия Petya пострадало множество частных лиц и

организаций, многие из которых безвозвратно потеряли ценную информацию.

С ростом популярности мобильной операционной системы Google Android к ней заметно повысился и интерес со стороны злоумышленников: первые вредоносные программы для портативных устройств под управлением Android появились в 2010 году. А в 2012 году специалисты российской антивирусной компании «Доктор Веб» выявили первый в истории ботнет, состоящий из компьютеров Apple с установленной на них операционной системой macOS, зараженных троянцем-бэкдором BackDoor.Flashback.39. В момент пика своей деятельности бот-сеть BackDoor.Flashback.39 насчитывала более 650 тысяч зараженных компьютеров Apple по всему миру.

Во второй половине «десятых» годов (и вплоть до 2020-го) вирусописатели стали проявлять повышенный интерес к операционным системам семейства Linux. Ранее Linux не пользовался особым «спросом» среди злоумышленников, прежде всего, из-за малой распространенности и особенностей архитектуры этой ОС. Гораздо проще было заражать Windows, работавшую на компьютерах миллионов пользователей по всему миру.

Все изменилось с появлением и широким распространением так недорогих «умных» устройств, лежащих в основе так называемого «интернета вещей» (Internet of Things, IoT). В эту категорию, в частности, входят многочисленные роутеры, IP-видеокамеры, телевизионные приставки, медиаплееры, сетевые накопители и хранилища. Действительно: вряд ли кому-то придет в голову искать вредоносную программу в недрах телеприставки. Вместе с тем, такие устройства часто подключены к высокоскоростному каналу Интернета, и могут использоваться киберпреступниками для массированных DDoS-атак, рассылки спама и демонстрации навязчивой рекламы.

Отдельную категорию вредоносных программ составляют опасные *скрипты*, способные заражать веб-сайты. Распространяются они чаще всего с использованием уязвимостей в системах управления контентом («движках») сайтов, а также в пиратских версиях платных шаблонов и плагинов для различных CMS — WordPress, Joomla, Drupal и других. Подобные скрипты используются их создателями, как правило, в рекламных целях: для размещения ссылок, показа рекламных баннеров на

инфицированном сайте, а иногда — для несанкционированной загрузки на сервер различных файлов и получения управления (шелла) на скомпрометированном сайте.

В настоящий момент в лаборатории антивирусных компаний поступает на анализ до миллиона файлов различных типов ежедневно, из них вредоносными признаются десятки и сотни тысяч. А вирусописатели непрерывно изыскивают все более изощренные способы избежать детектирования своих творений современными антивирусными программами.

ГЛАВА 2. СРАВНИТЕЛЬНАЯ ВИРУСОЛОГИЯ

В настоящий момент в мире отсутствует какая-либо общепринятая единая система классификации вредоносных программ: каждая антивирусная компания использует собственный метод их идентификации и наименования. Вместе с тем существует определенная исторически сложившаяся практика, позволяющая относить вредоносные приложения к различным типам и категориям.

КЛАССИФИКАЦИЯ ПО ТИПУ ОПЕРАЦИОННОЙ СИСТЕМЫ

Так, наиболее очевидной (и наименее точной) методикой классификации вредоносного ПО можно считать распределение угроз по системным платформам, на заражение которых ориентировано то или иное опасное приложение. В этом отношении абсолютным и безоговорочным лидером являются операционные системы семейства Microsoft Windows, на которые сегодня рассчитано порядка 95% всех существующих в мире вредоносных программ.

На втором месте по числу опасных приложений располагается мобильная платформа Google Android. Согласно статистическим данным, опубликованным российской антивирусной компанией «Доктор Веб», в 2010 году специалистам по информационной безопасности было известно всего лишь порядка 30 вредоносных, нежелательных и потенциально опасных Android-программ, к 2011 году их количество составило уже 630, еще через год оно возросло до 1267, к концу 2014 года превысило 5600, в июне 2015 года достигло 10144, а в начале 2017 года составило уже 61 413. К августу 2018 года количество известных вредоносных и потенциально опасных программ для платформы Android достигло 128017 и продолжило расти. Эти цифры наглядно демонстрируют, что прирост числа угроз для платформы Android постепенно принимает экспоненциальный характер.

Основная причина такого бурного роста числа угроз для Android в целом очевидна: вирусописатели заинтересованы в извлечении прибыли от распространения своих вредоносных приложений, а мобильные платформы — это живые деньги. Фактически современный смартфон представляет собой мобильный электронный кошелек, и простор для творчества со стороны

злоумышленников здесь поистине огромен: они могут рассылать платные СМС-сообщения, совершать в тайне от пользователя телефонные звонки на различные коммерческие номера, подписывать жертву на те или иные виртуальные услуги, за использование которых с ее счета будут ежедневно списываться денежные средства, крутить на телефоне рекламу, а если его владелец использует системы мобильного банкинга — просто красть с его счета деньги, перехватывая входящие СМС с одноразовыми паролями и кодами авторизации. Сегодня существуют даже блокировщики и троянцы-шифровальщики для Android. Потенциальная емкость этого теневого рынка огромна, и он будет освоен, причем стремительно. Сейчас мы как раз наблюдаем этот процесс вживую.

Безусловно, 99% современных троянцев для Android неискушенный пользователь запускает на своем устройстве самостоятельно, скачав их из Интернета под видом игры или какой-либо полезной программы. Казалось бы, достаточно проявлять элементарную осмотрительность, и можно гарантированно избежать опасности заражения. Однако не стоит также сбрасывать со счетов тот факт, что ОС Android — это массовая и чрезвычайно популярная операционная система, занимающая львиную долю на современном рынке мобильных устройств. Она рассчитана на обычного, рядового, среднестатистического владельца смартфона, который может даже не знать такого термина, как «системная платформа». Все, что от него требуется, — это умение интуитивно нажимать на кнопки. И потому, когда загруженная из Интернета игра, запускаясь на выполнение, потребует от него доступа к СМС, сети, списку контактов, внутренней памяти телефона, личным данным и холодильнику на кухне, он нажмет «Ок» не задумываясь, потому что просто не поймет, что все эти слова означают. Впрочем, он и не должен этого понимать, он — пользователь, от слова «использовать». Именно это «слабое звено» и эксплуатируют в своих неблаговидных целях киберпреступники.

Вместе с тем, троянские программы встречаются и в заводских прошивках Android-устройств (чаще всего этим страдают дешевые мобильные телефоны китайского производства), а также в официальном каталоге Google Play, куда они периодически проникают несмотря на все принимаемые корпорацией Google

меры безопасности. Помимо этого, в 2016 году были выявлены вредоносные программы, способные внедряться в системные процессы ОС Android. Отдельную угрозу представляют Android-загрузчики, сами по себе не имеющие опасных функций, но способные скачивать из Интернета и запускать на мобильном устройстве различных троянцев, в частности под видом обновлений установленных приложений и игр. Вредоносным программам для мобильных устройств в этой книге посвящена отдельная глава.

Третье место по распространенности в «дикой природе» занимают вредоносные программы для ОС семейства Linux. Интерес злоумышленников к данной платформе обусловлен тем, что эти операционные системы активно используются на различных «умных» устройствах, относящихся к категории IoT (*Internet of Things, «интернет вещей»*). К таковым относятся бытовые и промышленные wi-fi роутеры, точки доступа, сетевые хранилища, кабельные модемы, телевизионные приставки, камеры с веб-интерфейсом управления, и т. д. Чаще всего вредоносное ПО проникает на подобные девайсы из-за использования на них заводских настроек, установленных производителем по умолчанию: многие владельцы ТВ-приставок или роутеров даже не подозревают, что в них предусмотрена возможность поменять пароль. Еще две причины — использование простых, нестойких к подбору паролей, и наличие ошибок в заводской прошивке устройства (*firmware*), которую пользователь ленится вовремя обновлять. Наиболее распространенный метод атак — перебор паролей по словарю (*брутфорс*), цель — внедрение на скомпрометированные устройства троянцев для проведения DDoS-атак и загрузки по команде злоумышленников другого вредоносного ПО. Известны случаи, когда такие троянские программы подменяли настройки DNS, что приводило к автоматическому перенаправлению пользователей на вредоносные интернет-ресурсы при просмотре веб-сайтов или вызывало неожиданное появление рекламных сообщений в окне браузера. Некоторые вредоносные программы подобного типа реализовывали функции прокси-сервера, осуществляя фильтрацию пользовательского трафика с различными неблаговидными целями (обеспечение анонимности злоумышленников в Интернете, организация несанкционированных подключений, фишинг,

подмена поисковой выдачи, автоматическое перенаправление на вредоносные и мошеннические интернет-ресурсы и т. д.). Троянцам для IoT также будет посвящена отдельная глава.

Отдельную подкатегорию Linux-угроз составляют троянцы, предназначенные для заражения веб-серверов, а также FTP- и почтовых серверов в Интернете, благодаря чему атака на такую машину может открыть злоумышленникам, например, доступ к SMTP-серверу для организации массовых рекламных рассылок, к веб-серверу — для реализации автоматических перенаправлений посетителей сайта на интернет-ресурсы, распространяющие другое вредоносное ПО, в целях хищения пользовательских учетных данных, а также для организации массовых DDoS-атак. Причиной заражения во многих случаях является недостаточная компетентность администраторов Linux-серверов при настройке операционной системы, установка различного ПО из недоверенных репозиториев, а также использование ими «слабых» паролей, неустойчивых к взлому путем простого подбора по словарю или с применением методов «брутфорса» (подбор пароля полным перебором, также известен как взлом с использованием «грубой силы»).

Четвертую позицию по количеству известных угроз занимает операционная система Apple macOS (ранее — OS X), широко известная среди обывателей своей «неприступностью» и «безопасностью». Троянских программ для macOS и вправду существует сравнительно немного, но, тем не менее, они есть, причем первый троянец, ориентированный именно на OS X, был выявлен в 2006 году. Подавляющее большинство среди угрожающих пользователям Apple вредоносных программ составляют так называемые рекламные троянцы, многие из которых реализованы в виде надстроек (плагинов) для наиболее популярных браузеров: Safari, Chrome, Firefox. Их основное назначение заключается в демонстрации потенциальной жертве назойливой рекламы при открытии им в окне браузера различных веб-страниц. Существуют троянцы-майнеры для macOS, использующие вычислительные мощности компьютеров Apple для «добычи» электронных криптовалют путем сложных математических вычислений. Одним из немногих и весьма редких примеров вредоносной программы, способной проникнуть на «мак» вообще без участия пользователя и абсолютно незаметно для него,

является уже упоминавшийся мною ранее бэкдор BackDoor.Flashback.39, использовавший для своего распространения уязвимость в Java-машине OS X. Следует, между делом, отметить, что если бы инцидента с Flashback не случилось, пользователей «маков» наверняка постигла бы другая аналогичная эпидемия — например, спустя короткое время после выявления упомянутой выше угрозы был обнаружен троянец BackDoor.Sabpub.1, использовавший для своего распространения ту же самую уязвимость, что и Flashback. Беды удалось избежать лишь потому, что корпорация Apple вовремя выпустила «заплатку» для своей реализации Java (вовремя — это спустя два месяца после выхода аналогичного обновления от Oracle). Этих двух месяцев вполне хватило на то, чтобы BackDoor.Flashback успел инфицировать 600000 с лишним машин. По данным на июнь 2015 года во всем мире насчитывалось порядка 26700 компьютеров Apple, зараженных Flashback, однако к началу 2017 года этот ботнет практически прекратил свое существование.

Безусловно, с точки зрения архитектуры и разграничения прав доступа операционная система от Apple намного безопаснее Windows, однако абсолютно безопасных системных платформ не бывает в принципе. Любая ОС — это по большому счету конгломерат программ различного назначения, которые пишут люди. А им, в свою очередь, свойственно ошибаться. То, что вирусописатели всерьез заинтересовались macOS, — объективный признак роста распространенности системной платформы от Apple. Ни один создатель троянцев не станет писать вредоносное ПО под непопулярную ОС: чем больше число пользователей, тем выше шанс заразить чью-либо машину. Чистая математика, ничего личного. И эпидемия BackDoor.Flashback.39 — признак того, что macOS перешагнула некий незримый рубеж между «игрушкой для избранных» и «массовой системной платформой». И это — естественный процесс, получивший свое развитие еще в момент перехода Apple на аппаратную платформу Intel.

Угрозы для других системных платформ в количественном выражении составляют малозначительные величины. Так, по данным на 2020 год, число вредоносных программ, способных представлять опасность для пользователей мобильной платформы Apple iOS, вообще исчисляется единицами, да и те

представляют угрозу в основном для устройств, подвергшихся процедуре jailbreak (несанкционированного производителем устройства получения доступа к файловой системе).

КЛАССИФИКАЦИЯ ПО ВРЕДОНОСНЫМ ФУНКЦИЯМ

Более распространенной и логически правильной формой классификации вредоносных программ является их распределение по типам и подклассам согласно формальным признакам, определяющим их вредоносные функции. Ниже мы рассмотрим основные виды вредоносных программ по их деструктивному функционалу, архитектурным особенностям и практическому назначению, а также перечислим их основные характерные признаки.

Вирусы

Сегодня существует огромное количество различных программ, способных причинить вред вашему ноутбуку, компьютеру, смартфону или планшету, однако большинство пользователей отчего-то традиционно называет все их «вирусами». Это в корне неправильно. Возможно, некоторые читатели даже удивятся, если я скажу, что эпоха, ознаменованная преобладанием в «дикой природе» компьютерных вирусов, по большому счету, закончилась в начале «нулевых годов». В наши дни классические файловые вирусы пребывают в меньшинстве, составляя среди общего объема распространяющихся во всем мире вредоносных программ незначительные величины порядка нескольких процентов. Основной массив вредоносного ПО представляют различные разновидности троянцев, о которых мы поговорим позже.

Чтобы отнести вредоносную программу к категории *компьютерных (файловых) вирусов*, она должна отвечать двум важным критериям:

- обладать способностью к саморепликации — иными словами, способностью к распространению в автоматическом

режиме путем создания собственных копий без участия пользователя;
- обладать способностью заражения (инфицирования) файловых объектов.

Вот об этом следует поговорить чуть подробнее. Собственно, умением самостоятельно создавать собственные копии (самореплицироваться) обладает и еще одна категория вредоносных программ — *компьютерные черви (worms)*. А вот умение заражать файлы характерно в первую очередь для вирусов. Под заражением понимается технология, с использованием которой вирус внедряется непосредственно в файл исполняемого приложения (программы), не нарушая при этом ее основных функциональных возможностей. Запуская такую программу на исполнение, пользователь одновременно запускает и встроенный в нее вирус, который, загрузившись в память инфицированного компьютера, реализует заложенные в него создателями деструктивные функции. Таким образом, распространение вируса происходит в том числе вместе с зараженными программами, в которые успел встроиться вирус.

Помимо классических файловых вирусов принято различать еще несколько разновидностей компьютерных вирусов, которые кратко упоминались в предыдущей главе.

Полиморфные вирусы — вирусы, с целью затруднить свое обнаружение и уничтожение, способные «на лету» изменять свой код непосредственно в процессе его исполнения. Процедура, отвечающая за динамическое изменение кода вируса, тоже может меняться от заражения к заражению. Самый простой способ модифицировать структуру исполняемого файла вируса, не меняя алгоритм его работы, — добавить в программу «пустых операторов», пустых циклов, пустых строк и прочего «мусорного кода», не оказывающего существенного влияния на функциональные возможности вируса, но затрудняющие создание сигнатуры для его детектирования. Практически все современные файловые вирусы используют те или иные полиморфные технологии.

Стелс-вирусы — вирусы, способные полностью или частично скрывать свое присутствие на инфицированном компьютере, например, путем перехвата обращений операционной системы к зараженным файловым объектам, памяти или загрузочным

областям диска. Фактически, перехватив обращение операционной системы, стелс-вирус возвращает операционной системе недостоверную информацию о состоянии соответствующего объекта, например, о размере исполняемого файла, с целью сокрытия увеличения его объема в результате добавления в файл вирусного кода. Этот термин применялся по отношению преимущественно к вирусам, действовавшим в операционной системе MS-DOS, и потому сейчас его можно считать устаревшим. Вредоносные программы, использующие механизмы сокрытия своего присутствия в зараженной системе, принято называть *руткитами*.

Макровирусы — вирусы, написанные с использованием скриптовых языков, применяющихся для создания макросов в различных офисных приложениях, таких как Microsoft Office, в частности Microsoft Word.

Резидентные вирусы. Сам термин «резидентный» традиционно принято использовать в контексте «выполняющийся в фоновом режиме». Ранее «резидентными вирусами» называли вредоносные программы, действующие непосредственно в памяти зараженного компьютера параллельно с другими запущенными задачами и процессами. Иногда к этой же категории относили вирусы, не хранящиеся где-либо на диске в виде обособленного физического файла. Такие вирусы либо уничтожали исходный файл после своего запуска, либо хранили его в недоступных пользователю и операционной системе областях диска, либо «собирали» свое тело непосредственно в оперативной памяти из отдельных разрозненных компонентов (каждый из которых сам по себе не представлял какой-либо опасности). С появлением многозадачных операционных систем само понятие «резидентный вирус» принято считать устаревшим, а угрозы, действующие непосредственно в памяти инфицированного устройства и не хранящиеся на физических носителях в виде обособленных файлов, часто называют общим термином «*бесфайловые вредоносные программы*».

Черви

Компьютерные черви (*worms*) — разновидность вредоносных компьютерных программ, обладающих способностью к саморепликации, то есть к автоматическому

распространению без участия пользователя — по локальной сети, по каналам электронной почты, с использованием сменных носителей информации или иными методами. При этом считается, что большинство червей не способно заражать файловые объекты, хотя из данного правила имеются некоторые исключения.

В наши дни довольно широко распространены так называемые *почтовые черви* — эти вредоносные программы, запустившись на исполнение, отыскивают все хранящиеся на зараженном компьютере адреса электронной почты (некоторые сканируют для этих целей не только адресные книги почтовых клиентов, но также различные текстовые документы, локально хранящиеся на диске веб-страницы и документы Office), после чего рассылают свою копию по этим адресам в виде вложения в электронное письмо. Адрес отправителя такие вредоносные программы также зачастую заимствуют из списка контактов на зараженной машине. В этом отношении нельзя не отметить, что получатель подобного сообщения, увидев послание, якобы отправленное знакомым ему человеком, с большой долей вероятности попытается открыть такое письмо и в результате сам станет очередной жертвой почтового червя.

Многие черви распространяются, копируя себя на съемные носители информации. Например, некоторые вредоносные программы данного типа размещают в корневой папке съемного накопителя (флэшки или карты памяти) файл *autorun.inf*, обеспечивающий автоматический запуск червя при каждом обращении к инфицированному накопителю (в устаревших версиях операционных систем, где функция запуска и открытия содержимого смонтированных дисков не отключена по умолчанию, например, в ранних редакциях Windows XP, запуск червя может произойти автоматически при подключении зараженного диска к компьютеру). Другие черви перемещают все содержимое съемного носителя информации в скрытую папку, а вместо него размещают собственные копии или ярлыки с прежними именами папок и файлов. В результате пользователь, щелкнув мышью на значке такого файлового объекта или ярлыка, вместо открытия нужной ему папки или файла запустит на выполнение вредоносную программу.

Троянские программы (трояны или троянцы)

Это самый многочисленный и распространенный тип вредоносных программ. В отличие от вирусов и червей, троянцы не способны ни к саморепликации, ни к заражению файловых объектов.

Название данной категории угроз восходит к знаменитой легенде про деревянного коня, которого ахейцы преподнесли в подарок жителям осажденной Трои — напомню, что ночью из огромной деревянной статуи вылезли прятавшиеся внутри воины и открыли ворота спящего города атакующей армии. Троянские программы действуют практически аналогичным образом: маскируясь под какое-нибудь полезное приложение — утилиту для просмотра видео, компьютерную игру, даже под антивирус или безобидный, на первый взгляд, документ, присланный по электронной почте — например счет, квитанцию о штрафе или судебный исполнительный лист, — троянец при попытке запуска или открытия файла начинает свою вредоносную деятельность на зараженной машине. Выловить вредителя не так-то просто, особенно с учетом того, что многие троянские программы умеют «прятаться» в недрах операционной системы и даже хитроумно обходить некоторые типы антивирусной защиты.

Из сказанного выше можно сделать простой и очевидный вывод: троянские программы жертва запускает на своем компьютере самостоятельно, при этом злоумышленники заставляют ее сделать это различными хитроумными способами. Вот только один из нескольких сотен примеров возможного пути распространения троянцев. Однажды в папке личных сообщений пользователь социальной сети находит послание, отправленное кем-либо из его списка друзей (этот аккаунт, разумеется, был ранее взломан злоумышленниками). В сообщении содержится информация о том, что на сайте YouTube выложен компрометирующий пользователя видеоролик, а также приводится ссылка на соответствующую страничку. Перейдя по ссылке, пользователь попадает на принадлежащий киберпреступникам фишинговый сайт, полностью копирующий оформление сервера youtube.com. Подделка настолько похожа

на оригинальный YouTube, что отличить его от настоящего сайта популярного видеохостинга, на первый взгляд, довольно-таки сложно. На фишинговой веб-странице обозначено название видеоролика, в котором содержится имя жертвы, например, он может быть озаглавлен следующим образом: «*%username% is in the leading role. Shocking performance!*», что в переводе означает: «*%username% в главной роли. Шокирующее поведение!*», где вместо %username% автоматически подставляется имя потенциальной жертвы. Также страничка содержит несколько фальшивых комментариев от других пользователей сайта, которые «*были шокированы, ознакомившись с этим видео*». Чтобы просмотреть ролик, пользователю предлагается скачать и установить обновленный flash-проигрыватель, в роли которого выступает исполняемый файл с именем *Flash-Player.exe*. В этом файле и содержится троянец.

Однако чаще злоумышленники не изобретают подобных сложных схем, а массово рассылают троянцев в виде вложений в сообщения электронной почты под видом коммерческих предложений, сообщений почтовых служб и интернет-магазинов, скидочных купонов, бухгалтерских документов — счетов или счетов-фактур, различных договоров, судебных бумаг и т. д. Кроме того, очагами распространения троянских программ традиционно являются сайты категории «для взрослых», всевозможные коллекции пиратских и взломанных коммерческих программ, файлообменные интернет-ресурсы, сборники утилит для взлома лицензионных продуктов (всевозможные «кряки», «кейгены» и т. д.). Шанс получить вместе со «взломанной» программой или игрой бесплатный «подарок» в виде опасного троянца намного выше, чем в случае загрузки этого же приложения с сайта разработчиков или покупки такой программы у официальных дистрибуторов.

Бэкдоры

Если мы попытаемся перевести с английского языка слово «*бэкдор*» (*Backdoor*), то получится что-то вроде словосочетания «черный ход» или «задняя дверь». Вредоносные программы-бэкдоры, к которым относят и троянцев, и некоторые виды вирусов, как раз и выполняют на зараженном

устройстве подобные функции: они без ведома пользователя открывают злоумышленникам полный доступ к инфицированному компьютеру или смартфону, о чем его владелец зачастую даже не догадывается. С помощью бэкдоров киберпреступники могут не только увидеть все хранящиеся на устройстве личные файлы и по желанию скопировать их себе, но также полностью управлять зараженной машиной: запускать на ней различные программы, отдать команду на полное уничтожение всей имеющейся на устройстве информации, поместить на диск компрометирующие пользователя документы или фотографии, похитить деньги, если жертва пользуется системой «банк — клиент» или электронными платежными системами, даже использовать зараженный компьютер в качестве промежуточного звена в процессе интернет-атаки на банк или какой-нибудь другой сервер: в ходе последующего расследования служба безопасности или полиция могут выйти на владельца инфицированного устройства благодаря оставленным в сети следам, а настоящие преступники останутся неузнанными.

Наиболее простые с архитектурной точки зрения модификации бэкдоров могут транслировать поступающие извне (от принадлежащего злоумышленникам командного центра) директивы командному интерпретатору операционной системы (*cmd* в Microsoft Windows, *bash* или *sh* в Linux), более «продвинутые» обладают собственной системой команд, позволяющей выполнять различные операции с файловыми объектами и самой операционной системой. Данные, которыми обменивается бэкдор с удаленным сервером злоумышленников, как правило, шифруются.

Буткиты

Буткиты (от англ. boot — «загрузка» и kit — «инструмент»), или так называемые *загрузочные вирусы* — это вирусы или троянцы, способные заражать загрузочную запись на диске компьютера, благодаря чему запускаются либо раньше операционной системы, либо одновременно с ней, но в любом случае перед загрузкой основных средств антивирусной защиты. Из этого логически вытекает основная сложность борьбы

с буткитами — поскольку они стартуют еще на раннем этапе загрузки компьютера, буткиты перехватывают некоторые функции управления операционной системой и, как следствие, могут парализовать запуск и нормальную работу антивирусных программ, а также блокировать попытки «вылечить» инфицированное устройство. При неудачном удалении такой угрозы может произойти повреждение логической структуры диска, благодаря чему система и вовсе перестанет загружаться, и сложное электронное устройство «превратится в кирпич». Этим они и опасны.

Существуют разновидности буткитов: некоторые из них заражают главную загрузочную запись диска (*Master Boot Record, MBR*), некоторые — загрузочную запись тома (*Volume Boot Record, VBR*). В общем случае алгоритм действия буткита таков: запустившись на инфицированном компьютере, он размещает свою копию в одной из свободных логических областей диска, а затем модифицирует существующую загрузочную запись, внедряя в нее собственный код, который получает управление при запуске операционной системы и загружает в оперативную память основное тело буткита. По окончании этого процесса буткит передает управление дальнейшей загрузкой оригинальной загрузочной записи, позволяя ОС стартовать в штатном режиме. На момент завершения загрузки операционной системы вредоносная программа уже находится в памяти и может выполнять различные деструктивные действия, например, перехватывать те или иные системные функции и предотвращать запуск антивирусных программ, а также блокировать пользователю доступ к сайтам их разработчиков.

Особая опасность буткитов заключается еще и в том, что, запускаясь вместе с операционной системой, эти вредоносные программы могут получить в ней максимальные привилегии (например, полномочия администратора), даже если текущий сеанс открыт пользователем с ограниченными системными правами. Таким образом, буткит имеет на зараженном компьютере поистине неограниченные возможности для реализации всевозможных вредоносных функций, включая полный доступ к файловой системе, компонентам ОС, памяти и драйверам.

Руткиты

Руткитами специалисты по информационной безопасности обычно называют вредоносные приложения, умеющие скрывать свое присутствие в зараженной операционной системе, а также активно противодействовать попыткам их поиска или удаления. Кроме того, некоторые руткиты специально разработаны с целью сокрытия на инфицированном компьютере деятельности других вредоносных программ. В подобной ситуации несколько опасных приложений действуют как бы в связке: одно из них «обеспечивает прикрытие», а второе — реализует свой вредоносный функционал.

Обобщая, можно сказать, что руткиты предназначены для реализации в инфицированной операционной системе следующих функций:

- сокрытия различных объектов (файлов, папок, запущенных приложений и / или загруженных драйверов);
- сокрытия происходящих в системе процессов (запуск и выгрузка приложений, загрузка динамических библиотек, встраивание в запущенные процессы, заражение файловых объектов и т. д.);
- контроля над происходящими в системе процессами (перехват системных функций, обращений, контроль над иными системными событиями).

Многим руткитам для реализации своих функций необходимы полномочия администратора операционной системы, для чего некоторые из них стараются повысить собственные привилегии с использованием различных уязвимостей в архитектуре операционной системы. Подобные руткиты, действующие на уровне ядра операционной системы, принято называть *kernel-mode*, однако существуют руткиты и *user-mode*, действующие, с точки зрения используемых системных привилегий, на уровне пользователя. Например, руткит-модуль, встроенный в некоторые модификации широко известного банковского трояна семейства Zeus, является именно таким.

Для реализации своих деструктивных функций руткиты используют различные методы: перехват системных функций, захват таблиц вызовов, использование драйверов

или непосредственную модификацию системных объектов и объектов ядра ОС (Direct kernel object manipulation) и другие.

Биоскиты

Биоскиты — это вредоносные программы, способные модифицировать содержимое микросхем BIOS инфицированного компьютера. Подобные угрозы встречаются в природе достаточно редко, точнее, на сегодняшний день практически совсем не встречаются. Последняя такая программа, получившая наименование *Trojan.Bioskit.1*, была обнаружена специалистами антивирусной компании «Доктор Веб» еще в сентябре 2011 года, и по всем признакам она походила скорее на экспериментальную и незавершенную разработку академического характера, чем на реальную угрозу.

Trojan.Bioskit.1 использовал для заражения микросхемы BIOS стороннюю утилиту производства компании Phoenix Technologies, при этом теоретической опасности подвергались лишь компьютеры, оборудованные материнской платой, которая использует BIOS с прошивкой от Award Software, да и то при соблюдении ряда условий. В случае успешной перепрошивки содержимого BIOS уже сама эта микросхема могла стать источником повторного заражения компьютера даже после успешного лечения инфекции на уровне операционной системы.

Безусловно, не стоит недооценивать опасность такого рода угроз, особенно с учетом того, что в будущем подобные технологии могут получить дальнейшее развитие. Однако на момент написания этой книги можно смело сказать, что биоскиты не имеют широкого распространения и не представляют серьезной опасности для пользователей.

Боты

Ряд вирусов, троянцев и бэкдоров иногда называют *ботами* (от английского слова bot, сокращенного от robot) — это вредоносные программы, наделенные способностью объединяться в *ботнеты* (или бот-сети). Ботнетом называют сеть зараженных устройств, дистанционно управляемых злоумышленниками, например, с использованием одного или нескольких командных

серверов, и умеющих обмениваться информацией. Для чего киберпреступники создают ботнеты? В частности, для централизованной атаки на различные серверы Интернета. Скажем, если к какому-нибудь веб-сайту по команде одновременно обратится несколько сотен тысяч компьютеров, при этом они станут посылать ему запросы с интервалом в несколько микросекунд, у сервера попросту не останется ресурсов для обслуживания других клиентов и он откажется работать под такой неожиданно возросшей нагрузкой. Подобный тип атак называется «атака на отказ в обслуживании» — *Distributed Denial of Service*, или, сокращенно, *DDoS-атаками*. Подобным способом некоторые не слишком разборчивые в средствах владельцы коммерческих сайтов иногда «устраняют» своих конкурентов, а киберпреступники получают прибыль, продавая подобные услуги на подпольных форумах. Другой пример — массовая рассылка спама: «подцепив» где-нибудь троянскую программу, входящую в спаммерскую бот-сеть, ПК жертвы может стать одним из миллиона компьютеров, рассылающих по электронной почте рекламу виагры или сайтов для взрослых, при этом содержание рекламных писем и список адресов для рассылки поступают спам-ботам по команде из единого управляющего центра.

Существуют боты, способные самостоятельно настраиваться на различные управляющие серверы и автоматически менять их в случае необходимости (например, если какой-либо из таких серверов внезапно перестанет работать). Есть ботнеты, вообще обходящиеся без командных центров: они формируют так называемые *Peer-to-Peer*, или *P2P-сети*, — одноранговые и децентрализованные, уничтожить которые очень непросто. А, например, троянцы семейства IRC.Bot используют в качестве управляющего сервера … чаты для обмена текстовыми сообщениями, созданные с использованием протокола Internet Relay Chat (IRC). Подключаясь к одному из таких чат-каналов, специально созданных злоумышленниками, IRC.Bot ожидает получения от киберпреступников команд, которые те отправляют в виде простого текстового послания. Некоторые боты для мобильной платформы Android, например представители многочисленного семейства Android. SmsBot, могут получать управляющие директивы в СМС-сообщениях, поступающих на зараженный мобильный телефон. Подробнее о ботнетах мы поговорим в одной из следующих глав.

Шпионы (Spyware)

Шпионское программное обеспечение (spyware) чрезвычайно широко распространено в современном мире — значительная часть шпионских программ реализована в виде классических троянцев. Предназначение таких программ вполне очевидно: они способны следить за пользователем и передавать злоумышленникам информацию, получаемую с его устройства.

Один из наиболее популярных типов программ-шпионов — это так называемые *кейлоггеры*, то есть приложения, считывающие и сохраняющие в специальный журнал коды нажимаемых пользователем клавиш, а потом передающие эту информацию злоумышленникам. Таким образом, можно, например, похищать вводимые жертвой в различные экранные формы логины и пароли, а также любую иную набираемую на клавиатуре информацию. К слову, для хищения вводимых в экранные формы данных на различных веб-страницах существует специальная категория утилит, которые называют *грабберами*, — в отличие от кейлоггеров, фиксирующих все нажатия клавиш, они способны получать только интересующие злоумышленника значения, например путем анализа передаваемых от клиента серверу с помощью протокола HTTP GET и POST-запросов или при помощи перехвата используемых браузером API-функций.

Другие троянцы-шпионы могут создавать и отправлять киберпреступникам *скриншоты* — снимки содержимого экрана зараженного компьютера, осуществлять скрытую съемку с использованием встроенной видеокамеры устройства. Кроме того, программы-шпионы, ориентированные на мобильную операционную систему Google Android, могут транслировать злоумышленникам географические GPS-координаты текущего положения зараженного устройства, передавать журнал звонков, фотографии, выполнять несанкционированную фото- и видеосъемку, записывать телефонные разговоры и даже использовать встроенный микрофон мобильного устройства для скрытой диктофонной записи с последующей передачей полученных звуковых файлов на удаленный управляющий сервер.

Очевидно, что присутствие на персональном компьютере, ноутбуке или мобильном устройстве программы-шпиона является прямой угрозой конфиденциальности и частной жизни пользователя.

Нежелательные и нерекомендуемые приложения

Помимо вредоносных программ существует широчайший ассортимент разнообразных *потенциально опасных* и *нежелательных приложений*, которые сами по себе вредоносного потенциала не несут, но при определенных обстоятельствах способны доставить пользователю кучу ненужных проблем. В частности, к такой категории относятся приложения класса *Adware* — утилиты, которые, возможно, и имеют какую-то полезную нагрузку, но при этом помещают во все просматриваемые пользователем веб-страницы назойливую рекламу. Работать в Интернете на компьютере с установленными приложениями категории Adware становится практически невозможно: рекламные баннеры высвечиваются в самых неожиданных частях веб-страниц, выпрыгивают в отдельных окнах, браузер самостоятельно открывает дополнительные вкладки, произвольно меняется его стартовая страница…

С одной стороны, смертельной опасности такое поведение компьютера для пользователя не представляет, с другой — доставляет ему значительное неудобство.

Среди нежелательных программ отдельно можно отметить приложения семейства *Fakealert*: если перевести с английского это наименование, получится что-то вроде «ложного предупреждения». К этой категории относятся всевозможные поддельные антивирусы, утилиты для оптимизации системного реестра, «ускорения» Интернета и «лечения» компьютера. Выполнив «проверку» пользовательского устройства, они обнаруживают различные угрозы и неполадки, за устранение которых требуют оплаты. Разумеется, на самом деле никакой проверки не выполняется, а все выявленные таким приложением проблемы и неисправности существуют только в буйной фантазии разработчиков этой программы. Цель они преследуют лишь одну: как следует напугать пользователя и заставить его заплатить как можно больше денег.

КЛАССИФИКАЦИЯ ПО СТЕПЕНИ ОПАСНОСТИ

Именно этот принцип лежит в основе классификаций, используемых большинством антивирусных компаний в современном мире. Данный способ классификации во многом напоминает предыдущий, однако определяющей точкой в нем является не набор деструктивных функций, реализуемых той или иной вредоносной программой, а степень опасности, которую представляет поведение каждой конкретной угрозы на инфицированном компьютере.

При этом в подобной классификации реализуется принцип поглощения классов в виде определенной иерархической последовательности, например «Файловые вирусы -> Черви -> Бекдоры -> Троянцы». Пример такой иерархической структуры демонстрирует следующая иллюстрация.

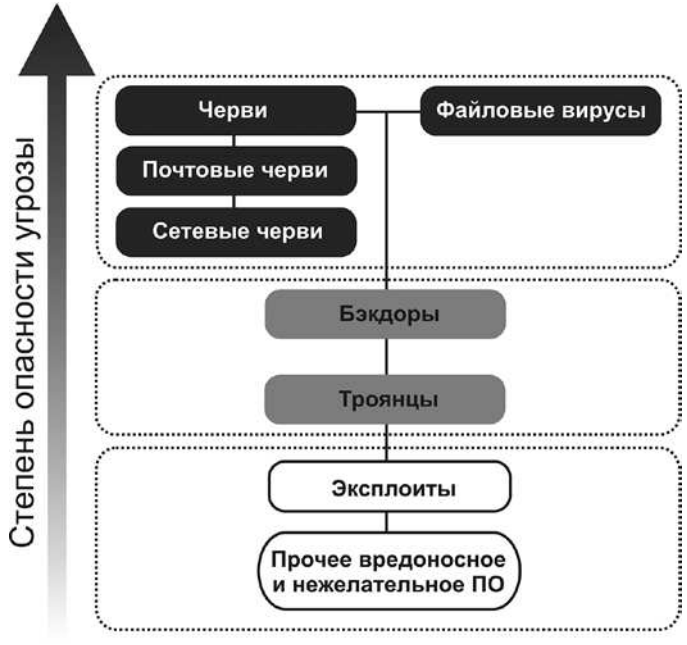


Рис. 1. Пример классификации вредоносных программ по степени опасности поведения

Почему такая система классификации является наиболее распространенной? Дело в том, что многие современные вредоносные программы являются многофункциональными и обладают формальными признаками сразу нескольких классов. Например, некий поступивший на анализ в вирусную лабораторию образец наделен способностью заражать файловые объекты (признак файлового вируса), но при этом может распространяться, создавая свои копии в общедоступных сетевых папках (признак сетевого червя), а также умеет выполнять поступающие от злоумышленников команды (признак бэкдора). В этом случае перед вирусным аналитиком возникает нелегкая задача: к какой категории отнести такой сложный экземпляр? Здесь на помощь приходит принцип поглощения классов, проиллюстрированный на представленной выше схеме, который позволяет отделить второстепенные по уровню опасности признаки от первостепенных. На указанной схеме мы видим, что наиболее опасное поведение находится в верхней части иерархии, а наименее опасное — в нижней. Так, способность предоставлять злоумышленникам несанкционированный доступ для управления зараженной машиной (бэкдор) менее опасна, чем умение самостоятельно распространяться без участия пользователя (червь). А оно, в свою очередь, представляет примерно равную опасность со способностью заражать файловые объекты (вирус): обе эти функции находятся в верхней части иерархической цепочки. Таким образом, аналитик скорее всего отнесет этот образец к классу файловых вирусов с признаками червя. Однако возможна ситуация, при которой основная опасность данной конкретной угрозы будет заключаться как раз в способности быстро и бесконтрольно распространяться по сети путем создания собственных копий, в то время как заложенный в программу механизм заражения файлов может срабатывать только при наступлении строго определенных условий или в ряде конкретных случаев. Тогда вирусный аналитик может отнести указанный образец к классу сетевых червей, и мы получим, ради разнообразия, червя, обладающего функцией заражения исполняемых файлов.

Тот же принцип будет распространяться, скажем, на программу, реализующую на зараженном компьютере сервер, который «слушает» определенный порт в ожидании поступающих

от злоумышленников команд или установки входящего соединения (признак бэкдора), и при этом может похищать пароли от различных приложений (это умеют многие троянцы). Такой образец скорее всего будет отнесен к классу бэкдоров, поскольку функция предоставления несанкционированного удаленного управления опаснее кражи конфиденциальных данных. Вредоносная программа с разнообразным набором менее деструктивных функций может считаться троянцем.

Итак, с основными типами и категориями вредоносных программ мы разобрались. В следующей главе обсудим основные разновидности троянцев и поговорим о том, какую опасность они могут представлять для пользователя.

ГЛАВА 3. ВНИМАНИЕ, ОПАСНОСТЬ!

Фантазия создателей вредоносных приложений поистине безгранична, однако все они так или иначе преследуют одну цель: тем или иным способом заработать деньги на пользователях, ставших жертвами их программных творений.

Про троянцев, рассылающих с инфицированного компьютера спам или подключающих его к подпольной сети для атаки на веб-сайты, я уже рассказывал ранее. В этом случае злоумышленники получают доход не непосредственно от владельца зараженного компьютера, а от заказчиков массированных атак или рекламных рассылок.

Троянцы-бэкдоры, предоставив злодеям доступ к зараженной машине, открывают перед ними возможность делать с данным устройством все что угодно. В итоге инфицированный компьютер может быть использован в качестве промежуточного звена при атаке на какой-либо банковский или правительственный сервер, для хранения краденых паролей или детского порно, для организации на нем архива с рецептами изготовления наркотиков и т. д. Устройство, на котором работает бэкдор, своему номинальному владельцу больше не принадлежит. Многие троянцы способны красть на зараженных машинах конфиденциальную информацию, такую как логины и пароли для доступа к электронной почте, социальным сетям, номера банковских карт, сведения из адресной книги. Всю добытую информацию злоумышленники потом продают за неплохие деньги: адреса электронной почты востребованы у спамеров; получив доступ к социальным сетям, мошенники смогут отправлять от имени жертвы ее же друзьям рекламные ссылки и просьбы пополнить счет мобильного телефона, с помощью ворованных реквизитов банковской карты можно купить какой-нибудь товар в интернет-магазине, а изображение паспорта пригодится для оформления сим-карты или открытия на имя владельца паспорта счета, на котором будут аккумулироваться деньги от продажи нелегального контента (причем узнает он об этом, только когда к нему домой нагрянет с обыском полиция). Не думайте, что хранящиеся на вашем устройстве файлы, документы и фотографии никому не нужны: на любой товар всегда найдется свой покупатель.

Каким еще образом киберпреступники утоляют свою жажду наживы? Об этом мы и поговорим в рамках настоящей главы, представляющей собой краткий обзор деструктивных функций наиболее распространенных в настоящее время вредоносных программ.

Троянцы-блокировщики (винлокеры)

Троянские *программы-блокировщики*, или *винлокеры*, принято относить к общей категории *программ-вымогателей*, которые в англоязычных источниках называют также термином *ransomware*. Первые случаи заражения вредоносными программами данного семейства были зафиксированы еще в конце 2007 года, а в период с ноября 2009-го по февраль 2010 года их распространение приняло буквально эпидемический характер. Винлокеры блокировали Рабочий стол Windows графическим окном-баннером, в котором демонстрировался текст с требованием выкупа за разблокировку системы и описанием возможных способов оплаты.

Первые модификации таких программ-вымогателей весьма успешно маскировались под встроенный механизм активации операционной системы Windows XP (*Microsoft Product Activation, MPA*). Окно программы, блокировавшее Рабочий стол Windows, имитировало своим оформлением интерфейс MPA, при этом пользователю демонстрировалось сообщение о том, что на его ПК установлена нелицензионная копия операционной системы, которую предлагалось активировать, отправив платное СМС-сообщение. Прервать работу этих ранних версий винлокеров было относительно несложно, удалив соответствующий процесс в Диспетчере задач, изменив в системном реестре значение оболочки по умолчанию или воспользовавшись утилитой Восстановление системы, для чего следовало загрузить Windows в безопасном режиме. Однако программа стремительно совершенствовалась: достаточно быстро винлокеры научились отслеживать нажатия сочетаний «горячих» клавиш, блокировать запуск Диспетчера задач, Редактора реестра, утилиты Восстановление системы, Командной строки, антивирусных приложений и некоторых апплетов Панели управления Windows. Троянец вносил определенные изменения в файл

%SYSTEMROOT% \ System32\drivers\etc\hosts, чтобы пользователь не мог открыть в браузере сайты разработчиков наиболее популярных антивирусных программ. Модифицировалось и оформление окна, нарушающего нормальную работу компьютера: оно могло демонстрировать различные непристойные изображения или сведения о том, что пользователь посещал сайты порнографического характера, либо обвинения в незаконной загрузке информации, запрещенной законодательством.

Таким образом, троянцы-блокировщики использовали для достижения своих целей принципы *социальной инженерии*: многие владельцы компьютеров, в действительности посещавшие порнографические интернет-ресурсы либо использовавшие на своем ПК контрафактное ПО, отправляли злоумышленникам платные СМС, либо пополняли указанные ими счета мобильных операторов, опасаясь огласки. Кроме того, некоторые версии винлокеров в случае отказа от оплаты либо попыток разблокировать компьютер иными методами угрожали уничтожением всей хранящихся на жестких дисках информации.

В 2011 году появились разновидности блокировщиков, не вносивших изменения в конфигурацию операционной системы, а модифицировавших главную загрузочную запись (Master Boot Record), делая невозможным запуск самой ОС. Такой троянец вносит изменения в Master Boot Record, однако оригинальная загрузочная запись и таблицы разделов обычно сохраняются. При каждом последующем включении питания компьютера вымогатель блокирует загрузку операционной системы, считывает из соседних секторов жесткого диска в память основной код винлокера и демонстрирует на экране требование заплатить выкуп за разблокировку компьютера.

А уже в 2014 году появился первый троянец-блокировщик для мобильных телефонов и планшетов, работающих под управлением операционной системы Google Android. Такие блокировщики попросту отключают сенсорный дисплей или выводят перекрывающее доступ к системе окно с требованием выкупа, которое невозможно удалить с экрана никакими стандартными методами. На сегодняшний день известно несколько десятков различных модификаций подобных мобильных троянцев, и их число постепенно растет.

Троянцы-шифровальщики (энкодеры)

Троянцы-шифровальщики, или *энкодеры*, — одна из наиболее опасных компьютерных угроз. Такие троянцы также относятся к условной категории *программ-вымогателей*. Энкодеры впервые появились в 2009 году, и на текущий момент специалистам компьютерной безопасности известно огромное количество их разновидностей.

Запустившись на инфицированном компьютере, энкодеры зашифровывают хранящиеся на дисках пользовательские файлы с использованием различных криптостойких алгоритмов, после чего требуют у жертвы выкуп за их расшифровку. Отказавшись выплачивать требуемое злоумышленниками денежное вознаграждение, пользователь рискует в одночасье потерять все свои данные. Аппетиты у злоумышленников могут быть различными: вымогаемый ими выкуп может составлять эквивалент и нескольких десятков, и нескольких тысяч долларов. К сожалению, в некоторых случаях восстановить поврежденные данные бывает практически невозможно, даже несмотря на обещания злоумышленников. Оплата требуемого киберпреступниками вознаграждения не дает никакой гарантии того, что поврежденные троянцем-энкодером файлы будут когда-либо расшифрованы. Подавляющее большинство троянцев данного типа ориентировано на устройства, работающие под управлением Microsoft Windows, но в 2014 году появились первые энкодеры и под Android, способные зашифровывать содержимое внутренней памяти мобильного устройства, а в ноябре 2015 года был обнаружен первый шифровальщик для Linux.

До недавнего времени троянцы-шифровальщики не обладали какими-либо механизмами саморепликации и проникали на атакуемый компьютер преимущественно в виде вложений в сообщения электронной почты, либо под видом каких-либо «полезных» приложений или игр. Иными словами, в подавляющем большинстве случаев потенциальные жертвы сами запускают вредоносную программу на своем ПК. Можно отметить следующие характерные пути распространения троянцев-энкодеров:

- вредоносные почтовые рассылки, содержащие во вложении самого троянца (например, под видом важного документа), либо троянца-загрузчика, выполняющего скачивание и запуск шифровальщика;
- загрузка троянца пользователем из Интернета под видом полезного приложения (различные кодеки, утилиты, игры, проигрыватели медиафайлов);
- непосредственный запуск троянца на инфицированном компьютере, к которому злоумышленники имеют несанкционированный удаленный доступ (благодаря наличию на таком ПК бэкдора или компрометации учетной записи пользователя операционной системы).

Однако в 2017 году случилось несколько массовых эпидемий, причинами которых стали энкодеры, способные распространяться по сети самостоятельно, то есть, обладающие функциями червя. Так, шифровальщики WannaCry и Petya использовали для своего распространения уязвимость в сетевом протоколе SMB операционных систем Windows и могли заражать компьютеры самостоятельно, без участия пользователя.

Энкодеры применяют около двух десятков различных алгоритмов шифрования пользовательских файлов. Определенные модификации троянцев — даже несколько поочередно, с целью затруднить последующую расшифровку информации. Широкому распространению подобных программ способствует то обстоятельство, что злоумышленники нередко выставляют на продажу на подпольных форумах исходные коды троянцев-шифровальщиков, помимо этого существуют специальные программы-конструкторы, с помощью которых вирусописатели могут создавать новые версии троянцев, даже не обладая знаниями в области криптографии и программирования. Еще один аргумент, свидетельствующий об опасности таких приложений, заключается в том, что некоторые версии подобных вредоносных программ содержат ошибки, поэтому расшифровать пострадавшие от их действия файлы иногда не в состоянии даже сами создатели троянца. После успешного запуска на компьютере жертвы энкодер в общем случае выполняет перечисленную ниже последовательность действий:

- автоматически генерирует по определенному алгоритму ключ шифрования либо получает его с принадлежащего злоумышленникам удаленного сервера;
- осуществляет поиск на дисках зараженного компьютера файлов, удовлетворяющих заданным вирусописателями критериям;
- шифрует все удовлетворяющие условиями файлы;
- создает и сохраняет на диске документы с перечислением действий для последующей расшифровки файлов и условиями выкупа.

После того как файлы оказываются зашифрованными, троянцы-энкодеры, в зависимости от версии, могут изменить фон рабочего стола атакованного компьютера на графическое изображение с указанием дальнейших инструкций для своей жертвы. Требуемая злоумышленниками сумма может варьироваться от десятков до нескольких тысяч долларов. В целях конспирации некоторые модификации шифровальщиков размещают свои управляющие серверы в анонимной сети TOR, что значительно затрудняет их идентификацию и последующую расшифровку данных.

Рассмотрим принцип работы троянца-шифровальщика на примере нашумевшего энкодера Troldesh. В самом начале марта 2019 года компания Group-IB сообщила об очередной массированной атаке с использованием этого трояна. Энкодер распространялся с помощью сообщений электронной почты, отправленных от имени известных на российском рынке компаний.

Театр, как известно, начинается с длительного поиска свободной парковки, а распространение большинства вредоносных программ — с почтовой рассылки. Среди поддельных отправителей подобных писем исследователи из Group-IB отмечали «Всероссийский банк развития регионов» и «Группу компаний ПИК», а журналисты РБК пополнили список пострадавших брендов «Ашаном», «Магнитом» и «Славнефтью», упомянув при этом, что злоумышленники прикр ывались именами более пятидесяти фирм, названия которых у всех на слуху. Лично мне попадались аналогичные письма, в строке *From* которых значился автомобильный дилер «Рольф», «Бинбанк», «KIA Motors», «Ресо Гарантия» и некая контора «Инженер Строй» (рис. 2).

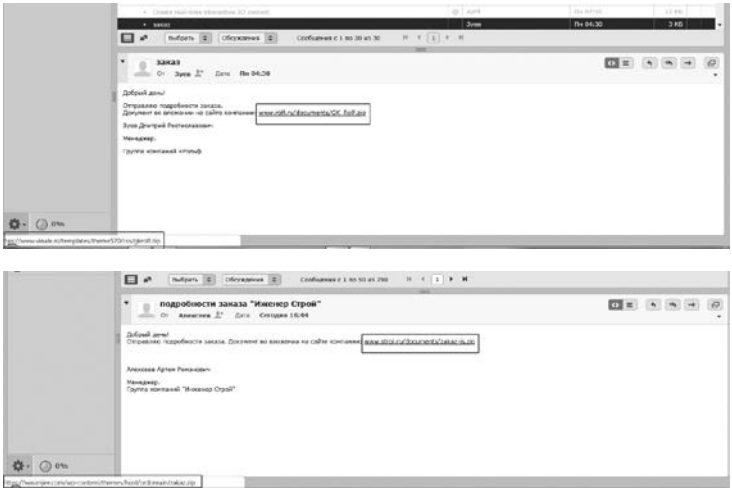


Рис. 2. Письма, рассылавшиеся в ходе атаки Troldesh

Очевидно, такая стратегия продиктована общеизвестными постулатами социальной инженерии: у сообщения от известного отправителя больше шансов быть открытым, чем у послания, автор которого адресату совершенно незнаком. Во всех без исключения изученных мной письмах ссылки вели на взломанные сайты, работающие под управлением WordPress или Joomla. И скрипт-загрузчик, и собственно энкодер злодеи хранили в открытых на запись папках *<site>/content/icons/*, *<site>/templates/*, *<site>/wp-admin/css/* или *<site>/wp-content/themes/*. Вероятно, чтобы размещать вредоносное содержимое, они приобрели на каком-то из хакерских форумов базу скомпрометированных аккаунтов либо воспользовались одним из известных эксплоитов для этих популярных CMS.

При нажатии на линк с удаленного хоста скачивался ZIP-архив, внутри которого хранился скрипт-загрузчик, написанный на JScript. Чтобы скрипт выполнился, пользователь должен самостоятельно извлечь из архива и попытаться открыть этот файл. Загрузчик представляет собой обфусцированный и зашифрованный файл JSE размером 5,7 Кбайт. В случае с сообщением, отправленным якобы от имени компании «Рольф», скрипт имел имя *Группа компаний Рольф подробности.jse*,

при этом злодеи даже не попытались подменить стандартный значок сценария, что опытного пользователя должно хотя бы немного насторожить.

Принцип действия загрузчика в целом стандартен для подобных вредоносов. Для выполнения кода на языке JScript в Windows используется приложение *wscript.exe*, именуемое Windows Script Host. Запустившись с помощью этого приложения, сценарий расшифровывает строки, в которых хранится URL сайта с полезной нагрузкой, скачивает файл шифровальщика в *%TEMP%* и запускает его в скрытом окне. В качестве «дембельского аккорда» загрузчик открывает свернутое окошко cmd и отправляет туда команду на самоудаление.

Исследователи из Group-IB в своей публикации сообщили, что Troldesh имеет «творческие псевдонимы»: Shade, XTBL и Trojan.Encoder.858. ESET детектирует его под именем Win32/Kryptik, «Антивирус Касперского» — как Trojan-Ransom.Win32. Shade.psq. Что нам известно об этом энкодере? Первые случаи заражения Trojan.Encoder.858 датируются еще 2015 годом, при этом вредонос, скорее всего, является потомком другого шифровальщика — Trojan.Encoder.686, распространение которого началось на год раньше, в июле 2014-го. 686-я модификация, названная его создателями CTB-Locker, успешно продавалась на одном из хакерских форумов. Троянец активно использует возможности OpenSSL и эллиптическую криптографию, а для генерации случайных данных применяет CryptoAPI. Зашифрованные файлы получали расширение *.ctbl*.

Файл шифровальщика, о котором идет речь в публикации Group-IB, имеет размер 1,05 Мбайт. Внутри хранится библиотека для работы с Tor. Энкодер упакован с использованием написанного на.NET кастомного пакера, под которым находится двоичный файл, сжатый UPX. Кроме того, строки в трояне зашифрованы с использованием симметричного алгоритма AES.

После запуска и инициализации шифровальщик создает свою копию в папке *%ALLUSERSPROFILE%\application data\windows* под именем *csrss.exe*. Затем он регистрируется в автозагрузке. Он ищет ветку реестра *[<HKCU>\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\]* и записывает в нее значение *'Client Server Runtime Subsystem' = '"%ALLUSERSPROFILE%\Application Data\Windows\csrss.exe»'*.

На следующем этапе энкодер собирает информацию об инфицированной машине, а именно определяет версию операционной системы, серийный номер дискового раздела, из которого запущен исполняемый файл, имя компьютера и тип установленного на нем процессора. На основе этих данных формируется уникальный для каждой инфицированной машины ключ. Этот ключ сохраняется в ветви реестра *[<HKLM|HKCU>\SOFTWARE\System32\Configuration\]* в параметре *i*. Туда же записывается версия вредоносной программы.

Затем троянец инициализирует Tor-клиент и пытается соединиться с одним из бридж-релеев (bridge relay), адрес которого в зашифрованном виде хранится в его теле. По этому адресу шифровальщик отправляет запрос на регистрацию, содержащий его ID, а в ответ получает данные, необходимые для шифрования файлов на зараженной машине. В том числе — ключ RSA длиной 2048 бит и его MD5-хеш для проверки.

Файлы на всех локальных и подключенных к машине сетевых дисках шифруются с использованием алгоритма AES (Advanced Encryption Standard) в режиме CBC (Cipher Block Chaining), при этом для каждого файла создается отдельный ключ при помощи генератора псевдослучайных чисел. Этот ключ шифруется полученным через Tor ключом RSA и сохраняется в зашифрованном файле. Для файлов, имеющих атрибут Read Only, перед шифрованием указанный атрибут сбрасывается. На прощание троян удаляет все точки восстановления системы.

Казалось бы, угроза давно известная и хорошо изученная, раз уж распространяется она как минимум четыре года, только вот есть одна загвоздка. Trojan.Encoder.858, о котором говорят авторы публикации Group-IB, присваивает зашифрованным файлам расширение *.xtbl*, а троянец, заражающий компьютеры в ходе рассматриваемой нами атаки, использует другое расширение — *.crypted000007*. Вывод: исследователи ошиблись. Это не Trojan.Encoder.858.

На самом деле речь, скорее всего, идет о более свежей модификации 858-го, которая в номенклатуре Dr.Web носит гордое наименование Trojan.Encoder.10507. Эта модификация энкодера датируется 2017 годом и почти не отличается от своего предшественника, но кое-какие нововведения все же имеются.

В теле вредоноса хранится 100 публичных ключей RSA длиной 3072 бита каждый. Перед началом шифрования энкодер случайным образом выбирает один из них, номер этого ключа сохраняется в текстовом файле с требованиями вымогателей. Каждый файл шифруется с отдельным ключом длиной 256 бит, его имя — другим ключом такой же длины, после чего оба сессионных ключа шифруются ранее выбранным публичным ключом RSA, а результат дописывается в конец зашифрованного файла. По завершении шифрования троянец создает на диске текстовые файлы README с порядковым номером от 1 до 10, в которых содержатся требования выкупа. Затем вредонос меняет обои рабочего стола Windows, и жертва вирусописателей наблюдает жизнерадостную картину, показанную на рис. 3.

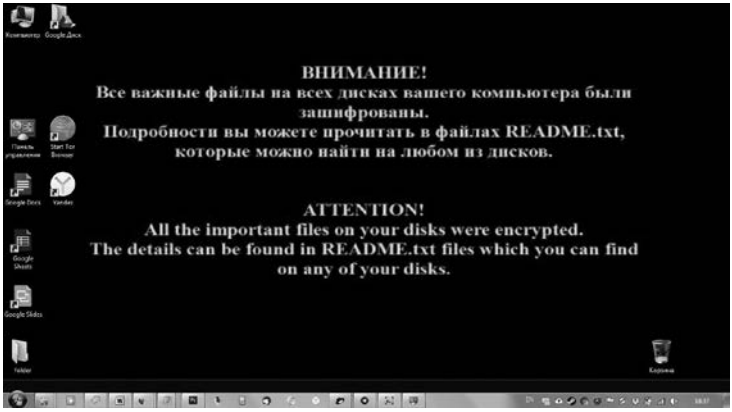


Рис. 3. Веселенькие обои рабочего стола, устанавливаемые шифровальщиком Troldesh

Все, файлы зашифрованы. Помимо прочего, в процессе своей работы троянец пытается прочитать файлы *%APPDATA%\thunderbird\profiles.ini* и *%APPDATA%\mozilla\firefox\profiles.ini*, вероятно, надеясь получить доступ к паролям и настройкам почтового клиента пользователя, чтобы обеспечить свое дальнейшее распространение.

На сегодняшний день наиболее эффективным методом противодействия троянцам-энкодерам является использование современного антивирусного ПО, обладающего механизмами

превентивной защиты, и своевременное *резервное копирование* всей актуальной информации на независимые носители, хранящиеся отдельно от основного компьютера пользователя — в качестве таковых могут, в том числе, выступать отключаемые облачные хранилища.

Банковские троянцы

Банковские троянцы предназначены для кражи денег со счетов жертвы, пользующейся системами *дистанционного банковского обслуживания* — то есть системами «банк — клиент», или другими электронными платежными инструментами. Сегодня практически любой банк оказывает своим клиентам услуги дистанционного банковского обслуживания (ДБО): как правило, для частных лиц они сводятся к предоставлению доступа на специальный защищенный сайт, где клиент может проверить состояние своего счета, перевести средства с одного счета на другой, а также оплатить ряд услуг в организациях, с которыми у банка имеется соответствующий договор. В простейших случаях доступ к подобным системам осуществляется по протоколу HTTPS с использованием логина и пароля. Также в качестве меры безопасности некоторые банки применяют подтверждение входа по СМС или с использованием электронной цифровой подписи клиента. Порой для организации связи применяется специализированное программное обеспечение.

Безусловно, использование ДБО крайне удобно для большинства клиентов: не нужно тратить время на поездки в офис банка или поход до ближайшего банкомата, чтобы уточнить баланс счета — простые операции с безналичными средствами выполняются одним щелчком мыши. Однако подобные системы неизбежно привлекают внимание злоумышленников.

Пик распространения так называемых *банковских троянцев* — вредоносных программ, предназначенных для кражи учетных данных и необходимых для организации доступа к системам ДБО файлов — пришелся на 2011 год, однако инциденты, связанные с хищением денежных средств держателей банковских счетов при помощи таких вредоносных программ, происходят с завидной регулярностью и сегодня. К категории наиболее опасных можно отнести сразу несколько банковских троянцев — это

Trojan.Carberp, Trojan.PWS.Ibank, Trojan.PWS.Panda (также известная под именами Zeus и Zbot) и Trojan.PWS.SpySweep (также известный под именем SpyEye).

Запускаясь на инфицированной машине, банковский троянец предпринимает целый ряд действий для того, чтобы уйти от всевозможных средств контроля и наблюдения. Так, например, Trojan.Carberp внедряется в другие работающие приложения, а свой основной процесс завершает; таким образом, вся дальнейшая работа происходит частями внутри сторонних приложений, что является его характерным свойством и затрудняет лечение заражения. Среди команд, которые способен выполнять Trojan.Carberp, имеются директивы запуска произвольных файлов на инфицированном компьютере, команда установки сеанса «удаленного рабочего стола» по протоколу RDP, и даже удаления на зараженном ПК операционной системы. Кроме того, в этом троянце скрыта возможность деактивации антивирусов, уничтожения «конкурирующих» банковских троянцев, а также кража паролей от множества различных программ: браузеров, мессенджеров, FTP-клиентов, почтовых программ и т. д. Помимо прочего, благодаря расширяемой архитектуре данная троянская программа имеет возможность скачивать специальные встраиваемые дополнения (плагины) для выполнения других деструктивных действий.

Для перехвата связанной с работой ДБО информации Trojan.Carberp использует различные методы: это логирование нажатий пользователем клавиш, вклинивание в HTTP-трафик в поисках учетных данных и передаваемых значений экранных форм, встраивание в процессы программ системы «банк — клиент», создание снимков экрана в моменты ввода важной информации, перехваты отдельных функций, которые могут участвовать в передаче данных, поиск и похищение цифровых сертификатов и ключей. Вредоносные программы семейства Trojan.Carberp способны объединяться в ботнеты, координируемые из одного (или нескольких) командных центров.

Троянцы семейства Trojan.PWS.Ibank также позволяет выполнять на инфицированном компьютере поступающие от удаленного командного центра директивы, включая команду уничтожения ОС. Причем адрес командного сервера троянец вычисляет динамически, используя для этого

специальный алгоритм. Для перехвата важной информации используются анализатор трафика, система мониторинга сетевой активности банк-клиентов, утилита для снятия снимков экрана, сборщики ключевой информации для различных систем банк-клиент. Основное назначение троянца — перехват данных, вводимых в экранные формы, сбор файлов сертификатов систем защищенного документооборота, а также перехват сетевого трафика и направление полученной информации злоумышленникам.

Нельзя обойти вниманием такого знаменитого троянца, как Trojan.PWS.Panda, известного также под именами Zeus и Zbot. Основной его функционал заключается в краже пользовательских паролей, хотя этот троянец обладает достаточно обширными возможностями, включая стандартный функционал бэкдора. Созданный еще в 2007 году, Zeus представляет определенную опасность для пользователей и по сей день. Этот троянец способен работать во всех версиях Windows, он умеет перехватывать информацию, вводимую пользователем во всех распространенных на сегодняшний день браузерах, красть пароли большинства FTP-клиентов.

Среди иных вредоносных функций Zeus следует отметить способность устанавливать и удалять в инфицированной системе цифровые сертификаты, файлы cookies, подменять «домашнюю страницу» в браузерах, блокировать доступ к различным URL, загружать и запускать программы, по команде с удаленного сервера выключать и перезагружать компьютер, удалять на жестких дисках любые файлы. Иными словами, функционал этой вредоносной программы обширен.

Еще один троянец, созданный со схожими целями, — Trojan.PWS.SpySweep, также известный под именем SpyEye. Как и Trojan.Carberp, SpyEye внедряет свой образ в адресные пространства других процессов. Функционал его в целом схож с возможностями Zeus: SpyEye способен выполнять поступающие от злоумышленников команды, похищать конфиденциальную информацию, загружать и запускать на инфицированном компьютере различные приложения.

В 2011 году появился и первый банковский троянец для платформы Android: это Android.SpyEye.1. Риску заражения вредоносной программой Android.SpyEye.1 подвергались в первую

очередь пользователи, компьютеры которых уже инфицированы троянской программой SpyEye. При обращении к различным банковским сайтам, адреса которых присутствуют в конфигурационном файле троянца, в просматриваемую пользователем веб-страницу осуществлялась инъекция постороннего содержимого, которое может включать различный текст или веб-формы.

Таким образом, ничего не подозревающая жертва загружала в браузере настольного компьютера или ноутбука веб-страницу банка, в котором у нее открыт счет, и обнаруживала сообщение о том, что банком введены в действие новые меры безопасности, без соблюдения которых пользователь не сможет получить доступ к системе «банк — клиент», а также предложение загрузить на мобильный телефон специальную программу, якобы содержащую «электронные ключи безопасности», а в самом деле представляющую собой троянца. Некоторые банковские троянцы внедряют в веб-страницы банка целые подробные инструкции по установке таких мобильных приложений. В альтернативном варианте пользователю будет предложено ввести в специальное поле номер своего мобильного телефона, на который он получит СМС-сообщение со ссылкой для скачивания «необходимой» программы, являющейся в действительности вредоносной.

После загрузки и инсталляции на мобильном устройстве Android.SpyEye.1 перехватывает и отправляет злоумышленникам все входящие СМС-сообщения. Таким образом злоумышленники осуществляют эффективный обход так называемой *двухфакторной аутентификации* — способа проверки подлинности платежа при помощи передаваемых через СМС одноразовых паролей (mTAN-кодов).

В наши дни для обхода банковских систем двухфакторной аутентификации злоумышленники используют более «продвинутые» и гибко настраиваемые мобильные вредоносные приложения, например троянцев семейства Android.SmsSpy (некоторые из них также известны под именем Perkele).

Как видим, принцип работы и функциональные возможности большинства современных банковских троянцев в целом схожи, да и цели, которые преследуют вирусописатели, также практически одинаковы, различается только конкретная техническая реализация этих задач.

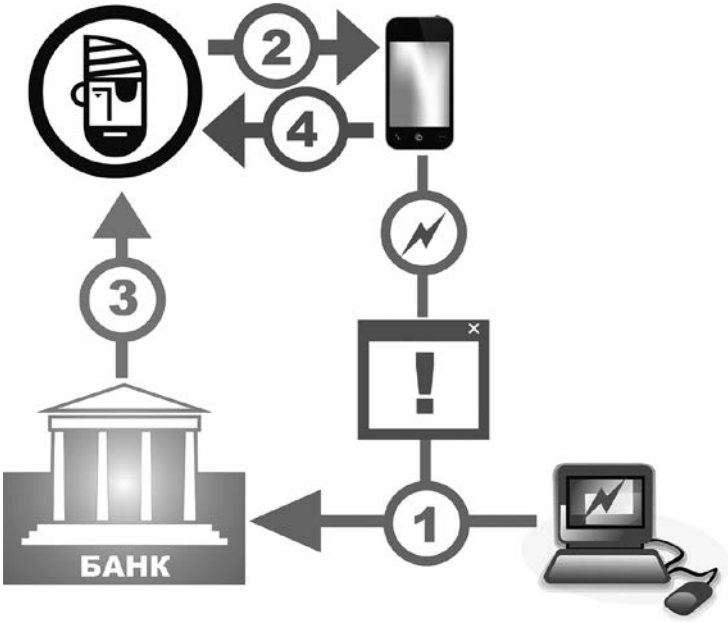


Рис. 4. Пример обхода банковской системы двухфакторной аутентификации с использованием одноразовых паролей (mTAN-кодов): 1 — в момент захода пользователя в систему «банк — клиент» действующий на его компьютере троянец требует для продолжения работы ввести во внедренное на веб-страницу банка поле номер мобильного телефона клиента; 2 — на мобильный телефон клиента приходит СМС со ссылкой на установку банковского троянца для Android; 3 — клиент вводит свои учетные данные в форму на странице банка, вся информация из которой передается троянцем злоумышленникам; 4 — мобильный троянец перехватывает поступающее на телефон СМС-сообщение с одноразовым паролем (mTAN-кодом) и также пересылает его злоумышленникам.

Отдельную категорию угроз составляют мобильные банковские троянцы для ОС Android, паразитирующие на программах класса «мобильный банк». Наиболее примитивные из них просто заменяют собой настоящее мобильное банковское приложение или «рисуют» поверх него собственную форму для ввода учетных данных, которые незамедлительно передаются злоумышленникам. Разумеется, мобильные банковские троянцы способны перехватывать и передавать киберпреступникам все входящие СМС-сообщения, в том

числе содержащие одноразовые пароли и коды авторизации. Более «продвинутые» троянцы данного типа умеют не просто красть аутентификационные данные, а похищать средства с расчетного счета пользователя, управляя транзакциями с использованием СМС-команд или интерфейса самого банковского приложения. При этом с точки зрения самого банка подобные транзакции будут выглядеть легитимными, поскольку осуществлялись они с телефона, номер которого «привязан» к счету клиента, а кроме того в процессе выполнения банковских операций была зафиксирована успешная авторизация мобильной банковской программы на сервере банка. Вернуть похищенные таким образом средства жертве киберпреступников становится впоследствии очень затруднительно. Подробнее о мобильных банкерах мы поговорим в следующей главе.

Веб-инжекты

Веб-инжекты — это не отдельный вид вредоносных программ, а специальная технология, используемая различными вирусами и троянскими программами для достижения своих целей, например банковскими троянцами и троянцами-вымогателями. Эта технология уже кратко упоминалась в предыдущем разделе при описании одного из способов обхода двухфакторной аутентификации в процессе хищения средств с банковских счетов пользователей.

Под веб-инжектом принято понимать встраивание вирусом или троянцем постороннего содержимого в просматриваемую пользователем в окне браузера веб-страницу. Отдельно следует упомянуть, что этот процесс осуществляется на стороне *пользователя*, то есть непосредственно на инфицированном компьютере работающей на нем вредоносной программой. Иными словами, жертва веб-инжекта наблюдает правильный URL — корректный адрес веб-страницы в адресной строке браузера, если соединение выполняется с использованием защищенного протокола HTTPS, это условие также соблюдается, а вот содержимое веб-страницы будет отличаться от оригинального. С какой целью вирусописатели добавляют в свои творения функционал для осуществления веб-инжектов?

Прежде всего этой технологией активно пользуются многочисленные банковские троянцы для реализации различных схем финансового мошенничества. Например, в веб-страницу сайта банка может быть встроена фальшивая форма для ввода логина и пароля (данные из которой впоследствии передаются злоумышленникам), сообщение со ссылкой о необходимости установить для входа в систему «банк — клиент» дополнительное приложение, «сертификат безопасности» или программу для мобильного телефона, под видом которых распространяется мобильное вредоносное ПО и т. д.

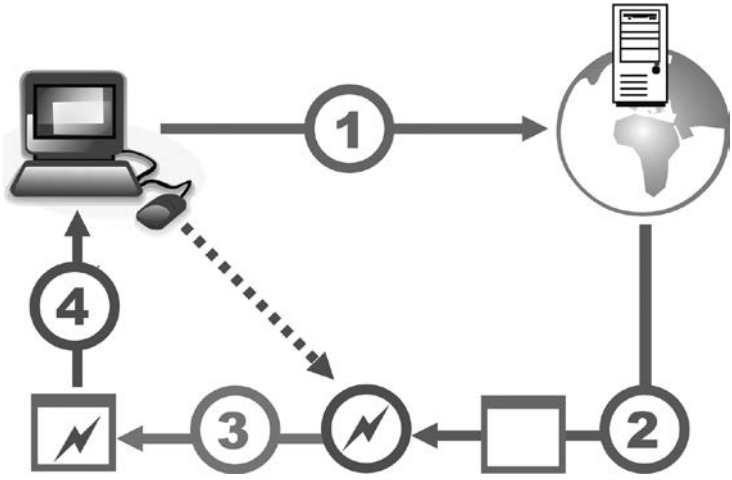


Рис. 5. Стандартная схема веб-инжекта: 1 — компьютер клиента запрашивает у сервера веб-страницу; 2 — сервер передает веб-страницу на локальный клиентский компьютер; 3 — заразивший этот компьютер троянец встраивает в веб-страницу постороннее содержимое; 4 — модифицированная веб-страница отображается в окне браузера.

Полученное браузером с сервера содержимое веб-страницы модифицируется троянцем перед загрузкой и демонстрацией ее в окне браузера и происходит для потенциальной жертвы совершенно незаметно. При этом отображаемый вредоносной программой контент гибко настраивается с помощью специального конфигурационного файла, который троянец может загрузить с принадлежащего злоумышленникам узла. Это позволяет

киберпреступникам, во-первых, оперативно менять оформление встраиваемых в веб-страницу элементов, включая содержащиеся в них изображения и текст, а также путем однократного внесения изменений на управляющем сервере настроить параметры веб-инжекта сразу на всех зараженных компьютерах.

С технической точки зрения наиболее ранние реализации механизма веб-инжектов были довольно примитивными: например, на страницах популярных интернет-магазинов троянец мог разместить поддельную форму для ввода реквизитов банковской карты, используя которые, злоумышленники впоследствии совершали несанкционированные держателем карты покупки (приобретенные таким способом дорогостоящие товары впоследствии сбывались различными способами, например, через системы онлайн-аукционов или электронных досок объявлений). Позже встраиваемое в веб-страницы содержимое стало использовать различные интерактивные элементы, в частности, сценарии на языке JavaScript или функции библиотеки jQuery для имитации действий пользователя, либо сокрытия последствий хищения.

Так, после проведения несанкционированной транзакции с использованием системы «банк — клиент» некоторые банковские троянцы в течение определенного времени демонстрировали пользователю прежнее значение баланса по счету, чтобы усыпить его бдительность. В другом, известном специалистам по информационной безопасности, случае троянец в момент входа в систему «банк — клиент» отображал на экране пользователя сообщение о том, что на его счет была ошибочно зачислена некая денежная сумма с просьбой вернуть ее отправителю на указанный номер счета. Одновременно с этим при помощи веб-инжекта вредоносная программа показывала информацию о состоянии счета, на котором действительно отражалась эта «лишняя» сумма — в самом деле указанная информация являлась, конечно, недостоверной. Жертва сама переводила злоумышленникам деньги, даже не подозревая об обмане.

Кроме банковских троянцев веб-инжектами пользуются и различные мошеннические вредоносные программы. Так, троянцы семейства Trojan.Mayachok применяют веб-инжекты совершенно в иных целях. При попытке владельца зараженного этим троянцем компьютера зайти на сайт какой-либо из популярных социальных сетей, Trojan.Mayachok демонстрирует

ему на экране сервисное сообщение якобы от имени данной социальной сети (оформленное в ее фирменном стиле) с информацией о том, что его личная страничка была взломана. Для восстановления доступа пользователю предлагается ввести в специальную форму номер мобильного телефона, а затем — пришедший в ответном СМС-сообщении код подтверждения. Обычно потенциальная жертва не задумываясь выполняет предложенные действия, поскольку входящие СМС у большинства операторов сотовой связи — бесплатные, и даже не подозревает о том, что сообщение в ее браузере сгенерировано коварным троянцем. Указав в предложенной форме код, пользователь соглашается с условиями платной подписки. Теперь с его мобильного счета будет ежедневно сниматься определенная денежная сумма до тех пор, пока прозревший владелец телефона не отменит навязанную тайком услугу, либо пока счет не опустеет.

Рис. 5. Веб-инжект, выполненный в веб-страницу социальной сети троянцем Trojan.Mayachok

Помимо широко известных социальных сетей троянцы данного типа успешно «охотились» и на пользователей других

популярных серверов — почтовых служб mail.ru, yandex.ru, видеохостинга youtube.com, сайтов знакомств и т. д.

Кроме сайтов популярных интернет-сервисов, некоторые модификации Trojan.Mayachok просто-напросто полностью блокировали своим жертвам доступ в Интернет из любых браузеров, выводя на экран сообщение якобы от имени обслуживающего пользователя интернет-провайдера примерно следующего содержания: «*Канал вашего района перегружен, и мы вынуждены ограничить загрузку некоторых сайтов на время. Если работа в сети Интернет на текущий момент для вас критична, вы можете подключить резервный канал вашего района. Чтобы подтвердить намерение и крайнюю необходимость перейти на резервный канал, введите ваш номер телефона и ответьте на входящее СМС-сообщение*». Результатом этой операции, как и в других аналогичных случаях, была потеря денег на счете мобильного телефона пользователя.

Еще одним примером нестандартного применения веб-инжекта может служить троянская программа Trojan.Mayachok.18831. Она не просто блокирует пользователю доступ к сайту социальной сети под предлогом «взлома», а видоизменяет его анкету прямо на локальном компьютере. Заглянув однажды на свою собственную страничку «В Контакте», пользователь с ужасом замечает, что вместо его фотографии в профиле социальной сети отображается снимок порнографического характера, а в списке «интересов» перечислены весьма нестандартные увлечения, включая склонность к нетрадиционным сексуальным связям. При попытке отредактировать профиль на экране появляется якобы сервисное окно социальной сети, в котором от жертвы требуется указать ее номер мобильного телефона и ввести проверочный код из ответного СМС.

Разумеется, все «признания» пользователя о его необычных эротических предпочтениях, равно как и непристойные снимки, в веб-страницу «В Контакте» встраивал троянец: если бы жертва зашла в социальную сеть с любого другого незараженного компьютера, она увидела бы свою обычную анкету без каких-либо визуальных изменений. Ну, а выполнив требования злоумышленников, пользователь незамедлительно терял средства на своем мобильном счете.

Троянцы-загрузчики

Создатели троянцев-загрузчиков зарабатывают установкой на инфицированный компьютер других вредоносных программ. Распространители вирусов платят им за «раздачу» ничего не подозревающим пользователям какого-то количества бэкдоров, банковских троянцев или энкодеров, после чего заказчики уже сами начинают зарабатывать на потенциальных жертвах. Вот почему после заражения троянцем-загрузчиком компьютер жертвы очень быстро может превратиться в настоящий «зоопарк», кишащий всевозможной вредоносной «живностью». Антивирусные программы в процессе сканирования порой «вылавливают» на таких зараженных машинах до нескольких сотен различных разновидностей вредоносных программ, включая кейлоггеры, бэкдоры, банковские троянцы и майнеры. А началось-то все с одного маленького и незаметного троянца-загрузчика.

Другие разновидности загрузчиков стараются не связываться с откровенно криминальным вредоносным ПО, а вместо этого скачивают и инсталлируют на зараженную машину различные сомнительные приложения, за установку которых им платят вознаграждение многочисленные проекты, гордо именующие себя «партнерскими программами по монетизации файлового трафика». В самом же деле все эти «партнерские программы» являются обычными жульническими схемами по несанкционированному «впариванию» пользователям всяких «ускорителей Интернета», «очистителей реестра», «помощников по поиску», «улучшателей графики» и прочего программного хлама. Подцепив где-нибудь троянца-загрузчика, жертва сетевых жуликов однажды обнаруживает, что ее компьютер полностью замусорен всякими рекламными баннерами, десятком непонятных браузеров, кучей «оптимизаторов» и прочих программ, значительно замедляющих работу операционной системы. Причем их удаление стандартными средствами ОС обычно не помогает: через пару дней весь этот зверинец возвращается на свои места, даже заметно прибавив в количестве и ассортименте. Единственным эффективным выходом из подобной ситуации является поиск и уничтожение изначального источника проблемы: прячущегося где-то на диске троянца-загрузчика.

Майнеры

Первым в истории электронным платежным средством, которое можно отнести к категории криптовалют, стала виртуальная денежная единица Bitcoin, изобретенная криптографом Сатоси Накамото в 2009 году. Для защиты транзакций в системе Bitcoin используются различные криптоалгоритмы, а сама система является децентрализованной — иными словами, в ней отсутствуют какие-либо обособленные управляющие или процессинговые центры. Особенность подобных платежных систем заключается в том, что объем имеющей в них хождение виртуальной валюты строго ограничен, причем пользователи могут не только «обмениваться» электронными «монетами», покупая на них всевозможные товары, но и добывать их, как старатели в реальном мире добывают золото, превращающееся потом в платежное средство. Для этого компьютер с помощью специальной программы должен выполнить ряд очень сложных математических вычислений. Этот процесс называется «*майнингом*» (от англ. mining, «добыча»).

Собственно, троянцы-майнеры, заразив компьютер, и нагружают его под завязку подобными расчетами, заставляя систему поминутно «тормозить» и «зависать». Ну, а все добытые таким образом деньги отправляются, естественно, не владельцу инфицированного устройства, а жулику-вирусописателю.

Существуют троянцы-майнеры не только для ОС Windows, но также для macOS и Google Android. И хотя на первый взгляд они и не представляют серьезной опасности для инфицированной системы, деятельность майнеров заметно замедляет скорость работы компьютера или мобильного устройства, а также вызывает перегрев процессора или чипа видеокарты (существует категория майнеров, использующих для вычислений мощности графических процессоров современных видеоадаптеров). В случае физических неполадок в системе охлаждения компьютера это может привести к фатальным последствиям вплоть до выхода из строя соответствующего оборудования.

Во второй половине 2017 и в 2018 году майнеры приобрели особую популярность у вирусописателей, заметно потеснив на подпольном рынке даже такие опасные вредоносные программы, как шифровальщики. Расширился и ассортимент

используемых киберпреступниками технологий: появились майнеры, реализованные с использованием скриптовых языков, таких как JavaScript. Достаточно внедрить подобный сценарий в веб-страницу, и при открытии ее в браузере компьютер пользователя начинает автоматически добывать криптовалюту, расходуя на это свои аппаратные ресурсы. А если запустить скрипт в свернутом, или даже в скрытом окне, жертва с высокой долей вероятности не заметит ничего подозрительного. Подобные сценарии активно внедрялись в веб-страницы не только самими владельцами сайтов. Чаще появление таких скриптов становилось результатом взлома интернет-ресурса. Причем задачу взломщикам зачастую облегчали сами администраторы сайтов, устанавливая в системе управления контентом (CMS, Content Management System) бесплатные либо видоизмененные коммерческие шаблоны оформления, а также компоненты, содержавшие различные «закладки». Были зафиксированы случаи распространения скриптов-майнеров и через автоматизированные рекламные сети, позволяющие обмениваться объявлениями и ссылками. А в последние годы стали широко известны троянцы-майнеры, атакующие Интернет вещей — различные «умные» устройства вроде сетевых хранилищ, телеприставок и роутеров. Здесь вирусописатели чувствуют себя вольготно: вряд ли простому пользователю придет в голову, что его кофеварка в свободное от основной работы время трудится на злоумышленников, добывая им криптовалюту. Троянцам для интернета вещей посвящена пятая глава.

Клиперы

Эта разновидность вредоносных программ, известная довольно давно, но получившая «вторую жизнь» в эпоху популярности криптовалют и электронных платежных систем, не имеет никакого отношения к парусным кораблям. Свое название троянцы-клиперы получили от английского термина «*clipboard*» — «буфер обмена». Собственно, в этом наименовании и раскрывается их предназначение: клиперы отслеживают состояние буфера обмена и похищают скопированную туда информацию. В частности, они могут подменять помещенные в буфер обмена

имена кошельков криптовалют и платежных систем на данные, заранее подготовленные злоумышленниками. В результате этого жертва, желающая перевести кому-нибудь денежные средства, сама того не подозревая отправляет платеж напрямую киберпреступникам. К слову, в 2018 году специалистами по информационный безопасности был обнаружен первый троянец-клипер для мобильной платформы Android.

Стилеры

Троянцев-стилеров (от англ. Stealer — «вор») можно отнести к категории шпионского ПО. Как это следует из их наименования, стилеры предназначены для кражи на инфицированном компьютере различной информации. Прежде всего, это файлы cookies, реквизиты банковских карт, сохраненные пароли и данные для автоматического заполнения форм в браузерах, а также файлы, в которых хранится информация учетных записей от различных приложений — мессенджеров, FTP-клиентов, клиентов электронной почты. Многие стилеры целенаправленно передают злоумышленникам все электронные документы, хранящиеся на Рабочем столе Windows.

Троянцы для любителей игр

В 2017-2018 годах высокую популярность среди вирусописателей обрели стилеры, ворующие файлы из клиентов игровой платформы Steam. Steam — это система цифровой дистрибуции, созданная компанией Valve Corporation. Она предназначена для распространения компьютерных игр и программ. Пользователи этой платформы имеют доступ к личному кабинету, в котором собрана информация обо всех приобретенных ими ранее играх и программах. Помимо этого они могут совершать покупки в специальном магазине Steam, где предлагается различный цифровой контент, а также продавать и обмениваться игровыми предметами с другими пользователями.

Ряд современных многопользовательских электронных игр представляет собой настоящие виртуальные миры с собственными социальными и экономическими законами. В таких вселенных игрок может приобретать различные виртуальные

предметы и ресурсы — например доспехи, магические заклинания или дополнительную броню, — которые меняют внешний вид игрока или дают ему ряд тактических преимуществ перед другими игроками. Причем многие подобные артефакты можно купить за реальные деньги (на чем и зарабатывают некоторые издатели компьютерных игр, делая саму игру бесплатной), а ставшие ненужными игровые предметы — продать другим пользователям.

Именно поэтому игровые предметы имеют среди игроков высокую ценность: некоторые артефакты можно продать за сотни и тысячи вполне реальных американских долларов. Сегодня существует обширный подпольный рынок купли-продажи краденых игровых предметов, большая часть которых была похищена с использованием троянцев.

Так, летом 2014 года у пользователей игры CS:GO стал таинственным образом пропадать игровой инвентарь, о чем они писали встревоженные сообщения на Reddit. Непосредственно перед самим инцидентом игрок получал в чате Steam сообщение от другого пользователя с предложением обменяться виртуальными предметами. Послание содержало скриншот предлагаемого к обмену инвентаря, при этом сама сделка выглядела достаточно выгодной. После успешного завершения операции пользователь заходил в игру и с удивлением обнаруживал, что часть его наиболее ценного имущества исчезла в неизвестном направлении.

Благодаря проведенному аналитиками расследованию удалось установить первопричину «трагедии». Ею оказался троян SteamBurglar. Пока ничего не подозревающий игрок разглядывал в окне чата дорогой предмет, предложенный ему для обмена на какую-нибудь посредственную безделушку, троянец находил в памяти компьютера процесс Steam и вытаскивал из него информацию об имеющейся в арсенале пользователя амуниции. Затем по этому списку выполнялся поиск с использованием ключевых слов rare, mythical, immortal, legendary, arcana и key (список ключевиков можно настраивать в административной панели трояна) — таким образом SteamBurglar выбирал наиболее ценный инвентарь. Найденное барахло троян тут же выставлял на продажу через Steam по весьма выгодной цене. Вырученные деньги поступали на счет вирусописателя.

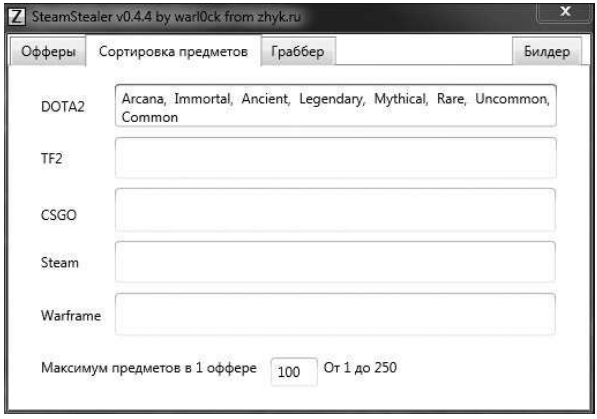


Рис. 7. Так выглядит билдер — программа для конструирования троянца-стилера SteamBurglar

Сам троян и билдеры (программы для его изготовления) успешно предлагались на читерских форумах, причем трой позволял воровать предметы не только из CS:GO, но и из других игрушек: Dota 2, Team Fortress 2, Warframe. Для рассылки сообщений пользователи SteamBurglar применяли сторонние инструменты, но в декабре 2014 автор выкатил обновление трояна, позволявшее спамить в чаты прямо из приложения-админки. В ответ на возмущенные сообщения пострадавших игроков администрация Steam поначалу отмалчивалась, предлагая обокраденным юзерам самостоятельно искать на страницах маркета аккаунты злодеев и жаловаться на них в техподдержку. Однако под давлением общественности они все-таки изменили процедуру продажи игровых предметов, после чего для совершения подобных сделок потребовалось обязательное подтверждение по электронной почте.

Осенью того же года по Сети начал разгуливать новый троянец, SteamLogger.1, с тем же самым функциональным назначением — кража предметов у игроков Dota 2, CS:GO и Team Fortress 2. Но устроен он был гораздо более замысловато.

Дроппер трояна распространялся с помощью ссылок на читерских сайтах, в социальных сетях и в личных сообщениях. Потенциальной жертве предлагалось купить по дешевке или обменять игровой инвентарь, а подробности сделки она должна была получить по ссылке, при нажатии на которую на компьютер

скачивался дроппер троянца. Внутри дроппера в зашифрованном виде хранился сам троян и его сервисный модуль. При запуске исполняемого файла образ дроппера загружался в память, его содержимое расшифровывалось и сохранялось на диск: сервисный модуль в папку *%TEMP%* под именем *update.exe*, а тело трояна подгружалось в память с помощью метода *Assembly.Load ()*. Сразу же после этого SteamLogger.1 скачивал с управляющего сервера и показывал на экране картинку с изображением якобы предлагаемого к продаже товара, чтобы усыпить бдительность жертвы.

Рис.8. Вот такую картинку показывал игроку троянец SteamLogger.1

Дальше к работе подключался сервисный модуль. Он искал в папке *ProgramFiles (x86)\Common Files* подпапку с именем *Steam* (если не находил — создавал ее), сохранял в нее файл *SteamService.exe*, присваивал ему атрибуты «системный» и «скрытый», после чего запускал его, предварительно зарегистрировав это приложение в отвечающей за автозагрузку ветви реестра. Собрав информацию о зараженной машине (включая серийный номер системного раздела, версию и разрядность ОС), сервисный модуль отсылал ее на управляющий сервер. При этом использовались прокси, адреса которых хранятся в самой программе. Основное предназначение сервисного модуля — обновление трояна.

Основной модуль SteamLogger.1 висит в памяти зараженной машины, внимательно отслеживает состояние процесса игрового клиента и ждет, пока пользователь авторизуется в Steam. Как только это произойдет, троян перехватывает используемые для входа в учетную запись данные, определяет, используются ли защитные механизмы SteamGuard, steam-id, security token, и передает все эти сведения на управляющий сервер. В ответ он получает список аккаунтов, на которые можно передать украденные у жертвы игровые предметы, и необходимые для совершения «сделки» параметры. Затем троян ищет в папке steam-клиента файлы, в именах которых содержится строка *ssfn**, собирает содержимое подпапки *config*, после чего формирует из полученных файлов большой массив, дописывает в его конец данные об аккаунте жертвы и шифрует все это с помощью Base64. Результат отсылается на управляющий сервер. Наконец, SteamLogger.1 проверяет, включена ли в клиенте Steam функция автоматического входа в аккаунт, и, если нет, запускает кейлоггер, который записывает и передает злодеям коды нажимаемых на зараженной машине клавиш. Любопытно, что кейлоггер не сохраняет результат своей работы в файл на локальной машине, а формирует специальный POST-запрос и передает его на управляющий сервер с интервалом в пятнадцать секунд. Этот запрос обрабатывается и логируется уже на стороне сервера.

Предметы, которые троянец планирует украсть, он ищет в инвентаре жертвы по ключевым словам Mythical, Legendary, Arcana, Immortal, Container и Supply Crate. При этом SteamLogger.1 проверяет, не выставил ли сам пользователь что-либо из списка на продажу, и, если это так, снимает с продажи интересующий его предмет. После чего все найденные предметы передаются на один из аккаунтов Steam, реквизиты которых троян получил ранее с управляющего сервера. Для перепродажи краденого киберпреступники создали несколько интернет-магазинов.

С тех пор новые вредоносы, предназначенные для угона аккаунтов Steam и различного игрового инвентаря, стали появляться регулярно. Распространению способствовало и появление троянцев, продававшихся как услуга — по принципу *malware as a service*. Несколько таких стилеров активно распространялись летом 2018 года. Стоило пользователю зараженной машины выставить

для обмена какой-либо игровой предмет на одной из предназначенных для этого площадок, такой троянец дожидался запроса от желающего обменять артефакт пользователя, отклонял его, а затем использовал аватар и ник игрока, чтобы направить жертве аналогичное предложение, но уже от имени учетной записи злоумышленника. При обмене инвентаря на официальном портале steamcommunity.com троян с помощью веб-инжекта менял изображения игровых предметов. Игроку казалось, что он приобретает дорогой и очень ценный артефакт, в то время как на самом деле он получал дешевую «безделушку».

Подводя итог, можно сказать, что весь существующий ныне ассортимент «игровых троянцев» условно делится на несколько категорий. Наиболее простые из них крадут файлы из клиента Steam либо воруют учетные данные пользователя — для этого применяется кейлоггинг и поддельные формы авторизации. Продвинутая малварь использует анализаторы трафика и веб-инжекты для перехвата критичных параметров безопасности и подмены игровых предметов при совершении онлайновых сделок обмена или купли-продажи. А в будущем вирусописатели наверняка придумают какие-нибудь новые методы отъема ценного виртуального имущества у любителей игр: там, где речь идет о деньгах, без этого не обходится никогда.

Фишинг

Весьма распространенным термином «*фишинг*» (от англ. phishing, производное от fishing — «выуживание») называется несанкционированное получение учетных данных пользователя для доступа к какой-либо приватной информации, например логинов и паролей.

Зачастую сетевые мошенники создают поддельные веб-сайты, имитирующие своим оформлением страницы популярных социальных сетей или почтовых сервисов, и всеми силами завлекают туда посетителей, в частности, с использованием так называемых методов *социальной инженерии* — обмана пользователя без применения сложных технических средств. Так, жулики рассылают по электронной почте сообщения якобы от имени различных сетевых служб знакомств, публикуют тизерную рекламу, имитирующую всплывающее сообщение социальной

сети о добавлении нового контакта и т. д. Таким образом жертва попадает на поддельный сайт, URL которого мошенники также стремятся сделать похожим на оригинальный — например, «odonlkassinki.ru» вместо «odnoklassniki.ru». Далеко не все замечают, что ссылка отличается от «оригинала» одним или несколькими символами.

Вероятно, читатель уже догадался, что нажатие на такую ссылку неизбежно приведет пользователя на поддельный сайт, внешне копирующий оформление одной из популярных социальных сетей, но фактически таковой не являющийся. Достаточно ввести в соответствующие поля формы авторизации свой логин и пароль, чтобы ими завладели злоумышленники. После этого от имени скомпрометированной учетной записи в социальной сети могут распространяться как безобидные рекламные сообщения, так и вредоносные программы. Распространены ситуации, когда мошенники, завладев чужой учетной записью, просят у знакомых «одолжить» им некоторую сумму, пополнив счет мобильного телефона злоумышленника. Шансы на то, что пользователь «купится» на такое сообщение, достаточно высоки: ведь с психологической точки зрения степень доверия к информации, полученной от одного из известных пользователю контактов, значительно выше, чем к сообщению, пришедшему от постороннего человека.

Однако одними лишь методами социальной инженерии киберпреступники не ограничиваются: в ход идут и вредоносные программы, предназначенные для фишинга — яркими их представителями является, в частности, семейство Trojan.Hosts. В операционных системах семейства Microsoft Windows для получения IP-адреса интернет-ресурса по введенному пользователем в адресную строку браузера URL используются DNS-серверы, однако перед отправкой запроса на них система обращается к локальному файлу *%SYSTEMROOT%\system32\drivers\etc\hosts*, в котором некоторыми приложениями сохраняются записи соответствия определенным URL IP-адресов в целях ускорения доступа к ним. Обосновавшись в операционной системе, Trojan.Hosts помещает в этот файл строки, подменяющие адреса сайтов популярных социальных сетей и поисковых систем на IP-адреса принадлежащих злоумышленникам веб-страниц. Таким образом, набрав в адресной строке браузера, например, «vk.com»,

потенциальная жертва попадает на сайт жуликов, имитирующий данную социальную сеть. Введенные на этой страничке логин и пароль тут же передаются киберпреступникам.

Другое семейство троянцев подобного типа — Trojan.DnsChange — подменяет корректные адреса DNS-серверов в сетевых настройках Windows на IP-адреса принадлежащих злоумышленникам DNS-серверов. Таким образом, в ответ на запросы, направляемые операционной системой на этот сервер, компьютер жертвы будет получать IP-адреса мошеннических интернет-ресурсов, в том числе страниц, с которых распространяется другое вредоносное ПО.

Случается и так, что программа-вредитель сама ворует логин и пароль для входа в социальную сеть, не заставляя пользователя вводить их вручную. Примерно так действовал наделавший много шума червь Win32.HLLW.AntiDurov, эпидемия которого разразилась в российском сегменте Интернета в 2008 году. Пользователи социальной сети «В Контакте» получали от друзей ссылку на забавную картинку deti.jpg, при открытии которой на компьютер загружался вредоносный файл *deti.scr*. Вслед за этим червь прописывался под именем *Vkontaktesvc.exe* в папку *Application Data* текущего пользователя и запускался в качестве фоновой службы с именем «Durov VKontakte Service». Затем червь анализировал хранящиеся на зараженном компьютере файлы cookies в поисках учетных данных для входа в социальную сеть «В Контакте», и если таковые обнаруживались, по списку «друзей» жертвы рассылалось сообщение со ссылкой на тот же рисунок, загружающий вредоносный файл. Деструктивная функция червя заключалась в том, что 25 числа каждого месяца в 10 часов утра он выводил на экран компьютера сообщение: «*Работая с ВКонтакте.РУ Вы ни разу не повышали свой рейтинг и поэтому мы не получили от Вас прибыли. За это Ваш компьютер будет уничтожен!*» и начинал удалять файлы с жесткого диска. Этот пример показывает, насколько опасными могут быть вредоносные программы, распространяющиеся через социальные сети.

Рекламные троянцы

Про рекламных троянцев я уже рассказывал в предыдущей главе, здесь стоит лишь кратко упомянуть о них, как об одном

из наиболее распространенных типов угроз. Рекламные троянцы позволяют своим создателям зарабатывать деньги за счет рекламодателей, заставляя пользователей просматривать во время путешествий по Интернету назойливые объявления, которые выпрыгивают на экран, как чертики из табакерки, в совершенно неожиданных местах и мешают нормальному просмотру веб-страниц. Опасность таких троянцев заключается еще и в том, что рекламируют они, как правило, всевозможные финансовые пирамиды, «лохотроны», опасные для здоровья диеты, несертифицированные медицинские препараты и биологически активные добавки, а также другие товары и услуги сомнительного толка. Некоторые такие троянцы «умеют» подменять стартовую страницу браузера, самостоятельно открывают новые вкладки при просмотре веб-страниц, а также изменяют выдачу в поисковых системах, «подсовывая» пользователю вместо ссылок с результатами обработки поискового запроса в Google, «Яндексе» и других поисковиках ссылки на всевозможные мошеннические ресурсы.

Распространяются рекламные троянцы с использованием так называемых *партнерских программ* — специальных сайтов, позволяющих недобросовестным создателям веб-страниц зарабатывать деньги на посетителях своих ресурсов: чем больше пользователей установит такую вредоносную программу на свой компьютер, тем большее вознаграждение получит участник «партнерской программы». Поэтому вредоносную программу предлагают скачать под видом различных игр, «полезных» утилит, музыкальных записей, бесплатных электронных книг и т. д. Как правило, оформлены они либо в виде отдельного приложения, либо в виде надстроек (плагинов) к популярным браузерам.

Узкоспециализированные вредоносные программы

Помимо троянцев, использующих «традиционные» схемы отъема денег у пользователей, существуют узкоспециализированные вредоносные программы, созданные для осуществления так называемых *таргетированных атак*, то есть для реализации некой конкретной и четкой цели. Например, BackDoor.Dande, проникнув на компьютер, в первую очередь проверял, не установлена ли на нем специальная программа, предназначенная

исключительно для заказа на оптовом складе лекарств и используемая в основном аптеками, больницами и другими медицинскими учреждениями. Если такая программа отсутствовала, троянец самоуничтожался, не причиняя никакого вреда. Если же она неожиданно обнаруживалась, бэкдор аккуратно собирал все сведения о совершенных пользователем этого компьютера закупках медикаментов и отправлял их на сервер своих разработчиков. Предположительно эта вредоносная программа была создана по заказу какой-то крупной фармацевтической компании для оценки рынка сбыта лекарств, потребительского спроса и слежки за конкурентами.

Другой троянец, являющийся характерным примером реализации технологий промышленного шпионажа, заражал только те компьютеры, на которых была установлена инженерная система для разработки чертежей AutoCad. Все копии созданных и обнаруженных на атакованном компьютере чертежей троянец упаковывал в архив и незамедлительно отправлял на сервер, расположенный в Китае. А ведь многие удивляются: как китайцам удается создавать точные копии электронных устройств, телефонов, компьютеров, автомобилей и истребителей, причем порой даже раньше, чем на свет появляется сам оригинал?

В 2016 году вирусные аналитики компании «Доктор Веб» обнаружили троянца-бекдора BackDoor.Crane.1, целенаправленно воровавшего конфиденциальную информацию у сотрудников двух российских компаний, занимающихся производством портальных и грузоподъемных кранов. Особый интерес вредоносная программа проявляла к договорам, финансовым документам и почтовой переписке своих жертв, кроме того, она периодически делала снимки экрана зараженного компьютера и отправляла их на сервер злоумышленников.

Все сказанное выше лишь подтверждает тезис, озвученный в самом начале этой главы: практически все существующие сегодня вредоносные программы рассчитаны исключительно на получение прибыли за счет своих жертв.

ГЛАВА 4. МОБИЛЬНЫЕ ВРЕДОНОСНЫЕ ПРОГРАММЫ

Как известно, операционные системы разрабатываются людьми, а люди склонны совершать ошибки. Так, в мобильной платформе Google на сегодняшний день обнаружено множество ошибок, некоторые из них представляют собой полноценные уязвимости и могут использоваться как для несанкционированного доступа к файловой системе смартфона, так и для распространения вредоносного ПО. Существуют и вредоносные программы для продукции корпорации Apple, способные работать в самых современных версиях iOS.

УЯЗВИМОСТИ В ANDROID

Самая первая уязвимость Android была обнаружена еще в октябре 2008 года в прошивке коммуникатора HTC T-Mobile G1. При просмотре веб-страниц с определенным содержимым ошибка в ПО позволяла выполнить вредоносный код, отслеживающий использование клавиатуры гаджета. Теоретически таким образом можно было реализовать кейлоггер, фиксирующий нажатия кнопок, и собирать вводимую пользователем при веб-серфинге информацию. Эта уязвимость представляла опасность только для одной-единственной модели коммуникатора, но само ее наличие наглядно показало: Android — не настолько безопасная и защищенная система, как считалось ранее.

Согласно информации с сайта cvedetails.com, на март 2020 года в Android насчитывается 2565 уязвимостей, при этом число выявленных багов начало экспоненциально расти примерно с 2014 года. Не так просто оценить, сколько из перечисленных устройств вовремя получили патчи безопасности, которые закрывают уязвимости, но это явно далеко не все из них. Мало того: не все уязвимости вообще оказываются закрытыми, тем более в старых версиях, официальная поддержка которых прекращена. Проблему усугубляют производители устройств, которые зачастую не торопятся выпускать обновления. С ростом популярности Android энтузиасты и исследователи отыскивали все новые и новые ошибки в различных ее версиях.

МОБИЛЬНЫЕ БАНКОВСКИЕ ТРОЯНЦЫ

Одним солнечным апрельским утром мой завтрак был прерван телефонным звонком приятеля — предпринимателя,

занимавшегося грузовыми перевозками. Срывающимся голосом он рассказал, что с его банковского счета куда-то испарились два миллиона рублей. А служба поддержки банка развела руками, отправив приятеля писать заявление в полицию, поскольку денежные переводы были совершены с помощью мобильного приложения и подтверждены по SMS, что по всем признакам соответствует вполне легальной финансовой операции. «Ты ж программист, — простонал в трубку мой приятель, — посоветуй, что делать». Увы, что-либо делать было уже поздно, ибо инструментом для кражи послужил банковский троянец, обосновавшийся на смартфоне моего товарища задолго до этого инцидента. И предотвратить потерю денег было можно, лишь заранее изучив принципы работы и методы борьбы с такими вредоносными программами. Чем мы прямо сейчас и займемся.

Первенцы

Первые полноценные банковские трояны для мобильной платформы Android были обнаружены еще в 2011 году. Нет, вредоносы, способные передавать злоумышленникам входящие SMS-сообщения, в том числе содержащие mTAN-коды (коды аутентификации транзакций), существовали и до этого. Кроме того были известны троянцы, умеющие оперировать USSD-командами. Они могли перевести заданную злодеями сумму с «привязанной» к телефону банковской карты, пополнив баланс чужого мобильного телефона, или узнать остаток средств на счете. Но полноценными банкерами такие трояны, конечно же, не были, поскольку заметно уступали по функциональным возможностям своим десктопным аналогам.

Все изменилось с появлением Android.SpyEye. Этот троян работал в связке с вредоносом SpyEye для Windows, благодаря чему обрел способность обходить двухфакторную аутентификацию. Действовал он следующим образом. Как только пользователь зараженной Windows открывал в браузере банковский сайт, работающий на компе троян выполнял веб-инжект, встраивая в страницу кусок HTML-кода, который он подгружал из конфигурации. Поскольку инжект осуществлялся на стороне клиента, URL банковского сайта в адресной строке браузера оказывался корректным, а соединение было установлено по протоколу

HTTPS. Поэтому содержимое веб-страницы не вызывало у жертвы никаких подозрений.

Текст, встроенный трояном в банковский сайт, гласил, что банк внезапно изменил условия работы, и для авторизации в системе банк-клиент необходимо установить на мобильный телефон небольшое приложение размером около 30 Кбайт, скачав его по предложенной ссылке — «в целях безопасности». Приложением, естественно, был мобильный трой Android.SpyEye. Эта вредоносная программа не создавала никаких значков, ее можно было отыскать только в списке работающих процессов под названием «Система». Основная задача трояна — перехват всех входящих SMS-сообщений и пересылка их на управляющий сервер, адрес которого вредонос брал из XML-файла.

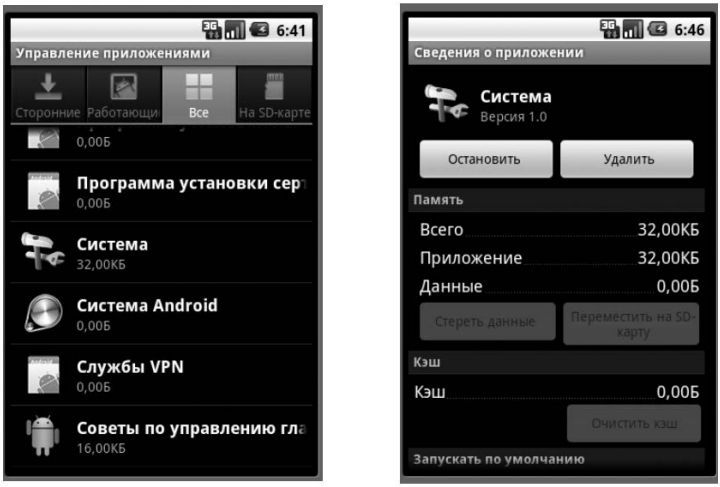


Рис. 9. Так выглядел один из первых мобильных банковских троянов — Android.SpyEye

Когда жертва вводит логин и пароль на банковском сайте в окне браузера, Windows-троян SpyEye перехватывает и отправляет их киберпреступникам. После этого злоумышленники в любой момент могут авторизоваться с помощью этих данных в системе банк-клиент на сайте банка, однако сервер обязательно отправит владельцу счета проверочный код в SMS, который нужно ввести в специальную форму. Это сообщение

будет перехвачено мобильной версией SpyEye и передано вирусописателям. Используя перехват SMS, они смогут выполнять любые операции по счету, например опустошить его подчистую.

Узким местом этой довольно сложной схемы была необходимость синхронизации работы банковского и десктопного компонентов троянской связки, однако указанную проблему вирусописателям удалось успешно решить. Несколько месяцев SpyEye наводил шорох среди пользователей банковских сервисов, пока не попал в базы всех популярных антивирусов, после чего его деятельность постепенно сошла на нет.

Как работают мобильные банкеры

Спустя какое-то время сотрудники IT-отделов банков понемногу освоили веб-программирование, и банк-клиенты окончательно перекочевали с настольных компьютеров в мобильные телефоны в виде Android-приложений. Это значительно облегчило жизнь вирусописателям: у них отпала необходимость мучиться с троянами под Windows, и они смогли наконец полностью сосредоточить свои усилия на разработке мобильных банкеров.

Как и прочие вредоносы для Android, банкеры распространялись под видом каких-либо полезных программ — «универсальных видеокодеков» или проигрывателей Flash, в том числе через официальный каталог Google Play. Троянская функциональность таких приложений, естественно, не афишировалась разработчиками, и проявлялась она либо спустя какое-то время либо после загрузки очередного обновления. Так, в одном из случаев банкер раздавался в виде программы, якобы объединяющей в себе возможности банк-клиентов сразу нескольких крупных кредитных организаций. Зачем вам куча отдельных приложений, когда вместо них можно скачать одно, с трояном? Также известны случаи, когда вредоносы встраивались в подлинные приложения некоторых банков, модифицированные злоумышленниками. Такие приложения распространялись с поддельных банковских страниц, оформленных в точности как настоящие, а жертв на них завлекали рассылками рекламных писем.

Еще один вектор распространения мобильных банковских троянов — фишинговые SMS-рассылки. Обычно это происходит так. Пользователю, зарегистрированному на одном

из сайтов бесплатных объявлений, приходит SMS-сообщение с предложением обмена. При этом получателя называют по имени, что должно усыпить его бдительность, — вирусописатели предварительно обработали базу пользователей этого сайта, вытянув оттуда всю полезную информацию. При переходе по короткой ссылке из сообщения потенциальная жертва направляется на промежуточную страницу, где определяется, что пользователь зашел на сайт именно с мобильного устройства под управлением Android, и выявляется его мобильный оператор, после чего происходит перенаправление на поддельную страницу с сообщением о поступлении MMS, оформленную в стилистике соответствующего оператора сотовой связи. После нажатия на кнопку начинается загрузка трояна.

Рис. 10. Пример мошеннической рассылки, SMS-рассылки, цель которой — распространение банковского трояна

Первые мобильные банкеры работали очень просто. Если для функционирования вредоноса были нужны права администратора, он настойчиво демонстрировал на экране окно с требованием выдать ему соответствующие полномочия, до тех пор, пока измученный пользователь не согласится на это действие.

Но иногда вирусописатели шли на различные ухищрения, чтобы обмануть потенциальную жертву. Например, банкер Android.BankBot.29 маскировал окно запроса прав администратора под сообщение приложения Google Play: «Ваша версия устарела, использовать новую версию?» При попытке пользователя нажать на экранную кнопку «Да» layout трояна исчезал, и тап попадал на кнопку Accept диалогового окна DeviceAdmin, в результате чего вредонос получал администраторские привилегии.

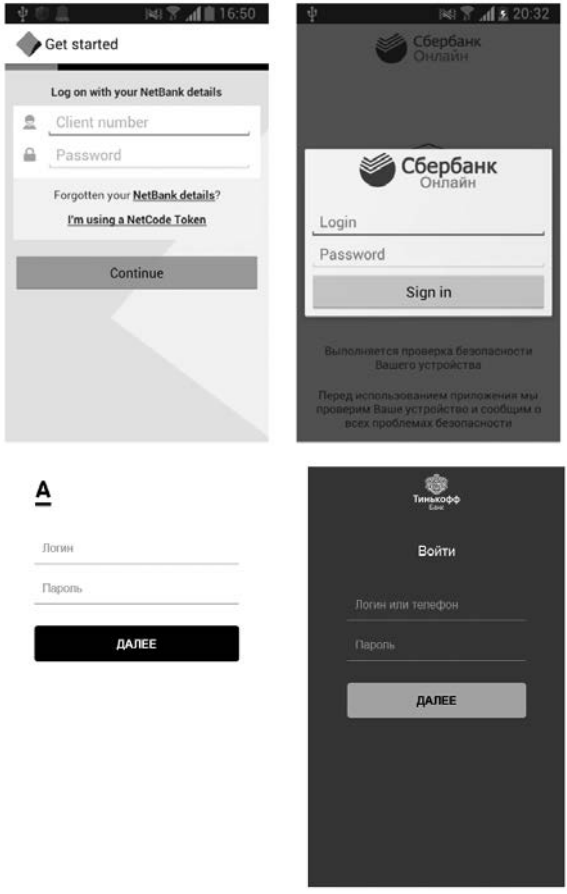


Рис. 11. Мобильные банкеры могут рисовать поддельные окна авторизации популярных банков

Еще один банкер доставал пользователей запросом на включение режима Accessibility Service — специальных возможностей для людей с ограничениями по здоровью. А получив такое разрешение, сам включал для себя администраторские привилегии. После этого троян просто сидит в памяти мобильного телефона, ожидая запуска мобильного банковского приложения. При наступлении этого события он определяет, какое именно приложение запущено, и рисует поверх него соответствующую поддельную форму ввода логина и пароля, а введенные данные тут же пересылаются на управляющий сервер по HTTP в виде JSON или на заданный телефонный номер SMS-сообщением. Конфигурация мобильного банкера может содержать HTML-код нескольких десятков форм с различным оформлением, копирующим интерфейс приложений наиболее популярных банков. После этого остается только перехватить и отправить в том же направлении SMS с одноразовыми паролями, чтобы предоставить киберпреступникам полный доступ к банковскому счету. Входящие сообщения от банков при этом обычно скрываются, чтобы не вызывать у жертвы подозрений.

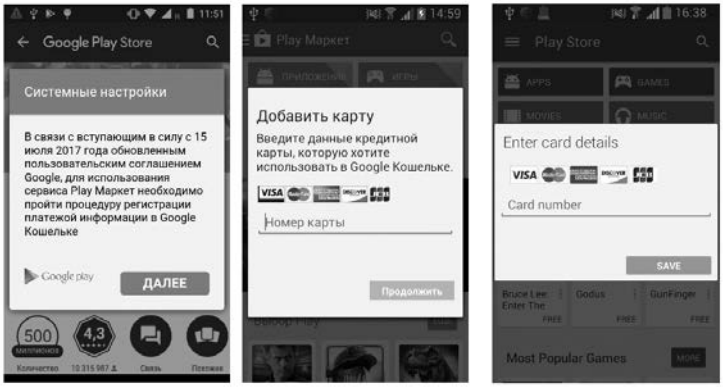


Рис. 12. Для хищения реквизитов банковских карт многие мобильные трои используют поддельные окна Google Play

Сколько денег подобным образом было похищено со счетов пользователей Android, сказать трудно, но суммы здесь наверняка фигурируют шестизначные. Даже если троянам по какой-то причине не удавалось получить доступ к банковскому

счету, они благополучно похищали реквизиты банковских карт. Для этого, например, широко использовались поддельные окна привязки карты к приложению Google Play.

Приобрести что-либо ценное в приличных интернет-магазинах с использованием ворованных реквизитов непросто, а вот оплатить онлайн-игры или покупку музыки в каком-нибудь сервисе вполне возможно. Подобные сайты редко выполняют серьезную проверку платежных реквизитов, поскольку транзакции там обычно копеечные. Чем и пользуются злоумышленники.

Банкботы

Банкботы — это побочная ветвь эволюции мобильных банкеров. Если обычные банковские трояны работают более-менее автономно, то банкботы способны получать различные управляющие команды и выполнять их на зараженном девайсе.

Команды могут передаваться по HTTP, например в формате JSON, по SMS, а в некоторых случаях даже через специальный Telegram-канал. Большинство банкботов по команде включают или отключают перехват входящих SMS-сообщений, могут скрывать полученные SMS (прятать можно сообщения с определенных номеров или с заданными ключевыми словами), отключать звук мобильного телефона, отправлять сообщения на указанный злоумышленниками номер с заданным содержимым или выполнять USSD-команды. Также ботовод может изменить адрес управляющего сервера или системный номер телефона, на который будет пересылаться информация, если ее не удалось передать по HTTP.

Многие банкботы также могут скачивать и устанавливать на мобильном устройстве APK-файлы, ссылку на которые укажет в команде оператор. В результате на зараженный девайс попадают другие трояны, имеющие более широкий ассортимент функций. Также некоторые банкботы умеют отображать на экране смартфона активити с присланными злодеем параметрами — это открывает широчайшие возможности для фишинга и реализации самых изощренных мошеннических схем. Ну и почти все такие вредоносы умеют передавать на командный сервер адресную книгу, SMS-переписку

и прочие конфиденциальные данные, а также переадресовывать входящие звонки на заданный в команде телефонный номер. Отдельные экземпляры троянов ко всему прочему обладают функциями самозащиты: они отслеживают имена работающих в системе процессов и при обнаружении антивируса пытаются выгрузить его, используя права администратора. Практически все банкботы используют веб-панель администрирования, предоставляющую операторам подробную статистику по инфицированным девайсам и похищенной на них информации.

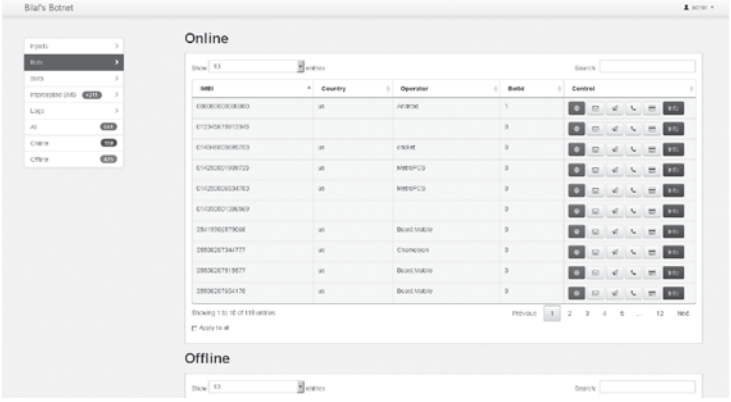


Рис. 13. Типичная панель управления мобильного банковского бота

Криминальная индустрия

С распространением мобильных устройств на Android производство троянов для этой платформы стало понемногу превращаться в самую настоящую подпольную индустрию. В полной мере коснулось это и банкеров. В даркнете стали появляться объявления о сдаче банковских троянов под Android в аренду, с предоставлением клиенту админки и технической поддержки. А затем начали распространяться билдеры, с использованием которых любой желающий без каких-либо навыков программирования мог соорудить банковского трояна, маскирующегося под выбранное приложение или определенную систему банк-клиент.

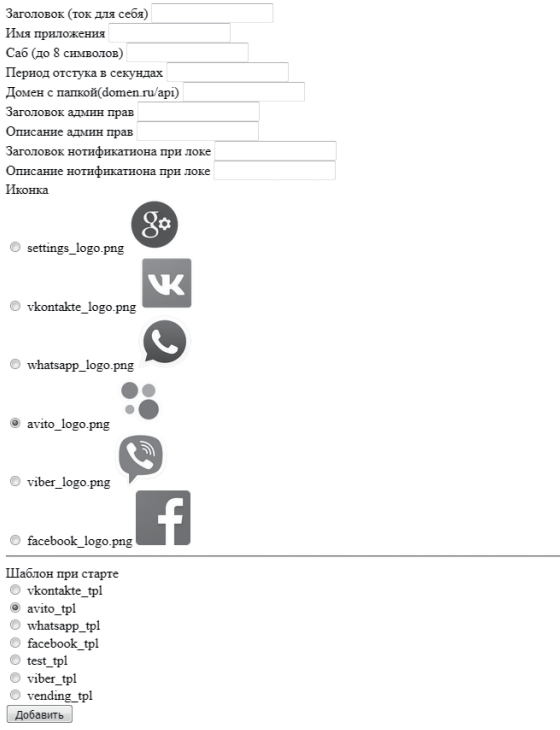


Рис. 14. Так выглядит билдер для создания банковского троянца под Android

Благодаря этому количество банковских троянов примерно с 2017 года начало расти если не в геометрической прогрессии, то весьма заметно. И шансы подцепить подобную заразу у пользователей смартфонов на Android тоже значительно выросли. А с учетом того, что большинство подобных вредоносов работает с привилегиями администратора, удалить их с устройства не так-то просто: для этого в лучшем случае придется запустить систему в безопасном режиме, в худшем — сбросить девайс к заводским настройкам со всеми вытекающими последствиями.

Доказанный факт: даже отключение на телефоне возможности установки приложений из сторонних источников далеко не всегда защищает пользователя от проникновения банкеров.

Случаев загрузки подобных вредоносов даже из официального каталога Google Play известно множество: технология проверки размещаемых там приложений все еще несовершенна. Кроме того, операционную систему Android отличает значительное количество уязвимостей, которые могут использоваться вирусописателями в своих, отнюдь не благородных целях. Защитить устройство от несанкционированного проникновения зловредов способны антивирусы, но вот устанавливать их или нет — личное дело самих пользователей Android. По крайней мере, мой приятель-коммерсант после инцидента с похищением денег решил больше не испытывать судьбу и скачал на свой телефон такую программу: лишней не будет.

ВРЕДОНОСНЫЕ ПРОГРАММЫ ДЛЯ IOS

Мобильная платформа iOS — одна из немногих операционных систем, для которых не существует антивирусов. Несмотря на высокую популярность айфонов и айпадов, разработчики антивирусных программ не в состоянии освоить эту привлекательную для них нишу в силу архитектурных особенностей iOS: платформа просто не предоставляет прикладным программам доступ к файловой системе, без которого никакая антивирусная проверка невозможна в принципе. Кроме того, большинство владельцев мобильных устройств от Apple уверены, что вредоносных программ для iOS не существует в природе. Так ли это?

Немного теории

Общеизвестный факт: все приложения в iOS выполняются в так называемой *песочнице* (sandbox) — изолированной среде, из которой они не могут получить непосредственный доступ к компонентам операционной системы и других программ. Это обеспечивает высокую безопасность ОС: при работе в sandbox приложение взаимодействует только с собственными данными и ресурсами, поэтому вредоносной программе попросту негде будет разгуляться. Кроме того, Apple разрешает установку приложений на устройства с iOS только из собственного каталога

App Store, куда они попадают после тщательной проверки. «Несчастные» владельцы айфонов лишены даже привычной пользователям Android функции «разрешить установку приложений из неизвестных источников» — если нужной программы нет в App Store, ее, скорее всего, не будет и на вашем смартфоне.

С другой стороны, подобные жесткие ограничения лишают владельцев «яблочных» девайсов целого ряда полезных возможностей. Если ваш айфон относится к устаревшему модельному ряду и его операционная система уже успела «выйти на пенсию», рано или поздно вы столкнетесь с полной невозможностью установить или обновить нужную программу через App Store. Однажды ваше любимое приложение откажется запускаться, сообщив, что разработчики давным-давно выпустили для него новую версию, которую пора поставить вместо текущей. По нажатию на кнопку «Обновить» запустится программа App Store и радостно сообщит, что для установки новой версии интересующей вас программы требуется операционная система посвежее. Наконец, отправившись в раздел «Настройки», вы с удивлением обнаружите, что на вашем устройстве уже стоит самая актуальная версия iOS, а чтобы использовать более современную, придется сбегать в соседнюю лавку за новым айфоном. Круг замкнулся, как любил говорить один бывший джедай.

Выходов из такого тупика существует ровно два. Первый — продать на черном рынке почку и таки купить себе новый телефон с надкусанным яблоком на корпусе, затаив надежду, что он не утратит своей актуальности в течение следующих нескольких лет. Второй путь — джейлбрейк. Под этим непонятным словом подразумевается банальный взлом операционной системы, позволяющий получить не санкционированный производителем телефона доступ к файловой системе, а кроме того, устанавливать приложения из сторонних репозиториев или в некоторых случаях напрямую с компьютера.

Джейлбрейк, конечно же, не превратит ваш четвертый айфон в десятый и не даст возможности использовать на древнем железе самую последнюю редакцию iOS. Зато позволит отыскать и установить на смартфон старую версию нужной программы, которая пусть и не сможет похвастаться современным набором функций, но будет хотя бы стабильно работать. Существует и еще одна лазейка, позволяющая почти официально

устанавливать на айфоны и айпады различный софт в обход App Store. Называется она *Mobile Device Management* (MDM). Это набор инструментов, дающий возможность управлять устройствами с iOS в корпоративной среде. Используется он, в частности, для установки на «яблочные» девайсы сотрудников фирм различных «внутренних» приложений, не предназначенных для широкого распространения вне компании. Такие программы можно доставлять на устройства с iOS без необходимости загружать их в App Store и проходить мучительную проверку. Очевидно также, что любое приложение на айфон этим способом установить не получится: метод имеет целый ряд естественных ограничений, призванных исключить возможное его использование злоумышленниками.

Означает ли все это, что существование вредоносов для iOS невозможно в принципе и пользователи девайсов от Apple могут чувствовать себя в полной безопасности? Нет. Опасные программы для iOS как тот легендарный суслик: не видны, но все-таки есть. Я расскажу о самых известных технологиях распространения такого ПО.

Шпионские игры

К 2013 году, когда мобильные телефоны производства Apple уже прочно заняли свою нишу на мировом рынке, а в розничной продаже появился iPhone 5s, специалистам по информационной безопасности было известно около 50 шпионских программ для iOS. Практически все они предназначались для аппаратов с джейлбрейком, и практически все распространялись через «пиратский» репозиторий Cydia — альтернативный каталог приложений для взломанных «яблочных» устройств.

Пользователь мог обрести шпиона на своем телефоне либо по незнанию (некоторые были представлены в каталоге под вполне нейтральными названиями), либо в результате целенаправленной установки, если какой-то доброжелатель решил, например, помочь неофиту с настройкой телефона. Упомянутые программы обладали вполне традиционным для spyware набором функций: кража SMS и истории звонков, контактов и фотографий, истории браузера, передача GPS-координат. Более продвинутые шпионы фиксировали информацию о совершенных

звонках, могли записывать аудио с помощью встроенного микрофона, делать по команде с сервера фотоснимки, копировать сообщения электронной почты и переписку в социальных сетях, которую пользователь вел при помощи приложений-клиентов. Вся информация отсылалась на управляющий сервер, где ее можно было получить в удобной и доступной форме.

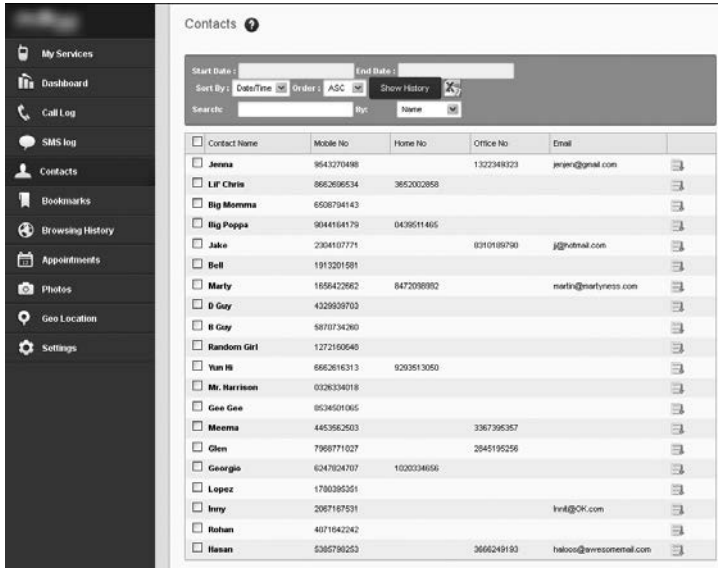


Рис. 15. Интерфейс панели управления программой-шпионом для iOS

Такие шпионы не создавали в системе значок приложения. По желанию в них можно было активировать только часть функций, чтобы снизить вероятность обнаружения программы пользователем. Наиболее популярным spyware для iOS в те времена считались SpyBubble, TopSpy, Tracker, OwnSpy, TruSpy и FlexiSpy. Однако все эти приложения никак нельзя было назвать полноценными троянами, поскольку, во-первых, их необходимо было устанавливать на устройство вручную, во-вторых, для этого требовался джейлбрейк и, в-третьих, многие из них продавались в интернете фактически легально в качестве средств родительского контроля или контроля над сотрудниками предприятия.

Технология MDM

Из-за параноидальных механизмов безопасности, которые Apple применяет в архитектуре своей мобильной ОС, создание полноценных вредоносных программ для этой платформы оказалось делом трудозатратным, однако сюрпризом для специалистов все-таки не стало. Если в iOS нельзя зайти через дверь, можно вломиться через окно — примерно так подумали злоумышленники и для распространения троянов начали применять тот самый механизм дистрибуции приложений MDM с использованием корпоративных сертификатов.

Работает это, вкратце, так. Для начала требуется развернуть специальный MDM-сервер, получить для него сертификат Apple Push Notification Service (APNs-сертификат) и установить его на этом сервере. Затем необходимо создать специальный конфигурационный профиль, фактически представляющий собой видоизмененный plist-файл, который следует доставить на устройство с iOS. Устройство получает с сервера push-уведомление, устанавливает с сервером TLS-соединение и после проверки сертификата авторизуется на нем. Далее сервер может передать устройству набор настроек (MDM Payload), привязанный к его конфигурационному профилю. При этом MDM-сервер необязательно должен находиться в одной сети с мобильным устройством, достаточно, чтобы он был доступен извне по протоколу HTTPS. В результате с использованием MDM-сервера становится возможным управлять iOS-устройством и устанавливать на него приложения в обход App Store.

Все это подразумевает серьезные пляски с бубном, но теоретически открывает лазейку для MITM-атак. С использованием этой технологии вполне можно реализовывать таргетированные «точечные» атаки. Кроме всего прочего, злоумышленником может оказаться, например, обиженный сотрудник компании, использующей MDM, если он имеет доступ к серверу. Говорят, именно таким способом у одной известной американской фирмы была похищена клиентская база и переписка с контрагентами, которая впоследствии утекла к конкурентам вместе с одним из бывших работников. Кажется, что массовое распространение вредоносов с привлечением технологии MDM организовать невозможно. Однако это не так.

С использованием MDM в 2015 году была устроена массовая раздача трояна YiSpecter, прятавшегося в клиентском приложении — видеоплеере для просмотра порнографии. Распространялся он преимущественно в Китае под видом версии популярного в этой стране порноплеера QVOD, разработчиков которого в 2014 году арестовала китайская полиция. Следуя инструкции, найденной на просторах интернета, пользователи сами копировали на свой айфон необходимые профили и сертификаты, чтобы получить возможность бесплатно скачивать на телефон коммерческие или «запрещенные» в их стране приложения, за что и поплатились.

Кроме того, разработчики трояна организовали целую партнерскую программу, в рамках которой платили по 2,5 юаня за каждую установку протрояненного софта. Предложением тут же воспользовались многочисленные «подвальные» сервис-центры по ремонту айфонов и фирмочки, специализирующиеся на продаже восстановленных «яблочных» устройств, и стали втихаря добавлять в систему благодарным клиентам необходимые компоненты. С паршивой овцы, как говорится, хоть шерсти клок. В результате обладателями айфонов «с сюрпризом» стали тысячи ни о чем не подозревающих китайцев.

YiSpecter устанавливал на устройство специальный модуль, который докачивал компоненты вредоноса, если их удаляли, и по команде ботоводов мог втихую удалять с телефона любые программы, заменяя их троянизированными копиями. Нужный софт он получал с собственного MDM-сервера. Пользователь не замечал подвоха, поскольку подмененные трояном клиенты социальных сетей и мессенджеры работали как обычно, время от времени отстукиваясь на управляющий сервер и скидывая туда конфиденциальную информацию, включая переписку и учетные данные жертвы. Кроме всего прочего, троян мог показывать на экране зараженного устройства полноэкранную рекламу, благодаря чему, собственно, и был обнаружен. В общем-то понятно, что таким способом трудно организовать массовое заражение «яблочных» устройств, хотя у китайцев это почти получилось. Важно другое: YiSpecter был одним из первых настоящих вредоносов, способных заразить айфон без джейлбрейка.

Технология DRM

Трояны, не использовавшие для распространения MDM, появились оттуда же, откуда является большинство оригинальных разработок в сфере IT, — из Китая. Здесь сошлись воедино алчное стремление Apple зарабатывать как можно больше на распространении приложений для iOS и непреодолимая страсть некоторых пользователей к халяве.

Как известно, приложения для iPhone следует в обязательном порядке приобретать в официальном магазине App Store — по крайней мере, так считают в Apple. Если программа честно куплена на этом ресурсе и числится на аккаунте пользователя, он может установить ее на телефон позже, присоединив последний к компьютеру при помощи шнура USB-Lighting и воспользовавшись программой iTunes. При запуске программа проверит Apple ID пользователя и запросит код авторизации, чтобы убедиться, что устанавливаемое на мобильный девайс приложение было действительно приобретено этим пользователем законным образом. Для этого используется разработанная Apple технология Digital Rights Management (DRM).

Для обхода этой проверки хитрые китайцы придумали специальную программу, которая эмулирует действия iTunes. Купив приложение в App Store, создатели этой программы перехватывают и сохраняют код авторизации при помощи уязвимости в реализации DRM, после чего передают его всем остальным пользователям своего приложения. В результате те получают возможность установить на свой айфон или айпад программу, за которую не платили. Одно из таких приложений носит наименование Aisi.

Aisi позволяет пользователям «яблочных» устройств не только устанавливать нелегальный софт, но и обновлять и создавать резервные копии прошивки, делать джейлбрейк, закачивать на телефон рингтоны и различный мультимедиа-контент. Предварительно китайские вирусописатели создали и разместили в App Store небольшую утилиту, позволяющую менять обои на устройстве с iOS. Утилита благополучно прошла все проверки Apple, даже несмотря на то, что скрывала в себе одну потенциально опасную функцию, которая активизировалась, правда, при совпадении ряда внешних условий — наверное,

потому ее и не заметили. После присоединения айфона к компьютеру и включения режима «доверия» между устройствами Aisi с помощью описанной выше технологии скрытно устанавливала в iOS ту самую утилиту, сообщив девайсу, что она якобы была ранее куплена пользователем в App Store. При запуске приложение требовало ввести данные учетной записи Apple ID. Эта информация тут же передавалась на управляющий сервер.

Рис. 16. Программа Aisi

Дальше, в общем-то, с устройством можно сотворить много интересного: утечка Apple ID открывает перед потенциальными злоумышленниками массу возможностей. Например, можно сменить пароль, заблокировать устройство и потребовать у его владельца выкуп за разблокировку. А можно получить доступ к хранилищу iCloud и полюбоваться чужими фотографиями из отпуска. Это в лучшем случае.

Несмотря на то что iOS действительно очень защищенная и безопасная операционная система, мы видим, что вирусописатели смогли отыскать лазейки и в ней. Правда, все они без исключения представляют собой тесный симбиоз технических приемов и социальной инженерии. Разработчики вредоносов

для айфонов и айпадов работают именно в этом направлении — играя на наивности или алчности владельцев «яблочных» телефонов.

Впрочем, интерес к ним неудивителен: считается, что покупатели недешевых телефонов производства корпорации Apple достаточно состоятельны, чтобы у них было чем поживиться. Поэтому пользователям таких устройств нужно проявлять особую бдительность и осторожность: в любой момент они могут оказаться под прицелом злоумышленников, ведь технологии не стоят на месте.

ГЛАВА 5. ВРЕДОНОСНЫЕ ПРОГРАММЫ ДЛЯ «ИНТЕРНЕТА ВЕЩЕЙ»

Пора отправлять в архивы истории старый анекдот о том, что вредоноса для Linux нужно сначала собрать из исходников, убедиться в том, что в системе есть все нужные библиотеки, и только после этого пытаться запустить, выдав ему предварительно права суперпользователя root. Сейчас на рынке полно «умных устройств» с Linux, и они стали одной из любимых целей вирусописателей. На чем основан этот интерес и как работают такие вредоносы?

Элементарная логика подсказывает: чтобы современный троян успешно выполнял свои функции, он должен без особого труда проникать в систему, иметь стабильное подключение к интернету и по возможности оставаться незамеченным как можно дольше. Всем этим критериям прекрасно соответствуют разнообразные девайсы категории «интернета вещей» — роутеры, телевизионные приставки, сетевые накопители, медиацентры, а также устройства, собранные на основе одноплатных компьютеров. Как правило, все они сидят на «толстом» интернет-канале, что превращает подобное оборудование в удобный инструмент для организации DDoS-атак.

С проникновением тоже зачастую не возникает серьезных проблем. По статистике, большинство владельцев умных устройств — это простые пользователи, которые не страдают от избытка знаний в сфере высоких технологий. Они относятся к девайсам IoT как к обычным бытовым приборам вроде соковыжималки или тостера, зачастую даже не подозревая, что у них вообще есть какие-то настройки, которые при желании можно изменить. Включили, работает, и ладно. Поэтому установленные на заводе дефолтные логины и пароли на подобных устройствах превращают взлом в задачу, посильную любому школьнику.

Наконец, скрытность. С этим совсем все просто: кому, скажите на милость, придет в голову искать трояна в телевизионной приставке или в недрах роутера, тем более что антивирусы для большинства таких девайсов в дикой природе практически не водятся, а проверить их дистанционно — та еще задача для простого пользователя? Да и многие непростые пользователи иногда по инерции уверены, что вредоносов для Linux не существует в принципе. Поэтому трояны для IoT могут чувствовать себя в системе совершенно вольготно: искать их там вряд ли будут, а если и будут, не факт, что найдут.

Производитель часто виноват не меньше пользователя: многие умные устройства, имеющие известные уязвимости

в прошивке, не получают от производителя обновлений, которые могли бы закрыть дыры. Это значительно повышает эффективность использования даже устаревших эксплоитов. Вероятность «пробива» таких устройств в среднем намного выше, чем даже компьютеров с регулярно обновляющейся операционной системой, не говоря уже о серверах или рабочих станциях с актуальным «Линуксом» на борту.

С какой целью создатели вредоносных программ устанавливают их на умные устройства? «Профессий» у большинства подобных троянов традиционно три. Прежде всего, это уже упомянутые выше DDoS-боты, начинающие флудить на указанный адрес сетевыми пакетами по команде с управляющего сервера. Конечно, одна бабушка с телевизионной приставкой уничтожить вражеский сервер тебе не поможет, но десяток старушек, как гласит народная мудрость, — уже рубль. Во-вторых, на IoT-девайсах часто поднимают SOCKS-proxy-сервер, который можно использовать разными способами, прежде всего — для обеспечения анонимности. Ну и наконец, в роутерах периодически заводятся трояны, подменяющие в настройках адреса DNS-серверов, чтобы показывать на клиентских машинах рекламу. При этом сам компьютер, на котором вдруг из ниоткуда начинают появляться баннеры с рекламой такси, онлайн-казино и девушек легкого поведения, остается девственно чистым, что подтвердит пользователю любой антивирус. Смена настроек DNS вручную ничем не поможет: после перезагрузки роутера все вернется на круги своя. Спасти ситуацию сможет разве что сброс устройства к заводским установкам или его перепрошивка, что для большинства обывателей выглядит как тайная магия высшего уровня. Иногда на Linux-устройствах попадаются майнеры, но в последнее время интерес к ним на фоне общего спада рынка криптовалют понемногу снижается.

МАТЧАСТЬ

Изучение логов ханипотов (специально созданных серверов — «приманок»), с помощью которых специалисты по информационной безопасности анализируют распространение

троянов для «интернета вещей» в живой природе, показывает следующий путь их доставки на целевое устройство. Обычно атакующие соединяются с привлекшим их внимание устройством по протоколу SSH или Telnet, подбирают пароль по словарю и в случае успешной авторизации отключают утилиту iptables, чтобы заблокировать работу файрвола. Дальше им остается только отыскать открытую на запись папку, сохранить в нее нужное приложение, соответствующее архитектуре устройства, и запустить его.

В арсенале атакующих обычно имеются готовые файлы малвари под различные архитектуры: ARM, MIPS, SPARC, M68K, SuperH, PowerPC, SH-4 и прочие типы железа. На некоторых девайсах можно организовать автозагрузку трояна, просто отредактировав файл */etc/rc.local*. Периодически проверять, работает ли нужный процесс, и при необходимости перезапускать его можно, например, с использованием */etc/cron.minutely*. Поскольку многие пользователи умных устройств не утруждают себя сменой дефолтных логинов и паролей, словарь для брутфорса (подбора пароля) в большинстве случаев выглядит весьма тривиально. Вот наиболее популярные логины и пароли для взлома девайсов по протоколу SSH.

ЛОГИН	ПАРОЛЬ
pi	rasberry
informix	informix
root	nagiosxi
nagios	nagios
root	synopass
cactiuser	cacti
admin	articon

Также при бруте по SSH очень часто используются такие логины, как ubnt, user, oracle, bin, support, postgres. А вот наиболее популярные сочетания логинов и паролей для взлома девайсов по протоколу Telnet:

ЛОГИН	ПАРОЛЬ
root	xc3511
root	vizxv
root	anko
root	5up
root	XA1bac0MX

Эта табличка наглядно демонстрирует, что по Telnet злодеи традиционно пытались авторизоваться в админке различных камер видеонаблюдения, таких как Anko, TP-Link, Dahua или CNB. Кроме того, для перебора паролей по этому протоколу очень часто берут широко используемые логины вроде admin, test и telnet.

Однако технологии не стоят на месте, и с определенного момента вирусописатели взялись за автоматизацию процедуры взлома. Так, если трояна удалось запустить в скомпрометированной системе, он может попытаться вытащить из различных источников вроде файла *ssh/known_hosts* или *bash_history* сведения о хостах, к которым устройство ранее получало доступ, и попытаться взломать их. Другие трояны поступают гораздо проще: генерируют диапазон IP-адресов и в цикле пытаются постучаться на стандартные для SSH и Telnet порты. Плюс такого подхода заключается в его относительной простоте, недостаток — в значительном объеме создаваемого паразитного трафика.

Примерно три года назад в ассортименте малвари для умных устройств было не менее десятка разных модификаций троянов Fgt, Mrblack, Mirai, разнообразных флудеров и DDoS-ботов. Однако после того как исходники Mirai в сентябре 2016 года оказались в свободном доступе на сайте hackforums.com, бесчисленные клоны этого трояна практически вытеснили с рынка остальных конкурентов. BrickBot, Sora, Wicked, Omni, Owari — все это вариации на тему Mirai с дополнениями все новых и новых соавторов.

MIRAI

По данным исследователей Сэма Эдвардса и Иоанниса Профетиса из Rapidity Networks, в 2016 году троянец Linux. Mirai заразил более 380 тысяч умных устройств по всему миру, став причиной множества нашумевших DDoS-атак. После того как исходники этого трояна были выложены в паблик, изучением принципов его работы тут же заинтересовались не только специалисты по информационной безопасности, но и тысячи анонимных вирусописателей.

Сам по себе исходный код Mirai тоже имеет ряд заимствований из других источников и во многом базируется на архитектуре одного из DDoS-ботов, распространявшихся весной 2016 года. Этот бот, в свою очередь, имеет схожие черты с троянами семейства Fgt, известного аж с 2014 года, — оттуда была явно скопирована функция флуда и генератор псевдослучайных данных для заполнения пакетов, отсылаемых на атакуемые хосты. Такое «перекрестное опыление», в целом характерное для разработчиков open source софта для Linux, по всей видимости, прижилось и в среде ориентирующихся на эту платформу вирусописателей. Все ранние версии Mirai для разных аппаратных архитектур были статически скомпилированы и использовали стандартную библиотеку *uClibc.so*, предназначенную для встраиваемых систем на базе uCLinux.

При запуске троян выделял область памяти, в которую загружал необходимый для его работы конфиг. Некоторые конфигурационные строки хранились в зашифрованном виде в самом файле ELF и перед загрузкой в память расшифровывались. В частности, таким образом передавался адрес и порт управляющего сервера, User-Agent и другие параметры HTTP-запросов. Отсюда становится понятен способ, с помощью которого исследователи определили адрес командного центра этого ботнета. Еще одна примечательная функция ранних версий Mirai — способность трояна в отдельном потоке убивать работающие процессы других конкурирующих ботов, читая в непрерывном цикле содержимое папки */proc/*. Список названий процессов хранился в конфиге. А вот в обработчике конфига авторы Mirai допустили небольшую ошибку: список процессов читался блоками по 2048 байт, после чего поиск названия выполнялся

внутри этого буфера. Если искомое значение располагалось на границе блока, оно игнорировалось, и такой процесс избегал общей печальной участи. Чтобы предотвратить случайную выгрузку собственного процесса, троян размещал в его папке файл *.shinigami* — при обнаружении файла с этим именем процесс автоматически исключался из «расстрельного списка». Однако в самом коде зловреда был зашит лимит жизнедеятельности на зараженном устройстве: по истечении недели непрерывной работы процесс Mirai автоматически завершается. Описываемая здесь ранняя версия Mirai умела выполнять с десяток команд, среди которых — команды на установку и завершение TCP-соединения с указанным хостом и различные виды флуда по протоколам HTTP, UDP и UDP over GRE. Также с помощью этого трояна можно было организовать DNS flood и TSource Query Flood, то есть целый комплекс атак на отказ в обслуживании.

При соединении с управляющим сервером троян сообщал ему IP-адрес и MAC-адрес сетевого интерфейса, а также идентификатор архитектуры зараженного устройства. В ответ он ожидал поступления команд, для выполнения каждой из которых создавал форк собственного процесса, имеющий заданную в специальной переменной продолжительность жизни. Если за один цикл поступало сразу несколько команд, они выполнялись по очереди. При этом команды на начало DDoS-атаки могли содержать несколько целей, перечисленных в массиве. Содержимое команды разбиралось специальным парсером, затем зловред в определенном порядке формировал заголовки пакетов, их содержимое (либо при необходимости брал их из конфига), отправлял эти пакеты заданному хосту и после подсчета контрольной суммы переходил к следующей цели. Вся эта процедура продолжалась в непрерывном цикле до тех пор, пока выполняющий команду процесс не завершался по таймеру. В случае DNS-флуда Mirai отправлял на атакуемый DNS-сервер по сто случайных запросов доменных имен за один цикл, что при наличии хотя бы сотни активных ботов гарантированно укладывало сервер на лопатки.

С помощью специального набора команд оператор мог очень гибко настраивать параметры атаки: определять значения TCP-флагов, указывать такие характеристики пакетов, как *maximum segment size* и *timestamp*, отправлять пустые пакеты,

пакеты со случайными значениями или информацией из конфигурации трояна, задавать целевой порт и размер отсылаемых данных, а также управлять прочими TCP-параметрами. Кроме всего прочего, Mirai позволял флудить на заданный узел инкапсулированными пакетами GRE с использованием Transparent Ethernet Bridging. В этом случае встроенный пакет включал в себя полноценный Ethernet-фрейм, в котором MAC-адреса как получателя, так и отправителя Mirai генерировал случайным образом.

В общем, у разработчиков получился довольно мощный DDoS-комбайн, позволяющий реализовывать разные сценарии атак и объединять зараженные умные девайсы с почти любой аппаратной архитектурой в полноценные ботнеты. Один из важных функциональных модулей Mirai — сканер уязвимых хостов, с которыми можно соединиться по протоколу Telnet. Этот сканер был практически полностью позаимствован из исходников Linux-трояна Fgt. В общих чертах алгоритм работы модуля состоит из четырех этапов: поиск уязвимых хостов в сети, перебор логинов и паролей по словарю, после успешного соединения — отправка на скомпрометированный девайс sh-скрипта, который в качестве финального аккорда выкачивает с управляющего сервера троянский бинарный файл под нужную архитектуру.

А теперь по порядку. Для начала сканер создает 256 структур данных, содержащих случайный IP-адрес, параметры сокета и другую полезную информацию. Из списка генерируемых адресов исключаются все локальные и частные диапазоны. Затем с каждым из них троян пытается установить соединение, при возникновении ошибки отсылая на собственный управляющий сервер сообщение. В ответах атакуемого узла сканер ищет строки *user, login, pass, dvrdvs* или *name*, указывающие на требование передачи логина, который он и отправляет, воспользовавшись словарем из конфига. Если логин успешно угадан, на управляющий сервер уходит сообщение *login prompt found* с указанием номера структуры и порядкового номера задачи, в противном случае троян рапортует об ошибке или тайм-ауте. Затем в ответе хоста ищутся строки, указывающие на запрос пароля. Если они обнаружены, хосту передается строка из словаря, а полученный ответ проверяется на наличие значений *invalid, incorrect, fail, again, wrong, accessdenied, error, bad, success, busybox, shell* или *dvrdvs*. Они

свидетельствуют об успехе или провале авторизации. Соответствующий отчет также отправляется на командный сервер. После этого процедура брута повторяется в цикле для всех 256 структур. Таким образом, за каждый проход цикла сканер может «окучить» до 256 случайных IP-адресов.

Если авторизация прошла успешно, на взломанный хост отправляется команда wget, которая выкачивает с управляющего сервера и сохраняет в папку */tmp/* sh-скрипт. Он, в свою очередь, загружает из сети бинарный файл трояна, соответствующий архитектуре устройства. С помощью этого нехитрого алгоритма организуется автоматическое распространение Mirai и заражение новых умных устройств, из которых впоследствии формируется ботнет.

«НАСЛЕДНИКИ» И МОДИФИКАЦИИ

Любой программный продукт эволюционирует со временем: его создатели устраняют выявленные ошибки и понемногу добавляют новые функции. Это в полной мере относится и к Mirai.

В следующей версии, распространявшейся летом 2016 года, создатели трояна отказались от динамического выделения памяти для хранения конфига в пользу статической области — видимо, чтобы повысить стабильность работы программы в целом. Поксоренные строки в конфиге после расшифровки и чтения зашифровывались обратно прямо в памяти. Наконец, был исправлен баг с пропуском имен убиваемых процессов, если те оказывались на границе блока данных, — теперь парсер научился обращаться к ним по индексу, совпадающему с позицией строки в массиве. Был полностью переписан генератор псевдослучайных значений, а при запуске Mirai определял IP-адрес зараженного устройства, постучавшись на DNS-сервер Google (раньше он обращался для этого к собственному управляющему серверу). Троян научился взаимодействовать с обработчиками сигналов и игнорировать SIGINT посредством *sigprocmask* с очевидной целью не дать пользователю принудительно остановить вредоносный процесс.

В новой версии Mirai была реализована функция смены адреса управляющего сервера с помощью SIGTRAP, а поступающие команды стали обрабатываться строго по одной во избежание формирования очереди. Для удобства работы с сокетами в составе трояна появился собственный локальный сервер. Так, берклиевский сокет *PF_INET* биндился на порт *48101* локалхоста и переходил в режим ожидания входящего соединения. Если вредоносу не удавалось создать сокет и подключиться к нему, троян находил владеющий сокетом процесс и убивал его: таким образом исключалось зависание программы при циклических попытках установить связь посредством функции bind с занятым сокетом. Вот почему для защиты от Mirai некоторые специалисты по информационной безопасности рекомендовали заблокировать на устройстве TCP-порт *48101*, если он не используется, — это нарушало нормальную работу встроенного в троян сервера. В обновленной версии была полностью пересмотрена самозащита: вместо костыля с файлом *.shinigami* троян научился определять PID и не убивал процесс, идентификатор которого совпадал с текущим. И без того богатые функции Mirai пополнились возможностью устраивать атаки типа DNS Amplification, зато в списке исчез флуд по протоколу HTTP — впрочем, в одной из следующих версий он вернулся на свое законное место.

Модификация трояна, которую создатели опубликовали на hackforums, несколько отличалась от предыдущих редакций Mirai. Прежде всего, программа обрела способность отключать сторожевой таймер watchdog, чтобы, если система зависнет, она автоматически не перезагрузилась. Чтобы затруднить поиск трояна на устройстве, имя его процесса выбиралось случайным образом в виде последовательности символов латинского алфавита и цифр. Еще один интересный момент: сразу после запуска Mirai удалял с диска собственный исполняемый файл. Наконец, при обнаружении процесса с именем *.anime* троян не просто выгружал его из памяти, но еще и уничтожал соответствующий исполняемый файл — чем-то создатели этого бота явно насолили разработчикам. Помимо всего прочего, в исходниках сентябрьской версии Mirai для архитектуры x86 обнаружился примечательный баг: из-за неверно указанного имени бинарного файла конфигурационная структура для него корректно

не заполнялась, однако затем троян предпринимал попытки чтения из конфига. Но, несмотря на эту ошибку, троян был вполне работоспособным, а богатый арсенал его функций покрывал практически все потребности ботоводов.

На основе выложенных в Сеть исходников Mirai было создано огромное количество клонов, дополнивших базовую разработку другими полезными функциями. Вскоре после утечки в паблик на свет появилась версия Mirai, оснащенная механизмом генерации имен управляющего сервера (DGA), позаимствованным из трояна Ranbyus. Это весьма существенное новшество. Если адрес управляющего сервера жестко зашит в теле вредоносной программы или в ее конфиге, прекращение работы сервера ведет к утрате работоспособности всего ботнета. В случае использования DGA адреса управляющих серверов генерируются трояном автоматически по специальному алгоритму, и при падении одного из них ботнет просто подключается к следующему, что повышает живучесть сети.

В одной из последующих версий Mirai появился локальный прокси-сервер, работающий в отдельном процессе, в других была добавлена функция самообновления, а передаваемые управляющему центру сообщения стали шифроваться с помощью полученного ранее с этого же сервера четырехбайтового ключа. Менялись используемые Mirai порты, чтобы обеспечить дальнейшую работу трояна, если владелец устройства заблокирует ранее известные. Затем Mirai начал использовать эксплоиты для известных уязвимостей в Linux-прошивках IoT-устройств. На основе кода Mirai некоторые энтузиасты попытались разработать собственные версии троянов для Linux вроде широко известного в узких кругах изделия мексиканских программистов под названием BrickBot, таскавшего с собой эксплоиты для уязвимостей CVE-2018-10561 и CVE-2018-10562. Этот троян получил заслуженную популярность тем, что скачивал и запускал на зараженном девайсе shell-скрипт, который порой благополучно «окирпичивал» устройство. Благодаря доступности исходного кода другим троянам для IoT очень сложно тягаться по популярности с Mirai. Конкуренцию вредоносам этого семейства может составить разве что уникальная разработка вирусописателей под названием Hajime.

HAJIME

5 октября 2016 года на одном из ханипотов, принадлежащих исследовательской группе Rapidity Networks, был обнаружен подозрительный трафик. По всем признакам выходило, что в заботливо приготовленную специалистами по информационной безопасности ловушку угодил очередной клон Mirai — как минимум симптомы заражения казались очень похожими. Однако стоило исследователям взглянуть на находку чуть пристальнее, чтобы прийти к выводу: они имеют дело с чем-то принципиально новым.

Вредонос оказался не просто трояном для работающих под управлением Linux умных устройств, а самым настоящим сетевым червем, способным объединять зараженные девайсы в ботнеты. Трой получил имя Hajime — это слово используется в японских единоборствах в качестве команды к началу поединка. Схватка между Linux.Hajime и интернетом вещей получилась увлекательной, но главное — она продолжается до сих пор.

Взлом устройства

Как и в случае Mirai, в архитектуре Hajime используется генератор случайного диапазона IP, из которого исключаются локальные и служебные адреса, после чего полученный массив данных передается сканеру. Тот последовательно стучится на 23-й TCP-порт по каждому из адресов, пробуя установить Telnet-соединение. Если попытка увенчалась успехом, Hajime начинает брутить атакуемый хост с использованием словаря, зашитого в самом трояне. Список логинов и паролей в словаре аналогичен тому, который использует Mirai, разве что к нему добавились пары значений *root/5up* и *Admin/5up*, с помощью которых Hajime атакует ряд моделей роутеров TP-Link и Atheros с дефолтной прошивкой. Главное отличие кроется в том, что Mirai пытается авторизоваться на удаленном устройстве, перебирая логины и пароли в случайном порядке, в то время как Hajime строго следует списку, причем после каждой неудачной попытки авторизации он закрывает текущее Telnet-соединение и создает новое. Список используемых трояном логинов и паролей приведен в следующей таблице.

Логин	Пароль
root	xc3511
root	vizxv
root	klv123
root	root
guest	guest
root	admin
admin	admin
admin	password
root	Zte521
admin	<None>
guest	12345
admin	smcadmin

Подобно разработчикам Mirai, создатели Hajime предполагали, что пользователи умных устройств далеко не всегда меняют заводские настройки, поэтому словарь содержит набор дефолтных логинов и паролей для разных девайсов. Взлом будет успешен только в том случае, если владелец аппарата поленился изменить предустановленные на заводе-изготовителе параметры авторизации.

Если брут удался, Hajime отправляет устройству команду *enable*, чтобы получить доступ к привилегированному режиму интерфейса командной строки. За ней следует команда *system* для перехода в меню системных опций, а затем команды *shell* и *sh* запускают командный интерпретатор. Чтобы проверить, запустился ли нужный для его работы шелл, Hajime передает на атакуемый хост строку */bin/busybox ECCHI*. Специфические оболочки не смогут обработать эту команду, в то время как стандартный *sh* запустит *BusyBox*, который вернет сообщение об ошибке в аргументе — *ECCHI: applet not found*. Это позволит Hajime понять, что он на верном пути.

Исследование устройства

Окончательно убедившись в том, что он попал в Linux-окружение и имеет доступ к командной строке, Hajime начинает исследовать взломанное устройство. Для начала он получает из файла */proc/mounts* список смонтированных файловых систем и ищет открытые на запись папки. Обнаружив первую такую папку, отличную от */proc*, */sys* или */*, Hajime проверяет, действительно ли в нее разрешена запись и не хранится ли уже в ней троянский бинарный файл. В дальнейшем эта папка будет использоваться в качестве рабочей директории.

Затем Hajime исследует заголовок файла */bin/echo*, чтобы определить тип процессора скомпрометированного устройства. В зависимости от аппаратной архитектуры на девайс будет скачан соответствующий ELF-файл, в котором реализован инфектор, доставляющий в систему полезную нагрузку. Hajime поддерживает архитектуры ARMv5, ARMv7, MIPS и, конечно же, Intel x86–64.

Инфектор

Выяснив, какой процессор установлен на взломанном устройстве, Hajime отправляет командному интерпретатору директиву *wget* для загрузки бинарника для соответствующей архитектуры. Этот ELF-файл занимает менее 500 байт и изначально написан на ассемблере. Образцы бинарника, разработанные под различную аппаратную конфигурацию, отличаются друг от друга незначительно, в частности имеют разную структуру *sockaddr* размером 6 байт, в которой сохраняется IP-адрес и номер порта девайса, откуда изначально выполнялся взлом скомпрометированного устройства. В этом и есть одна из особенностей Hajime: адрес для получения полезной нагрузки записан в структуре *sockaddr* самого инфектора, а не определяется динамически.

Инфектор устанавливает TCP-соединение с указанным хостом и принимает оттуда поток байтов. Этот поток перенаправляется на стандартный вывод *stdout* и по конвейеру сохраняется в файл, который будет запущен на выполнение. Так на взломанное устройство попадает основной модуль трояна.

Где-то с середины 2017 года создатели некоторых версий Hajime перестали заморачиваться с ассемблерными инфекторами и вместо этого начали качать полезную нагрузку с помощью Wget или TFTP. Процесс заражения стал проще, но при этом несколько потерял в надежности.

Основной модуль трояна

Запустившись в системе, основной модуль Hajime пытается убить все процессы, имеющие входящие и исходящие соединения с 23-м портом, для чего анализирует содержимое файлов */proc/net/tcp* и */proc/net/tcp6*. Затем троян модифицирует iptables, чтобы перекрыть доступ к портам 7547, 5555, 5358, и удаляет цепочку CWMP_CR, которая используется в части роутеров Movistar:

```
iptables -A INPUT -p tcp --destination-port 7547-j DROP
iptables -A INPUT -p tcp --destination-port 5555-j DROP
iptables -A INPUT -p tcp --destination-port 5358-j DROP
iptables -D INPUT -j CWMP_CR
iptables -X CWMP_CR
```

После инициализации Hajime отправляет NTP-запрос к серверу *pool.ntp.org*, чтобы определить временную зону устройства, а также корректное значение текущей даты. Если запрос не дал результата, используется локальное время. Точное определение времени и даты очень важно для синхронизации ботнета, а некоторые умные устройства, где используются установленные по умолчанию параметры авторизации, имеют неправильную конфигурацию системного времени. Если пользователь не изменил логин и пароль, с чего бы ему менять другие настройки? Создатели Hajime учли этот тонкий момент.

Затем командой unlink троян удаляет собственный файл из системы, после чего при помощи функции *strcpy* меняет символьную строку *argv [0]*, в которой хранится имя программы, на *telnetd*. Наконец, с использованием системного вызова *prctl (PR_SET_NAME, argv [0])* он меняет имя своего процесса. Таким хитрым способом Hajime пытается замаскироваться под стандартный демон Telnet, чтобы не вызывать у пользователей подозрений. Дальше управление передается модулю, который отвечает за работу DHT-протокола Kademlia,

предназначенного для организации одноранговых децентрализованных файлообменных сетей. Непосредственно для приема и передачи данных ботнет использует транспортный протокол uTorrent. Это, в частности, позволяет зараженным девайсам успешно работать под NAT.

Ботнет

Маршрутизация в ботнете Hajime базируется на модифицированном проекте KadNode, который поддерживает шифрование и инфраструктуру открытых ключей (PKI). Передаваемые файлы сжимаются при помощи модифицированного алгоритма LZ4, но некоторые файлы могут транслироваться и в несжатом виде.

После инициализации протокола Hajime устанавливает соединение с пирами Torrent-ботнета и скачивает актуальный конфигурационный файл. Для опознавания пиров в сети используются уникальные идентификаторы ботов, генерируемые на основе текущей даты и хеша SHA-1, полученного от имени файла трояна. Наличие свежего конфигурационного файла на других узлах ботнета Hajime проверяет с интервалом в десять минут. Типичный конфиг содержит обозначение процессорной архитектуры, для которой собраны исполняемые файлы, имена этих файлов и timestamp, позволяющий трояну определить их версию. Если файл в сети свежее того, информация о котором сохранена в локальной конфигурации вредоноса, он скачивает бинарник для соответствующей аппаратной конфигурации и запускает его в качестве своего дочернего процесса. Аналогичным образом работает самообновление Hajime. P2P-ботнет, созданный по такой схеме, получается одноранговым, а значит, децентрализованным и отказоустойчивым. Он не зависит от наличия управляющих серверов, следовательно, не прекращает свою деятельность, если какое-то количество инфицированных устройств вдруг «вылечится» или перестанет работать. Да и перехватить управление таким ботнетом физически невозможно.

Обосновавшись в системе, Hajime запускает цикл генерации и опроса IP-адресов, чтобы продолжить заражать уязвимые сетевые устройства. При этом сам хост выступает в роли

сервера, с которого скачивается исполняемый файл инфектора и тело троянца. Если для заражения какого-то удаленного хоста требуется файл с поддержкой другой аппаратной архитектуры, Hajime может подтянуть его из пиринговой сети.

ЦЕЛИ И ВЫВОДЫ

В Hajime по умолчанию не предусмотрены какие-либо деструктивные функции за исключением одной: троян может скачивать и запускать на инфицированном устройстве любые приложения. Ключевое слово здесь — «любые». Поэтому для оператора не составит никакого труда при необходимости установить на все зараженные устройства модуль для реализации DDoS-атак, бэкдор, майнер или просто продавать инсталлы всем желающим, зарабатывая за счет других вирусописателей. Иными словами, готовый ботнет можно монетизировать множеством различных способов. Но как бы то ни было, о назначении трояна до сих пор строятся догадки и предположения.

Существует несколько методов защиты от Hajime, Mirai и им подобных опасных программ. Можно закрыть на потенциально уязвимом устройстве порт 4636, через который троян качает полезную нагрузку. Можно блокировать все входящие соединения на порт 23, если в запросе присутствует строка */bin/busybox ECCHI* — явный индикатор атаки. Но лучше всего правильно настроить параметры авторизации по протоколам Telnet и SSH, используя сложные пароли: это защитит девайс от брута по словарю, который применяют Hajime и Mirai, а владельцу такого устройства позволит сберечь нервы.

ГЛАВА 6.
БОТНЕТЫ

Термином «ботнет», или «бот-сеть», принято обозначать сети, созданные с использованием автономно действующих дистанционно управляемых вредоносных программ (ботов). Такие сети могут выполнять централизованно отдаваемые киберпреступниками команды и имеют различную архитектуру: как с использованием удаленных командных серверов (Command and Control Server, C&C), так и без них. Злоумышленники строят ботнеты с различными целями. Как правило, это рассылка спама, организация массированных атак на отказ в обслуживании (DDoS-атаки) или загрузка других вредоносных приложений на инфицированные компьютеры (сети, состоящие из троянцев-загрузчиков). Некоторые ботнеты способны выполнять сразу несколько деструктивных функций. Владельцы бот-сетей обычно предлагают свои услуги другим злоумышленникам на различных подпольных форумах, извлекая таким образом из своей деятельности прибыль. Иными словами, как сами ботнеты, так и предоставляемые ими нелегальные услуги зачастую становятся объектом купли-продажи.

ИСТОРИЯ ВОПРОСА

В их современном виде бот-сети возникли на рубеже «нулевых» годов XXI века, хотя и до этого вирусописатели предпринимали успешные попытки собрать в единую систему зараженные компьютеры, доступные для удаленного управления — например системы, инфицированные *бэкдорами*. Исторически первым массовым ботнетом принято считать бот-сеть, созданную злоумышленниками в 2004 году с использованием почтового червя Beagle, заразившего порядка 230000 компьютеров по всему миру. Червь Beagle мог инфицировать все существовавшие на тот момент версии Microsoft Windows и с использованием собственной реализации механизма SMTP рассылал свои копии по электронной почте. Beagle обладал руткит-модулем, способным скрывать собственное присутствие в системе, а также умел завершать процессы некоторых антивирусных программ.

В 2006 году заявил о себе массовый ботнет Rustock, предназначенный для рассылки спама. Инфицированные Rustock компьютеры могли рассылать до 25000 почтовых сообщений в час, а в периоды максимальной активности — до 192 писем в минуту. Бот-сеть успешно действовала до 2011 года, когда была практически полностью уничтожена в результате тщательно спланированной кампании, проведенной экспертами из корпораций Microsoft, FireEye и специалистами из университета Вашингтона в сотрудничестве с федеральными агентами безопасности США.

С тех пор ботнеты различного функционального назначения стали появляться на свет с завидной регулярностью. 2007 год отметился появлением сети Cutwail (также известен как Bulknet). По различным подсчетам, Cutwail заразил в общей сложности порядка 1,5 млн компьютеров по всему миру. Этот ботнет также был предназначен для рассылки спама и имел довольно-таки

простую структуру: действующие на зараженном компьютере вредоносные программы соединялись напрямую с командным сервером и получали оттуда все данные, необходимые для рассылки рекламных писем, а после успешного выполнения задачи боты передавали злоумышленникам отчет, содержащий подробную статистику о проделанной «работе».

2008 год стал периодом бурного распространения червя Conficker (также известного под именем Kido), заразившего в общей сложности более 10,5 млн компьютеров (по другим данным — порядка 12 млн) в 200 странах мира. По сей день этот ботнет считается одним из наиболее масштабных. Используя уязвимости в операционных системах семейства Microsoft Windows, червь загружал себя на атакуемый компьютер из Интернета, а также распространялся с использованием съемных носителей информации. В феврале 2009 года корпорация Microsoft объявила награду в 250000 долларов за любую информацию о создателях этого червя. Кроме того, данная вредоносная программа обладала возможностью получать собственные обновления непосредственно с других зараженных узлов, минуя управляющий сервер, что значительно затрудняло борьбу с ним.

Начиная с 2010 года стали возникать бот-сети, состоящие из компьютеров, инфицированных банковскими троянцами (в частности, TDL / TDSS). В 2011 году получил широкое распространение файловый вирус Rmnet (Ramnit), также способный самоорганизовываться в ботнеты и инфицировавший в общей сложности порядка 3 млн компьютеров (по другим данным — более 5 млн).

Злоумышленники продолжают создавать и активно эксплуатировать ботнеты. В частности, в 2014 году киберпреступники встроили вредоносную программу в приложение, предназначенное для поисковой оптимизации сайтов, в результате чего сумели создать бот-сеть для рассылки спама, состоящую из более чем 250 тыс. зараженных узлов. Не обделяют своим вниманием злоумышленники и рынок портативных платформ. Еще в 2010 году был зафиксирован первый ботнет, состоящий из мобильных устройств под управлением ОС Android, инфицированных троянцем Geinimi. А всего пару лет спустя, в 2013 году, специалисты компании «Доктор Веб»

зафиксировали крупнейший мобильный ботнет, состоящий из смартфонов и планшетов, инфицированных троянцами семейства Android.SmsSend — заражению подверглись 200000 с лишним устройств, при этом более 128000 из них принадлежали российским пользователям.

Поскольку эксплуатация ботнетов приносит злоумышленникам определенную прибыль, можно предположить, что этот вид незаконного бизнеса будет процветать и в дальнейшем.

АРХИТЕКТУРА БОТНЕТОВ

Простые ботнеты

Исторически самые первые бот-сети имели довольно примитивную структуру. Все вредоносные программы, инфицировавшие компьютеры своих жертв, подключались к единому управляющему серверу, с которого получали конфигурационные файлы, необходимые им для дальнейшей работы, обновления и собственно команды для последующего выполнения. В качестве управляющего узла мог выступать и чат-сервер, предназначенный для организации обмена текстовыми сообщениями по протоколу IRC (Internet Relay Chat) — этой технологией, в частности, пользовались троянцы семейства BackDoor.IRC.Bot. Боты этого типа соединялись с соответствующим IRC-сервером, подключались к определенному чат-каналу и начинали «слушать» его в ожидании входящих команд. Адрес управляющего сервера был, как правило, «зашит» в теле самого бота и в лучшем случае был зашифрован в целях затруднить его поиск и анализ. Бот расшифровывал адрес командного центра непосредственно во время своего выполнения. Таким образом, ранние ботнеты имели звездообразную структуру и обладали существенным архитектурным недостатком: при отключении или перехвате управляющего сервера вся бот-сеть становилась нефункциональной, и усилия, приложенные злоумышленниками на распространение вредоносного ПО, оказывались затраченными впустую.

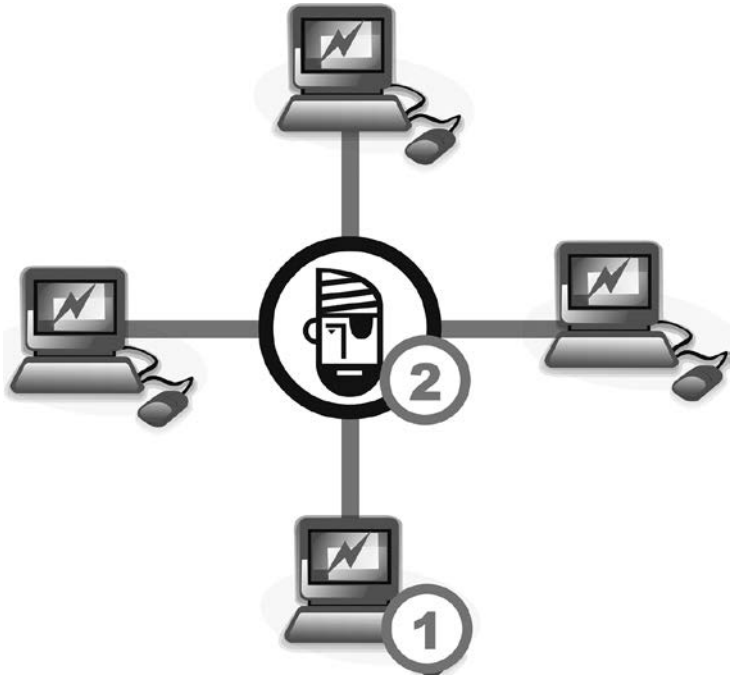


Рис. 17. Первые ботнеты имели простую звездообразную структуру: 1 — зараженные компьютеры; 2 — управляющий сервер

Ботнеты, использующие DGS

Следующим эволюционным этапом в развитии бот-сетей стало появление *технологии генерации доменных имен контрольно-управляющих серверов (DGS, domain generation system)*. Создана эта система была в первую очередь в целях борьбы с противодействием работе бот-сетей со стороны антивирусных компаний и для повышения «живучести» ботнетов.

Вместо адреса контрольно-управляющего сервера или списка таких адресов в состав ботов злоумышленники стали включать специальный алгоритм, генерирующий адрес управляющего сервера по определенной схеме «на лету». Таким образом формировался перечень подобных адресов. Далее бот опрашивал полученные адреса по очереди, и, если с одного из них

приходил заранее обусловленный ответ, этот сервер становился для бота управляющим. Например, некоторые троянцы составляли адреса серверов из последовательности латинских символов и цифр, добавляя к ним обозначение домена первого уровня *.com* или *.org*. Зная используемый вредоносной программой алгоритм генерации имен, злоумышленник мог оперативно зарегистрировать такой домен для управляющего сервера. Если этот домен по каким-либо причинам оказывался заблокированным, киберпреступник попросту регистрировал новый.

Отчасти задачу владельцам ботнетов упрощало и то обстоятельство, что некоторые крупные зарубежные регистраторы доменов в рекламных целях практикуют отсрочку платежей — зарегистрированный домен становится доступным для использования сразу, а оплату за него можно внести спустя определенный срок, например через месяц — в случае отказа от платежа делегирование домена автоматически прекращалось. Используя эту особенность, некоторые киберпреступники регулярно меняли домены управляющих серверов для своих ботнетов, не платя никому ни цента. Кроме того, даже в случае утраты доступа к управляющему серверу (например, из-за его блокировки хостинг-провайдером по жалобе пользователей или антивирусной компании) злоумышленнику ничто не мешает в считанные минуты развернуть еще один управляющий сервер на новой площадке и прилинковать к нему новый (или уже существующий) домен.

Некоторые владельцы бот-сетей использовали сразу несколько параллельных командных центров, разбивая ботнет на отдельные независимо управляемые подсети, что также повышало их устойчивость к различным внешним воздействиям. Таким образом, структура бот-сетей, построенных с применением технологии DGS, немного усложнилась.

С одной стороны, DGS сделала бот-сети более управляемыми и устойчивыми к внезапному отказу командных центров, с другой — значительно упростила их перехват с использованием метода *sinkhole* (о нем мы побеседуем чуть позже). В целях борьбы с этим методом перехвата управления ботнетами злоумышленники стали заметно усложнять механизм коммуникации бот-сети с управляющим центром: протоколы обмена данными стали активно использовать шифрование, управляющие

сервера для подтверждения своей подлинности обменивались с вредоносной программой специальным образом сформированной цифровой подписью. Например, один из представителей семейства вирусов Win32. Virut способен генерировать до 100 адресов управляющих серверов в сутки. Последовательно опрашивая их, вирус ожидает поступления ответного пакета, содержащего цифровую подпись домена. Если проверка цифровой подписи проходит успешно, бот считает сервер с таким доменом управляющим.

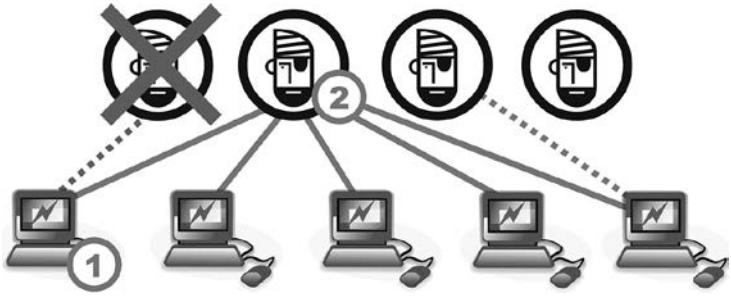


Рис. 18. Структура ботнета, построенного с использованием технологии генерации доменных имен контрольно-управляющих серверов (DGS): 1 — инфицированные компьютеры; 2 — управляющие серверы

P2P-ботнеты

Еще одной категорией бот-сетей являются так называемые *P2P (Peer-To-Peer)*, или *пиринговые одноранговые сети*. Такие сети вовсе не используют управляющих серверов, вместо этого они «общаются» с другими инфицированными компьютерами, передавая команды по сети от «точки к точке». Одноранговые бот-сети являются децентрализованными и потому их невозможно вывести из строя, уничтожив одним метким «ударом» управляющий сервер — за полным отсутствием такового. Ярким примером пиринговой вредоносной сети можно считать ботнет, созданный киберпреступниками с использованием файлового вируса Win32. Sector, заразившего в общей сложности более миллиона компьютеров. Этот вирус может загружать из P2P-сети и запускать на зараженной машине другие

вредоносные программы, останавливать работу некоторых антивирусов и предотвращать пользователям зараженной машины доступ к сайтам их разработчиков.

Общеизвестно, что подключенный к интернету компьютер может иметь собственный внешний IP-адрес, либо не иметь его в случае, если в его локальной сети используется механизм NAT (Network Address Translation). NAT позволяет передавать пакеты за пределы локальной сети, направляя их от узлов с «внутренним» IP-адресом, недоступным из интернета, внешнему получателю и обратно путем замены в заголовке пакетов «внутреннего» IP-адреса на реальный адрес шлюза или роутера, через который осуществляется подключение всей локальной сети к интернету.

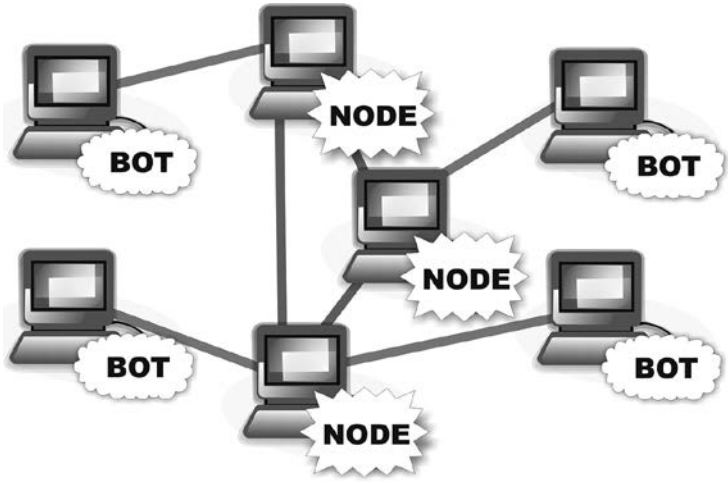


Рис. 19. Организация одноранговой бот-сети на примере Win32. Sector: «Bot» — зараженные компьютеры, не имеющие внешнего IP-адреса; «Node» — зараженные компьютеры, имеющие внешний IP-адрес и выполняющие функции маршрутизатора для компьютеров «Bot»

Заразив компьютер, Win32. Sector проверяет, имеет ли он внешний IP-адрес или нет. Если имеет, такой бот, условно называемый «Node», начинает играть роль маршрутизатора для других зараженных машин («Bot»), не имеющих реального внешнего IP-адреса. Инфицированные компьютеры типа «Bot»

начинают «общаться» с интернетом и другими зараженными узлами через него. Каждый инфицированный узел такой сети получает начальный список из 100 IP-адресов других ботов, с которыми он пытается установить соединение, причем этот список периодически обновляется. При подобной структуре сети потеря одного «Node» (например, если владелец компьютера вылечит его от вируса) ничем не грозит всей системе в целом: «Bot» просто подключится к другому «Node» для получения дальнейших управляющих команд. Так, на 20 мая 2014 года специалисты компании «Доктор Веб» насчитали в ботнете Win32. Sector 1197739 зараженных компьютеров, из них внешний IP-адрес имели только 109 783.

Ботнеты смешанного типа

Примером более сложной бот-сети смешанного типа может служить ботнет Trojan.Dridex.49. Этот троянец умеет встраиваться в запущенные на инфицированном компьютере процессы, а все сообщения, которыми он обменивается в сети, шифруются. Основное назначение этих вредоносных программ — выполнение *веб-инжектов*, при помощи которых они могут похищать различную конфиденциальную информацию, в том числе получать доступ к системам дистанционного банковского обслуживания.

Для связи с управляющими серверами Trojan.Dridex.49 использует сложную по своей архитектуре одноранговую ботсеть, состоящую из двух промежуточных слоев прокси. Заразив компьютер, Trojan.Dridex.49 может принять на себя одну из трех возможных ролей:

- роль «Bot» — троянцы этого типа работают на компьютерах, не имеющих внешнего IP-адреса. «Bot» осуществляет связь с управляющим сервером через троянцев с ролью «Node»;
- роль «Node» — троянцы этого типа работают на компьютерах, имеющих внешний IP-адрес, и передают данные от троянцев с ролью «Bot» троянцам с ролью «Admin Node», а также в обратном направлении;
- роль «Admin Node» — троянцы этого типа работают на компьютерах, имеющих внешний IP-адрес, и осуществляют

связь друг с другом, а также непосредственно с управляющим сервером.

Цепочка связи инфицированного компьютера, не имеющего внешнего IP-адреса, с управляющим сервером ботнета, выглядит в общем случае следующим образом: Bot -> Node -> Admin Node -> другие Admin Node -> Управляющий сервер. При этом в целях обеспечения безопасности соединения троянцы обмениваются между собой цифровыми ключами. Схематически структура этой сети показана на рис. 20.

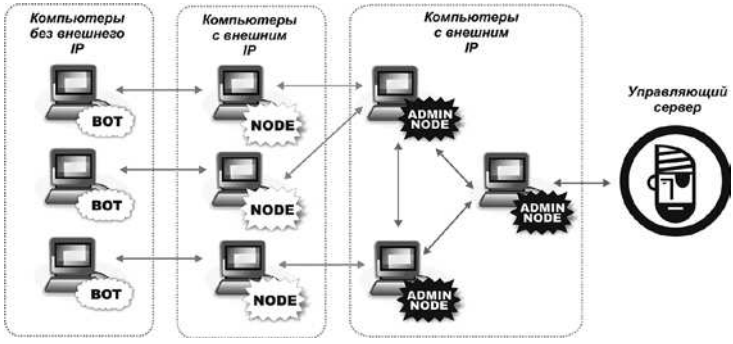


Рис. 20. Архитектура бот-сети смешанного типа на примере Trojan.Dridex.49

Так, чтобы получить с управляющего сервера новый перечень IP-адресов ботов с ролью «Node» или конфигурационных данных, необходимых для выполнения веб-инжектов, троянцы с ролью «Bot» передают запрос троянцу с ролью «Node», тот переправляет его троянцу с ролью «Admin Node», который, в свою очередь, может перебрасывать его другим «Admin Node» до тех пор, пока запрос не достигнет управляющего сервера. Передача запрошенных данных от управляющего сервера троянцу «Bot» осуществляется в обратном иерархическом порядке. Аналогичным образом осуществляется доставка дополнительных функциональных модулей вредоносной программы: «Node» запрашивают их у управляющего сервера через «Admin Node», а «Bot» получают эти модули по запросу у троянцев с ролью «Node». Такая многоуровневая и запутанная система значительно затрудняет отслеживание управляющих серверов

бот-сети и перехват информации, которой обмениваются инфицированные машины.

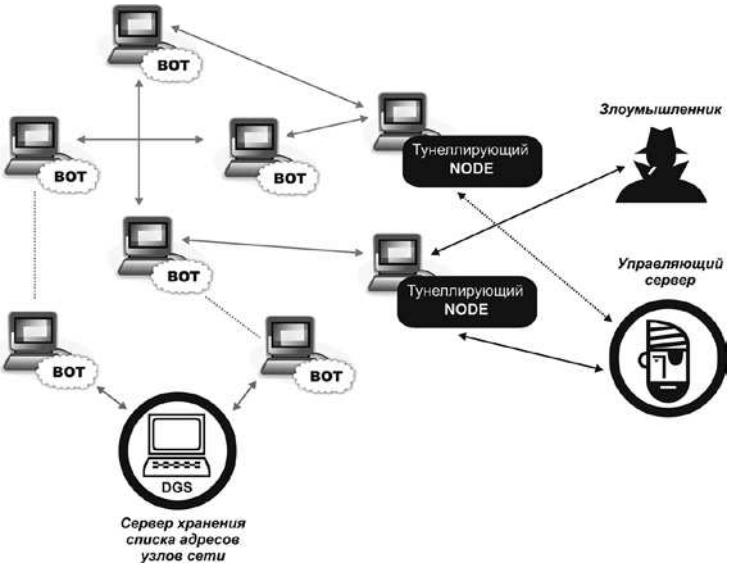


Рис. 21. Схема сложной комбинированной P2P-сети на примере банковского троянца Zeus

Некоторые современные вредоносные программы, например определенные модификации банковского троянца Zeus, используют еще более сложную и изощренную структуру P2P-ботнета с применением туннелирования. Например, запустившись на целевом компьютере, такой троянец может расшифровать из собственного конфигурационного файла список других ботов с ролью «Node» и попытаться установить с ними соединение, а если наладить связь не удалось, он сгенерирует перечень доменов с использованием алгоритма DGS и попытается получить актуальный список «Node» — узлов ботнета оттуда. При этом ресурсы на доменах, содержащие списки узлов сети, не являются управляющими серверами — они просто хранят данные перечни IP-адресов «Node» — узлов, которые отдают ботам по запросу. Связь с истинным управляющим сервером реализуется в такой сети через отдельные «Node» — узлы,

работающие как туннелирующий сервер. Сам злоумышленник, управляющий ботнетом, также передает на управляющий сервер список команд не напрямую, а через туннель — в целях затруднения своей локализации и идентификации. Общая схема такой сети показана на рис. 21.

Ботнеты с использованием TOR и «облаков»

В последнее время в связи со стремительным развитием антивирусных технологий злоумышленники начали размещать управляющие серверы ботнетов в сети TOR, а также активно использовать облачные технологии.

TOR (от англ. The Onion Router —«луковый маршрутизатор») — система так называемой «луковой маршрутизации», состоящая из многослойной структуры прокси-серверов и позволяющая устанавливать анонимное соединение, защищенное от стороннего прослушивания и слежения. TOR реализует модель анонимных виртуальных тоннелей с использованием шифрования и благодаря распределенной сети узлов связи позволяет пользователю сохранять приватность, препятствуя работе всевозможных механизмов перехвата и анализа трафика.

Структура сети TOR позволяет создавать скрытые веб-сервисы в псевдодоменной зоне.onion, к которой имеют доступ только пользователи TOR-сети и в которой невозможно отследить реальное физическое местоположение сервера. Этим и пользуются злоумышленники, размещая в сети TOR управляющие серверы ботнетов. В случае отсутствия на инфицированной машине TOR-клиента вирусописатели используют для обмена данными с командным центром соответствующие шлюзы, например сервис TOR2WEB.

В частности, управляющий сервер в сети TOR использовал ботнет Mevade (альтернативное наименование — Sefnit), а также довольно крупный ботнет Skynet, предназначенный для проведения массированных DDoS-атак и добычи (майнинга) криптовалют. По данным различных источников, в 2012 году в эту бот-сеть входило порядка 12-15 тыс. зараженных компьютеров.

Активно используют управляющие серверы в TOR и различные троянцы-энкодеры, осуществляющие через «скрытую

сеть» обмен ключами шифрования и принимающие оплату за расшифровку файлов в Bitcoin с помощью сайта, размещенного в TOR-сети на домене.onion. К таким энкодерам относится, например, широко распространенный Critroni (альтернативное название — CTB-Locker).

Для затруднения обнаружения и фильтрации вредоносного трафика ботнета вирусописатели все чаще стали использовать в своих целях облачные технологии и социальные сети. Так, еще в 2008 году были выявлены троянцы, получающие команды с определенной учетной записи в социальной сети Twitter и в специально созданной вирусописателями ветке открытых дискуссионных форумов Google Groups.

Для хранения компонентов вредоносных программ, которые могут быть загружены и установлены на инфицированной машине, киберпреступники нередко используют облачные сервисы Google Docs и Google Drive. Кроме того, известны случаи задействования публичных облачных сервисов для организации коммуникации вредоносной программы с ее командным центром. Например, обнаруженная в 2012 году вредоносная программа-бэкдор BackDoor.Makadoc использовала для обмена данными со своим управляющим сервером облачный сервис Google Docs. Встроенное в этот сервис веб-приложение Google Docs Viewer обрабатывает все входящие запросы как обычный открытый прокси-сервер, при этом такие запросы не вызывают «подозрений» у встроенного брандмауэра Windows, чем и воспользовались злоумышленники. В процессе обработки запросов Google Docs Viewer даже не требует у клиента обязательной авторизации в сервисе Google Docs, просто перенаправляя их на командный сервер, что еще более облегчило задачу киберпреступникам.

Нетрадиционные схемы

Помимо описанных выше «традиционных» способов организации взаимосвязи зараженного компьютера и управляющего сервера, злоумышленники порой изобретают более замысловатые схемы, проявляя недюжинную фантазию. Одним из примеров может служить ботнет, в который на конец сентября 2014 года входило порядка 18,5 тыс. инфицированных

компьютеров Apple с уникальными IP-адресами, работающих под управлением операционной системы Mac OS X. Причиной заражения стал троянец Mac.BackDoor.iWorm, активно использующий в своей работе различные криптографические алгоритмы.

Для получения списка адресов управляющих серверов Mac.BackDoor.iWorm обращался с запросом к встроенной поисковой системе популярного публичного новостного сайта reddit.com, в котором зарегистрированные пользователи могут оставлять различные ссылки и краткие сообщения. Поисковая строка включала в себя шестнадцатеричные значения первых 8 байт хэш-функции MD5 от текущей даты. В ответ поисковая система Reddit выдавала троянцу ссылки на оставленные злоумышленниками комментарии со списком управляющих серверов ботнета и портов для установки соединения.

Mac.BackDoor.iWorm последовательно перебирал полученный список, отправляя запросы в случайном порядке на первые 29 адресов из этого перечня, и обменивался с удаленными узлами пакетами данных, проверяя с помощью сложного математического алгоритма подлинность каждого сервера. Если проверка завершалась успешно, троянец сообщал на удаленный узел свои идентификационные данные и ожидал от него поступления управляющих команд.

Еще один бэкдор, известный под именем BackDoor.Zetbo.1, на сей раз — угрожающий пользователям Windows, использовал не менее оригинальный способ получения конфигурационных данных, необходимых ему для осуществления вредоносной деятельности. Управляющий сервер, с которого загружала свои параметры эта поделка турецких вирусописателей, внешне выглядел вполне безобидно: на нем размещалось несколько веб-страниц, содержащих графические кнопки с различными гиперссылками. Анализируя HTML-код этих кнопок, троянец вычленял из него значения определенных атрибутов HTML-тегов и путем ряда преобразований получал из них всю интересующую информацию, то есть осуществлял *парсинг* (синтаксический разбор) кода веб-страницы. Для стороннего же наблюдателя данный сайт не представлял никакой угрозы и не вызывал подозрений — внешне он выглядел, как самая обычная веб-страничка, которых в интернете сотни миллионов.

Схожим образом действовал один из троянцев-загрузчиков, активно распространявшихся осенью 2012 года, — с его помощью, в частности, на компьютеры пользователей попадали банковские троянцы семейства Zeus (Zbot, Trojan.PWS.Panda). Заразив атакуемый компьютер, загрузчик расшифровывал из своего тела список адресов удаленных веб-сайтов и обращался к каждому из них по протоколу HTTPS. В ответ троянец получал хранящуюся в корневой папке каждого сайта веб-страницу и осуществлял ее парсинг в поисках тега вставки изображения: *<img src=«data:image / jpeg;base64… >*. В качестве аргумента этого тега веб-страница содержала зашифрованный исполняемый код вредоносной программы, который загрузчик извлекал, расшифровывал и запускал на исполнение на инфицированной машине.

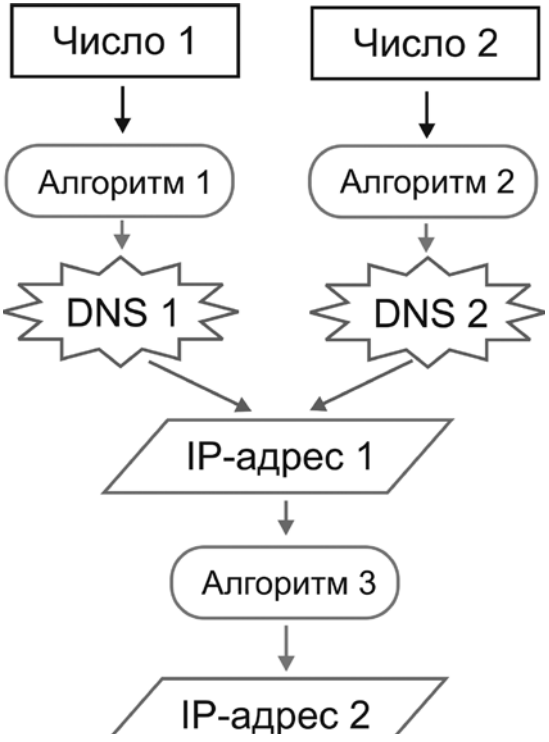


Рис. 22. Алгоритм генерации имени командного сервера троянцем Linux.Sshdkit

Другая троянская программа, Linux.Sshdkit, предназначенная для заражения работающих под управлением Linux серверов, использовала следующий алгоритм получения имени командного сервера. Троянец по специальному алгоритму генерировал два IP-адреса, которые являлись адресами принадлежащих злоумышленникам DNS-севреров. Если оба этих DNS-севрера при обращении к ним ссылались на один IP-адрес, этот адрес с помощью специального математического алгоритма преобразовывался в другой IP-адрес. Он-то и являлся истинным адресом командного центра, на который вредоносная программа загружала всю похищенную на зараженном сервере информацию.

КОМАНДНАЯ СИСТЕМА БОТНЕТОВ

Как уже упоминалось ранее, некоторые боты, фактически совмещающие в себе функции бэкдоров, могут транслировать поступающие извне директивы встроенному командному интерпретатору операционной системы, другие используют собственный набор команд, при этом весь передаваемый в бот-сети трафик обычно шифруется.

Наиболее часто административный интерфейс управляющего сервера ботнета реализован в виде набора сценариев на языке PHP или с использованием другого скриптового языка, и в целом напоминает административный раздел стандартных систем управления контентом (CMS, Content Management Systems). Авторизовавшись на сервере, злоумышленник получает доступ к статистике ботнета — на специальной страничке, как правило, отображается общее количество установленных и активных ботов, перечень последних выполненных ботами команд, данные об успешности / провале завершения этих операций, в некоторых случаях (если ботнет связан с какой-либо «партнерской программой») — полученный злоумышленниками совокупный доход. В отдельном разделе «админки» киберпреступник может отдать ботам соответствующие команды, просто набрав их в специальной экранной форме.

Однако способ трансляции команд ботнету может значительно различаться в зависимости от используемого бот-сетью

для передачи данных протокола. Все команды можно условно разделить на две категории:

- директивы, общие для всего ботнета;
- команды, отдаваемые отдельным подсетям или ботам.

В частности, практически все ботнеты поддерживают команду на загрузку с удаленного сервера и установку обновлений ботов, получение конфигурационных данных: спам-боты загружают в качестве таковых текст рассылаемых сообщений и адреса почтовых серверов для отсылки писем, предназначенные для выполнения веб-инжектов троянцы — встраиваемый в веб-страницы код и список узлов, при установке соединения с которыми осуществляется веб-инжект. Ботнеты, предназначенные для проведения массированных атак на отказ в обслуживании (DDoS-атак) могут получать команду на начало или окончание атаки определенного типа с использованием поддерживаемых вредоносной программой протоколов. Большинство современных DDoS-бот-сетей обладают возможностью осуществлять следующие виды атак.

- SYN Flood — отсылка определенным образом сформированных пакетов на атакуемый узел до тех пор, пока тот не перестанет отвечать на запросы.
- UDP Flood — установка ботом соединения с атакуемым сервером по протоколу UDP, после этого бот отправляет большое количество «мусорных» дейтограмм.
- Ping Flood — формирование троянцем эхо-запроса с использованием протокола ICMP, который направляется с определенным временным интервалом на атакуемый узел.
- DNS Amplification — рассылка троянцами массированных запросов на указанные злоумышленниками серверы DNS с целью вызвать их отказ.
- NTP Amplification — отправка массированных запросов на серверы NTP с целью вызвать их отказ.

Большинство банковских троянцев может выполнять команду на полное уничтожение операционной системы на инфицированном компьютере: в результате после успешного хищения средств с банковского счета компьютер жертвы перестает загружаться. Это дает злоумышленникам временную фору до того момента, пока пострадавший от действия

вредоносной программы пользователь не начнет бить тревогу и обратится в банк с требованием заморозить несанкционированную транзакцию.

Существуют и оригинальные виды команд, которые могут выполнять некоторые вредоносные программы. Например, предназначенный для рассылки спама и организации DDoS-атак многокомпонентный троянец Trojan.Tofsee обладает некоторыми функциями антивируса: он умеет искать на диске зараженного компьютера файлы по полученному с управляющего сервера списку, отслеживать запущенные в Windows процессы и проверять содержимое системного реестра с одной любопытной целью — отыскать и уничтожить конкурирующие угрозы. Таким образом, создатели троянца реализуют своего рода естественный отбор в экосистеме инфицированной машины — устраняя «чужие» вредоносные программы, так или иначе проявляющие активность, Trojan.Tofsee борется за собственную «выживаемость». Следует отметить, что это далеко не единственный подобный случай: специалистам антивирусных компаний известно множество вирусов и троянцев, способных устранять на зараженном ПК другие работающие вредоносные программы в целях конкурентной борьбы за аппаратные ресурсы машины, а также в попытках скрыть от пользователя сам факт заражения.

МЕТОДИКА ПЕРЕХВАТА УПРАВЛЕНИЯ БОТНЕТАМИ (SINKHOLE)

Как уже упоминалось ранее, технология DGS обладает рядом архитектурных особенностей, позволяющих специалистам по информационной безопасности успешно перехватывать управление ботнетами. При этом под «перехватом» понимается получение полного управления и контроля над бот-сетью с возможностью не только осуществлять мониторинг и отслеживать состояние сети, но и отдавать команды ботам. Один из наиболее распространенных методов носит наименование *sinkhole* (англ., «выгребная яма»).

Суть метода кроется в самой реализации механизма DGS. Исследовав образец вредоносной программы методом

реверс-инжиниринга, то есть дизассемблировав ее и изучив исходный код, аналитики получают в свое распоряжение алгоритм, с использованием которого бот генерирует адреса управляющих серверов. Зарегистрировав несколько таких адресов, специалисты создают с их помощью собственные управляющие серверы ботнета, способные в ответ на запрос вредоносной программы отправить ей корректный отклик и успешно пройти проверку на подлинность (всю необходимую информацию, как правило, можно почерпнуть из декомпилированного и расшифрованного кода самой вредоносной программы). После этого достаточно лишь тем или иным способом ликвидировать действующие управляющие центры бот-сети. Утратив связь с командным сервером, боты начинают генерировать адреса альтернативных управляющих серверов с использованием уже известного специалистам по информационной безопасности алгоритма DGS, устанавливают связь с принадлежащим им «поддельным» управляющим центром и после успеха этой операции прекращают поиски альтернативных командных узлов. Управление ботнетом захвачено.

В частности, с использованием этого метода сотрудникам российской антивирусной компании «Доктор Веб» удалось перехватить управление несколькими подсетями ботнета Rmnet, бот-сетью Linux.Sshdkit, а также угрожавшим пользователям компьютеров Apple обширным ботнетом BackDoor. Flashback.39.

Почему специалисты по информационной безопасности в некоторых случаях предпочитают перехватить у злоумышленников управление бот-сетью, вместо того чтобы просто уничтожить ее управляющие центры? Потому что в большинстве случаев это не принесет никакого результата: злоумышленники оперативно запустят новые командные серверы, а сами зараженные компьютеры можно избавить от угрозы только в том случае, если их владельцы осознают необходимость установки и своевременного обновления антивирусных программ — убедить в целесообразности этого шага несколько миллионов пользователей инфицированных машин, очевидно, невозможно. В то же самое время перехват управления ботнетом позволяет «заморозить» его, предотвратив дальнейшую вредоносную деятельность сети: злоумышленники больше не могут отдавать

команды ботам, и, следовательно, использовать бот-сеть в своих противоправных целях.

В случае с бот-сетями, использующими пиринговые технологии, использование описанного здесь метода, безусловно, невозможно. Вместе с тем, у специалистов по информационной безопасности остается возможность отправлять таким ботам различные команды, например, запрос, в ответ на который вредоносная программа вернет список IP-адресов других зараженных машин, что позволяет оценить размеры ботнета и его географию. Также можно передать ботам список IP-адресов, принадлежащих вирусным аналитикам компьютеров, выполняющих в такой бот-сети функцию маршрутизаторов (узлов Node) с тем, чтобы к ним обращались инфицированные машины, не имеющие выделенного IP-адреса или расположенные в сети, использующей NAT. Это нельзя назвать «перехватом управления» бот-сетью, однако подобный метод позволяет получить достоверное представление о масштабах распространения угрозы.

ГЛАВА 7. ТЕХНОЛОГИИ ПРОНИКНОВЕНИЯ

Существует несколько возможных путей распространения вредоносных программ, причем все они известны довольно давно и хорошо изучены, однако это ничуть не уменьшает масштабов вирусных эпидемий и числа заражений пользовательских устройств троянцами и вирусами. В целом можно сказать, что самая распространенная и основная причина вирусных заражений — это неграмотность самих пользователей, которые не только не желают соблюдать очевидные меры безопасности, но даже не подозревают о том, в каких ситуациях их устройство может быть инфицировано. Давайте кратко рассмотрим основные пути распространения вредоносных программ и дадим им соответствующую характеристику.

СМЕННЫЕ НОСИТЕЛИ ИНФОРМАЦИИ

Файловые вирусы и черви умеют распространяться самостоятельно, заражая переносные накопители информации — флэшки, sd-карты, даже мобильные телефоны и планшеты, подключаемые к зараженному компьютеру для обмена файлами, например, чтобы скачать на ПК фотографии, или, наоборот, закачать на переносное устройство музыку. Впоследствии достаточно подключить такое инфицированное устройство к компьютеру, и его заражение с большой долей вероятности произойдет автоматически. Выбор устройства для заражения троянец обычно осуществляет путем перечисления имен дисков, содержащих значение «USB», а также смонтированных в системе разделов и логических дисков в поисках съемных накопителей.

Некоторые вредоносные программы просто копируют себя на съемный носитель, а затем создают в его корневой папке файл *autorun.inf*, автоматически запускающий троянца или червя при обращении к такому накопителю. В некоторых версиях Windows (например, ранних версиях Windows XP) запуск мог осуществиться и вовсе в момент подключения такого устройства к компьютеру.

Другие вредоносные программы могут, например, переместить в Корзину (или скрытую папку с атрибутами «system» и «hidden») все находящиеся на съемном носителе файлы и директории, а затем создать вместо них ссылающиеся на исполняемый файл троянца пусковые ярлыки с теми же именами, по щелчку мышью на которых сначала запускается троянец, а потом открывается уже исходный файловый объект. Щелкнув мышью на таком ярлыке в попытке, например, открыть папку или просмотреть фотографию, не слишком искушенный пользователь запустит троянца на исполнение.

Некоторые черви, активно распространявшиеся ранее, аналогичным образом заражали системную папку Windows, использовавшуюся в качестве буферной при записи компакт-дисков (*%userprofile%\Local Settings\Application Data\Microsoft\CD Burning*). Таким образом, все CD или DVD-диски, записанные на зараженном компьютере, также автоматически оказывались инфицированными и содержали файл *autorun.inf*, запускавший троянца при помещении диска в оптический привод. В наши дни, в связи с распространением флэш-накопителей и карт памяти, технология CD / DVD постепенно отмирает, и этот вектор атак сейчас уже можно назвать архаичным.

ВРЕДОНОСНЫЕ ПОЧТОВЫЕ РАССЫЛКИ

Пожалуй, это самый популярный на сегодняшний день метод распространения вредоносных программ. Злоумышленники рассылают червей и троянцев под видом каких-либо документов: счетов, извещений о почтовых отправлениях, сообщений с просьбой подтвердить заказ в интернет-магазине, даже уведомлений о штрафах и судебных исках. Нередко вирусы и троянские программы приходят по электронной почте под видом предложения о знакомстве с приложенными «фотографиями», вместо которых, как правило, оказывается вредоносное вложение. При этом в тексте таких сообщений нередко присутствует ссылка якобы на профиль отправителя на сайте знакомств или в социальной сети. Переход по такой ссылке, как правило, также чреват заражением компьютера.

Пользователи мобильных устройств зачастую получают подобные рассылки с использованием СМС, причем злоумышленники задействуют в своих целях сервисы сокращения ссылок, вследствие чего потенциальная жертва зачастую не может заранее определить, что именно ожидает ее при переходе по предложенной ссылке. Однако можно смело сказать, что там ее не ожидает ничего хорошего. Входящие СМС маскируются под сообщения систем электронного банкинга, различных биллинговых систем, а также под уведомления о доставке MMS-сообщений.

Вот один из примеров рассылаемого злоумышленниками письма, во вложении к которому распространялся опасный троянец-энкодер, зашифровывавший все хранящиеся на компьютере жертвы файлы и требовавший выкуп за их расшифровку (орфография и синтаксис оригинала):

Тема: Процесс № 315

Уведомление о начале судебного разбирательства
Здравствуйте.
Поскольку ваша финансовая задолженность не была урегулирована в добровольном порядке, новый кредитор вынужден прибегнуть к принудительным мерам взыскания, предусмотренным законодательством Российской Федерации: направлению выездных групп по адресу регистрации, судебному разбирательству, инициированию исполнительного производства до полного погашения задолженности. Напоминаем Вам, что в случае Вашего неучастия в процессе, решение суда может быть вынесено заочно, что может повлечь за собой принудительное исполнение решения суда. Всю информацию о ходе предварительного судебного производства и сроках рассмотрения, включая копию искового заявления, Вы можете получить ознакомившись с приложенной к настоящему письму документацией или перейдя по ссылке находящейся в конце этого письма.

Это письмо создано автоматической системой и не требует ответа.

Как правило, вредоносные сообщения содержат вложение либо в виде zip-архива (в целях затруднения его детектирования некоторыми антивирусными программами, которые способны анализировать почтовый трафик), либо в виде исполняемого. exe,.bat или.scr-файла. В ряде случаев сообщение может содержать крошечный VBS-скрипт, который скачивает исполняемый файл троянца из Интернета и запускает его на исполнение. Также злоумышленники нередко рассылают настоящие документы в формате Microsoft Office (.doc /.docx,.xls /.xlsx) или PDF-файлы, запускающие вредоносный компонент с использованием уязвимостей в прикладном ПО.

УЯЗВИМОСТИ

Общеизвестно, что и операционные системы, и прикладные программы создаются программистами, а программистам, как и всем людям, свойственно допускать ошибки. Некоторые скрытые ошибки в коде приложений или самой системы, называемые *уязвимостями*, злоумышленники и используют в своих целях, при этом заражение зачастую происходит автоматически, без участия жертвы, при выполнении каких-либо рутинных действий, например, в момент открытия веб-страницы в окне браузера или текстового документа — в окне Microsoft Word. Яркий пример — уязвимость в ранних версиях Android, позволявшая сохранять внутри установочного.apk-файла с дистрибутивом программы два элемента с одинаковым именем, при этом сама ОС Android в процессе инсталляции такой программы проверяла первый элемент, а устанавливала и запускала — второй.

Все приложения для Android распространяются в формате. APK и представляют собой ZIP-архив с тем отличием, что они имеют специальную цифровую подпись. Внутри находятся необходимые для работы компоненты, которые в процессе установки приложения извлекаются, а их контрольные суммы проверяются по эталонным значениям. С помощью уязвимости Extra Field злоумышленник может изменить содержимое установочного пакета APK, не повредив его цифровую подпись. Внутри архива.APK располагается файл *classes.dex*, в котором содержится скомпилированный код приложения и набор служебных полей. Среди них есть:

- поле, хранящее имя файла с расширением;
- размер файла;
- поле Extra Field, в котором записан сам исполняемый код;
- таблица со списком используемых им классов.

Если в поле заголовка записать исходное значение без первых трех байт, значение длины поля Extra Field также изменится, благодаря чему появляется возможность дописать туда произвольный код, например, перечислить классы, используемые троянской частью приложения. После этого можно добавить в архив, помимо оригинального classes.dex, его вредоносную копию, часть кода которой будет храниться в «расширенном» поле Extra Field оригинального classes.dex. При установке программы

система прочитает содержимое видоизмененных полей, и, поскольку в них перечислены классы из модифицированного classes.dex, на устройство будет установлен именно этот файл.

Таким образом, уязвимость позволяет «подсадить» троянца в любое легитимное приложение с валидной цифровой подписью, разве что размер вредоносного модуля будет ограничен максимальным размером файла classes.dex в 65533 байт. Уязвимость была обнаружена в начале июля 2013 года и была устранена в версиях Android, выпущенных позже этой даты.

Другая уязвимость в Android оперировала цифровыми сертификатами. Как уже упоминалось, все .APK-файлы в Android используют цифровую подпись. Подпись приложения может быть взаимосвязана с цифровой подписью издателя программы. Все эти подписи используют инфраструктуру открытых ключей PKI (Public Key Infrastructure). С помощью цифровой подписи операционная система определяет, какие возможности и привилегии могут быть у приложения, с какими компонентами ОС оно может взаимодействовать, какие системные функции использовать, имеет ли оно право скачивать и устанавливать обновления и так далее.

Применяемые при проверке подписи приложения цифровые сертификаты (электронные документы, в которых хранится цифровой ключ) издаются специальными удостоверяющими центрами. Если система доверяет удостоверяющему центру, она автоматически доверяет и всем изданным им сертификатам, которые использует приложение. При проверке (валидации) цифровой подписи приложения операционная система использует открытый ключ разработчика программы. Чтобы убедиться в действительности этого ключа, требуется выполнить проверку соответствующего сертификата удостоверяющего центра. Это называется проверкой цепочки сертификатов. Уязвимость заключается в том, что в процессе установки приложения ранние версии Android не выполняли такую проверку вовсе.

В качестве практической реализации уязвимости можно привести такой пример. Если приложение подписано двумя цифровыми сертификатами: подлинным и поддельным, то при его установке будет создана цифровая подпись, использующая оба сертификата. В результате приложение сможет, например, скачивать и устанавливать вредоносные обновления, которые

не будут проверяться на безопасность, если разработчик подпишет их с помощью того же недостоверного сертификата.

Некоторые приложения Android могут использовать учетные данные пользователя для автоматической авторизации в различных интернет-сервисах. В этом случае пользователю достаточно указать свои логин и пароль один раз, после чего они регистрируются в специальном разделе системных настроек «Аккаунты», к которому приложение обращается всякий раз, когда ему необходимо пройти аутентификацию. При создании такой учетной записи ОС передает приложению различные параметры, среди которых имеется параметр *PendingIntent*. Из-за ошибки в реализации вызываемого при регистрации аккаунта метода addAccount в Android 4.0-4.4 система не проверяет значения этого поля, поэтому злоумышленник может передать в PendingIntent фактически любую команду, которая будет выполнена с теми же привилегиями, что и направившее его приложение «Настройки», — системными. Например, можно сформировать команду на удаление хранящихся на устройстве файлов или последовательность байтов, которая будет воспринята системой как входящее SMS-сообщение. Так, если в параметре PendingIntent будет передана команда *android.intent.action.MASTER_CLEAR*, Android послушно выполнит полный системный сброс с уничтожением всей хранящейся на устройстве информации.

А вот пример очень хитрой уязвимости. Эта уязвимость, получившая название ToastOverlay, была обнаружена в 2017 году и затрагивает все версии Android с 4.0 по 7.1.2 включительно. Ошибку разработчики допустили в подсистеме оверлеев — окон, способных отображаться поверх других экранных форм.

Использующему уязвимость приложению достаточно объявить в манифесте только одно разрешение — *BIND_ACCESSIBILITY_SERVICE*. В обычных условиях для отображения окон типа *TYPE_TOAST*, предназначенных для показа системных уведомлений, приложению требуется отправить запрос *SYSTEM_ALERT_WINDOW*, однако благодаря ошибке в обработчике проверки разрешений Android AOSP вредоносная программа может обойтись без подобных формальностей. Компонент просто не выполняет проверку доступа (permission check) и операции (operation check) при обработке запроса

SYSTEM_ALERT_WINDOW для *TYPE_TOAST*. В результате использующее уязвимость приложение может безнаказанно рисовать свои окна поверх окон других программ и фиксировать нажатия на экран. Фактически оно получает полный контроль над окном *TYPE_TOAST*. Какое содержимое будет отображаться в этом окне, зависит только от фантазии вирусописателей.

В Android есть встроенный фреймворк SIM Application Toolkit (STK), который позволяет SIM-карте выполнять в системе определенный набор команд. Таким образом, в частности, формируется SIM-меню оператора связи. Уязвимость SinToolkit позволяет перехватывать команды, отправляемые SIM-картой операционной системе, а также подменять их. Вредоносное приложение может передать классу *com.android.stk.StkCmdReceiver* специально созданный объект *parcelable*. Получатель не проверяет подлинность отправителя, при этом действие *android.intent.action.stk.command* не объявлено в манифесте как защищенное, благодаря чему можно эмулировать отсылку команд SIM-картой.

Например, если SIM-карта формирует на экране устройства сообщение с подтверждением действий пользователя, оно будет содержать кнопку OK. Такие сообщения используются для подтверждения отправки USSD-запросов, транзакций, действий с хранящимися на карте контактами и так далее. Вредоносное приложение может вызвать действие *android.intent.action.stk.command* и отобразит на экране поверх настоящего поддельное окно, содержащее произвольный текст. При нажатии кнопки OK вызывается метод *sendResponse()* с флагом *true*, и это событие — нажатие кнопки — передается SIM-карте, ожидающей реакции пользователя. При этом событие будет обработано так, как если бы оно поступило от настоящего диалогового окна. Это открывает широчайшие возможности для создателей вредоносных программ.

Описанная выше ошибка в SDK Android — далеко не единственная. Так, уязвимость Cloak and Dagger актуальна для Android вплоть до 7.1.2. Из-за ошибки в SDK вредоносное приложение, используя разрешения *SYSTEM_ALERT_WINDOW* и *BIND_ACCESSIBILITY_SERVICE*, может получить практически полный контроль над операционной системой и доступ к конфиденциальной информации пользователя, а также

фиксировать нажатия клавиш. Вкратце суть сводится к тому, что разрешение *SYSTEM_ALERT_WINDOW* позволяет вывести на экран «системное окно» — View-элемент, который отобразится поверх любого другого элемента интерфейса, даже если это будет Activity из стороннего приложения. При этом перекрытые Activity об этом не узнают и продолжат работать так, как будто ничего и не произошло. Это может сделать любое приложение, если разрешение *SYSTEM_ALERT_WINDOW* заявлено в его манифесте. Разместив несколько «невидимых» системных окон друг над другом и обрабатывая нажатия на них, злоумышленник может создать кейлоггер. А с помощью разрешения *BIND_ACCESSIBILITY_SERVICE* вредоносная программа способна получить доступ к другим объектам ОС и хранящимся на устройстве данным.

Уязвимости встречаются и в программной прошивке аппаратных устройств. Благодаря одной такой ошибке существует возможность взломать практически любое мобильное устройство производство компании Apple начиная с iPhone 5 и заканчивая самыми современными версиями iPhone и iPad. Речь идет об известной уязвимости checkm8 (checkmate), самой «свежей» на сегодняшний день уязвимости в мобильной технике Apple, наделавшей много шума в конце 2019 — начале 2020 года. Рассмотрим ее подробнее, чтобы понять, как работают и используются на практике подобные уязвимости.

Название «checkm8» произносится по-английски примерно как checkm-eight, что созвучно со словом checkmate — «шах и мат», символизирующим окончание шахматной партии. Отсюда и характерный логотип одноименного эксплоита в виде опрокинутой фигуры шахматного короля. «Игра окончена, ребята из Купертино, — как бы намекают нам авторы сплоита, — you lose». Самое интересное во всей этой истории с checkm8 то, что уязвимость была обнаружена не на программном, а на аппаратном уровне яблочной техники, причем охватывает она очень большой диапазон моделей, начиная с самых древних устройств на чипе A5 вроде iPhone 4S и заканчивая вполне современным iPhone X. «Дыра» прячется в механизме BootROM, который играет ключевую роль в процессе загрузки айфонов и айпадов. Причем исправить ее программными заплатками невозможно: для того чтобы решить проблему, нужно пересмотреть

аппаратную конфигурацию самого устройства, чего, как вы понимаете, за пару месяцев никак не сделать.

Для пользователя загрузка айфона выглядит крайне просто: нажал на кнопочку — и спустя пару секунд на экране появляется привычный интерфейс iOS. С технической точки зрения все немного сложнее. За начальный этап запуска яблочного устройства отвечает так называемый SecureROM, он же BootROM. Это — самый первый код, который запускается при холодной загрузке в Application Processor. Фактически он представляет собой урезанную и упрощенную версию загрузчика iBoot. Основная задача SecureROM — получить образ загрузчика из энергонезависимой памяти и передать ему управление. Этот код хранится непосредственно в чипе на аппаратном уровне, доступен только на чтение и потому не может быть изменен никаким образом извне. SecureROM — это самый доверенный код в Application Processor, который выполняется без каких-либо проверок. Он же отвечает за переход устройства в сервисный режим восстановления DFU (Device Firmware Update), активизируемый нажатием специальной комбинации кнопок при включении девайса. Для нас важно, что в режиме DFU доступна загрузка на устройство файлов через интерфейс USB.

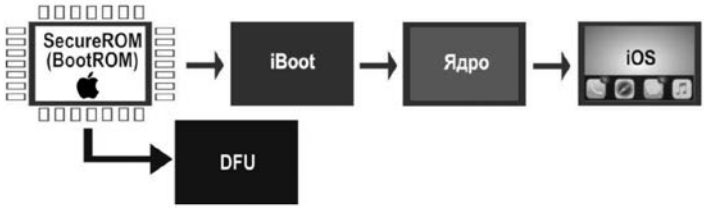


Рис. 23. Так выглядят «этапы большого пути» — загрузки устройства с iOS

Архитектурно SecureROM представляет собой первое звено цепочки безопасной загрузки, придуманной Apple для защиты от самого главного врага «яблочных» мобильных устройств — вредоносных программ и джейлбрейков. В SecureROM вшит криптографический ключ Apple, используемый для расшифровки образов, которые задействованы на последующих этапах загрузки, а также имеется необходимый инструментарий

для работы с криптоалгоритмами. Получив управление от SecureROM, загрузчик iBoot расшифровывает и запускает ядро операционной системы, после чего загружается образ самой iOS с графическим интерфейсом пользователя. Однако все эти этапы запуска iPhone или iPad выполняются, только если инициализация SecureROM прошла успешно.

Именно поэтому все существовавшие до открытия checkm8 джейлбрейки старались всячески обойти этот механизм. Ведь их первоочередная задача — загрузить видоизмененный образ iOS, допускающий установку программ из сторонних источников, чего не должно происходить с использованием SecureROM, стоящего на страже безопасной загрузки. Именно полный контроль над процессом запуска операционной системы гарантирует невозможность проникновения на устройство всевозможных буткитов, руткитов и прочей подобной малвари, отсутствием которой всегда и славилась мобильная платформа от Apple.

Как уже упоминалось, режим восстановления яблочного девайса DFU используется, если невозможна штатная загрузка айфона или айпада, и допускает обмен данными между компьютером и устройством через интерфейс USB. Для организации этого обмена в Apple придумали специальный протокол DFU. С его помощью можно залить на «окирпиченный» айфон новую прошивку, восстановить телефон из резервной копии или обновить операционную систему. Протокол DFU загружает с компьютера на яблочный девайс блоки данных с образом прошивки по запросу *0×21, 1*. Когда загрузка полностью завершается, запрашивается состояние устройства, после чего соединение по USB разрывается, устройство выходит из режима DFU, перезапускается и пытается загрузить полученный образ прошивки. Это если процесс протекает в штатном режиме. Однако исследователи заметили, что выйти из DFU можно и другими способами, например по запросу *0×21, 4* (DFU abort). В этом случае выполняется форсированное завершение режима восстановления устройства.

При передаче данных протокол DFU использует режим, который носит наименование USB Control Transfer. Соединение инициализируется с использованием установочного пакета (Setup Stage) длиной 8 байт, структура которого показана на иллюстрации 24.

Рис. 24. Структура установочного пакета USB Control Transfer

Назначение всех полей этого пакета нам сейчас не принципиально — кроме самого последнего. Если значение *wLength* ненулевое, сразу за установочным пакетом следует стадия данных (Data Stage), то есть данные пойдут с компьютера на устройство или обратно (направление определяется значением *bmRequest Type*). Эти данные передаются последовательно фрагментами размером от 8 до 64 байт в зависимости от скорости USB-соединения. Сессия передачи данных завершается стадией проверки статуса транзакции (Status Stage), на которой стороны обмениваются пакетами нулевой длины.

В USB-стеке iBoot временный буфер выделяется в момент инициализации USB и пакеты, передаваемые в фазе данных, загружаются в него непосредственно «на входе». Важный момент состоит в том, что USB-стек включается сразу, как только устройство переходит в режим DFU. Выделенный буфер используется для временного хранения информации на стадии данных. После передачи управления указатель на этот буфер (и размер ожидаемых данных) копируется в глобальную переменную, которую USB-стек использует как место назначения для пакетов, поступающих на устройство в фазе данных. Как только устройство выходит из режима DFU, USB-стек снова выключается. Однако глобальная переменная при этом не обнуляется! Таким образом исследователи нащупали классическую уязвимость типа Use-after-Free (UaF).

В этом и кроется ошибка, лежащая в основе checkm8. Если отправить на устройство запрос Setup Stage, инициировать передачу данных, но, не начав эту самую передачу, отправить девайсу запрос DFU abort (*0×21, 4*) на форсированный выход из DFU, то устройство попытается снова запуститься в режиме DFU и завершить прерванную сессию. При этом состояние памяти останется инициализированным и мы получаем возможность передать на устройство, загрузить в память и выполнить произвольный код по адресу буфера, выделенного до момента предыдущего выхода устройства из DFU. Поскольку вся

программа, обеспечивающая выделение буфера, работу с кучей (heap) и структурами задач, хранится в SecureROM и исполняется на аппаратном уровне, исправить эту ошибку попросту невозможно. Шах и мат!

Примечательно, что на девайсах с 32-разрядным ROM (A7, A10, A10X, A11 и A11X) указанный механизм не работает, поскольку буфер там аллоцируется всякий раз в одном и том же месте при каждой инициализации USB-стека. Тем не менее, обнаруживший данную уязвимость хакер axi0mX нашел способ обойти такую предопределенность с использованием правильно подобранного сценария эксплуатации Use-after-Free. Для этого он использовал то обстоятельство, что система одновременно может инициализировать несколько USB-передач. Например, в ответ на некоторые запросы устройство не сможет отправить данные, если получатель занят, до тех пор, пока конечная точка (endpoint) не освободится или не будет сброшен USB, то есть не будут устранены условия блокировки. Отправленные в таком состоянии запросы попадают в очередь. После устранения блокировки выполняется обход очереди и все запросы поочередно завершаются. Информация о конечной точке (endpoint) обнуляется, а запросы нулевой длины остаются в куче. Управляя запросами и тайм-аутами, теоретически можно создать такие условия формирования кучи, которые в итоге повлияют на следующее выделение памяти при создании буфера.

Обобщая, можно сказать, что из-за найденной в SecureROM ошибки в механизме создания и уничтожения USB-стека происходит утечка памяти, которая может использоваться для формирования состояния кучи, дающего возможность управлять выделением памяти при размещении буфера. В результате с помощью UaF можно выполнить запись в выделенную память для получения контролируемого косвенного перехода (controlled indirect branch) при выходе из DFU. Это, в свою очередь, позволяет взломать устройство еще до момента загрузки iOS и получить полный контроль над файловой системой.

В целом, природа уязвимостей может быть различной: некоторые из них связаны с недостатками архитектуры операционной системы или приложения, другие — с ошибками разработчиков ПО. Например, какая-либо программа может выполнять проверку недостаточно эффективно или вовсе не проверять

обрабатываемые ею и загружаемые в память данные, чем могут воспользоваться злоумышленники, либо передавать в буфер информацию без фактической проверки его границ (атака на переполнение буфера). Для закрытия уязвимостей разработчики операционных систем и прикладных программ периодически выпускают *обновления* своих продуктов, также порой называемые в обиходе «заплатками», или «патчами» (от англ. patch — «заплатка», «пластырь»). Среди уязвимостей принято особо выделять *критические уязвимости* — с помощью которых имеется возможность полностью нарушить работоспособность операционной системы или программы, а также *уязвимости нулевого дня* — уязвимости, уже известные злоумышленникам, для которых производитель ПО или ОС еще не успел выпустить соответствующего обновления, закрывающего данную брешь в безопасности.

Злоумышленники специально выискивают уязвимости в операционных системах и прикладных программах с целью распространения вирусов и троянцев. Самый простой метод такого поиска — анализ выпускаемых производителями софта обновлений. Например, киберпреступник может проанализировать содержимое опубликованного корпорацией Microsoft критического обновления безопасности для ОС Windows, заменяющего несколько системных динамических библиотек. Сравнив «старую» версию этих файлов с обновленной, злодей может без труда выяснить, в чем именно заключалась ликвидируемая обновлением уязвимость. До тех пор, пока все пользователи Windows не установят это обновление, они являются беззащитными для целевой атаки в данном направлении. А это сотни миллионов компьютеров по всему миру.

ЭКСПЛОЙТЫ

Программа, файл, электронный документ, фрагмент исполняемого кода или последовательность команд, использующая ту или иную уязвимость для реализации различных вредоносных функций, называется *эксплойтом*, *эксплоитом*, или *сплоитом* (от англ. exploit, «эксплуатировать»). Например, возможность

устаревших версий текстового редактора Microsoft Word автоматически запускать встроенные в документы.doc макросы — это уязвимость. А сам документ, в момент открытия сохраняющий при помощи макроса на диск компьютера и запускающий опасного троянца — это эксплойт.

В качестве еще одного примера можно назвать уже давно устаревший метод атаки, при которой злоумышленники рассылали пользователям Outlook письма в формате HTML, «прячущие» в себе скрытый фрейм. Поскольку эта почтовая программа ранних версий не умела правильно обрабатывать такие письма, из скрытого фрейма вызывался вредоносный скрипт, который загружал из Интернета троянца на пользовательскую машину и запускал его на выполнение. Это и был эксплойт — один из самых распространенных, с которыми многие наши соотечественники сталкивались на практике в начале «нулевых». Можно сказать, что притчей во языцех стали и многочисленные эксплойты, реализованные в виде документов в формате Adobe PDF или встраиваемых в веб-страницы интерактивных элементов в формате Adobe Flash — подобные творения злоумышленников появляются с завидной регулярностью, поскольку некоторые (особенно устаревшие) продукты корпорации Adobe с точки зрения информационной безопасности являются лакомой приманкой для злоумышленников.

Эксплойты являются популярным предметом купли-продажи на различных подпольных форумах, где общаются представители компьютерного андеграунда. Например, с использованием крайне распространенного у вирусописателей комплекта эксплойтов «Black Hole» («черная дыра») было инфицировано несколько миллионов компьютеров по всему миру. Один из частных случаев применения на практике данной схемы распространения вредоносного ПО выглядит следующим образом. При просмотре некоторых веб-сайтов, не несущих в себе какого-либо подозрительного контента, внезапно происходит открытие нового окна браузера, в котором загружается веб-страница, имитирующая своим оформлением интерфейс популярной службы «Ответы@Mail.ru». Разумеется, данная веб-страница физически располагается на другом сервере, адрес которого может отдаленно напоминать URL службы mail.ru. Уже одного факта открытия этой странички в браузере достаточно, чтобы

антивирусное ПО начало выдавать предупреждения о попытках загрузки и установки на компьютер вредоносных программ и запуска подозрительных скриптов. Теоретически вредоносная страница может быть любой и нести в себе совершенно иное визуальное оформление.

Анализ HTML-кода такой странички показывает, что она содержит несколько скрытых фреймов, в которых запускается скрипт, выполняющий переадресацию пользователя на принадлежащий злоумышленникам сайт. Этот сайт в свою очередь эксплуатирует набор эксплойтов «Black Hole», реализующих загрузку на компьютер жертвы вредоносного ПО с использованием известных уязвимостей браузеров и операционной системы. Какие именно троянские программы попадут при этом на компьютер жертвы, зависит от используемой ею версии браузера, версии операционной системы и наличия на ней установленных обновлений безопасности, наличия или отсутствия в атакуемой системе брандмауэра и современного антивирусного ПО.

Особо следует упомянуть об опасности, которой подвергаются пользователи операционной системы Google Android. По данным на март 2020 года, для различных версий данной мобильной платформы известно в общей сложности 2565 уязвимостей (2563 на уровне компонентов самой операционной системы), притом некоторые из них время от времени эксплуатируются разработчиками вредоносных программ. Однако поскольку сама ОС Android является, по большому счету, встраиваемой и распространяется в виде предустановленной заводской прошивки для смартфонов и планшетов, далеко не все производители этих устройств озабочены регулярным обновлением операционной системы. Скорее дела обстоят прямо наоборот: обновления прошивки выпускаются, в основном, для флагманских продуктов наиболее известных и крупных корпораций, а большинство бюджетных моделей мобильных гаджетов, особенно китайского производства, не обновляется вообще. Соответственно, пользователь не в состоянии сам закрыть подобные уязвимости, поскольку они содержатся в системных компонентах, к которым он в обычных условиях не имеет доступа. Самое забавное, что и антивирусные программы для Android также не в состоянии закрыть такие уязвимости, поскольку они

не имеют возможности модифицировать компоненты ОС. Антивирусы могут лишь предупредить пользователя о наличии бреши в защите и обезвредить вредоносные программы при попытке запуститься на защищаемом устройстве с использованием той или иной уязвимости.

Кроме того, специалистам широко известны случаи, когда вредоносные программы были внесены в Android-прошивку телефона самим его производителем, и смартфоны поступали в продажу уже зараженными. В особенности это касается дешевых устройств (зачастую — имитирующих дорогостоящие модели известных брендов), выпущенных никому не известными небольшими китайскими заводами. Именно таким образом распространялся, например, Android.Becu.1.origin — компонент этого вредоносного приложения, способного скрытно загружать и устанавливать по команде с управляющего сервера различные вредоносные программы, «прятался» в одном из системных каталогов, имел цифровую подпись самой операционной системы и потому обладал на инфицированном устройстве поистине неограниченными полномочиями.

Еще один встроенный в прошивку некоторых дешевых китайских телефонов троянец, Android.Oldboot.1.origin, действовал как буткит, загружаясь еще до момента запуска операционной системы. С его помощью злоумышленники инфицировали более чем 350000 Android-устройств. Нередки случаи заражения при установке на телефоны и планшеты так называемых кастомных (созданных сторонними разработчиками) Android-прошивок: многие из них таят в себе очень неприятные сюрпризы. Таким образом, при выявлении очередной уязвимости в ОС Android устройство в большинстве случаев остается подверженным заражению вредоносными программами, и спасти ситуацию может, по большому счету, только замена такого устройства на более современную модель с более новой версией ОС, в которой уязвимость уже устранена разработчиками.

Эксплоиты могут быть очень сложными с архитектурной точки зрения и реализовывать эксплуатацию уязвимости в несколько последовательных этапов. Рассмотрим в качестве примера, как работает эксплоит для уже упомянутой выше уязвимости в устройствах Apple — checkm8.

По принципу своего действия использующий эту уязвимость эксплоит — типичный буткит. Основная его задача состоит в том, чтобы дать устройству нормально загрузиться, но при этом скомпрометировать каждое звено в цепочке загрузки после того, как отработает BootROM. В нормальном режиме BootROM передает управление загрузчику (iBoot), который загружает в память ядро iOS и передает управление на точку входа. Цель создателей эксплоита — пропатчить ядро до того, как этот процесс завершится.

Во время работы iBoot использует специальный режим (его называют boot trampoline), который ненадолго возвращает процессор в «особое» состояние: кеш сброшен, все регистры установлены в ноль, MMU отключен. Команда хакеров под названием checkra1n, трудившаяся над созданием эксплоита, разработала особый метод размещения в памяти устройства шелл-кода и научилась использовать хуки (перехват вызова функций) при вызове некоторых функций загрузчика, чтобы заставить его выполнить полезную нагрузку. В итоге загрузчик подготавливает ядро и переводит процессор в состояние boot trampoline перед вызовом точки входа. Но вместо нее с помощью хука управление передается на заранее загруженный в память шелл-код. Теперь можно спокойно патчить ядро.

Однако просто пропатчить ядро недостаточно: когда оно загрузится, необходимо сохранить код в пользовательском режиме, чтобы можно было выполнить джейлбрейк. Для этого шелл-код создает в памяти маленький виртуальный диск, чтобы перехватить выполнение команд в режиме EL0. Это позволяет изменить дерево устройств и структуру аргументов загрузки ядра и потом использовать его в качестве корневого устройства. Для реализации этой идеи хакерам даже пришлось написать собственный динамический компоновщик. Для монтирования корневой файловой системы поверх / используется *syscall (3)*. При этом применяется каскадно-объединенное монтирование (union mounting), чтобы случайно не обрушить vnode. После всех этих манипуляций можно запускать произвольный код с идентификатором PID 1, прежде чем будет запущен launchd. Такой код инициализируется ядром iOS перед всеми последующими процессами и является для них родительским. Само собой, он обладает соответствующими системными привилегиями.

Нюанс в том, что на данном этапе смонтированная корневая файловая система доступна только для чтения, а для нормальной работы джейлбрейка нужно дропнуть в ФС несколько файлов, чтобы получить доступ к шеллу iOS и позволить пользователю установить менеджер пакетов. То есть необходимо получить доступ к */private/var*, для чего сначала инициализировать механизм Data Protection, за который отвечает launchd. Чтобы добиться нужного результата, с использованием того же union mounting поверх */usr/libexec/* монтируется еще один образ.dmg с целью переопределить какой-нибудь из системных демонов. В качестве жертвы был выбран sysstatuscheck, поскольку этот демон запускается в различных версиях iOS в нужный момент — в начале загрузочного процесса, но достаточно поздно, чтобы включить Data Protection. Когда задача выполнена,. dmg-образ принудительно размонтируется. Дальше можно позволить launchd продолжить загрузку в штатном режиме. После включения usbmux и загрузки всех необходимых утилит, позволяющих пользователю установить правильный bootstrap в своей корневой файловой системе, можно запускать демон SSH и выполнять джейлбрейк. Дело сделано: айфон успешно взломан!

ЗАГРУЗЧИКИ

Об опасности троянцев-загрузчиков я уже неоднократно упоминал. Множество вредоносных программ может проникнуть на устройство, уже инфицированное троянцем-загрузчиком, основное назначение которого — закачка из Интернета и запуск других вредоносных программ. Вот почему, однажды допустив заражение, пользователь рискует превратить свой компьютер в настоящий рассадник вирусов и троянцев.

Вредоносные программы-загрузчики нередко объединяются в ботнеты, услуги которых злоумышленники продают на различных подпольных форумах. Создатели банковских троянцев, троянцев-шпионов, рекламных троянцев и троянцев-вымогателей платят им за определенное число успешных загрузок своих

творений на компьютеры потенциальных жертв. Именно поэтому данный вектор распространения вредоносных программ является весьма популярным и актуальным.

СОЦИАЛЬНАЯ ИНЖЕНЕРИЯ

Под термином *социальная инженерия* принято понимать комплекс используемых злоумышленниками психологических приемов, позволяющих обмануть пользователя или ввести его в заблуждение, чтобы заставить выполнить те или иные нужные киберпреступнику действия, например, передать пароли от аккаунта в социальных сетях, сообщить реквизиты банковской карты или запустить на компьютере вредоносную программу. Так, от одного из друзей в социальной сети (аккаунт которого был ранее взломан) пользователю может прийти личное сообщение примерно следующего характера: «Ты с ума сошел, выкладывать такое видео? Я в шоке!» — далее в послании присутствует ссылка якобы на видеоролик, выставляющий получателя в неприглядном свете. Порой троянцы, рассылающие подобные послания, могут сгенерировать целую переписку, создавая поддельные комментарии пользователей, якобы успевших ознакомиться с видео. Однако для просмотра ролика потенциальной жертве предлагается скачать и запустить специальный кодек или приложение-проигрыватель, в котором скрывается… догадались что?

Само понятие социальной инженерии, по утверждениям ряда источников, ввел в обиход известный американский хакер Кевин Митник. Согласно рассказам его коллег и современников, сам Митник не обладал глубокими техническими познаниями в сфере информационных технологий, однако являлся при этом неплохим психологом. Он не брезговал покопаться в мусорных контейнерах на заднем дворе крупных офисных центров, куда сотрудники могли выбросить содержащие конфиденциальную переписку или важные пароли бумаги, распечатки внутренних документов и другие ценные источники информации. Он мог несколькими телефонными звонками выведать у работников различных фирм те или иные

интересующие его сведения. Например, представившись новым руководителем одного из подразделений компании, Митник мог поинтересоваться по телефону у одного из сотрудников техподдержки, где найти системного администратора, чтобы подключить на его рабочем месте принтер, и заодно узнать, как его зовут. Затем, представившись тем самым системным администратором, он выведывал у ничего не подозревающей секретарши пароль от ее электронного почтового ящика.

Вот лишь несколько простых примеров применяющихся на практике приемов социальной инженерии, используемых современными злоумышленниками:

- «забытые» в людных местах флэшки, которые при подключении к компьютеру заражают его вредоносной программой;
- телефонные звонки якобы от имени службы безопасности банка по поводу блокировки пластиковой карты, для разблокирования которой у жертвы просят назвать ее реквизиты, CVV-код и одноразовый пароль из СМС (впоследствии с использованием этих данных мошенники могут похитить деньги с банковского счета);
- звонки якобы от имени попавшего в беду родственника с просьбой перечислить деньги на номер мобильного телефона;
- звонки от имени службы технической поддержки провайдера (или компании) с сообщением о возникшей на рабочем месте проблеме с предложением для ее решения выполнить определенные (требуемые злоумышленнику) действия или сообщить конфиденциальную информацию (логин, пароль, секретный код из СМС, иные учетные данные).

Существуют и иные различные способы мошенничества, когда киберпреступники, например, рассылают пользователям популярных онлайновых служб занятости выгодные приглашения на работу, однако требуют оплатить какие-либо курсы обучения или внести определенную сумму якобы за оформление необходимых документов. Разновидностью социальной инженерии являются и классические «нигерийские письма», получившие свое название благодаря тому, что этим видом

мошенничества промышляли в основном жители солнечной Нигерии. Они отправляли потенциальным жертвам электронные послания на ломаном английском, представляясь адвокатами якобы недавно умершего богатого родственника получателя. Для оформления наследства требовалось всего ничего: оплатить «адвокату» небольшую сумму, чтобы правильно завизировать все бумаги или заплатить взятку коррумпированным нигерийским чиновникам.

Отдельной категорией сетевого мошенничества, использующего методики социальной инженерии, является так называемый dating scam — под этим термином понимается технология обмана пользователей многочисленных сайтов знакомств, к которым злоумышленники втираются в доверие с целью обогащения.

В последние годы традиционные способы мошенничества «на доверии», когда сетевые жулики под видом одинокой барышни знакомились на форумах с иностранцами, расписывали ужасы жизни в каком-нибудь провинциальном городке и слезно просили «выслать немного денег в голодную Россию», уже не работают, да и зарубежные граждане научились относиться к «русским красавицам» с определенной долей осторожности. Теперь индустрия обмана использует более продвинутые технологии, а сам процесс поставлен на конвейерный поток.

На подпольных хакерских форумах значительной популярностью пользуются комплекты не студийных фотографий, на которых изображена не слишком примелькавшаяся в Интернете симпатичная девушка. Стоимость набора из полутора сотен фото и нескольких видеороликов может составлять от $400 до $1000, в зависимости от качества «натуры» и выдвигаемых заказчиком требований. Нередко жуликам требуются услуги «девушки с поддержкой» — в этом случае модель должна время от времени передавать мошенникам фотографии, отснятые в заранее оговоренном антураже или видеоролики, в которых барышня произносит заранее оговоренные фразы.

Не менее высоким спросом пользуются услуги профессиональных переводчиков и копирайтеров, способных составлять от имени девушки грамотные и нешаблонные письма. Обычно мошенники до мелочей продумывают биографию для своего персонажа, включая имя девушки, место ее рождения

и проживания, ее историю, увлечения, вкусы и предпочтения. Как правило, копирайтерам заказывают 20-30 шаблонов писем постепенно увеличивающегося объема, со сквозным сюжетом и строгими требованиями к стилистике.

Для девушки мошенники создают веб-сайт или страницу в социальной сети, где размещают несколько заранее подготовленных фотографий. Согласно сценарию, девушка обычно проживает в небольшом провинциальном российском городе и имеет творческую профессию — художника, фотографа или дизайнера. В письмах мошенники рассказывают о каких-то забавных случаях или ситуациях, связанных с ее работой и жизнью в небольшом российском городке. Ради придания переписке романтического ореола (и с целью избежать обвинений в легкомысленности) жулики обычно упоминают о том, что девушка ранее переписывалась с иностранным мужчиной, но он обманул ее ожидания, оказавшись женатым пенсионером с кучей детей, — по этой же причине она избегает телефонных звонков и видеосвязи, надеясь сначала узнать своего избранника получше. На определенном этапе жулики выясняют почтовый адрес мужчины и высылают ему небольшой трогательный подарок — например, фотографию девушки с нарисованным помадой сердечком. Когда жертва окончательно растает от потоков романтики и знаков внимания, на работе у девушки внезапно происходит какая-либо катастрофа: разбивается дорогой зеркальный фотоаппарат или любимый дизайнерский планшет. Далее события могут развиваться по двум сценариям: доверчивую жертву «раскручивают» на приобретение дорогостоящего устройства, которое затем успешно продается, либо заманивают на сайт поддельного интернет-магазина, где можно приобрести недорогую, но «уникальную» вещь — в этом случае влюбленный мужчина рискует и вовсе расстаться со всеми средствами на счете своей банковской карты, добровольно передав мошенникам ее реквизиты. С учетом того, что одни и те же шаблоны сообщений могут использоваться в процессе переписки сразу с несколькими десятками адресатов, злоумышленники могут получить весьма внушительный нелегальный доход.

Согласно сообщениям, регулярно публикуемым на различных форумах, последнее время процветает еще один вид

преступлений, связанных с общением в Интернете. Злоумышленники находят на сайтах знакомств одинокую женщину или мужчину, старающихся наладить личную жизнь, завязывают романтическую переписку и, выяснив, что жертва проживает в своей квартире одна, предлагают ей приехать в другой город для личной встречи. Зачастую мошенники даже приобретают для нее билеты на поезд или самолет (как правило, воспользовавшись для этого краденными платежными реквизитами) и высылают по указанному адресу. И пока жертва в предвкушении счастливой встречи с избранником мчится в другой конец страны, квартирные воры, никуда не торопясь, выносят из ее жилища все ценные вещи и деньги.

Социальная инженерия активно используется и для взлома электронных почтовых ящиков. Например, в социальной сети потенциальной жертве может поступить сообщение от незнакомого человека, в котором он, крайне вежливо и обходительно извинившись за беспокойство, сообщает, что, возможно, учился в одном классе с матерью получателя. Чтобы убедиться в этом, он просит уточнить ее девичью фамилию. Нередко такие письма поступают пользователям и от отправителей, якобы озабоченных поиском дальних родственников. Отвечая на такие послания, самое время задуматься: а не является ли вопрос «девичья фамилия матери» ключевым для восстановления пароля в используемом жертвой почтовом сервисе? Если это так, злоумышленники очень быстро завладеют ее почтовым ящиком, а с помощью него — незамедлительно получат доступ к аккаунтам во всех социальных сетях и других подобных сервисах, допускающих восстановление пароля доступа при помощи электронной почты.

Распространение вредоносных программ с применением методов социальной инженерии применяется чрезвычайно широко. Практика показывает, что заставить не слишком опытного пользователя запустить на своем компьютере полученную по электронной почте вредоносную программу — не слишком сложная задача. Киберпреступники активно используют в своих целях и фишинг, заманивая пользователя на поддельные сайты, которым он доверяет, — с них ему предлагают скачать вредоносное ПО под видом новой версии браузера или необходимых обновлений.

ПОДДЕЛЬНЫЕ САЙТЫ

Этот транспорт, которым пользуются злоумышленники для распространения вредоносных программ, также можно условно отнести к категории фишинга.

В Интернете чрезвычайно распространены всевозможные сайты файлового обмена и торренты, с которых можно скачать полезные программы, игры и музыку, а также сайты из разряда «Вопрос — ответ», где пользователи могут разместить запрос о поиске какой-либо программы и получить соответствующую ссылку. Злоумышленники не гнушаются подделывать такие сайты, размещая там ссылки на вредоносные приложения.

Например, пользователю нужно переконвертировать снятый портативной видеокамерой клип из формата.MOV в другой формат, скажем,.AVI. Он вводит соответствующий запрос на сайте поисковой системы и получает ссылку на страничку, где другой пользователь уже задавал похожий вопрос: «Всем привет! Моя камера сохраняет видео в формате mov, а мне нужно записать ее на диск в avi. Как это делается?». И тут же получает ответ: «Для этого тебе нужна специальная программа-конвертер, вот ссылка. Я тоже долго искал такую и нашел — там все просто! Не благодари!».

Сайт, на котором опубликован этот диалог, вроде бы, известный и надежный (в действительности злоумышленники создали его копию-двойник), поэтому потенциальная жертва с радостью нажимает на предложенную ссылку… И машина в считаные секунды оказывается заражена троянской программой.

БЕСПЛАТНЫЕ И ВЗЛОМАННЫЕ ПРИЛОЖЕНИЯ

Как известно, бесплатный сыр бывает только в мышеловке, да и то — исключительно для последней по счету мыши. В Интернете распространяется множество различных бесплатных программ. Однако истинных альтруистов среди их разработчиков не так уж и много, программистам тоже хочется есть, поэтому в комплекте с некоторыми бесплатными приложениями часто идут различные «коммерческие» дополнения, от установки

которых, впрочем, как правило, можно отказаться, сбросив соответствующие флажки на этапе принятия лицензионного соглашения. Однако лицензионные соглашения обычно никто никогда не читает, поэтому вместе с нужным приложением на компьютер иногда инсталлируется и несколько нежелательных утилит. Этим пользуются многочисленные «партнерские программы», позволяющие разработчикам монетизировать свои бесплатные приложения. Так, в частности, распространяются рекламные троянцы. Достаточно невнимательно прочитать условия использования бесплатной программы или игры, и на экран вашего компьютера отовсюду будут выпрыгивать назойливые рекламные объявления.

Весьма распространенным способом внедрения на компьютер опасного, нежелательного и откровенно вредоносного ПО являются взломанные версии платных коммерческих приложений, распространяемых через торренты и файлообменные ресурсы. Троянцами зачастую оказываются и всевозможные «лекарства» для взлома коммерческого ПО — различные «кряки», «кейгены», «активаторы» и прочие пиратские утилиты.

Известны случаи распространения троянцев даже под видом музыки, видео и книг в популярных форматах, в частности FB2 — внутри скачанного архива вместо нового бестселлера известного писателя вполне может оказаться опасный исполняемый файл.

СИСТЕМЫ TDS

В целях максимального охвата потенциальной аудитории злоумышленники используют всевозможные системы управления трафиком (*TDS, Traffic Distribution Systems*). С их помощью осуществляются автоматические перенаправления пользователей (редиректы) на различные вредоносные ресурсы в зависимости от заданных киберпреступником условий. Например, встроенный злоумышленниками в веб-сайт TDS-сценарий может определить по IP-адресу географическое местоположение посетителя, и пользователя из России отправить

на русскоязычную страницу, с которой ему будет предложено скачать вредоносную программу под видом популярной игры, а иностранного посетителя — на англоязычную версию этого же сайта.

Определив версию браузера, TDS-скрипт автоматически перенаправит пользователя на страницу, содержащую эксплойт конкретно для его версии Opera, Firefox или Chrome. Наконец, по параметру User Agent, определяющему тип клиентского приложения, пользователи «настольной» версии Windows будут автоматически перенаправлены на страницу с троянцем для этой системы, пользователи мобильных устройств под управлением Android — на страницу загрузки мобильной вредоносной программы, пользователи Apple — на сайт, с которого можно скачать троянца для macOS. Иными словами, TDS позволяют заразить максимальное количество посетителей какого-либо вредоносного сайта, не потеряв привлеченный на него трафик впустую. К слову, комплекты скриптов для реализации TDS являются весьма популярным и востребованным товаром на всевозможных подпольных форумах.

РЕСУРСЫ «ДЛЯ ВЗРОСЛЫХ»

Порнографические ресурсы являются одним из самых массовых источников «компьютерной заразы». Подцепить троянскую программу при просмотре порно — гораздо более вероятно, чем не подцепить ее. В процессе посещения порносайтов у пользователя обычно притупляется критическое мышление, поэтому он с большей долей вероятности нажмет на какую-нибудь кнопку во всплывающем окне, загрузит исполняемый файл под видом фотографии или попытается скачать кодек, якобы необходимый для просмотра видеоролика. Этим и пользуются многочисленные распространители вирусов и троянцев.

Число случаев заражения вредоносным ПО среди пользователей ресурсов «для взрослых», по статистике, значительно выше по сравнению с людьми, не посещающими интернет-ресурсы подобной категории.

ВЗЛОМАННЫЕ САЙТЫ

В последние годы значительно увеличилось количество успешных взломов различных веб-сайтов, работающих с использованием популярных систем управления контентом (CMS, Content Management Systems) с открытым исходным кодом, таких как, например, Wordpress или Joomla. CMS позволяют администраторам сайтов менять опубликованный на собственном интернет-ресурсе контент с использованием удобной административной панели, не прибегая к необходимости вручную править структуру веб-страниц или код разметки гипертекста. Во встроенном в CMS редакторе можно добавлять на страницы сайта текст, иллюстрации, таблицы, менять оформление ресурса, добавлять или удалять разделы, выполнять другие редакторские операции.

Поскольку такие системы управления контентом, как Wordpress и Joomla, установлены на миллионах сайтов в Интернете, а их исходный код общедоступен и открыт для изучения, злоумышленники зачастую отыскивают в нем уязвимости, которые впоследствии используют для незаконного взлома таких сайтов. Вот лишь некоторые пути, используемые злоумышленниками для достижения этих целей:

- использование уязвимостей в системе авторизации административного интерфейса CMS или подсистеме обработки сессий;
- SQL-инъекции (внедрение в запросы к используемой «движком» реляционной базе данных некорректно обрабатываемых данных);
- создание и распространение бесплатных плагинов и компонентов для CMS, включающих бэкдоры или намеренно оставленные уязвимости;
- создание и распространение бесплатных шаблонов дизайнерского оформления для CMS, включающих намеренно оставленные уязвимости;
- хищение логинов и паролей для доступа к сайтам по протоколу FTP с использованием троянцев-шпионов и других вредоносных программ;
- фишинг — рассылка администраторам сайтов на адреса электронной почты, полученные с помощью общедо-

ступной системы Whois, писем якобы от имени службы техподдержки хостинг-провайдера с просьбой сменить пароль и ссылкой на поддельную страницу авторизации;
- взлом методом грубой силы (брутфорс).

Существуют и троянцы, специально разработанные для целенаправленного взлома веб-сайтов. Например, вредоносная программа Trojan.WPCracker была ориентирована на взлом систем управления конкретных сайтов путем последовательного перебора паролей, при этом список целевых ресурсов передавали ей злоумышленники с управляющего сервера.

Взломав сайт, киберпреступники обычно загружают на него *шелл-скрипт*, открывающий доступ к файловой системе атакованного интернет-ресурса, либо размещают в различных файлах, являющихся компонентами CMS, сценарии для перенаправления посетителей сайта на вредоносные ресурсы (элементы TDS) или для непосредственной загрузки троянцев. Отыскать такие «закладки» порой не под силу даже самим администраторам атакованного интернет-ресурса, поскольку вредоносный код обычно оказывается обфусцирован и спрятан глубоко в недрах служебных файлов CMS. Нередко преступники продают доступ к взломанным сайтам третьим лицам (как правило, другим злоумышленникам) с целью получения непосредственного дохода.

АТАКИ ТИПА MITM

Атаки типа MITM (Man in the middle — «человек посередине») — это способ атак, при которых злоумышленник внедряется в канал связи между отправителем и получателем информации и может видоизменять эту информацию «на лету» непосредственно в процессе ее передачи.

Примером успешной реализации такой атаки в целях распространения вредоносного ПО можно назвать инцидент с использованием троянца OnionDuke, заражавшего выходные узлы сети TOR, в результате чего весь трафик, транслировавшийся через эти узлы, оказывался инфицированным. При попытке пользователя скачать какую-либо программу из сети TOR через

скомпрометированный узел она автоматически оказывалась зараженной вирусом.

Случаи практического применения технологии MITM конкретно в целях распространения угроз достаточно редки, однако злоумышленники часто используют ее в целях *сниффинга* — анализа сетевого трафика для извлечения оттуда различной полезной информации, например логинов и паролей для доступа к тем или иным интернет-ресурсам.

ГЛАВА 8.
ТЕХНОЛОГИИ ЗАРАЖЕНИЯ

О том, какие именно деструктивные функции выполняют на инфицированном компьютере различные вредоносные программы, а также о том, какой транспорт они используют для проникновения в атакуемую систему, мы побеседовали в предыдущих главах. Настало время кратко обсудить методики, используемые вирусами и троянцами для заражения ПК.

Эта глава содержит определенный объем сугубо технической информации, поэтому читатель, не желающий вникать в различные наукообразные сложности, может пропустить данный раздел

Дроппер

Несмотря на довольно скромные размеры (средний размер вредоносной программы составляет 60-100 килобайт) ряд современных вредоносных приложений, как правило, включает в себя несколько модулей, реализующих различные функции. Один из таких модулей принято называть дроппером, он может быть реализован как в виде отдельного файла, так и в качестве составного компонента сложной угрозы.

Например, широко распространенные троянцы семейства MulDrop представляют собой троянцев-дропперов: основное (и единственное) назначение этих вредоносных программ — установка на атакуемый компьютер другого троянца, который хранится внутри контейнера Trojan.MulDrop, как одна матрешка в другой (как правило — в упакованном и зашифрованном виде). В отличие от троянцев-загрузчиков (Trojan.Download), Trojan.MulDrop не скачивает «полезную нагрузку» из Интернета, а извлекает ее «из себя».

Иными словами, дроппер — это объект, осуществляющий извлечение содержащихся в нем основных модулей вируса или троянца, их распаковку и установку в операционной системе. Основные компоненты вредоносной программы при этом могут помещаться дроппером, например, в одну из папок на диске или загружаться непосредственно в оперативную память. Основная функция дроппера — сохранить в атакуемой системе все компоненты вредоносного приложения, при необходимости зарегистрировать их в автозагрузке и передать им управление.

Инфектор

Еще один компонент вредоносных приложений называют *инфектором*. Инфектор — это модуль, осуществляющий

заражение файловых объектов (например исполняемых файлов или динамических библиотек) либо загрузочной записи компьютера путем изменения их внутренней структуры.

Инжектор

Инжектором принято называть функциональный модуль вредоносной программы, реализующий встраивание вредоносных компонент в запущенный процесс другого приложения (инжект). После успешного осуществления инжекта вредоносный компонент вируса или троянца выполняется в контексте инфицированного процесса.

Объект для инжекта во многом определяется целью злоумышленников и функциональным назначением самой вредоносной программы. Наиболее распространенным объектом для инжектов в Windows являются запущенные в системе процессы браузеров или различные системные процессы, такие как *svchost.exe* или *explorer.exe* (проводник).

Лоадер

Лоадером называют компонент вредоносной программы, осуществляющий загрузку других модулей (например, динамических библиотек) в оперативную память компьютера и (в ряде случаев) настройку этих компонентов в памяти. Для загрузки компонент лоадер может использовать как функции API, так и различные собственные механизмы.

ПРОЦЕСС ЗАРАЖЕНИЯ

Соответственно, различные компоненты вредоносной программы в целях инфицирования операционной системы могут действовать совместно.

В качестве примера такой функциональной интеграции можно рассмотреть механизм заражения Microsoft Windows одной из ранних версий вредоносной программы, принадлежащей к семейству Ramnit (Rmnet).

Первым на атакуемом компьютере запускается дроппер, который извлекает из своего тела, расшифровывает, а потом помещает во временную папку инфектор загрузочной записи и несколько модулей, реализованных в виде динамических библиотек. Затем дроппер запускает инфектор с помощью одной из системных функций, после чего файл инфектора удаляется.

После успешного запуска инфектор модифицирует основную загрузочную запись компьютера (MBR, Master Boot Record) с целью обеспечения запуска модулей Rmnet до старта средств антивирусной защиты, если таковые установлены на инфицированной машине. Оригинальная MBR вместе с частью вирусной загрузочной записи перемещается в конец диска.

При запуске операционной системы вредоносная программа получает управление, имеющийся в ее архитектуре инжектор встраивает модули Rmnet в один из запущенных процессов, в контексте которого они начинают выполняться. Один из модулей предназначен для саморепликации вредоносной программы, второй выполняет функции бэкдора, третий осуществляет мониторинг сетевого трафика, и, наконец, последний реализует функцию кражи паролей от различных приложений, в частности FTP-клиентов. Описанная здесь схема заражения представлена на рис. 25.

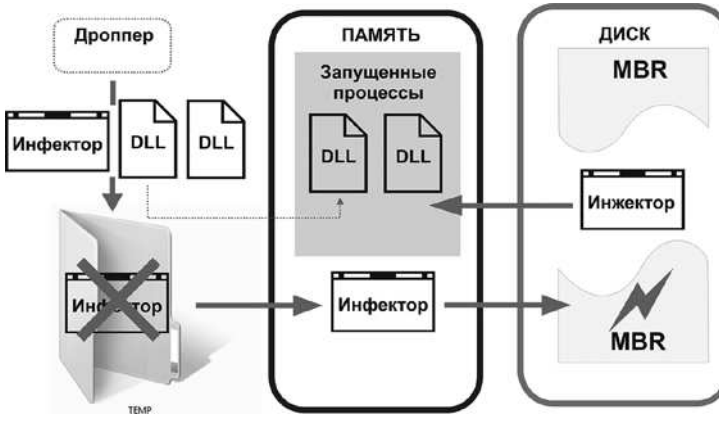


Рис. 25. Одна из реализаций схемы заражения ОС Windows

В зависимости от модификации и типа вредоносного приложения механизм заражения может значительно меняться:

представленный выше весьма упрощенный алгоритм является лишь одним из множества возможных вариантов и достаточно схематичен. Современные вредоносные приложения могут использовать различные руткит-технологии, позволяющим им эффективно скрывать свое присутствие в операционной системе, а также применять всевозможные методы антиотладки.

ИНФИЦИРОВАНИЕ ФАЙЛОВЫХ ОБЪЕКТОВ

Современные файловые вирусы реализуют различные методы заражения файлов, иногда достаточно изощренные. В общем случае механизм заражения файловых объектов можно упрощенно представить следующим образом. Определив по тем или иным заданным вирусописателем критериям подходящий для заражения файл, вирус модифицирует его, дописывая в его структуру собственное тело (иногда — в зашифрованном виде, в этом случае при выполнении вредоносного кода задействуется специальный механизм его расшифровки). В частности, вирус может поместить свою копию в начало файла, а оригинальное содержимое перенести в его конец. Затем вирус изменяет код приложения таким образом, чтобы при его запуске первым управление получил встроенный в него вредоносный компонент (например, размещая в соответствующей секции файла команду перехода на начало вируса). После выполнения основного тела вируса управление передается обратно инфицированному приложению, и оно выполняет свои функции так, как и было изначально задумано его разработчиками. Пользователь в большинстве случаев не замечает того, что в момент старта программы произошло что-то необычное. Такой метод заражения можно условно представить в виде схемы (рис. 26).

Существует вариант заражения, при котором оригинальный файл, наоборот, в том или ином виде сохраняется внутри вредоносной программы и в момент запуска вируса отображается в память компьютера или сохраняется в какой-либо папке на диске и запускается оттуда. Схожий алгоритм использовали, например, вирусы семейства Neshta.

В процессе заражения машины вирус сохранял свою копию в папке установки Windows под именем svchost.com, а затем изменял параметр ключа системного реестра *[HKCR \ exefile \ shell \ open \ command]*, записывая в него значение *"(default)»* = *"%WinDir% \ svchost.com "%1» %*"*. В результате при попытке запуска пользователем любой программы на инфицированном компьютере в первую очередь запускалось тело вируса из файла svchost.com, которому передавалось имя нужного пользователю приложения в виде параметра. Если этот исполняемый файл еще не был инфицирован ранее, Neshta сохранял его имя в специальном журнале для последующего заражения и запускал оригинальную программу. Если же пользователь пытался запустить ранее зараженный файл, вирус, получив управление, распаковывал оригинальный исполняемый файл нужной пользователю программы, помещал его во временную папку текущего пользователя Windows и запускал оттуда.

Рис. 26. Один из возможных алгоритмов заражения файловых объектов вирусом

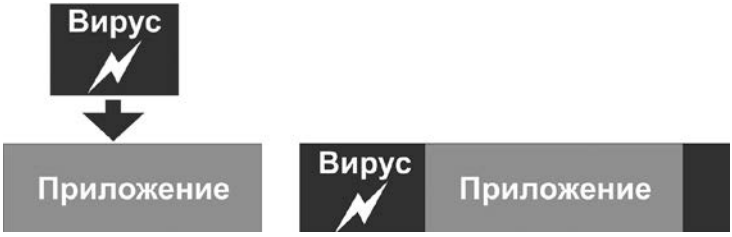


Рис. 27. Альтернативный алгоритм заражения файловых объектов вирусом

С целью избегания повторного заражения вирусы, как правило, используют специальные уникальные идентификаторы — в процессе заражения файла инфектор проверяет наличие в структурах файла соответствующей строки, и в случае ее обнаружения заражения не происходит. Описанная выше схема представлена на рис. 27.

Существуют и альтернативные методы заражения, используемые злоумышленниками значительно реже. Например, известно семейство вредоносных программ (иногда их называют общим термином «вирус-компаньон»), действовавших достаточно просто: они сохраняли свою копию в той же папке, где расположен исходный файл приложения, причем с тем же самым именем, а оригинальный исполняемый файл переименовывали. При запуске такой программы пользователем сначала запускался вирус, а после того, как он отрабатывал, управление передавалось оригинальному приложению.

МЕТОДЫ ОБЕСПЕЧЕНИЯ АВТОМАТИЧЕСКОГО ЗАПУСКА

Собственно, способов, позволяющих вредоносной программе автоматически запуститься на инфицированном компьютере, не так уж и много, и все они зависят в первую очередь от версии операционной системы, в которой действует вредоносная программа, а также от ее типа. Например, троянцы в macOS обеспечивают собственную автозагрузку с использованием файлов PLIST (Property List). Буткиты, о которых мы уже беседовали ранее, модифицируют для этих целей загрузочную запись, получая, таким образом, возможность запуститься одновременно с операционной системой (либо даже до завершения процесса ее загрузки), однако в любом случае — до момента старта действующих на компьютере антивирусных средств защиты.

На ранних этапах эволюции антивирусных программ для ОС Windows троянцы зачастую просто создавали скрытый ярлык, указывающий на исполняемый файл вредоносной программы, в системной папке автозагрузки Windows. Большинство современных угроз для этой цели модифицируют

отвечающую за автозапуск приложений ветвь системного реестра, как правило, выбирая в этом качестве ветвь [*HKLM* или *HKCU* \ *Software* \ *Microsoft* \ *Windows* \ *CurrentVersion* \ *Run*]. Однако до сих пор существуют троянцы, действующие по старинке: определив место расположения папки автозагрузки для текущего пользователя Windows (для этого троянец получает значение ключа реестра [*HKEY_CURRENT_USER* \ *Software* \ *Microsoft* \ *Windows* \ *CurrentVersion* \ *Explorer* \ *Shell Folders* \ *Startup*]), вредоносная программа просто сохраняет в этой директории собственный исполняемый файл, устанавливая для него атрибуты «системный» и «скрытый».

Вредоносные программы для ОС Linux используют другие методы автозапуска, обусловленные особенностями архитектуры этой операционной системы. Так, некоторые Linux-троянцы задействуют для данных целей службу выполнения приложений по расписанию (планировщика задач) cron, создавая соответствующие записи в конфигурационном файле crontab. Другие изменяют содержимое служебной директории «*~ /.config / autostart /*». Один из троянцев, принадлежащих к семейству Linux.BackDoor.Fysbis, добавлял во все найденные в директории */ etc /* файлы *rc.local* строку со ссылкой на инфицированный файловый объект либо (если изменить содержимое файлов не получилось) пытался создать файл службы в папке */ usr / lib / systemd / system /* и установить ее в системе, выполнив соответствующую последовательность команд.

ИНЖЕКТЫ

Инжект (не следует путать с *веб-инжектом*) — это механизм, позволяющий вирусу или троянцу встраивать вредоносный объект в запущенный и уже работающий в операционной системе процесс другого приложения, после чего внедренный объект начинает выполняться в контексте данного процесса. Инжекты осуществляются вредоносными программами с несколькими возможными целями. Во-первых, внедрение в процесс приложения позволяет получить доступ к различным ресурсам, используемым данным приложением. Например, инжект в процесс

браузера открывает перед злоумышленником возможность обходить установленные в брандмауэре ограничения или перехватывать вызовы соответствующих функций API. Во-вторых, внедренный в работающую программу вредоносный объект не будет демонстрироваться в Диспетчере задач Windows в виде отдельного процесса и потому станет как бы невидимым для пользователя. Есть и еще один заметный «плюс» инжекта с точки зрения злоумышленника: вредоносный код выполняется в операционной системе с привилегиями программы-«носителя», то есть основного процесса приложения, в которое инжектирован вредоносный объект. Иными словами, если программа, например, запущена от имени Администратора, вирусный код также получит аналогичные права на зараженной машине и сможет выполнять действия, недоступные для ограниченной учетной записи простого пользователя.

В настоящее время известно порядка 20 различных практических методов выполнения инжектов в процессы, запущенные в ОС семейства Microsoft Windows. Я не буду описывать их подробно, поскольку соответствующую техническую информацию можно без труда отыскать в популярных учебниках по программированию. В общем случае (самый распространенный метод) последовательность действий вредоносной программы такова:

- выбор приложения, в которое будет осуществляться инжект;
- поиск и получение соответствующего дескриптора процесса;
- выделение в адресном пространстве процесса памяти для внедрения вредоносного объекта по соответствующему адресу;
- копирование вредоносного компонента в выделенную область памяти;
- создание нового потока в виртуальном адресном пространстве процесса, в который внедрен вредоносный код.

Многие разработчики инжектов используют для реализации данного механизма стандартные функции API, например, *WriteProcessMemory* для копирования кода в память процесса и *CreateRemoteThread* для запуска удаленного потока. Другие поступают иначе. Так, создатели некоторых образцов троянцев

семейства Trojan.Inject использовали альтернативную методологию: инжектор вредоносной программы создавал «замороженный» экземпляр процесса svchost.exe (хост-процесс для загружаемых из динамических библиотек служб), подготавливал адреса всех необходимых для работы встраиваемого кода функций API, затем выделял требуемое адресное пространство памяти и полностью копировал троянца в процесс, после чего «размораживал» его.

В некоторых, довольно редких, случаях инжекты выполняются с использованием различных уязвимостей или недокументированных возможностей тех или иных компонентов операционной системы. Так, сразу несколько троянских программ использовали в свое время уязвимость в компоненте explorer.exe, вернее, в его подсистеме GUI (графического интерфейса пользователя). Однако подобные случаи на практике все же встречаются нечасто.

ПЕРЕХВАТ ВЫЗОВОВ ФУНКЦИЙ

Для реализации различных практических задач многие вредоносные программы обладают механизмом перехвата вызовов API-функций в процессах других приложений. Иногда эту технологию называют «*хуками*».

Кратко процедуру «хукинга» можно описать следующим образом. Очень многие приложения, выполняющиеся в среде Microsoft Windows (если не сказать «почти все»), используют для своей работы так называемые динамические подключаемые библиотеки (Dynamic Link Library), физически представленные в виде файлов с расширением *.dll*. Динамические библиотеки в свою очередь содержат определенный набор используемых приложениями функций. Сделано это, чтобы «разгрузить» само приложение, избавить его от «лишнего» кода. Например, разные программы могут многократно использовать одну и ту же функцию. Вместо того чтобы размещать ее код в самом приложении, разработчики позволяют этой программе обращаться к соответствующей динамической библиотеке и вызывать нужную ей функцию оттуда. При этом подразумевается,

что одной стандартной библиотекой может пользоваться сразу несколько разных программ.

Когда программа запускается на выполнение, ОС Windows создает для нее отдельный процесс, выделяя для приложения определенный объем памяти. В начале каждого исполняемого файла перечислены имена динамических библиотек, которые использует это приложение: система осуществляет их поиск (в папке, где установлена сама программа, или в системных директориях Windows), *аллоцирует* (выделяет) память в процессе работающего приложения и загружает туда эти библиотеки. Когда программа вызывает какую-либо функцию, нужную ей для работы в данной момент времени, она автоматически определяет, в какой библиотеке и по какому адресу хранится данная функция, после чего строит специальную таблицу зависимости вида «имя функции — имя библиотеки — адрес функции внутри библиотеки» (она называется таблицей импорта функций), отыскивает нужный адрес в памяти процесса и передает на него управление (рис. 28).

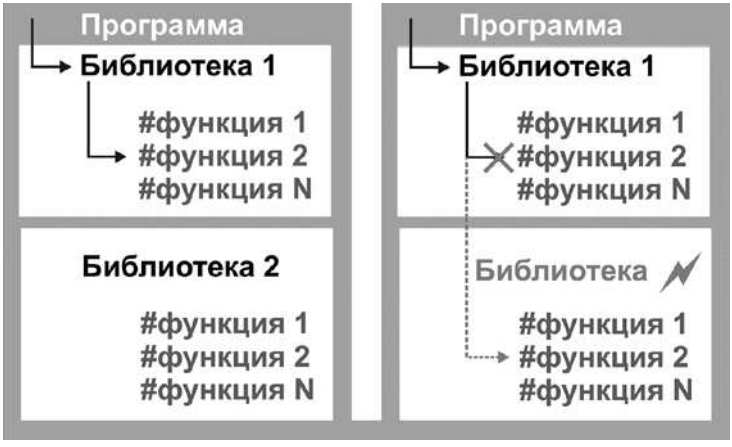


Рис. 28. Стандартный механизм работы приложения и перехват вызовов функций с передачей управления на вредоносный объект

Собственно, задача «хукинга» заключается в том, чтобы «убедить» программу, будто нужная ей функция хранится

не по этому адресу, а расположена в другом месте, например, в совершенно иной динамической библиотеке, которая была предварительно встроена в процесс методом инжекта.

Практически перехват вызовов функций осуществляется обычно перестановкой указателя, то есть, изменением содержимого этой самой «таблицы зависимостей» таким образом, чтобы имя вызываемой функции ссылалось в конечном итоге на нужный злоумышленнику адрес функции в памяти процесса.

Второй метод носит наименование code injection и заключается в том, что злоумышленник непосредственно в памяти процесса заменяет первые байты кода самой функции, вставляя туда инструкцию безусловного перехода на собственную, вредоносную функцию, после выполнения которой управление (в случае необходимости) возвращается обратно оригинальной функции. Этот метод чуть более сложен в реализации и потому встречается на практике реже.

Так, некоторые банковские троянцы при запуске снимают хуки с ряда функций системных библиотек с целью обхода различных инструментальных средств, позволяющих исследовать вредоносную программу — таким образом реализуется механизм антиотладки. Установка перехватов вызовов функций широко используется и в целях непосредственного выполнения приложением деструктивных действий.

Например, встроившись в процесс Проводника, некоторые троянцы могут перехватывать вызов функции *ZwQueryDirectoryFile* с целью сокрытия собственных файлов на диске инфицированного компьютера — при обращении к такому файлу будет автоматически установлен код ошибки *STATUS_NO_SUCH_FILE*. Нередко установка хуков на определенные функции применяется троянцами-шпионами для «просеивания» сетевого трафика (сниффинга) в поисках логинов и паролей.

Существует огромное количество практических задач, которые могут быть реализованы с использованием установки тех или иных хуков. Фактически перехват вызовов функций открывает перед злоумышленниками почти полный контроль над приложением. С помощью хуков, например, можно отслеживать действия пользователя (движения курсора и нажатия кнопок мыши), предотвращать создание или закрытие окон, выгрузку

компонентов приложения, перехватывать сообщения от других процессов, контролировать работу приложения с файловыми объектами и т. д. Например, если злоумышленник хочет, чтобы другая программа не завершила используемый им инфицированный процесс, он может установить хуки на функции открытия и завершения процессов (*OpenProcess*, *TerminateProcess*), и в момент вызова этих функций другими процессами осуществлять проверку, не пытаются ли они завершить процесс его вредоносного приложения.

С этой точки зрения горизонты применения технологии перехвата вызовов функций ограничены, по большому счету, только набором самих функций, естественными рамками архитектуры операционной системы и фантазией злоумышленника.

ГЛАВА 9. КТО ПИШЕТ И РАСПРОСТРАНЯЕТ ВИРУСЫ?

Современный мир компьютерной преступности включает огромное число различных представителей пестрого племени злоумышленников. Времена энтузиастов-одиночек давным-давно канули в прошлое: в наши дни разработкой и распространением вредоносных программ занимаются целые группы сетевых злодеев, хорошо структурированные и напоминающие по своей иерархии настоящие мафиозные кланы, в которых каждому участнику отведена строго определенная роль.

По долгу службы мне приходится проводить довольно много времени на различных полуподпольных интернет-форумах, в том числе в «глубоком Интернете», где собираются многочисленные представители компьютерного андеграунда: вирусописатели, хакеры, кардеры, спамеры и прочая творческая интеллигенция эпохи высоких технологий. Коллектив там подбирается, как правило, весьма разношерстный, но достаточно интересный. Да и деньги в этом криминальном бизнесе крутятся немаленькие, по крайней мере, тема из разряда «куда вложить лишние 100000 долларов» не является на подобных сайтах чем-то особенным. Чем же отличаются друг от друга различные представители «темной стороны» IT-технологий? Каковы источники дохода современных киберкриминальных деятелей? Давайте попробуем обобщить эти сведения в рамках настоящей главы.

ХАКЕРЫ И КИБЕРПРЕСТУПНИКИ

Всякий раз, когда в Интернете или традиционных СМИ я вижу новостные заголовки из разряда «Хакеры организовали очередную вирусную атаку» или «База данных интернет-магазина была похищена хакерами», мне нестерпимо хочется отыскать написавшего это журналиста и стукнуть его по голове чем-нибудь тяжелым. Когда-то, работая редактором компьютерного журнала, я безжалостно вымарывал подобные строки из поступавших в издательство материалов. «Почему ты с таким упорством защищаешь хакеров?» — недоумевали коллеги. «Потому что они ни в чем не виноваты», — всякий раз отвечал я.

Предполагается, что термин «хакер» зародился в кампусах и аудиториях Массачусетского технологического института

еще в 60-х годах XX века. Бытует мнение, что словечко попало в обиход компьютерщиков из жаргона хиппи, где глагол «to hack» означал отнюдь не «взламывать», как это считается сейчас, а «соображать», «врубаться». Собственно, в 70-х «хакерами» как раз и называли тех, кто «врубается» в принципы работы компьютеров, глубоко понимает происходящие в них процессы — то есть высококвалифицированных IT-специалистов.

Иными словами, в классическом понимании «хакерами» называли компьютерных гениев, тех самых косматые парни в очках, сквозь толстые стекла которых можно поджигать муравьев. Настоящие хакеры никогда не взламывали чужие программы или компьютеры ради денег, и уж тем более не совершали преступлений — разве что порой использовали свои знания для организации безобидных розыгрышей. Говорят, один предприимчивый парень, сконструировав «bluebox» — устройство, позволявшее «обманывать» аппаратуру телефонных сетей — однажды сумел дозвониться самому Папе Римскому. По большому счету, хакерами можно назвать Стива Возняка и Билла Гейтса, Линуса Торвальдса и Ричарда Столлмана. Даже создатель первой в истории человечества электронно-вычислительной машины Конрад Цузе был своего рода хакером, хотя в его времена такого понятия не существовало вовсе.

Для людей, взламывающих программы или удаленные серверы с целью заработка, а также разрабатывающих различные средства обхода систем лицензионной защиты, существует отдельное название — *крэкеры*, происходящее от английского глагола to crack, «взламывать». Специалистов по взлому телефонных сетей принято называть *фрикерами*. Злоумышленников, специализирующихся на хищении данных банковских карт, взломе других платежных инструментов и систем электронных платежей — кардерами.

Вредоносный код пишут *вирусописатели* или *вирмейкеры*, при этом бок о бок с ними действуют *криптеры*, которые упаковывают и шифруют вредоносные приложения, отдельные «специалисты» осуществляют раздачу вирусов и троянцев пользователям, а *ботоводы* реализуют на «черном рынке» различные «услуги» с помощью принадлежащих им ботнетов, например рассылку спама или DDoS-атаки.

Широко известны интернет-мошенники, выманивающие у доверчивых жертв деньги при помощи специально созданных сайтов или с использованием методов социальной инженерии. Такую противозаконную деятельность называют *фродом* от английского термина fraud — «мошенничество». Жуликов, использующих личную информацию потенциальной жертвы в целях обмана или вымогательства именуют *скамерами* — нередко они обитают на сайтах знакомств и промышляют в социальных сетях. *Спамеры* рассылают рекламу по электронной почте и с использованием СМС-сообщений. Иными словами, все эти категории киберпреступников имеют не только собственную специализацию, но и собственные «профессиональные» наименования.

НА ЧЕМ ЗАРАБАТЫВАЕТ КОМПЬЮТЕРНЫЙ АНДЕГРАУНД?

Итак, некоторая часть населения «подпольного Интернета» представлена собственно вирусописателями, один из источников заработка которых — создание вредоносного ПО. Собственно, сами вирусописатели нередко дают возможность заработать другой категории специалистов, которую можно условно назвать «впаривателями», путем привлечения их к участию в партнерских программах. В этом случае наблюдается своеобразный симбиоз между двумя различными группами киберпреступников: создатели троянцев весьма охотно «покупают загрузки», то есть оплачивают другим специалистам факты инфицирования пользовательских компьютеров их творениями. Те же, кто продает загрузки, то есть, собственно, получает деньги за каждый факт заражения, обычно владеют своей действующей бот-сетью или распространяют вредоносное ПО через собственные сайты, используя методы социальной инженерии, различные системы управления трафиком (TDS), взломанные ресурсы или уязвимости популярных браузеров (а чаще и то, и другое, и третье). Чем же выгодны злоумышленникам покупки «загрузок»? Таким образом, как правило, создаются бот-сети, услуги которых также можно успешно

монетизировать, рассылая спам или устраивая за соответствующее вознаграждение DDoS-атаки на неугодные серверы. Кроме того, создателям, скажем, банковских троянцев не нужно заботиться о том, как их вредоносная программа попадет на целевой компьютер — они просто покупают у «впаривателей» или «ботоводов» некоторое количество загрузок либо спам-рассылку, а потом изучают полученный «улов». Например, известны случаи, когда разработчики троянца-шифровальщика не просто предлагали всем желающим приобрести созданный ими энкодер «в аренду», но создавали для него автоматизированный управляющий сервер, на котором жертвы шифровальщика могли авторизоваться при помощи автоматически генерируемого троянцем ключа, дешифровать один пробный файл и оплатить выкуп, процент от которого тут же перечислялся распространителю этой вредоносной программы. Таким «сервисом» не могут похвастаться даже некоторые серьезные коммерческие компании!

Отдельную категорию платных (и весьма востребованных) услуг на подпольном киберрынке составляет так называемое криптование. Что это? Давайте разбираться. Принцип работы большинства антивирусных программ состоит в том, что для каждой выявленной угрозы создается уникальная сигнатура, которая помещается в вирусные базы. При последующем сканировании вредоносные файлы безошибочно определяются по этой самой сигнатуре и успешно удаляются, либо помещаются в карантин. С целью обхода защиты вирусописатели применяют метод переупаковки вредоносных программ с использованием различных упаковщиков исполняемых файлов, нередко нанизывая слои таких «оберток» друг на друга по принципу матрешки: в результате вредоносный файл становится не похож сам на себя, и антивирусная программа «не признает» его за угрозу. Такой троянец не будет распознаваться антивирусным ПО ровно до тех пор, пока его образец не попадет в вирусную лабораторию и соответствующая сигнатура не будет добавлена в базу. К тому же толковый «крипт» несколько затрудняет анализ угрозы: в дизассемблере грамотно упакованный троянец выглядит как запутанный набор инструкций. Не то чтобы распутать такой клубок невозможно, просто занимает это гораздо больше времени… Вот почему услуга по «навешиванию» на вредоносную программу «криптов» весьма востребована. Впрочем, это

совершенно не спасает вирусописателей от системы эвристического анализа угроз, так что с появлением подобных технологий ситуация стала понемногу меняться в лучшую сторону: хоть криптуй, хоть не криптуй, все равно получишь… детект. Разумеется, в сети активно продаются и сами троянцы-боты (под заказ их могут перекомпилировать, «зашив» в ресурсы ссылки на требуемые управляющие серверы), и даже исходники некоторых вредоносных программ — их можно приобрести по цене от двух до пяти тысяч долларов.

Весьма распространенный объект купли-продажи — наборы эксплойтов и скриптов для организации систем управления трафиком. Продаются и покупаются похищенные с компьютеров пользователей базы данных, содержащие логины и пароли для доступа к почтовым ящикам, страничкам в социальных сетях, форумам, различным сетевым ресурсам. Вся эта информация используется потом в целях мошенничества, рассылки спама, кражи конфиденциальной информации.

Довольно большим спросом пользуются так называемые дедики, или dedicated-servers, то есть выделенные серверы, которые можно использовать в различных криминальных целях: например, устанавливать на них специфическое ПО, применять в качестве прокси, использовать их как промежуточное звено при осуществлении атаки на другие сетевые ресурсы. «Дедики» добываются с использованием нехитрой технологии: сначала посредством специальных программ осуществляется сканирование выбранного диапазона IP-адресов на наличие открытых портов, затем методом подбора паролей реализуется попытка проникновения на сервер. Вы будете удивлены, узнав, сколько системных администраторов оставляет на серверах активные учетные записи с параметрами удаленного доступа из разряда admin / admin или Administrator / 1234. По статистике, за одну ночь соответствующие программы обнаруживают в худшем случае два-три таких «дедика», а при удачном стечении обстоятельств улов может быть и более крупным. «Живучесть» подобных серверов зависит от того, насколько аккуратно ими пользуется злоумышленник: если не перегружать процессор машины своими задачами, не создавать бешеный трафик, прятать в системе используемое ПО и регулярно «чистить» за собой логи, аккаунт может просуществовать достаточно долго. Ну,

а если злоумышленнику по каким-либо причинам неохота искать открытые dedicated-серверы самостоятельно, их можно просто купить: данные для доступа продаются в сети пачками по весьма приемлемой цене.

Еще одна востребованная услуга — предоставление абузоустойчивого хостинга, то есть хостинга, администрация которого не реагирует на жалобы пользователей и позволяет размещать в сети практически любой контент: порнографию, сайты, рекламируемые спам-рассылками, администраторские панели бот-сетей, мошеннические странички с массовыми «лохотронами» за СМС, а также веб-страницы, распространяющие всевозможное вредоносное ПО. Причем серверы таких хостинг-провайдеров далеко не всегда располагаются на Каймановых островах: гораздо чаще абузоустойчивый хостинг можно разыскать в сытой Европе. К слову, техподдержка таких провайдеров по уровню сервиса порой может дать фору даже саппорту легальных хостеров. Вот у кого следовало бы поучиться некоторым раскрученным провайдерам…

Следующим номером нашей программы является продажа трафика, в частности iframe-трафика. Требуется нагнать пару тысяч уникальных посетителей на какую-нибудь веб-страничку, чтобы накрутить счетчик? Или просто хочется раздать десятку тысяч пользователей по троянцу-даунлоадеру? К вашим услугам трафик с различных сайтов, часть из которых грешит редиректами, а некоторые оборудованы скрытыми элементами iframe и pop-up-кодом. Причем нередко заказчик может выбирать источник трафика по странам, по крайней мере, ему будет точно известно число посетителей из США, Канады, Китая, Европы. Очень удобно в целях проведения таргетинга при распространении вредоносных программ.

Вы когда-либо пытались взять кредит? Не обязательно в банке, вполне достаточно приобрести в каком-нибудь магазине мобильный телефон или стиральную машину в рассрочку. Будьте уверены: ваш комплект документов, включая отсканированную копию паспорта и водительского удостоверения (часто к этому добавляется еще и фото «клиента» с паспортом в руках) уже продается на подпольных форумах по цене примерно пять долларов за набор. Торговля сканированными комплектами документов — очень выгодная и широко развитая индустрия.

С помощью таких комплектов можно, например, получить персональный аттестат в платежной системе Webmoney, или зарегистрировать домен (хотя это нетрудно сделать и вовсе анонимно), открыть счет в некоторых платежных системах. В общем, широчайшее поле для деятельности.

Отдельной категорией «кибербизнесменов» являются мелкие жулики, торгующие «угнанными» номерами мессенджеров или взламывающие под заказ почтовые ящики — в основном используя методы социальной инженерии. Методы, позволяющие выяснить у пользователя данные для ответа на контрольный вопрос, необходимый при восстановлении «забытого» пароля, достаточно просты. Иногда для этого не нужно даже беспокоить самого пользователя: чтобы ввести правильный ответ на контрольный вопрос «имя моего домашнего животного» порой достаточно прошерстить фотоальбомы в социальной сети потенциальной жертвы в поисках снимка с подписью «я и Мурзик». Случаются и более идиотские ситуации: например, в середине «нулевых» пользователям мессенджера ICQ приходили сообщения якобы от имени девушки, работающей менеджером крупной фирмы, которой — внимание! — нужно «поднять рейтинг», для чего жертве предлагалось установить в настройке профиля «аськи» предложенный адрес электронной почты и получить за это кругленькую сумму. Денег жертва, разумеется, не получала, зато получала уникальную возможность «поднять рейтинг собственного IQ», поскольку указанный в профайле «аськи» адрес e-mail — единственный способ восстановить измененный злоумышленником пароль. Сама услуга по взлому электронной почты предоставляется обычно на весьма удобных условиях: заказчик сообщает взломщику адрес ящика, к которому требуется получить доступ, он выполняет все необходимые действия, после чего отправляет клиенту письмо заранее оговоренного содержания с захваченного адреса как свидетельство того, что работа выполнена. После оплаты заказчик получает логин и пароль.

Среди прочих способов заработка можно упомянуть целую категорию людей, промышляющих охмурением страждущих любви пользователей на многочисленных сайтах знакомств. Для этого им требуется не слишком заезженная фотография какой-нибудь очаровательной особы, а также некоторые

знания в области психологии. Пообщавшись с юной «куколкой» день-другой, жертва внезапно узнает, что эта милая, трогательная и красивая девочка вот буквально только что случайно потеряла мамин цифровой фотоаппарат, или отдала гопникам дорогой папин мобильник, и теперь родители обязательно ее убьют, как только узнают об этом печальном происшествии… Какой настоящий мужчина не захочет помочь барышне, которая поначалу даже будет отказываться от неожиданной щедрости своего поклонника, но потом все же согласится: волосатые красноглазые обитатели подпольных форумов очень любят пиво, а оно стоит денег… Ну, и иногда они любят пообсуждать между собой, как «этот лох классно развелся на блондинку».

Иными словами, компьютерный андеграунд — целый мир со своими принципами, традициями, сленгом, героями и антигероями, и естественно, со своими финансовыми потоками. В этом мире обитает множество людей, которые зарабатывают себе на пропитание сотнями различных способов, часто балансируя на грани закона и нередко преступая эту грань. Впрочем, для кого-то это уже давно стало привычным стилем жизни.

ТАК КТО ВСЕ-ТАКИ РАСПРОСТРАНЯЕТ ВИРУСЫ?

Обычно созданием и распространением вредоносных программ занимаются совершенно разные люди. Выше я уже упоминал о том, что киберпреступные сообщества в общем и целом напоминают настоящую мафиозную группировку, роли внутри которой четко поделены между ее участниками. При этом зачастую многие из них даже не знакомы друг с другом лично — все общение и взаимодействие осуществляется онлайн, с помощью распространенных в Интернете средств коммуникаций.

Ситуации порой складываются разные, но в общем случае схема разработки и распространения вируса или троянца может выглядеть следующим образом. Итак, программисты-вирусописатели создают вредоносный код. Далее он либо выставляется на продажу на различных подпольных интернет-площадках,

либо разработчики организуют «партнерскую программу», к участию в которой привлекаются другие участники киберпреступного сообщества. Иногда сам процесс разработки кода дробится на определенные этапы, каждый из которых реализуется разными людьми: например, один программист создает базовый код, включающий механизмы саморепликации, другой — модули, отвечающие за антиотладку, третий разрабатывает «полезную нагрузку» — компоненты, реализующие основной функционал вредоносного приложения. Нередко определенные элементы заимствуются из кода, попавшего ранее в открытый доступ, — в подобных случаях аналитики, изучающие попавший в вирусную лабораторию образец троянца, обнаруживают в нем давно знакомые черты. Затем специалисты соответствующего профиля криптуют исполняемый файл с тем, чтобы затруднить его распознавание антивирусными программами, и тестируют получившееся приложение, чтобы убедиться в отсутствии антивирусного детекта.

На следующем этапе вредоносное приложение попадает к распространителям, которые «раздают» его пользователям на тех или иных условиях, заранее согласованных с разработчиками. Иногда вирусописатели просто покупают у владельцев ботнетов, состоящих из инфицированных троянцами-загрузчиками компьютеров, какое-то количество установок вредоносной программы на ПК жертв, иногда приобретают наборы эксплойтов и доступ к некоторому количеству взломанных веб-сайтов, после чего занимаются распространением сами. В этом случае очевидно, что взломом интернет-ресурсов, поиском уязвимостей и разработкой эксплойтов занимаются отдельные группы киберпреступников. Иногда команда разработчиков просто «сдает» своего троянца «в аренду», размещая на подпольных форумах соответствующие объявления и предлагая потенциальным партнерам различные условия — либо фиксированную помесячную оплату за эксплуатацию созданной ими вредоносной программы, либо процент от прибыли, если им принадлежит управляющий сервер с функцией биллинга. Например, подобным образом распространяются некоторые троняцы-шифровальщики: их создатели предлагают распространителям саму программу и доступ к управляющему серверу, на котором жертвы энкодера могут получить ключ для расшифровки файлов,

оплатив выкуп. Прибыль делится между разработчиками и распространителями троянца.

Наконец, если вредоносная программа предназначена, например, для сбора конфиденциальной информации и шпионажа, полученные с ее помощью данные также нередко выставляются на продажу на подпольных форумах, и ею пользуются в своих целях другие злоумышленники. Так, кардеры приобретают дампы и реквизиты банковских карт с тем, чтобы впоследствии обналичить их или приобрести на них какой-либо товар и реализовать его через подставных лиц (так называемых дропов). Мошенники охотно покупают логины и пароли для доступа к учетным записям в социальных сетях, спамеры — базы действующих адресов электронной почты и т. д. Отдельные личности и группы разрабатывают поддельные веб-страницы, которые могут впоследствии использоваться для фишинга, а также их отдельные элементы, применяющиеся для осуществления веб-инжектов: весь этот «товар» охотно покупают вирусописатели и владельцы бот-сетей.

Иными словами, современный киберпреступный мир весьма неоднороден, роли в нем тщательно распределены, а подпольные форумы напоминают порой огромный восточный базар, где все продается и покупается.

КАК ВЫЧИСЛИТЬ ВИРУСОПИСАТЕЛЯ?

С одной стороны, поймать сетевого злоумышленника или вычислить вирусописателя — задача не из самых простых, поскольку Интернет в сочетании с различными средствами обеспечения анонимности предоставляет в распоряжение киберпреступников весьма защищенную среду, в которой они чувствуют себя как рыба в темном омуте. С другой стороны, это ощущение полной анонимности и вседозволенности порой и подводит сетевых злодеев, заставляя их терять бдительность и оставлять следы, по которым их могут впоследствии вычислить специалисты по информационной безопасности. Существует отдельная дисциплина — *форензика*, — являющаяся подразделом криминалистики и занимающаяся изучением

различных аспектов при проведении расследований киберпреступлений и всевозможных инцидентов в сфере информационной безопасности. Но иногда имеется возможность вычислить злоумышленника и без всякой форензики.

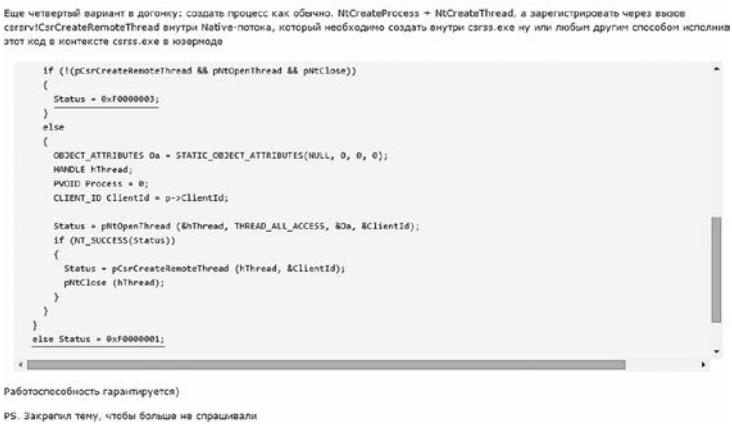
Так, известен случай, когда один из администраторов ботнета хранил на том же узле, где он разместил управляющий сервер своей бот-сети, личные фотографии из серии «я и моя девушка у меня дома». Снимки были выполнены с использованием смартфона Apple iPhone, который, как известно, вместе с изображением автоматически сохраняет данные о времени и географической точке, в которой был получен снимок. Вычислить злоумышленника при таких обстоятельствах не составило особого труда.

В другой подобной ситуации киберпреступник оказался более умен и не хранил на управляющем сервере какую-либо компрометирующую его информацию. Вместо этого он разместил среди обслуживающего ботнет программного обеспечения скрипт на языке PHP, через определенные промежутки времени сбрасывавший статистику по работе бот-сети СМС-сообщением на определенный номер мобильного телефона. Сим-карту, к которой был привязан этот номер, злоумышленник предусмотрительно приобрел с рук, и потому был уверен, что вычислить его персональные данные с ее помощью невозможно. Однако специалистам по информационной безопасности оказалось достаточно всего лишь «погуглить» данный телефонный номер, чтобы найти в сети несколько десятков объявлений по купле-продаже различных предметов и личных вещей, в которых нынешний владелец номера оставлял его в качестве контактного. Одним из таких объявлений оказалось сообщение о продаже автомобиля, содержащее фотографию, где был прекрасно виден государственный регистрационный знак с номером авто. Остальное было делом техники.

А вот еще один практический случай, позволивший аналитикам косвенным образом определить авторов одной из опасных вредоносных программ. В апреле 2014 года специалисты антивирусной компании «Доктор Веб» завершили исследование сложного многокомпонентного троянца BackDoor.Gootkit.112, о чем опубликовали на корпоративном сайте весьма подробный отчет. Из него, в частности, можно почерпнуть следующее.

Основная полезная нагрузка бэкдора реализована в виде исполняемого файла объемом около 5 Мбайт, базовая часть которого представляет собой интерпретатор языка JavaScript под названием Node.JS, создающего удобный интерфейс для работы с различными встроенными объектами. В архитектуре этого троянца был реализован редко применяемый злоумышленниками метод внедрения вредоносного кода в процессы запущенных приложений (инжектов). Аналогичный метод ранее был подробно описан на одном из интернет-форумов пользователем, скрывающимся под псевдонимом Great. В опубликованной им статье приведены примеры кода, использующего характерные статусы возврата — в точности такие же статусы обнаруживаются и в коде BackDoor.Gootkit.112. Можно предположить, что вирусописатель в процессе создания бэкдора просто-напросто позаимствовал готовое решение из открытых источников, однако здесь внимание аналитиков привлекло еще одно обстоятельство. В примерах кода, опубликованного на форуме wasm.ru, в одну из функций передается структура с уникальным именем *DRIVER_TO_SHELLCODE_PARAMETERS*. Структура с таким же в точности именем встречается в материале, опубликованном в личном блоге другого автора, описывающего данную технологию инжектов. При этом в указанной статье прямо сообщается, что этот код является собственной разработкой автора блога, созданной им совместно с другим программистом по имени Илья, также известным под ником Great.

В этом же блоге автор сообщает о своей искренней приверженности фреймворку Node.JS, возможности которого активно используются в коде троянца BackDoor.Gootkit.112. Так, автор опубликовал на своем сайте несколько весьма любопытных статей, содержащих практические примеры реализации кода, одна из которых называется «NodeJS \ C++: Нативное расширение для реестра» — в ней описываются методы работы с ветвью системного реестра Windows, имеющей характерное имя *SOFTWARE \ CXS*. В другой своей статье, озаглавленной «NodeJS: Spyware на Javascript?» владелец блога повествует об архитектуре специального шпионского модуля SpywareModule, все методы в котором имеют характерный префикс «sp». Удивительно, но факт: аналогичные структуры в практически неизменном виде встречаются в дизассемблированном


Еще четвертый вариант в догонку: создать процесс как обычно. NtCreateProcess + NtCreateThread, а зарегистрировать через вызов csrsrv!CsrCreateRemoteThread внутри Native-потока, который необходимо создать внутри csrss.exe ну или любым другим способом исполнив этот код в контексте csrss.exe в юзермоде

```
if (!(pCsrCreateRemoteThread && pNtOpenThread && pNtClose))
{
  Status = 0xf0000003;
}
else
{
  OBJECT_ATTRIBUTES Oa = STATIC_OBJECT_ATTRIBUTES(NULL, 0, 0, 0);
  HANDLE hThread;
  PVOID Process = 0;
  CLIENT_ID ClientId = p->ClientId;

  Status = pNtOpenThread (&hThread, THREAD_ALL_ACCESS, &Oa, &ClientId);
  if (NT_SUCCESS(Status))
  {
    Status = pCsrCreateRemoteThread (hThread, &ClientId);
    pNtClose (hThread);
  }
}
}
else Status = 0xF0000001;
```

Работоспособность гарантируется)

PS. Закрепил тему, чтобы больше не спрашивали

Рис. 29. Пример того, как простая публикация в блоге может

коде BackDoor.Gootkit.112. Как любил говорить один популярный телеведущий: «Совпадение? Не думаю».

Безусловно, размещение в Интернете статей, подробно раскрывающих некоторые технические аспекты работы вредоносных программ, само по себе еще ни о чем не говорит, однако определенные сомнения специалистов, расследующих подобные вирусозависимые инциденты, в некоторых случаях может развеять сравнение даты компиляции исполняемого файла изучаемого троняца и даты публикации текста, описывающего его внутреннюю архитектуру, известную только разработчику.

А вот еще пример, связанный с распространением вредоносной программы Trojan.BtcMine.218, предназначенной для скрытной добычи (майнинга) криптовалют. Этот троянец, обнаруженный специалистами антивирусной компании «Доктор Веб», устанавливался на компьютер жертвы с помощью дроппера, написанного на языке AutoIt. При этом дроппер содержал в своей структуре следующую характерную строку:

*FILEINSTALL («C: \ Users * ***Antonio*** *\ Desktop \ Glue \ Install.exe», @ TEMPDIR & " \ Setup_2.exe»)*

Из этой строки становится очевидно, что некий вирусописатель, имеющей на своем компьютере, работающем под управлением Windows, учетную запись пользователя с именем Antonio, собрал на Рабочем столе своего ПК дроппер в виде файла Install.

```
hCSRSRV = get_module_handle(&szCSRSRV_dll);
hNTDLL = get_module_handle(&szNTDLL_DLL);
if (hCSRSRV && hNTDLL)
    {
    CsrCreateRemoteThread = $int (_stdcall *)(int, int *))get_proc_addr_by_hash(hCSRSRV, 0x5846EB85);
    NtOpenThread = (int (_stdcall *)(int *, signed int, OBJECT_ATTRIBUTES *, int *))get_proc_addr_by_hash(
                                                                                    hNTDLL, 0xF88A31D1);
    NtClose = (void (_stdcall *)(int))get_proc_addr_by_hash(hNTDLL, 0x8B8E133D);
    NtTerminateThread = get_proc_addr_by_hash(hNTDLL, 0xAC3C9DC8);
    if (CsrCreateRemoteThread && NtOpenThread && NtClose)
    {
        v1 = *(_DWORD *)(inj_context + 8);
        from_ctx = *(_DWORD *)(inj_context + 4);
        from_ctx2 = v1;
        status = NtOpenThread(&hThread, 0x1FFFFF, &oa, &from_ctx);
        if (!status)
        {
            status = CsrCreateRemoteThread(hThread, &from_ctx);
            NtClose(hThread);
            }
        }
        else
        {
            status = 0xF9990003;
            }
        }
        else
        {
            status = 0xF0000001;
        }
        if (ntTerminateThread)
```

навлечь на автора подозрения в вирусописательстве

exe. Это приложение запускает на компьютере жертвы загрузчик, который, используя специальный конфигурационный файл в формате XML, скачивает из Интернета и устанавливает на атакуемую машину добытчик криптовалюты. В этом конфигурационном файле указано наименование учетной записи, в пользу которой троянец будет эксплуатировать аппаратные ресурсы зараженного ПК для майнинга криптовалюты и на которую будет перечисляться все добытое. Этого пользователя зовут Tonycraft.

Судя по всему, сам Antonio (Tonycraft) не обладает достаточно глубокими познаниями в программировании, поскольку основные модули троянца он поручил написать другому человеку. Однако и здесь Tony, судя по всему, решил немного сэкономить, выбрав в качестве своего партнера не слишком искушенного вирусописателя: создавая код троянца, тот по неосмотрительности оставил в модулях вредоносной программы следующую отладочную строку: «*c: \ Users * ***Кошевой Дмитрий*** *\ Documents \Visual Studio 2012 \ Projects \ Miner \ Instal \ obj \ Debug \ Instal.pdb*».

О чем говорит нам этот «автограф» вирусописателя? О том, что он пользуется компьютером под управлением Microsoft Windows с учетной записью «Кошевой Дмитрий», проект он назвал Miner и создал его с применением среды разработки Visual Studio 2012. Простой поиск по сети «В Контакте»

без труда позволил выявить пользователя, скрывающегося под псевдонимом Tonycoin, активно рекламирующего на своей страничке сайт, посвященный криптовалюте Bitcoin. По странному стечению обстоятельств, этот же самый сайт является ресурсом, с которого установщик загружает на зараженный компьютер основной модуль троянца Trojan.BtcMine.218. Ну, а среди контактов Tonycoin при определенной настойчивости отыскивается и программист «Дмитрий Кошевой».

Известна и вовсе феерическая ситуация, когда троянец, воровавший вводимые пользователем данные в формы авторизации на сайтах «В Контакте», «Одноклассники», «Mail.RU» и «Яндекс», передавал их скрипту, работающему на принадлежащем злоумышленнику сервере. Притом домен для этого сервера он умудрился зарегистрировать на свое настоящее имя, благодаря чему специалистам по информационной безопасности с помощью запроса к службе Whois удалось выяснить его телефон, имя, фамилию и город проживания, а по этим данным — отыскать его анкету на сайте знакомств, где злоумышленник собственноручно опубликовал остальную недостающую информацию: фотографию, рост, вес, дату и место рождения, а также сведения об образовании.

Все описанные случаи свидетельствуют о том, что вирусописателей чаще всего удается вычислить вследствие элементарных просчетов в области обеспечения анонимности, а также благодаря их самоуверенности, неопытности или стремлению к дешевой славе.

ГЛАВА 10. МЕТОДЫ БОРЬБЫ

Считается, что антивирусные программы появились на свет практически одновременно с первыми компьютерными вирусами, поскольку пользователи остро нуждались в адекватном средстве для борьбы с вредоносным ПО. История развития антивирусной индустрии насчитывает множество увлекательных страниц, однако о самой первой из них до сих пор ведутся ожесточенные споры. Кто придумал антивирусы? Когда появилась первая подобная программа? Об истории развития технологий информационной безопасности и некоторых особенностях работы антивирусных программ мы и поговорим в настоящей главе.

НЕМНОГО ИСТОРИИ

Существует мнение, что первое приложение для выявления и удаления вредоносного ПО разработал в 1984 году американский программист Энди Хопкинс — речь идет об утилите CHK4BOMB, позволявшей сканировать исполняемые файлы в поисках характерных фрагментов кода и текстовых сообщений. Другая утилита того же автора, BOMBSQAD, могла перехватывать операции записи в файл и команду форматирования. В течение длительного времени этих двух утилит было вполне достаточно для обеспечения информационной безопасности, поскольку вредоносных программ существовало не так уж и много. Например, в 1985 году по американским электронным доскам объявлений (BBS) гулял текстовый файл «Грязная дюжина» («The Dirty Dosen — An Unloaded Program Alert List»), представлявший собой составленный владельцем одной из BBS Томом Нельфом список опасных программ, которые пользователи загружали в файлообменники. Первоначально перечень включал всего лишь 12 вредоносных приложений, но он постоянно пополнялся. Ассортимент опасных программ ширился, и вскоре простого списка оказалось уже недостаточно для отражения все новых и новых угроз — пользователи понемногу осознали необходимость в постоянной антивирусной защите.

Спрос, как известно, рождает предложение, и первая резидентная антивирусная программа под названием DPROTECT, разработанная программистом и основателем американской софтверной компании GEE WIZ Sofware Company Ги Вонгом, появилась в начале 1985 года. Программа DPROTECT распространялась бесплатно, однако желающие могли сделать добровольное пожертвование в размере пяти долларов. Приложение не использовало в своей работе вирусные базы, содержащие

сигнатуры вредоносных программ, а отслеживало активность запущенных программ, перехватывая обращения к файловой системе. Таким образом резидентный монитор DPROTECT позволял обезопасить защищаемый компьютер от деструктивных действий вирусов и троянцев, которые могли, например, отформатировать диск или испортить загрузочную запись.

Вторая половина 80-х годов ознаменовалась широкой экспансией на международном рынке персональных компьютеров семейства IBM PC и ростом популярности операционной системы MS-DOS. К этому же периоду относятся и первые масштабные вирусные эпидемии: летом 1986 года множество компьютеров оказались заражены стелс-вирусом Brain. Написанный студентом из Новой Зеландии вирус Stoned, получивший распространение в том же 1987 году, заражал загрузочную запись ПК и блокируя запуск ОС. От этой напасти пострадало несколько тысяч компьютеров по всему миру. А в пятницу, 13 мая 1988 года пользователи познакомились с вирусом Jerusalem, уничтожавшим приложения при попытке их запуска. Эта вредоносная программа вызвала настоящую пандемию, распространившись на территории не только США, но также Европы и Азии. Разумеется, отыскались толковые парни, очень быстро осознавшие, что на компьютерной безопасности можно неплохо заработать — одним из них оказался простой программист из Lockheed Corporation по имени Джон Макафи, выпустивший в 1988 году коммерческий продукт под названием McAfee VirusScan, а в 1989 году основавший компанию имени себя. Вскоре, почувствовав запах прибыли, подтянулись и другие игроки: в 1991 году состоялся релиз первой версии приложения под названием Norton AntiVirus.

Ну, а на отечественном рынке истинным первопроходцем стал Дмитрий Лозинский, разработавший в 1988 году, практически одновременно с Макафи, антивирусную программу-сканер Aidstest, использовавшую технологию сигнатурного поиска угроз. Приложение, дистрибуцией которого занималась компания «Диалог Наука», быстро завоевало заслуженную популярность у пользователей, на долгие годы став своего рода стандартом антивирусного ПО. Дмитрий Николаевич и по сей день вносит ощутимый вклад в борьбу с вирусными угрозами — в качестве одного из ведущих сотрудников компании «Доктор

Веб». За минувшие годы мир изменился до неузнаваемости, появились троянцы для различных операционных систем и мобильных платформ, а разработка антивирусного ПО превратилась в целую индустрию. Однако идеи, впервые реализованные Энди Хопкинсом в далеком 1984 году, не утратили своей актуальности и по сей день.

КАК АНТИВИРУСНЫЕ КОМПАНИИ ПОПОЛНЯЮТ БАЗЫ?

Применительно к современным антивирусным технологиям само понятие «антивирус» — это скорее дань моде, нежели термин, правильно отражающий суть вещей. Классические файловые вирусы, то есть вредоносные программы, способные заражать исполняемые файлы или динамические библиотеки и распространяться без участия пользователя, сегодня очень большая редкость. Подавляющее большинство встречающихся сейчас в «дикой природе» вредоносов — это трояны, не способные ни к заражению файловых объектов, ни к саморепликации. Чуть реже в руки аналитиков попадаются черви: эти программы могут создавать свои копии на съемных носителях или сетевых дисках, «расползаться» по сети или каналам электронной почты, но файлы заражать не умеют. Все остальные традиционные категории вредоносного ПО отличаются друг от друга лишь базовым набором функций, но по своей архитектуре могут быть сведены к этим трем группам.

Как образцы вредоносов попадают в вирусные лаборатории? Каналов поступления новых семплов у антивирусных компаний традиционно несколько. Прежде всего, это онлайн-сервисы вроде VirusTotal, то есть серверы, на которых любой анонимный пользователь может проверить детектирование произвольного файла сразу десятком самых популярных антивирусных движков. Каждый загруженный образец вне зависимости от результатов проверки автоматически отправляется вендорам для более детального исследования.

Очевидно, что с подобных ресурсов в вирусные лаборатории прилетает огромный поток мусора, включая совершенно

безобидные текстовые файлы и картинки, поэтому на входе он фильтруется специально обученными роботами и только после этого передается по конвейеру дальше. Этими же сервисами успешно пользуются небольшие компании, желающие сэкономить на содержании собственных вирусных лабораторий. Они копируют в свои базы чужие детекты, из-за чего регулярно испытывают эпические проблемы, когда какой-нибудь вендор в шутку или по недоразумению поставит вердикт *infected* на тот или иной компонент такого антивируса, после чего тот радостно переносит в карантин собственную библиотеку и с грохотом падает, вызывая шок у пользователей и истерический хохот у конкурентов.

Второй канал — «самотек», подозрительные файлы, которые пользователи передают в вирлаб через сайт антивирусной компании, по запросу службы поддержки или выгружают из карантина. Третий канал — ханипоты, специальные приманки для вирмейкеров в виде виртуальных серверов с открытыми наружу портами и логинами-паролями вроде root/root, куда некоторые ботоводы радостно заливают свои творения, дивясь криворукости админов. Наконец, четвертый путь — обмен базами между самими вендорами, но в последние годы в силу обострившейся конкуренции на рынке и сузившейся кормовой базы кооперация между антивирусными компаниями практически сошла на нет.

После того как семпл попадает в вирусную лабораторию, он сортируется по типу файла и исследуется автоматическими средствами аналитики, которые могут установить вердикт по формальным или техническим признакам — например, по упаковщику. И только если роботам раскусить вредоноса не удалось, он передается вирусным аналитикам для проведения инструментального или ручного анализа.

КОМПОНЕНТЫ АНТИВИРУСНОЙ ПРОГРАММЫ

Антивирусные программы различных производителей включают в себя разное число компонентов, и даже более: одна

и та же компания может выпускать несколько версий антивируса, включающих определенный набор модулей и ориентированных на различные сегменты рынка. Например, некоторые антивирусы располагают компонентом Родительского контроля, позволяющего ограничивать доступ несовершеннолетних пользователей компьютера к сайтам определенных категорий, или регулировать время их работы в системе, а некоторые — нет. Так или иначе, обычно современные антивирусные приложения обладают следующим набором функциональных модулей.

- **Антивирусный сканер** — утилита, выполняющая поиск вредоносных программ на дисках и в памяти устройства по запросу пользователя или по расписанию.
- **Резидентный монитор** — компонент, выполняющий отслеживание состояния системы в режиме реального времени и блокирующий попытки загрузки или запуска вредоносных программ на защищаемом компьютере.
- **Брандмауер (фаервол)** — компонент, выполняющий мониторинг текущего соединения, включая анализ входящего и исходящего трафика, а также проверяющий исходный адрес и адрес назначения в каждом передаваемом с компьютера и поступающем на компьютер пакете информации — данные, поступающие из внешней среды на защищенный брандмауэром компьютер без предварительного запроса, отслеживаются и фильтруются. С функциональной точки зрения брандмауэр выступает в роли своеобразного фильтра, контролирующего поток передаваемой между локальным компьютером и Интернетом информации, защитного барьера между компьютером и всем остальным информационным пространством.
- **Веб-антивирус** — компонент, предотвращающий доступ пользователя к опасным ресурсам, распространяющим вредоносное ПО, фишинговым и мошенническим сайтам с использованием специальной базы данных адресов или системы рейтингов.
- **Почтовый антивирус** — приложение, выполняющее проверку на безопасность вложений в сообщения электронной почты и (или) пересылаемых по электронной почте ссылок.

- **Анти-руткит** — модуль, предназначенный для борьбы с руткитами (вредоносными программами, обладающими способностью скрывать свое присутствие в инфицированной системе).
- **Модуль превентивной защиты** — компонент, обеспечивающий целостность жизненно важных для работоспособности системы данных и предотвращающий опасные действия программ.
- **Модуль обновления** — компонент, обеспечивающий своевременное обновление других модулей антивируса и вирусных баз;
- **Карантин** — централизованное защищенное хранилище, в которое помещаются подозрительные (в некоторых случаях — определенно инфицированные) файлы и приложения до момента вынесения в их отношении окончательного вердикта.

В зависимости от версии и назначения антивирусной программы, она может включать в себя и другие функциональные модули, например компоненты для централизованного администрирования, удаленного управления и т. д.

СИГНАТУРНОЕ ДЕТЕКТИРОВАНИЕ

Современные антивирусные программы используют несколько различных методик обнаружения вредоносных программ в различных сочетаниях. Основная из них — это сигнатурное детектирование угроз.

Данный метод детектирования вредоносных программ основывается на создании так называемых *сигнатур* — уникальных цифровых идентификаторов файла, представляющих собой специальный набор байтов и получаемых на основе содержимого исследуемого файла. Фактически сигнатура представляет собой своего рода «отпечаток пальцев» файла — с помощью сигнатуры можно однозначно идентифицировать тот или иной файл или приложение.

Сигнатуры собираются в набор, называемый в*ирусными базами*. Вирусные базы антивирусных программ периодически

обновляются с целью добавления в них сигнатур новых угроз, исследованных за истекшее с момента последнего обновления время в лаборатории антивирусной компании.

В процессе проверки защищаемого устройства антивирусная программа исследует хранящиеся на дисках (или загружаемые из Интернета) файлы и сравнивает результаты исследования с сигнатурами, записанными в антивирусной базе. В случае совпадения такой файл считается вредоносным. Данная методика сама по себе имеет значительный недостаток: злоумышленнику достаточно изменить структуру файла на несколько байт, и его сигнатура изменится. До тех пор, пока новый образец вируса или троянца не попадет в вирусную лабораторию и его сигнатура не будет добавлена в базы, антивирус не сможет распознать и ликвидировать данную угрозу.

ПОВЕДЕНЧЕСКИЙ АНАЛИЗ

Помимо сигнатурного детектирования большинство современных антивирусных программ используют те или иные механизмы *поведенческого анализа*. Поведенческий анализ можно отнести к разновидности вероятностного анализа — как следует из наименования данного метода, антивирусная программа следит за поведением приложений, и если оно кажется ей «подозрительным», блокирует работу потенциально опасной программы.

Одним из методов безопасного исследования поведения приложения является его запуск в так называемой песочнице (sandbox) — защищенном изолированном виртуальном контейнере, из которого приложение не может получить доступ к компонентам ОС и файловой системе. Если поведение программы вызывает у антивируса подозрения — например, она пытается встроиться в процессы запущенных приложений (выполнить инжект), модифицировать загрузочную запись, изменить структуру исполняемого файла и т. д., она может быть признана потенциально опасной или вредоносной.

Одним из вариантов поведенческого анализа является, в частности, технология Origin TracingTM, активно применяемая

компанией «Доктор Веб» в некоторых программных решениях, например в Антивирусе Dr.Web для Android. В рамках данной технологии для каждого образца вредоносной программы создается специальная запись, описывающая поведение троянца в инфицированной системе. Если приложение, запустившись на устройстве, действует в соответствии с этим шаблоном, оно признается вредоносным. Такой подход позволяет, во-первых, заметно сократить объем вирусных баз (поскольку одной записью в этом случае детектируется целое семейство вредоносных программ), а во-вторых, успешно определять еще неизвестные аналитикам угрозы: если поведение приложения укладывается в эталонную модель, оно будет обезврежено вне зависимости от того, попадал раньше этот образец в вирусную лабораторию на анализ или нет. Нужно отметить, что в мобильной операционной системе Android, где у вредоносных программ (в отличие от Windows) не так много свободы действий, этот метод работает весьма эффективно. Все угрозы с префиксом «origin», детектируемые на смартфонах и планшетах Антивирусом Dr.Web для Android, обезвреживаются при помощи технологии Origin TracingTM.

ЭВРИСТИЧЕСКИЙ АНАЛИЗ

Эвристический анализ — разновидность вероятностного анализа вредоносных программ, основанная на логических алгоритмах, позволяющих выявить и обезвредить потенциально опасное приложение. Эвристический анализ приходит пользователям на помощь в тех случаях, когда угрозу не удается обнаружить с помощью сигнатурного детектирования.

Упрощая, основной принцип эвристического анализа можно описать следующим образом. Каждой функции, которую может реализовывать программа в операционной системе, назначается некий условный «рейтинг опасности». Какие-то действия приложения могут считаться менее опасными, другие — более. Если по совокупности выполняемых приложением действий оно превышает некий условный «порог безопасности», то признается потенциально вредоносным.

Например, если какая-то программа работает в фоновом режиме, не имеет графического интерфейса, последовательно опрашивает удаленные серверы, а потом пытается скачать с них и запустить в системе некое приложение, оно с высокой долей вероятности может оказаться троянцем-загрузчиком. Или утилитой обновления браузера Google Chrome, которая действует аналогичным образом. В этом, очевидно, и кроется основная «ахиллесова пята» эвристического метода анализа вирусных угроз — большая вероятность ложного срабатывания.

Другой метод эвристического анализа — эмуляция исполнения программы. Антивирус загружает подозрительное приложение в собственную буферную память, выполняет разбор кода на инструкции и выполняет их по очереди, последовательно проверяя результат.

Эвристический анализ применяется с целью выявления и обезвреживания угроз, еще неизвестных антивирусу, — то есть тех, сигнатуры которых на текущий момент отсутствуют в вирусных базах. Отсюда логически вытекает еще один недостаток эвристических алгоритмов: даже если неизвестную ранее угрозу удается обнаружить, ее далеко не всегда получается сразу «вылечить». Во многих случаях пользователю приходится ожидать появления очередного обновления вирусных баз, содержащего алгоритмы лечения конкретно для этой вредоносной программы.

ПРОАКТИВНАЯ ЗАЩИТА (HIPS)

Проактивную антивирусную защиту, HIPS (Host-based Intrusion Prevention System, англ. «система предотвращения вторжений»), также можно отнести к разновидности антивирусной защиты на основе поведенческого анализа. Антивирус следит за запущенными приложениями и информируют пользователя о тех или иных действиях программы. Решение о том, позволить или запретить программе выполнять какое-либо действие, принимает пользователь. Это — классический вариант реализации HIPS. Существует еще так называемый экспертный вариант, при котором антивирус самостоятельно

блокирует действия тех или иных приложений на основе набора заложенных в него правил и разрешений. Пользователь может при необходимости добавить какую-либо программу в список исключений, разрешив ей выполнение тех или иных действий в защищаемой системе.

МЕТОДИКИ ПРОТИВОДЕЙСТВИЯ АНТИВИРУСАМ

К сожалению, борьба вирусописателей и производителей антивирусных программ носит перманентный характер: первые непрерывно изобретают все новые способы обхода антивирусной защиты, вторые стараются совершенствовать алгоритмы поиска и обнаружения вредоносного ПО. Давайте перечислим основные методики, которыми пользуются злоумышленники для противодействия антивирусным приложениям.

Переупаковка

Это самый распространенный и популярный метод, активно применяемый вирусописателями для обхода сигнатурного детекта. Как я уже упоминал, сигнатура является своего рода аналогом «отпечатков пальцев» каждого конкретного файла, при этом она уникальна для файлового объекта. Соответственно, если в файл будут внесены даже незначительные изменения, антивирус не сможет «опознать» его с помощью сигнатуры, и такой файл не будет детектироваться антивирусом до тех пор, пока не попадет в исследовательскую лабораторию антивирусной компании.

Наиболее простой способ изменить структуру файла, не меняя его функциональных возможностей, — «накрыть» его программным упаковщиком. Программные упаковщики сжимают содержимое файла приложения и дописывают к нему код, необходимый для распаковки и выполнения программы. Некоторые из них к тому же включают различные функции шифрования, затрудняющие анализ и исследование подобного приложения. Этим и пользуются злоумышленники.

При каждой повторной упаковке файла его сигнатура меняется и он становится «невидимым» для системы сигнатурного детектирования антивируса. Некоторые вирусописатели для затруднения исследования вируса или троянца упаковывают и шифруют свои творения в «несколько слоев» — тогда под одним упаковщиком прячется другой сжатый и зашифрованный объект, под ним — еще один, и вся конструкция напоминает в итоге этакую логическую «матрешку», добраться до «сердцевины» которой бывает порой весьма непросто.

Иногда киберпреступники применяют и иной метод: на сервере, с которого жертвам раздается вредоносное ПО, устанавливается специальный сценарий. При активизации этого сценария (например, при переходе пользователя по ссылке) скрипт извлекает из соответствующей директории сервера бинарный файл вируса или троянца, упаковывает его «на лету» и только после этого «отдает» его пользователю. Таким образом каждая жертва получает свой собственный, уникальный, экземпляр вредоносной программы, гарантированно не детектируемой по сигнатуре.

Обфускация

Обфускация (от англ. obfuscate — «запутывать», «сбивать с толку») — сознательное запутывание, усложнение кода вредоносной программы с сохранением ее функциональности в целях затруднения ее исследования и анализа. Для обфускации вирусописатели иногда добавляют в приложение различный «мусорный» код, ненужные инструкции, множественные переходы или вызовы различных функций и т. д. Существуют специальные утилиты — *обфускаторы*, созданные для запутывания кода приложений.

Обфускация приложений затрудняет реверс-инжиниринг, то есть декомпиляцию вредоносной программы и изучение ее функциональных возможностей на уровне кода, однако одновременно с этим усложняет вирусописателям отладку приложения, а в некоторых случаях увеличивает его размер и снижает быстродействие.

Антиотладка

Большинство современных вредоносных программ оснащено различными мощными механизмами *антиотладки*, препятствующими их исследованию. Ряд вирусов и троянцев в момент начала работы проверяют, не пытаются ли их запустить в изолированной среде («песочнице»), под отладчиком или в виртуальной машине. Осуществляется это разными методами, например попытками получить имена работающих процессов (и их сравнением с заданным списком), поиском характерных строк в заголовках открытых окон и т. д. Если вредоносное приложение определяет попытку запуска в виртуальной среде или под отладчиком, оно завершает свою работу.

Аналогично многие троянцы и вирусы ищут среди установленных или запущенных программ приложения популярных антивирусов и пытаются завершить их, а если это не получается, прекращают свою работу. Бывают и более интересные варианты: так, троянец, известный под именем Trojan.VkBase.73, менял параметры загрузки Windows, устанавливал в системе специальную службу, которая при перезагрузке системы в безопасном режиме удаляла установленные на компьютере антивирусы. Затем троянец размещал в области уведомлений Панели задач значок соответствующего антивирусного приложения, которое было ранее им удалено. Таким образом, пользователь даже не догадывался, что его компьютер больше не имеет антивирусной защиты. После успешного удаления средств информационной защиты на экран выводилось сообщение на русском или английском языке (в зависимости от версии антивирусного ПО) следующего содержания: «*Внимание! Антивирус [название антивируса] работает в режиме усиленной защиты. Это временная мера, необходимая для моментального реагирования на угрозы со стороны вирусных программ. От вас не требуется никаких действий*». Данное сообщение демонстрировалось, чтобы пользователь не проявлял беспокойство, обнаружив, что значок антивирусной программы в Области уведомлений Панели задач Windows более не реагирует на щелчки мышью.

Для обхода «песочницы» некоторые вирусы или троянцы имеют специальные «механизмы замедления», которые притормаживают вредоносный функционал приложения или усыпляют

его на некоторый срок, активизируя деструктивный функционал по истечении определенного времени. Это позволяет усыпить бдительность защитной программы, которая, запустив приложение в «песочнице» и убедившись в его безопасности, дает ему «зеленый свет». Например, один из современных троянцев использует такой механизм обхода автоматизированных систем анализа: создает во временной папке файл, в которой миллион раз записывает по одному байту, а потом миллион раз читает из него также по одному байту. В результате таких безобидных длительных циклических действий процедура поведенческого анализа завершается раньше, чем троянец начинает реализовывать свой основной вредоносный функционал.

ЗАКЛЮЧЕНИЕ

Алгоритмы детектирования вредоносного ПО, так же как и методы борьбы с антивирусами, совершенствуются с каждым днем. Однако уникальные алгоритмы обхода антивирусной защиты появляются крайне редко — как правило, вирусописатели используют стандартные и давно проверенные на практике методики. Разработчики защитных программ, хорошо знающие все эти методики, часто оказываются на полшага впереди злоумышленников. Именно поэтому антивирусная защита все еще остается довольно эффективным методом борьбы с вредоносными и потенциально опасными программами — особенно для неопытных пользователей.

ГЛОССАРИЙ

adware — семейство рекламных приложений, традиционно относящихся к нежелательным.

bitcoin — старейшая в истории *криптовалюта*; электронная платежная система, допускающая проведение защищенных и анонимных транзакций.

DDoS (Distributed Denial of Service), «распределенный отказ в обслуживании» — разновидность сетевых атак, при которых к атакуемому узлу генерируется большое количество запросов в единицу времени, которые он не в состоянии обработать, что вызывает его отказ.

DGS, domain generation system — технология, позволяющая составляющим *ботнет* вредоносным программам автоматически генерировать имена управляющих серверов с использованием специальных алгоритмов.

fakealert — семейство программ, традиционно относящихся к «нежелательным»: они запугивают пользователя, «отыскивая» на компьютере различные несуществующие проблемы и требуют оплаты за их устранение.

jailbreak — несанкционированная разработчиком устройства процедура получения доступа к файловой системе (термин применяется в основном по отношению к смартфонам и планшетам производства корпорации Apple).

HIPS (Host-based Intrusion Prevention System) — система проактивной антивирусной защиты, отслеживающая различные действия приложений и выборочно блокирующая их.

MITM (Man in the middle — «человек посередине») — способ атак, при которых злоумышленник внедряется в канал связи между отправителем и получателем информации и может видоизменять эту информацию «на лету» непосредственно в процессе ее передачи.

P2P (Peer-To-Peer), или пиринговые одноранговые сети, — сети, в которых отсутствует управляющий сервер, а данные передаются непосредственно от одного узла к другому.

ransomware — общее наименование программ-вымогателей.

sinkhole — метод перехвата управления ботнетом, работающим с использованием технологии DGS, при котором осуществляется регистрация «поддельного» управляющего сервера бот-сети, а действующие серверы выводятся из строя.

TDS, Traffic Distribution Systems — системы управления трафиком, позволяющие злоумышленникам перенаправлять пользователей на различные сетевые ресурсы в зависимости от заданных условий.

TOR (от англ. The Onion Router, «луковый маршрутизатор») — система так называемой луковой маршрутизации, состоящая из многослойной структуры прокси-серверов и позволяющая устанавливать анонимное соединение, защищенное от стороннего прослушивания и слежения.

Whois — онлайн система, позволяющая установить по имени домена данные его администратора.

антивирусный сканер — утилита, выполняющая поиск вредоносных программ на дисках и в памяти устройства по запросу пользователя или по расписанию.

антиотладка — используемый вирусописателями комплекс мер, позволяющий оказывать противодействие антивирусным программам и различным методам исследования вирусов и троянцев.

атака на отказ в обслуживании — см. *DDoS (Distributed Denial of Service)*.

банковские троянцы — вредоносные программы, предназначенные для хищения учетных данных и файлов, необходимых для организации доступа к системам ДБО (дистанционного банковского обслуживания).

биоскит — вредоносные программы, способные инфицировать микросхемы BIOS.

блокировщик, винлокер — вредоносная программа-вымогатель, блокирующая нормальную работу операционной системы, и требующая денег за ее разблокировку.

бот — вредоносная программа, способная объединяться в ботнеты (бот-сети).

ботнет, бот-сеть — сеть зараженных устройств, умеющих обмениваться информацией и дистанционно управляемых злоумышленниками, например, с использованием одного или нескольких командных серверов.

брандмауер (фаервол) — компонент антивирусной программы, выполняющий мониторинг текущего соединения, включая анализ входящего и исходящего трафика, а также проверяющий исходный адрес и адрес назначения в каждом передаваемом

с компьютера и поступающем на компьютер пакете информации — данные, поступающие из внешней среды на защищенный брандмауэром компьютер без предварительного запроса, отслеживаются и фильтруются.

брутфорс — метод получения несанкционированного доступа к какому-либо ресурсу путем полного перебора паролей, «взлом методом грубой силы».

буткит — вредоносная программа, модифицирующая загрузочную запись с целью обеспечения своего запуска до (или одновременно с) загрузки операционной системы (но ранее запуска основных антивирусных средств защиты).

бэкдор — вредоносная программа, обладающая возможностью выполнять поступающие от злоумышленников команды, т. е. допускающая несанкционированное удаленное управление инфицированным компьютером.

веб-антивирус — компонент антивирусной программы, предотвращающий доступ пользователя к опасным ресурсам, распространяющим вредоносное ПО, фишинговым и мошенническим сайтом с использованием специальной базы данных адресов или системы рейтингов.

веб-инжект — встраивание вредоносной программой постороннего содержимого в просматриваемую пользователем в окне браузера веб-страницу.

винлокер — см. *блокировщик.*

вирус — вредоносная программа, способная к саморепликации (автоматическому распространению без участия пользователя) и заражению файловых объектов.

вирусная база — набор используемых антивирусной программой файлов, содержащих *сигнатуры* вирусов и алгоритмы лечения некоторых угроз

вымогатель — категория вредоносных программ, требующих у жертвы выкуп за определенное действие (например, разблокировку компьютера или расшифровку файлов).

граббер — вредоносная программа, позволяющая перехватывать и передавать злоумышленникам данные из заполняемых пользователем форм

ДБО, системы дистанционного банковского обслуживания — системы типа «банк — клиент», позволяющие осуществлять удаленное управление банковским счетом.

двухфакторная аутентификация — система безопасности, применяемая в некоторых ДБО, и подразумевающая дополнительную проверку подлинности при авторизации пользователя путем ввода одноразового пароля (mTAN-кода), отправляемого в СМС-сообщении.

дроппер — объект, осуществляющий извлечение содержащихся в нем основных модулей вируса или троянца, их распаковку и установку в операционной системе.

загрузочный вирус — см. *буткит.*

загрузчик, троянец-загрузчик — семейство троянцев, предназначенных для загрузки из Интернета и запуска на инфицированном компьютере других вредоносных программ.

инжект — механизм, позволяющий вирусу или троянцу встраивать вредоносный объект в запущенный и уже работающий в операционной системе процесс другого приложения, после чего внедренный объект начинает выполняться в контексте данного процесса.

инжектор — функциональный модуль вредоносной программы, реализующий встраивание вредоносных компонент в запущенный процесс другого приложения.

инфектор — модуль вредоносной программы, осуществляющий заражение файловых объектов (например, исполняемых файлов или динамических библиотек), либо загрузочной записи компьютера путем изменения их внутренней структуры.

кардер — киберпреступник, специализирующийся на мошеннических действиях с банковскими картами.

кейлоггер — программа-шпион, считывающая и сохраняющая в специальный журнал коды нажимаемых пользователем клавиш, а потом передающиая эту информацию злоумышленникам.

командный сервер (Command and Control Server, C&C) — управляющий сервер ботнета (бот-сети).

криптовалюта — электронное платежное средство, использующее для эмиссии и учета взаиморасчетов методы криптографии.

крэкер (от англ. to crack, «взламывать») — взломщик компьютерных систем, а также разработчик различных средств обхода систем лицензионной защиты.

лоадер — компонент вредоносной программы, осуществляющий загрузку других модулей (например, динамических библиотек)

в оперативную память компьютера и (в ряде случаев) настройку этих компонент в памяти.

логическая бомба —наименование категории вредоносных программ, срабатывающих при наступлении определенных условий.

макровирус — категория вирусов, написанных с использованием скриптовых языков, применяющихся для создания макросов в различных офисных приложениях, таких как Microsoft Office, в частности Microsoft Word.

обфускация (от англ. obfuscate — «запутывать», «сбивать с толку») — сознательное запутывание, усложнение кода вредоносной программы с сохранением ее функциональности в целях затруднить ее исследование и анализ.

песочница (sandbox) — защищенный изолированный виртуальный контейнер для безопасного запуска приложений.

поведенческий анализ — анализ поведения программ с целью выявления вредоносных функций.

полиморфизм вирусного кода — способность некоторых вирусов изменять собственный код в процессе выполнения.

полиморфные вирусы — вирусы, с целью затруднения своего обнаружения и уничтожения способные «на лету» изменять свой исполняемый код непосредственно в процессе его исполнения. Процедура, отвечающая за динамическое изменение кода вируса, тоже может меняться от заражения к заражению.

почтовый антивирус — приложение, выполняющее проверку на безопасность вложений в сообщения электронной почты и (или) пересылаемых по электронной почте ссылок.

превентивная защита — компонент антивируса, обеспечивающий целостность жизненно важных для работоспособности системы данных и предотвращающий опасные действия программ.

проактивная защита — метод антивирусной защиты, основанный на мониторинге деятельности работающих программ (см. *HIPS (Host-based Intrusion Prevention System)*).

процесс — виртуальное адресное пространство памяти, отведенное для выполнения программой или самой операционной системой каких-либо процедур.

реверс-инжиниринг — метод исследования вредоносных программ, основанный на их декомпиляции и изучении исходного кода.

резидентный вирус — вредоносные программы, действующие непосредственно в памяти зараженного компьютера параллельно с другими запущенными процессами. С появлением многозадачных операционных систем само понятие «резидентный вирус» принято считать устаревшим.

резидентный монитор — компонент антивируса, выполняющий отслеживание состояния системы в режиме реального времени и блокирующий попытки загрузки или запуска вредоносных программ на защищаемом компьютере.

руткит — вредоносная программа, использующая механизмы сокрытия своего присутствия в зараженной системе.

рутование — несанкционированная разработчиком устройства процедура разблокировки учетной записи суперадминистратора (root) (термин применяется в основном по отношению к смартфонам и планшетам под управлением ОС Android).

саморепликация — способность вредоносных программ к самостоятельному распространению в автоматическом режиме путем создания собственных копий без участия пользователя.

сигнатура — уникальный цифровой идентификатор файла, представляющий собой специальный набор байтов, получаемых на основе содержимого исследуемого файла.

сниффинг — анализ сетевого трафика с целью перехвата каких-либо данных.

социальная инженерия — комплекс используемых злоумышленниками психологических приемов, позволяющих обмануть пользователя или ввести его в заблуждение.

спам — не заказанная пользователем реклама, распространяющаяся по каналам электронной почты.

сплоит — см. *эксплойт*.

стелс-вирусы — вирусы, способные полностью или частично скрывать свое присутствие на инфицированном компьютере, например, путем перехвата обращений операционной системы к зараженным файловым объектам, памяти или загрузочным областям диска. Термин применяется в основном к угрозам, работающим в ОС MS-DOS, и потому считающийся устаревшим.

таргетированная сетевая атака — узконаправленная атака, рассчитанная на взлом/заражение какого-либо конкретного объекта.

тизерная реклама — назойливая реклама вызывающего, «дразнящего» содержания, либо рекламное сообщение, раскрывающее лишь часть информации о рекламируемом продукте.

троянская программа, троянец — вредоносная программа, не способная к заражению и саморепликации, распространяющаяся под видом каких-либо «полезных» приложений, файлов или документов.

уязвимость — недокументированная возможность или ошибка в программе, позволяющая злоумышленникам выполнить с ее использованием какое-либо деструктивное действие.

фишинг — обман пользователей с целью незаконного завладения конфиденциальной информацией (например, логином и паролем для доступа к какому-либо сайту) с использованием технических средств (в частности, путем создания поддельного сайта с аналогичным визуальным оформлением).

фрикер — киберпреступник, специализирующийся на взломе телефонных сетей.

фрод — сетевое мошенничество (от английского fraud — «мошенничество»).

хук — перехват вызовов API-функций в процессах приложений.

червь — вредоносная программа, способная к саморепликации, например, путем создания собственных копий на съемных носителях, в общедоступных сетевых папках, в виде файлов в пиринговой сети или по каналам электронной почты.

шифровальщик (энкодер) — вредоносная программа-вымогатель, шифрующая файлы на дисках компьютера и требующая у пользователя выкуп за их расшифровку.

шпионы (spyware) — вредоносные программы, предназначенные для хищения конфиденциальной информации.

эвристический анализ — разновидность вероятностного анализа вредоносных программ, основанная на логических алгоритмах, позволяющих выявить и обезвредить потенциально опасное приложение.

эксплойт (эксплоит, сплоит) — программа, файл, электронный документ, фрагмент исполняемого кода или последовательность команд, использующая ту или иную уязвимость для реализации различных вредоносных функций.

энкодер — см. *шифровальщик.*

СОДЕРЖАНИЕ

Серия книг

ПРОСТО

КУПИТЬ
ОТ ИЗДАТЕЛЯ
www.strata.spb.ru

ПРОСТО АРИФМЕТИКА
ПРОСТО ИГРА
ПРОСТО КОПИРАЙТИНГ
PRO ВИРУСЫ
PRO АНТИМАТЕРИЮ
PRO ТЕМНУЮ МАТЕРИЮ
ПРОСТО ХАОС
ДИНАМИЧЕСКИЙ ХАОС
ПРОСТО ФРАКТАЛ
СУПЕРФРАКТАЛ
АРТ-ФРАКТАЛ
ДУХ ЧИСЛА
ПРОСТО КИБЕРНЕТИКА
ОНИ ВНУТРИ НАС
УДИВИТЕЛЬНАЯ ОТНОСИТЕЛЬНОСТЬ
ПРОСТО ЭЛЕКТРИЧЕСТВО
PRO КВАНТОВЫЕ ЧУДЕСА
ПРОСТО ХИМИЯ АРОМАТА
PRO БОТАНИКУ
АРИСТОТЕЛЬ VS БУДДА
СИМВОЛ И АЛГОРИТМ
ПРОСТО ЭНТРОПИЯ
PRO ПАРАДОКСЫ НАУКИ
ПРОСТО ГРАФЕН
PRO ВРЕМЯ
ПРОСТО ГЕНОМ
ПРОСТО СТАТИСТИКА

Валентин Холмогоров

PRO ВИРУСЫ

Издание четвертое, переработанное и дополненное

Автор идеи и научный редактор серии «Просто…»
Сергей Деменок

Издательство «Страта»
195112, Санкт-Петербург, Заневский пр., 65, корпус 5
Тел.: +7 (812) 320-56-50, 320-69-60
www.strata.spb.ru

Подписано в печать 27.04.2020
Тираж 100 экз.

Мне кажется, компьютерные вирусы можно считать новой формой жизни. И это, пожалуй, кое-что говорит о природе человека, коль скоро единственная форма жизни, которую нам удалось создать к настоящему, несет только разрушения.

Мы сотворили жизнь по образу и подобию своему.

Стивен Хокинг

Серия книг «Просто...» — это наука, не зажатая в тиски формул, скрупулезность цифр и объемы пробирок. Не нужно надевать очки и белый халат, дабы понять, как что работает. Нейронные связи человеческого мозга и смартфоны, напичканные нанотехнологиями, микрочастицы в коллайдере и квантовые биты, фрактальное искусство, стратегия игры и теория Большого взрыва — все это не сложнее яичницы-глазуньи, если только увидеть соль. В серии «Просто...» мы готовим наши научные блюда так вкусно и сервируем так изысканно, что вам наверняка захочется добавки.

Валентин Холмогоров *— автор 38 книг по компьютерным технологиям, 7 художественных произведений и нескольких сотен публикаций в популярных газетах и журналах. В прошлом — заместитель Главного редактора журнала «Магия ПК», редактор «Компьютерной Газеты». В течение 7 лет являлся штатным сотрудником одной из ведущих российских антивирусных компаний. В настоящее время работает редактором легендарного журнала «Хакер».*

9 785907 314122

НАУЧНО-ПОПУЛЯРНОЕ ИЗДАТЕЛЬСТВО «СТРАТА»