ИССЛЕДОВАТЕЛЬСКИЙ ПРОЕКТ

ВРЕДОНОСНЫЕ ПРОГРАММЫ НОВОГО ПОКОЛЕНИЯ: ИСТОЧНИКИ РАЗРАБОТКИ, ЦЕЛИ, ХАРАКТЕР, ОСОБЕННОСТИ И ПОСЛЕДСТВИЯ ИХ ПРИМЕНЕНИЯ

(II этап, итоговый)

Москва – 2014

Аннотация

Данный исследовательский студенческий проект проводится совместно Институтом проблем информационной безопасности МГУ имени М.В.Ломоносова (ИПИБ МГУ) и Институтом информационных наук и технологий безопасности Российского государственного гуманитарного университета (ИИНТБ РГГУ).

Целью проекта является исследование, анализ и систематизация сведений о вредоносных программах и средствах их доставки до атакуемых объектов, появившихся за последнее время (2009-2014 г.г.).

В рамках проекта решались *следующие задачи*.

1) Выявление источника разработки исследуемых вредоносных программ (страны, разработчики, группы разработчиков) и целей разработки данных программ.

2) Анализ средств доставки вредоносных программ до объектов их атаки.

3) Анализ характера, особенностей действия и последействия исследуемых вредоносных программ.

4) Анализ последствий применения исследуемых вредоносных программ для объектов их атаки.

Руководитель проекта: Казарин О.В., *участники проекта*: Бадалян А.А., Вялова Н.В., Гаранина Н.А., Даминова О.А., Кулджанишвили С.Р., Менькин М.И., Никитинская Р.В., Петрова М.А., Торопчанин С.Д.

Принятые сокращения

БВП	–	боевая вредоносная программа
ВП	–	вредоносная программа
КС	–	компьютерная система
ПО	–	программное обеспечение
СДВП	–	средства доставки вредоносных программ

Содержание

Введение

Рассматриваемые в данном проекте вредоносные программы (ВП) – это программы, которые имеют внутренний механизм распространения по локальным и глобальным компьютерным сетям с некоторыми заданными заранее целями. Такими целями могут быть:

- проникновение на удаленные компьютеры с частичным или полным перехватом управления ими;
- запуск своей копии на компьютере;
- (возможно) дальнейшее распространение по всем доступным сетям, как локальным, так и глобальным.

К сетям, по которым передаются данные ВП, в первую очередь, можно отнести файлообменные и торрент-сети, локальные сети, сети обмена между мобильными устройствами, а также электронную почту и различные Интернет-сервисы.

В основном рассматриваемые ВП распространяются в виде файлов. Их прикрепляют в качестве вложений к электронным письмам и сообщениям, либо же различными способами пользователю предлагается пройти по определенной ссылке, закачать и запустить у себя на локальном компьютере некую полезную и бесплатную программу, фотографию и т.д. (вариантов маскировки сетевых ВП существует много). Электронная почта стала практически идеальной средой для распространения ВП.

Существуют также и так называемые «бесфайловые», «пакетные» ВП, которые распространяются в виде сетевых пакетов и проникают на компьютеры при помощи различных брешей и уязвимостей в операционной системе или установленном программном обеспечении (ПО).

Для проникновения на удаленный компьютер используются самые различные методы, начиная от методов социальной инженерии (когда пользователю приходит некое заманчивое письмо со ссылкой или вложенным файлом, призывающее либо открыть данный вложенный файл, либо пройти по

указанной ссылке), или же это проникновение осуществляется с помощью уязвимостей и «back-door» в используемом ПО. Также проникновение возможно при существующих недочетах в планировании и обслуживании локальной сети (примером может служить какой-либо незащищенный локальный диск).

В дополнение к своим основным функциям ВП довольно часто содержат и функции другого вредоносного ПО – вирусов, «троянских» программ, «логических бомб» и т.п.

Одной из целей данного проекта, в том числе, является подтверждение или опровержение некоторых фактов, которые сообщаются различными СМИ, IT-компаниями (в том числе, антивирусными компаниями) о таких вредоносных программах, как Stuxnet, Flame, Gauss, Duqu, Wiper и Icefog.

Уже сейчас по этому направлению исследований можно сделать следующие умозрительные выводы:

- информация, которая сообщается о ВП (та информация, которая непосредственно касается их функциональности) является с большой долей вероятности истинной. Если же какие-то неточности и имеются, то это не от желания обмануть неопытного аналитика, а от ошибок, допущенных при анализе;
- аргументация истинности такова: в наш век информационных технологий, человек, обладающий определенным упорством и желанием, почти всегда может найти нужную ему информацию по данной проблематике;
- образцы данных ВП находятся в открытом доступе в Интернете. Единственным ограничивающим фактором в проверке подлинности аргументов антивирусных компаний может служить отсутствие опыта и необходимых знаний для анализа этих ВП;

- в конце концов, другие антивирусные компании, конкурирующие с данной, могут обличить ее во лжи, тем самым в некоторой степени «подпортив» ее репутацию.

Таким образом, можно с некоторой долей уверенности пользоваться такой общедоступной информацией для анализа упомянутых ВП нового поколения.

1 АНАЛИЗ, КЛАССИФИКАЦИЯ И МЕТОДЫ ВНЕДРЕНИЯ ВРЕДОНОСНЫХ ПРОГРАММ

1.1 Боевые вредоносные программы

В настоящее время за рубежом в рамках создания новейших оборонных технологий и видов оружия активно проводятся работы по созданию так называемых средств нелетального воздействия, одной из разновидностей которого является информационное оружие, представляющее собой совокупность средств поражающего воздействия на информационный ресурс противника. Воздействию информационным оружием могут быть подвержены прежде всего компьютерные и телекоммуникационные системы противника. При этом центральными объектами воздействия являются программное обеспечение, структуры данных, средства вычислительной техники и обработки информации, а также каналы связи, а основной формой информационного оружия являются боевые вредоносные программы (БВП).

Сегодня чаще всего вредоносные программы (ВП) внедряются в компьютерные сети и аппаратные средства через Интернет[1]. Программное обеспечение для наступательных операций нацелено и на отдельные компьютеры, и на сети. Оно решает задачу проникновения в компьютеры и сети противника, используя известные и обнаруживаемые уязвимости, которые содержатся не только в программах и технических средствах, разработанных потенциальным противником, но и в аппаратном и программном обеспечении, используемом по всему миру известных компаний, большинство из которых находится в США.

Интернет является не единственной средой для «доведения» БВП до объекта атаки. Существует большое количество других способов и средств доставки вредоносных программ (СДВП). В том числе, агентурные, удален-

[1] Например, в США этим занимается специальное подразделение Агентства национальной безопасности – ТАО. Эта группа зачастую организует разработку боевых ВП применительно к каждой конкретной цели.

ные аппаратно-технические, в том числе, через способы доставки через различные периферийные устройства атакуемой компьютерной системы (КС), комбинированные способы[2].

Боевые ВП предназначены часто для того, чтобы не просто скрытно присутствовать в программном обеспечении «противника», но и сохраняться в нем даже в случае модернизации оборудования, либо обновления ПО. Боевые ВП призваны решать не только разведывательные задачи по сбору нужной информации, но и выводить атакуемые объекты из строя, либо перехватывать контроль и управление над ними.

1.2 Классификации вредоносных программ

Под вредоносной программой будем понимать внешнюю или внутреннюю по отношению к атакуемой компьютерной системе программу, обладающую определенными деструктивными функциями по отношению к этой системе. К таким функциям можно отнести:

- уничтожение или внесение изменений в функционирование ПО КС, уничтожение или изменение обрабатываемых в ней данных после выполнения некоторого условия или получения некоторого сообщения извне КС («логические бомбы»);
- превышение полномочий пользователя с целью несанкционированного копирования конфиденциальной информации других пользователей КС или создания условий для такого копирования («троянские» программы);
- подмена отдельных функций подсистемы защиты КС или создание «люков» в ней для реализации угроз безопасности информации в КС;

[2] См. следующий раздел.

- перехват паролей пользователей КС с помощью имитации приглашения к его вводу или перехват всего ввода пользователей с клавиатуры;
- перехват потока информации, передаваемой между объектами распределенной КС (мониторы, снифферы);
- сокрытия признаков своего присутствия в программной среде КС;
- реализации самодублирования, ассоциирования себя с другими программами и/или переноса своих фрагментов в иные (не занимаемые изначально указанной программой) области оперативной или внешней памяти;
- разрушения (искажения произвольным образом) кода программ в оперативной памяти КС;
- перемещения (сохранения) фрагментов информации из оперативной памяти в некоторые области оперативной или внешней памяти прямого доступа;
- искажения произвольным образом, блокировки и/или подмены выводимых во внешнюю память или в канал связи массивов информации, образовавшихся в результате работы прикладных программ или уже находящихся во внешней памяти, либо изменения их параметров и др.

Таким образом, можно рассматривать три основные группы деструктивных функций, которые могут выполняться ВП:

- сохранение фрагментов информации, возникающей при работе пользователя, прикладных программ, вводе/выводе данных, во внешней памяти (локальной или удаленной) в сети или выделенном компьютере, в том числе сохранение различных паролей, ключей и кодов доступа, собственно конфиденциальных документов в электронном виде, либо безадресная компрометация фрагментов ценной информации;

- изменение алгоритмов функционирования прикладных программ (т.е. целенаправленное воздействие во внешней или оперативной памяти) – происходит изменение собственно исходных алгоритмов работы программ;
- навязывание некоторого режима работы (например, при уничтожении информации – блокирование записи на диск, при этом информация, естественно, не уничтожается) либо замена записываемой информации данными, навязанными ВП.

1.3 Методы внедрения вредоносных программ

Можно выделить следующие основные методы внедрения (доставки на атакуемый объект) ВП.

Маскировка ВП под «безобидное» программное обеспечение. Данный метод заключается в том, что ВП внедряется в систему под видом новой программы, на первый взгляд абсолютно безобидной. Такая программа может быть внедрена в текстовый или графический редакторы, системную утилиту, компьютерную игру, хранитель экрана и т.д. После внедрения ВП ее присутствие в системе не нужно маскировать – даже если администратор заметит факт появления в системе новой программы, он не придаст этому значения, поскольку эта программа внешне совершенно безобидна.

Маскировка ВП под «безобидный» модуль расширения программной среды. Многие программные среды допускают свое расширение дополнительными программными модулями. Например, для операционных систем семейства Microsoft Windows модулями расширения могут выступать динамически подгружаемые библиотеки (DLL) и драйверы устройств. В таких модулях расширения может содержаться ВП, которая может потенциально внедрена в систему. Данный метод фактически является частным случаем предыдущего метода и отличается от него только тем, что ВП представляет собой не прикладную программу, а модуль расширения программной среды.

Подмена ВП одного или нескольких программных модулей атакуемой среды. Данный метод внедрения в систему ВП заключается в том, что в атакуемой программной среде выбирается один или несколько программных модулей, подмена которых фрагментами ВП позволяет оказывать на среду требуемые негативные воздействия. Такая программа должна полностью реализовывать все функции подменяемых программных модулей.

Основная проблема, возникающая при практической реализации данного метода, заключается в том, что программист, разрабатывающий ВП, никогда не может быть уверен, что созданная им ВП точно реализует все функции подменяемого программного модуля. Если подменяемый модуль достаточно велик по объему или недостаточно подробно документирован, точно запрограммировать все его функции практически невозможно. Поэтому описываемый метод целесообразно применять только для тех программных модулей атакуемой среды, для которых доступна полная или почти полная документация. Оптимальной является ситуация, когда доступен исходный текст подменяемого модуля.

Прямое ассоциирование. Данный метод внедрения в систему ВП заключается в ассоциировании ВП с исполняемыми файлами одной или нескольких легальных программ системы. Сложность задачи прямого ассоциирования ВП с программой атакуемой среды существенно зависит от того, является атакуемая среда однозадачной или многозадачной, однопользовательской или многопользовательской. Для однозадачных однопользовательских систем эта задача решается достаточно просто. В то же время при внедрении ВП в многозадачную или многопользовательскую программную среду прямое ассоциирование ВП с легальным программным обеспечением является весьма нетривиальной задачей.

Косвенное ассоциирование. Косвенное ассоциирование ВП с программным модулем атакуемой среды заключается в ассоциировании ВП с кодом программного модуля, загруженным в оперативную память. При косвенном

ассоциировании исполняемый файл программного модуля остается неизменным, что затрудняет выявление ВП.

Для того чтобы косвенное ассоциирование стало возможным, необходимо, чтобы инсталлирующая часть ВП уже присутствовала в системе. Другими словами, ВП, внедряемая в систему с помощью косвенного ассоциирования, должна быть составной.

Исследуемые в данном проекте БВП имеют сложный и комплексный по характеру воздействия и методом внедрения на объект атаки пул деструктивных функций, которые подробно рассматриваются в разделах 2 и 3.

2 ИССЛЕДУЕМЫЕ ВРЕДОНОСНЫЕ ПРОГРАММЫ

2.1 Вредоносная программа Stuxnet

2.1.1 Общие сведения о ВП Stuxnet

Специалисты белорусской антивирусной компании «ВирусБлокада» 9 июля 2010 года обнаружили вредоносную программу, которая была названа Stuxnet. У антивирусных компаний нет единого мнения о точной дате возникновения Stuxnet. По некоторым данным, распространение вируса происходило еще с января 2009 года. Речь идет не просто об обычном вирусе, а о чрезвычайно высокотехнологичном вредоносном ПО во всех его проявлениях. Stuxnet фактически является первой в истории ВП, переступившим через границу киберпространства в реальный физический мир. Это первая ВП, которая нацелена не на широкий спектр компьютеров, а на узкий круг корпоративных систем, способная «портить» не только данные и программный код, но и вполне реальные устройства и промышленные установки. Ее основной задачей является внедрение вредоносного функционала в промышленные информационные системы класса SCADA. Появление этой ВП выявило очередные уязвимости в операционных системах Microsoft. Раньше мало кто задумывался о безопасности промышленных систем, но с появлением Stuxnet взоры многих специалистов по информационной безопасности были направлены в эту абсолютно новую для них область.

К слову, некоторые компании в сфере информационной безопасности предупреждали об этом еще несколько лет назад. Данное явление можно объяснить фактом изолирования промышленных сетей как от сетей общего пользования, так и от внутренних сетей предприятия. В них применяется очень специфическое оборудование и ПО, все процессы четко упорядочены. Являясь профессиональными программистами, инженерами и специалистами, зная специфику работы с промышленными контроллерами и другим периферийным оборудованием, создателям Stuxnet удалось без труда обойти эту, казалось бы, самую надежную физическую защиту.

Создатели ВП Stuxnet – группа высококвалифицированных специалистов, которые хорошо ориентируются в слабых местах современных систем информационной безопасности. Эта ВП сделана таким образом, чтобы оставаться незамеченным как можно дольше.

2.1.2 Особенности работы ВП Stuxnet

Подробный анализ структуры кода Stuxnet (см. рис.1) свидетельствует о том, что эта вредоносная программа является самым изощренным и функционально насыщенным средством среди всех ранее известных компьютерных вирусов, троянов и червей. Чтобы представить себе все «богатство» функций и средств, которое имеет Stuxnet, приведем краткий список его основных особенностей:

- самокопирование посредством сменных механизмов, использующих уязвимость, позволяющую автовыполнение: Microsoft Windows Shortcut „LNK/PIF‟ Files Automatic File Execution Vulnerability (BID 41732);
- распространение по сети через уязвимость в буфере печати Windows: Microsoft Windows Print Spooler Service Remote Code Execution Vulnerability (BID 43073);
- распространение через SMB, используя Microsoft Windows Server Service RPC Handling Remote Code Execution Vulnerability (BID 31874).
- автокопирование и автовыполнение на удаленных компьютерах через разделение сетей;
- автокопирование и автовыполнение на удаленных компьютерах через сервер базы данных WinCC;
- автокопирование в проекты Step 7 таким образом, что она автоматически выполнялась, когда проект Step 7 загружен.

- самообновление через механизм соединения точка-к-точке в пределах ЛВС;
- использование в общей сложности всех четырех неисправленных уязвимостей Microsoft, две из которых выше упомянуты – для возможности самокопирования, и двух других – для повышения скрытности уязвимостей, которые должны все же быть обнаружены;
- связь с сервером контроля и управления, который позволяет злоумышленникам загружать и выполнять код, включая обновленные версии;
- содержит Windows rootkit, который скрывает двоичный код ВП;
- содержит варианты средств защиты;
- делает «слепок» определенной промышленной системы управления и изменяет код в ПЛК Siemens, чтобы потенциально осуществить саботаж системы;
- скрывает измененный код на ПЛК, что, по существу, представляет собой rootkit для ПЛК.

2.1.3 Принципы работы ВП Stuxnet

Основу ВП Stuxnet составляет большой .dll файл, который включает в себя множество разных экспортов и ресурсов. Кроме большого .dll файла Stuxnet содержит два шифрованных конфигурационных блока. Засылающий компонент (dropper) Stuxnet является программой-оболочкой, которая включает в себя все раньше указанные компоненты, хранимые внутри себя в разделе stub. Этот раздел stub является интегрирующим в работе Stuxnet. Когда данная программа выполняется, оболочка извлекает .dll файл из раздела stub, размещает его в памяти как модуль и вызывает один из экспортов. Указатель на исходный раздел stub передается этому экспорту как параметр. Этот экспорт, в свою очередь, извлекает .dll файл из раздела stub, который был пере-

дан как параметр, размещает его в памяти и вызывает другой экспорт из размещенного в памяти .dll файла. Указатель на исходный раздел stub снова передается как параметр. Это происходит постоянно, пока исполняется программа, так что исходный раздел stub постоянно передается различным процессам и функциям в качестве параметра, доходя до главной поражающей части.

Таким образом, каждый уровень данной программы всегда имеет доступ к главной .dll и конфигурационным блокам. В дополнение к загрузке .dll файла в память и вызова экспорта напрямую, Stuxnet использует также другой способ для вызова экспортов из главного .dll файла. Этот способ связан с внедрением исполняемого шаблона в другой процесс.

2.1.4 Способы распространения ВП Stuxnet

Программа Stuxnet устанавливает в систему два драйвера, один из которых является драйвером-фильтром файловой системы, скрывающим наличие компонентов этой ВП на съемном носителе. Второй драйвер используется для внедрения зашифрованной динамической библиотеки в системные процессы и содержит в себе специализированное ПО для выполнения основной задачи. Драйверы, которые ВП устанавливает в систему, снабжены электронными подписями, «украденными» у производителей легального программного обеспечения. Известно об использовании подписей, принадлежащих таким компаниям, как Realtek Semiconductor Corp. и JMicron Technology Corp. Злоумышленники используют электронную подпись для незаметной установки драйверов руткита в целевую систему. В системах безопасности многих производителей файлы, подписанные известными фирмами, заведомо считаются безопасными, и наличие подписи дает возможность свободно, не выдавая себя, производить действия в системе. Кроме того, Stuxnet располагает механизмами контроля количества заражений, самоликвидации и дистанционного управления.

Кроме распространения посредством внешних носителей Stuxnet успешно заражает компьютеры посредством соединения через локальную сеть. То есть, оказавшись на компьютере вне промышленной сети, он анализирует все активные сетевые соединения и «пробивается» к промышленной сети всеми возможными способами. После внедрения в систему вредоносное ПО ищет в ней присутствие SCADA-системы фирмы Siemens. Причем им атакуются только системы SCADA WinCC/PCS7. Данных о заражении другой SCADA-системы от Siemens – Desigo Insight, которая широко используется для автоматизации зданий и жилых комплексов, аэропортов и т.д., нет. Это говорит о «заточенности» Stuxnet на крупные промышленные стратегические объекты.

2.1.5 Отличительные особенности ВП Stuxnet

Можно выделить следующие особенности написания и функционирования ВП Stuxnet:

- программа Stuxnet содержит несколько модулей, написанных с использованием нескольких сред разработки и языков программирования;
- для обхода механизмов антивирусной защиты некоторые модули (драйверы) ВП имели электронную подпись, сделанную с использованием сертификатов компаний Realtek и JMicron, предположительно, «украденных»;
- несколько способов распространения – посредством USB-Flash накопителей и по сети. В версии 2009 года использовался широко применяемый способ запуска через autorun.inf (который, как правило, отключают из соображений безопасности), в версии 2010 года он был заменен на более эффективный – использование уязвимости обработки ярлыков MS10-046 (zero-day на тот момент). Для распространения через сеть использовались уязвимости MS08-067 (ранее использовалась в 2009 году ВП Kido,

что привело к массовым заражениям) и MS10-061 (zero-day на тот момент);

- для обеспечения работы производилось повышение привилегий до уровня администратора системы при помощи использования двух локальных уязвимостей (zero-day на тот момент) MS10-073 (Windows 2000 и XP) и MS10-092 (Windows Vista, включая версию x64), таким образом, был предусмотрен нормальный запуск ВП из-под ограниченных учетных записей;
- программа Stuxnet организует свою собственную peer-to-peer (P2P) сеть для синхронизации и обновления своих копий;
- присутствует функционал, позволяющий пересылать на удаленные сервера управления информацию, найденную на компьютере;
- необычная ненужная нагрузка ведет за собой нарушение нормальной работы системы автоматизации SIMATIC, производимой компанией Siemens, используемой, как правило, в различных промышленных системах управления производственными процессами.

2.1.6 Возможные целим атак ВП Stuxnet

Эксперты полагают, что Stuxnet мог быть разработан для применения «против» АЭС в Бушере (Иран). В качестве вероятных разработчиков может выступать Израиль и США. В основу версии легли следующие факты:

- Иран является одной из наиболее пострадавших от Stuxnet стран. Судя по отчетам о заражениях – примерно в мае-июне 2010 года Иран был лидером по числу заражений;
- Бушерская АЭС является одной из наиболее важных военных целей в Иране для США и Израиля;
- АЭС начали строить еще в 1970-е. В строительстве, принимала участие компания Siemens. В 1979 году, из-за революции в Иране, Siemens прекратила работы в этой стране. Впоследствии Siemens

«вернулась» в Иран, ставшей одной из крупнейших в этой стране поставщиков оборудования. В январе 2010 года компания Siemens снова заявила о прекращении сотрудничества с Ираном. Однако, летом она была уличена в поставке комплектующих в Бушер. Используется ли на АЭС программное обеспечение Siemens для управления процессами – официальная информация отсутствует. На одном из размещенных в сети Интернет снимков экрана компьютера, сделанного якобы внутри АЭС, можно видеть систему управления WinCC компании Siemens;

- Израиль является одной из наиболее заинтересованных в нарушении функционирования Бушерской АЭС стран. Израиль подозревает Иран в том, что на этой станции, под видом ядерного топлива, будут изготовляться запасы для производства собственного ядерного оружия, которое, наиболее вероятно, может быть использовано против Израиля;
- Израиль является одной из стран, которая обладает высококвалифицированными специалистами в области информационных технологий. Они способны использовать их как для атак, так и для шпионажа.

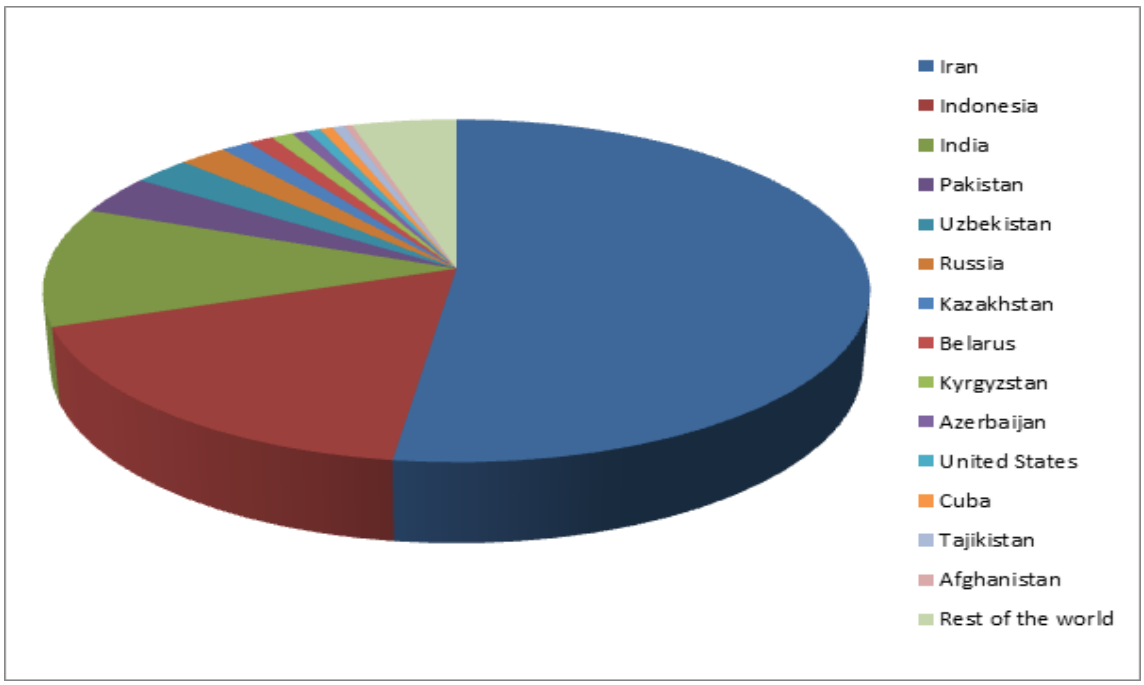


Рис. 2. Список стран, которые подверглись атаке Stuxnet

Еще одна из версий о цели атаки – производство по обогащению урана в г. Натанзе (Иран). Эту версию косвенно подтверждают следующие факты:

- производство по обогащению урана в Натанзе – это мощно укрепленный и спрятанный глубоко под землей объект. По свидетельствам экспертов она представляет собой намного большие риски с точки зрения производства ядерного оружия, нежели Бушерская АЭС;
- В июле 2009 г. один из источников, связанных с ядерной программой Ирана, конфиденциально сообщил о серьезной аварии, произошедшей незадолго до этого в Натанзе. Позднее, иранскими СМИ и британской BBC, было сообщено, что Голамреза Агазаде, глава Иранской организации по атомной энергии (IAEO), ушел в отставку. В это же время, согласно официальным данным, предоставляемым IAEO в контролирующие структуры, значительно упало количество

функционирующих центрифуг в Натанзе, что могло произойти из-за воздействия Stuxnet.

2.1.7 Интересные факты

Эксперты по компьютерной безопасности, занимающиеся расшифровкой кода Stuxnet, обнаружили, что в коде содержится намек на библейскую царицу – героиню Есфирь (Эстер) – спасительницу еврейского народа от козней персидского злодея Амана.

Расшифровка кода позволила предположить, что проект по созданию Stuxnet носил название «Мирт». Один из модулей кода назывался «Гуава» – плод растения из семейства миртовых. Германский эксперт Ральф Лангер первым заметил, что «мирт» на иврите – «адасса», и точно так же звучало еврейское имя Эстер (именно в честь нее названа знаменитая женская сионистская организация США).

В США в июне 2012 года вышла книга под названием «Confront and Conceal: Obama's Secret Wars and Surprising Use of American Power» («Конфронтация и сокрытие: Тайные Войны Обамы и удивительное использование американской мощи»), автором которой является журналист The New York Times Дэвид Сэнгер. В ней содержалась информация, что Stuxnet был разработан именно в США с участием израильских специалистов и целью была нейтрализация ядерной программы Ирана. Автор утверждает, что Stuxnet разрабатывался ещё в период президентства Джорджа Буша-младшего. Проект назывался «Olympic Games». Сначала это была программа по распространению шпионского ПО, благодаря которому удалось получить информацию об оборудовании иранского центра по обогащению урана в Натанзе. Уже после этого был разработан функционал, который воздействовал на программное обеспечение, которое управляет центрифугами очистки урана.

Ещё в прошлом году Дэвид Сэнгер и двое его коллег публиковали в New York Times статью, в которой говорилось, что Stuxnet – действительно

дело рук американских и израильских спецслужб и что испытывали его в секретном израильском центре «Димона» в пустыне Негев. Официально Израиль отказывается признавать существование собственной ядерной программы, однако авторы статьи ссылаются на неких осведомленных экспертов в разведывательной и военной областях, которые подтверждают, что в Димоне стоят центрифуги, практически идентичные тем, что стояли в Натанзе. Способность Stuxnet выводить их из строя была опробована и на них. По данным The Wall Street Journal, ФБР проводит расследование утечки информации, в результате которой стало известно о причастности правительства страны к кибератакам на ядерные объекты Ирана. Многие эксперты относятся к этой информации скептически. Они считают ее очередной «уткой», появившейся накануне президентских выборов в США.

В начале 2013 года корпорация Symantec объявила об обнаружении более ранней версии Stuxnet. Версия с номером 0.5 была активна с 2007 по 2009 гг., а применялся эта ВП могла еще с ноября 2005 г., когда был зарегистрирован ее первый C&C-сервер. В отличие от более поздней версии 1.001, в версии 0.5 использовался совершенно другой механизм атаки. Эксперты Symantec также обнаружили, что платформа, на которой была создана более ранняя версия та же, что и у ВП Flame.

Полученные экспертами Symantec в ходе исследования данные показали, что Stuxnet версии 0.5 был специально создан для атаки на промышленные установки Simatic 400 Station и Simatic H-Station (центрифуги для обогащения урана) иранского завода, расположенного около города Натанз. Действия ВП были построены через выведение из строя центрифуг путем создания пятикратных перепадов давления в центрифугах, которые были объединены в каскады. ВП управляла клапанами управления подачи и сброса газообразного гексафторида урана. При этом она маскировала свои действия, показывая оператору заранее сохраненный снимок монитора нормального состояния центрифуги.

2.1.8 Выводы и заключение

На основе вышесказанного, можно выделить следующие выводы:

- Stuxnet представляет собой тщательно спроектированное вредоносное ПО, которое разрабатывалось командой специалистов-профессионалов в разных областях;
- Stuxnet был внедрен в закрытую систему, не имеющую прямого подключения к общедоступным сетям, так как не было выявлено фактов его распространения посредством сети Интернет, только через USB-Flash и посредством локальной сети;
- функционал нарушения нормальной работы системы управления производственными процессами Siemens WinCC подразумевает, что разработчики Stuxnet для тестирования имели программно-аппаратную систему, идентичную той, на которую планировалась атака. Кроме того, они ориентировались на конкретную цель (использование данных от завербованного персонала внутри организации);
- разработка такого масштаба предполагает значительное финансирование – оплата труда группы программистов, организация кражи цифровых сертификатов, покупка или разработка 4-zero-day-уязвимостей, доступ к развернутой системе Siemens WinCC.

Все эти косвенные признаки могут указывать на участие в разработке Stuxnet силовых ведомств или спецслужб каких-либо государств. Главная функция данной ВП – распространение и автономная работа в замкнутой системе с последующим саботажем работы системы управления производ-

ственными процессами, как главной функции ВП, которая не свойственна «традиционным» киберпреступникам[3].

Именно по этим причинам Stuxnet можно считать определенной разновидностью кибероружия. Stuxnet является первой ВП, которая использовала четыре уязвимости 0-го дня, компрометировала два цифровых сертификата, первым внедрила код в системы управления производственным процессом и скрыла этот код от оператора, – поэтому ее можно считать самой значительной вехой в истории вредоносного кода. «Затмив» огромное большинство существующих деструктивных атак, Stuxnet может по праву считаться ВП, с которой началось новое поколение атак вредоносного кода на инфраструктуру реального мира.

2.2 Вредоносная программа Wiper[4]

История Wiper такова – в 2012 году на серверах десятка организаций Ирана было удалено большое количество баз данных. «Однако не было найдено ни одного образца вредоносной программы, использованной в этих атаках, что многих заставило усомниться в точности сведений, содержащихся в сообщениях СМИ. В связи с этими инцидентами Международный союз электросвязи обратился к компании «Лаборатория Касперского» с просьбой провести свое расследование и определить потенциальные деструктивные последствия активности этого нового вредоносного ПО. Аналитики «Лаборатории Касперского» проанализировали несколько жестких дисков, атакованных Wiper, и подготовили отчет о проделанной работе.

Цели анализа в настоящем проекте – выявление источников разработки ВП Wiper, целей кибератаки с ее использованием, последствия применения,

---

[3] Здесь под «традиционными» киберпреступниками будем понимать людей, которые обычно преследуют цели материальной выгоды и, как правило, используют ВП, созданные программистами-одиночками.

[4] В работе используются материалы отчета «Лаборатории Касперского» по результатам исследования ВП Wiper.

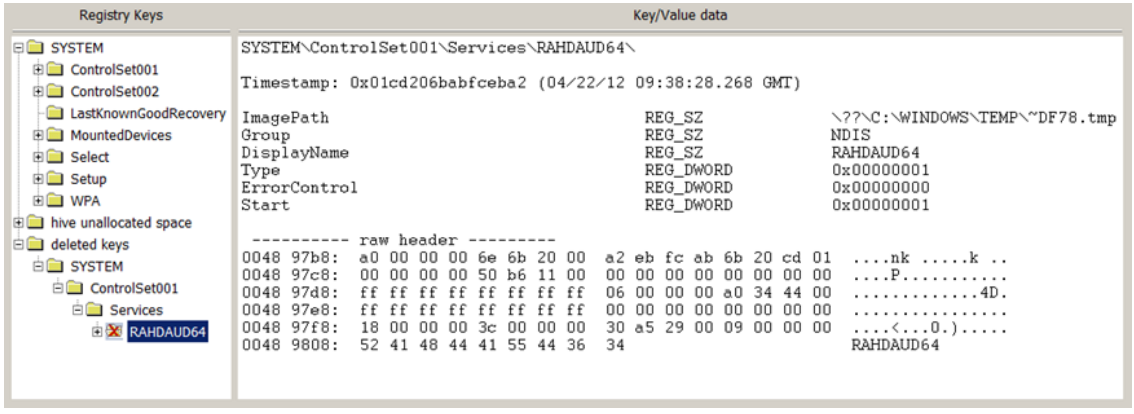
нахождение криптографических методов в функционале программы для выполнения своих задач (например, для сокрытия тела программы).

В ходе исследования было установлено, что эта ВП действительно существовала. «Мы можем с уверенностью утверждать, что инциденты действительно имели место и что ВП, использованная в этих атаках, существовала в апреле 2012 года…».

Помимо ВП Wiper, примерно в это же время были обнаружены такие программы (или комплекс программ), получивших название ВП Flame и Gauss.

Цель ВП Wiper – безвозвратное удаление данных с жесткого диска, и, в первую очередь, файлов, которые можно было бы потом использовать для анализа работы программы. «Поэтому в каждом из случаев, которые мы проанализировали, после активации Wiper от вредоносной программы не оставалось почти никаких следов. Здесь важно подчеркнуть, что именно «почти никаких», потому что кое-какие следы все же остались, и они позволили нам лучше понять, как осуществлялись эти атаки».

Из оставшихся данных на одном из жестких дисков сотрудниками «Лаборатории Касперского» удалось восстановить кусок реестра. «[Он] не содержал вредоносных драйверов или ключей автозапуска. Однако мы решили проверить свободную часть диска на наличие удаленных ключей реестра». Вот что ими было обнаружено:

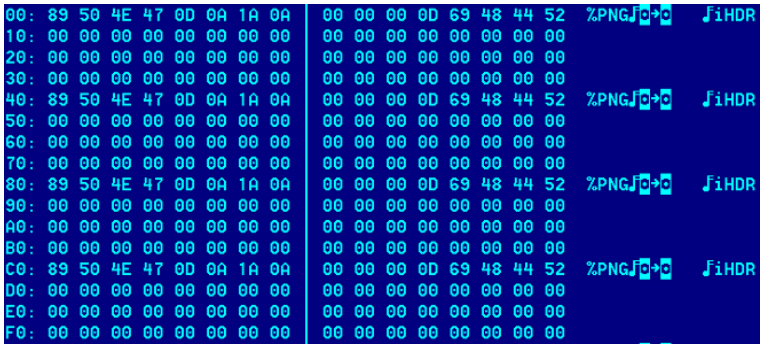


«Интересно, что 22 апреля, непосредственно перед тем, как система отказала, был создан, а затем удален один конкретный ключ реестра. Этот ключ ссылался на службу под названием RAHDAUD64[5]. Он указывал на файл ~DF78.tmp в папке C:\WINDOWS\TEMP на диске».

Такой же формат временных файлов использовал и в ВП Duqu. «Мы попытались восстановить файл ~DF78.tmp, но обнаружили, что физическое пространство на диске, где он ранее находился, заполнено мусором».

«Мы обнаружили, что аналогичная схема была использована для «затирания» данных еще в нескольких из проанализированных нами систем: служба под названием RAHDAUD64, удаленная непосредственно перед уничтожением содержимого диска, причем область, занимаемая ее файлом, была заполнена мусором. В этих системах ключ RAHDAUD64 указывал на файлы с различными именами, такими как ~DF11.tmp и ~DF3C.tmp. Не исключено, что имена файлов генерировались случайным образом».

Еще одна обнаруженная особенность процесса затирания содержимого диска – это шаблон, использованный для перезаписи файлов на диске мусором:



Данный шаблон повторяется снова и снова на всех перезаписанных файлах. В том случае, если длина файла превышала какое-то критическое значение, то перезаписи подвергались только заголовок файла и небольшая

[5] С иврита переводится примерно как «Злой Давид».

часть кода. Связано это с расчетом на достаточно быстрое удаление максимально большого количества данных.

«Основываясь на известном шаблоне, которым были перезаписаны файлы, мы собрали в Kaspersky Security Network (KSN) статистику о том, какие файлы подверглись уничтожению».

На основе полученных результатов удалось получить схему работы Wiper:

1) Поиск и перезапись файлов по их расширению[6].

«Особое внимание удалялось уничтожению PNF-файлов. Стоит отметить, что именно этот тип файлов использовался ВП Duqu и Stuxnet для хранения в зашифрованном виде своего основного кода».

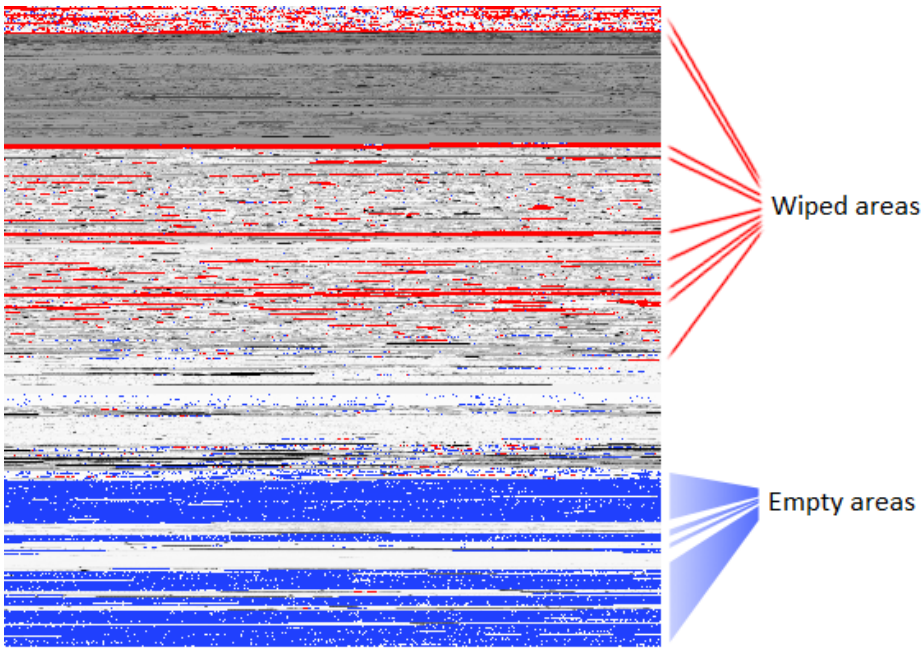
2) Поиск и перезапись всех файлов в определенных каталогах (например, в Documents and Settings, Windows, Program Files) и на всех доступных подключенных USB-дисках.
3) Перезапись секторов диска.

Приведенный выше алгоритм позволяет наиболее эффективно перезаписать требуемые данные без возможности их восстановления.

«Рассмотрим для примера следующий диск, содержимое которого было перезаписано Wiper. Мы использовали статистическое представление (энтропия Шеннона с блоками размером 256 Кб) для отображения энтропии на диске. Более светлые участки имеют более высокую энтропию, более темные – более низкую. Красные области имеют очень высокую энтропию (т.е. в них записаны данные с высокой степенью случайности)».

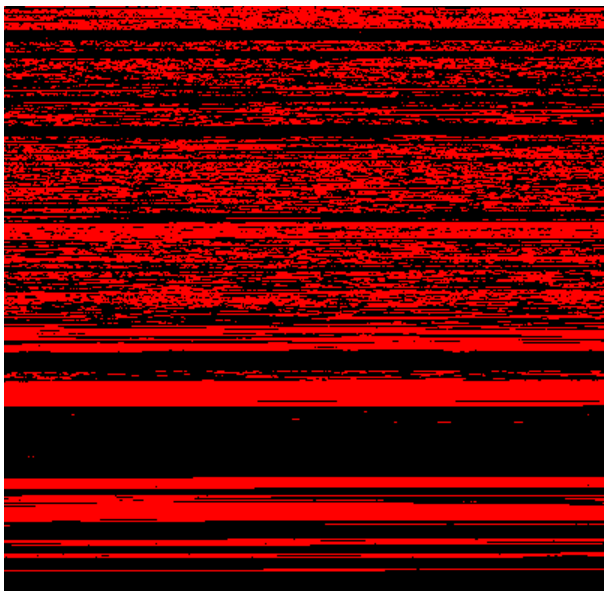
---

[6] Список расширений: accdb, acl, acm, amr, apln, asp, avi, ax, bak, bin, bmp, cdx, cfg, chk, com, cpl, cpx, dat, db, dbf, dbx, dll, dmp, doc, docx, dot, drv, dwg, eml, exe, ext, fdb, gif, H, hlp, hpi, htm, html, hxx, ico, inc, ini, jar, jpg, js, json, lnk, log, lst, m4a, mid, nls, one, pdf, pip, pnf, png, pps, ppt, pptx, pro, psd, rar, rdf, resources, rom, rpt, rsp, sam, scp, scr, sdb, sig, sql, sqlite, theme, tif, tiff, tlb, tmp, tsp, txt, vbs, wab, wav, wma, wmdb, wmv, xdr, xls, xlsx, xml, xsd, zip.



«Как видите, Wiper успешно уничтожил данные на большей части диска. На верхней части карты видна заполненная красным цветом полоса, обозначающая тщательно очищенную область. Четкий паттерн не вырисовывается, но значительная часть диска оказалась заполнена бесполезными данными. Очевидно, что процесс перезаписи сначала был сосредоточен на начале диска, затем он перешел на середину диска, после чего, наконец, произошел отказ системы».

«Другое представление можно получить, отобразив секторы, заполненные известным шаблоном %PNG / iHDR». Здесь красным цветом отмечены перезаписанные блоки секторов:



Исходя из выше приведенных данных можно заметить, что более 75% всего диска «затронуто» Wiper, причем почти все данные безвозвратно утрачены.

«В некоторых случаях Wiper «давал осечку» – например, мы столкнулись с одной 64-битной системой, в которой Wiper не сработал. В этом случае мы обнаружили два файла в папке %TEMP%, которые были перезаписаны знакомым паттерном PNG/iHDR, но диск при этом остался цел». Пример:

```
04/22/2012  09:57 AM            28,672 ~DF7EC2.tmp
04/22/2012  09:57 AM            28,672 ~DF7ED3.tmp
```

«Мы предполагаем, что именно эти два файла из тысяч, находившихся в папке %TEMP%, были уничтожены потому, что они содержали данные, которые были важны для атаки Wiper. В другой проанализированной нами системе кроме этих файлов размером примерно по 20 Кб мы обнаружили два файла размером 512 байтов с именами ~DF820A.tmp и ~DF9FAF.tmp. Эти файлы также были перезаписаны без возможности восстановления».

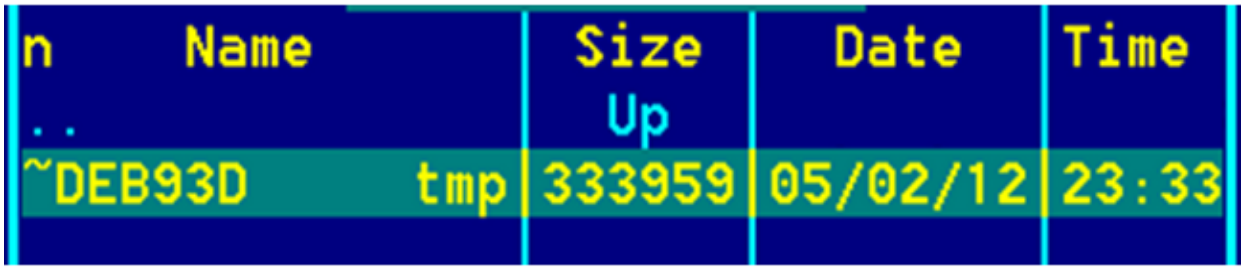
«Интересно, что в некоторых системах мы обнаружили, что все файлы с расширением PNF в подпапке INF папки Windows были перезаписаны с бо-

лее высоким приоритетом, чем другие файлы. Это еще один признак родства Wiper с Duqu и Stuxnet, которые хранили свое основное тело в зашифрованных файлах .PNF».

На основе полученных результатов можно сделать вывод, что одной из приоритетных целей создателей Wiper была невозможность самого обнаружения этой ВП, а потом уже удаление данных с жестких дисков. В связи с чем в первую очередь перезаписывались компоненты ВП, а потом уже другие файлы.

В ходе анализа Wiper были получены дополнительные сведения и о других вредоносных программах в т.ч. о ВП Flame.

«[Вероятно], что Wiper использует такие имена файлов, как ~DF*.tmp или ~DE*.tmp в папке TEMP, поэтому стали искать подобные имена с помощью KSN. В процессе поиска мы заметили, что файл под названием ~DEB93D.tmp встречается на необычно большом числе машин в регионе Ближнего Востока»:



Имя файла являлось индикатором принадлежности к той же платформе, на которой создавались ВП Duqu и Stuxnet.

«Совершенно случайно мы обратили внимание на то, что этот файл начинался с байтов 6F C8. Они также присутствовали в начале PNF-файла, содержащего зашифрованное основное тело Duqu, которое загружалось драйвером, скомпилированным 3 ноября 2010 г. Если бы не это, возможно, мы бы и не обратили внимания на файл ~DEB93D.tmp, поскольку его содержимое было, на первый взгляд, мусорным».

Несмотря на то, что файл был зашифрованным[7] сотрудникам «Лаборатории Касперского» удалось его расшифровать, и в результате получить нечто, напоминающее логи сниффера.

«Это заставило нас искать дальше, и мы обнаружили другие файлы, измененные в тот же день, с такими именами, как mssecmgr.ocx, EF_trace.log и to961.tmp. Остальное, как говорится, уже история: именно так мы обнаружили Flame».

Несмотря на отсутствие прямых доказательств существования исполняемого кода подобного типа «…нет никакого сомнения в том, что существовала вредоносная программа, известная как Wiper, которая атаковала компьютерные системы в Иране (и, возможно, в других частях света) до конца апреля 2012 года».

Также остается неизвестным конкретный разработчик Wiper. «Вредоносная программа была написана так профессионально, что, будучи активирована, она не оставляла после себя никаких данных. Поэтому, несмотря на то, что мы видели следы заражения, сама ВП остается неизвестной: мы не знаем ни о каких других инцидентах с перезаписью содержимого диска, произошедших по той же схеме, что при заражении Wiper; не зарегистрировано также обнаружения этого вредоносного ПО компонентами проактивной защиты, входящими в состав наших защитных решений».

2.3 Вредоносная программа Flame

Программа Flame – вредоносная программа, поражающая компьютеры под управлением операционной системы Microsoft Windows версий XP, Vista, 7.

Его обнаружил Роэль Шувенберг, старший научный сотрудник по компьютерной безопасности Лаборатории Касперского во время исследования вируса Wiper, атаковавшего компьютеры в Иране, о чем было объявлено

---

[7] Шифр оказался недостаточно надежным; скорее всего, это обыкновенный шифр гаммирования с длиной гаммы 4096 байт.

28 мая 2012 года. Наиболее пострадавшими странами являются Иран, Израиль, Судан, Сирия, Ливан, Саудовская Аравия и Египет.

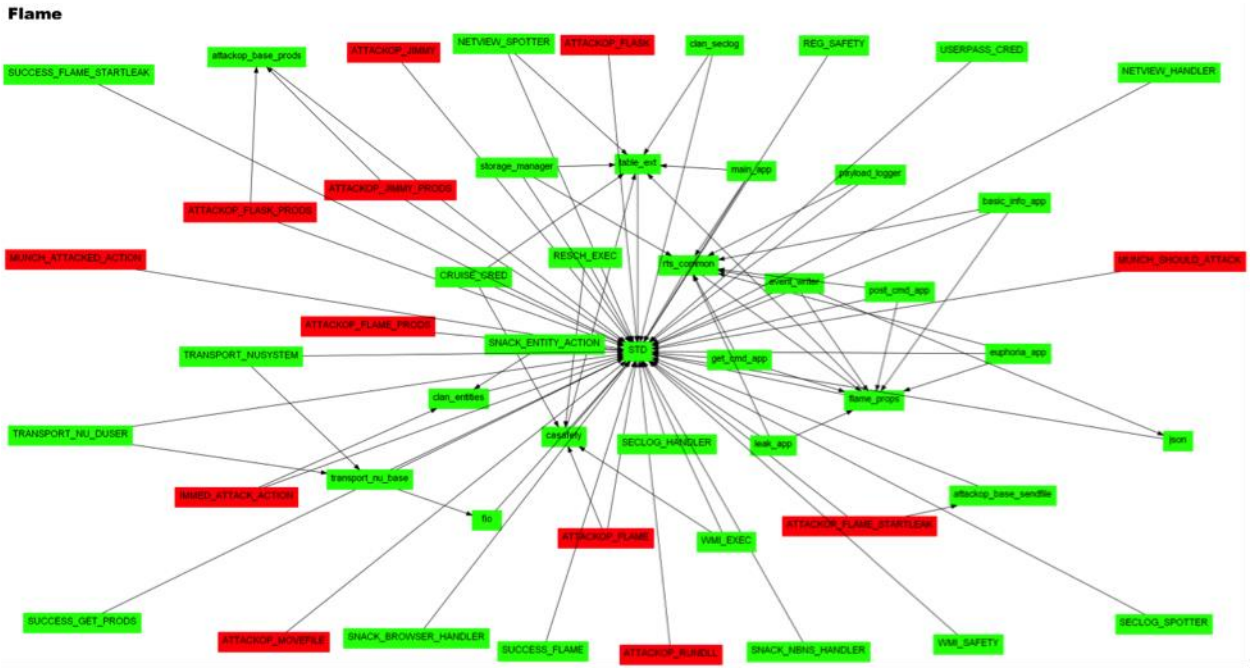
Так как роль «первооткрывателя» данной ВП принадлежит Лаборатории Касперского, то при анализе «опираться» будем на информацию, фигурирующую в блогах данной компании.

ВП Flame представляет собой весьма хитрый набор инструментов для проведения атак, значительно превосходящий по сложности ВП Duqu. Это троянская программа – «backdoor, имеющая также черты, свойственные червям и позволяющие ей распространяться по локальной сети и через съемные носители при получении соответствующего «приказа» от ее хозяина.

Исходная точка входа Flame неизвестна – возможно, что первоначальное заражение происходит путем целевых атак, однако найти исходный вектор атаки пока не удалось. Подозревается, что используется уязвимость MS10-033, однако в данный момент подтвердить это не представляется возможным.

После заражения системы ВП Flame приступает к выполнению сложного набора операций, в том числе к анализу сетевого трафика, созданию снимков экрана, аудиозаписи разговоров, перехвату клавиатурных нажатий и т.д. Все эти данные доступны операторам через командные серверы Flame.

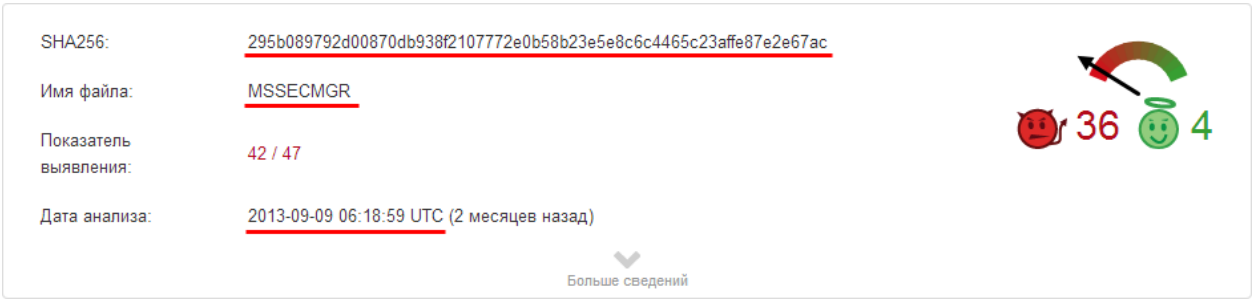
В дальнейшем операторы могут принять решение о загрузке на зараженные компьютеры дополнительных модулей, расширяющих функционал ВП Flame. Всего имеется около 20 модулей, назначение большинства которых в данный момент изучается.



Исходный вектор атаки остался неизвестным, таким образом, заражение системы воспроизвести не удалось. Скорее всего, остается только статический анализ найденных в сети Интернет-образцов. Был найден сайт – vx-archiv.at. На данном сайте хранятся образцы некоторых ВП. Скачанный файл представляет собой архив, защищенный паролем, внутри которого содержится образец ВП.

«Для того чтобы убедиться, что нами был скачан именно – Worm.Win32.Flame.a – загружаем скачанный файл на сайт – www.virustotal.com. VirusTotal – бесплатная служба, осуществляющая анализ подозрительных файлов и ссылок (URL) на предмет выявления вирусов, червей, троянов и всевозможных вредоносных программ».

Иными словами, после загрузки указанного файла на данный сайт – файл проверяется всеми известными антивирусами на предмет его вредоносности.



Обращается внимание на то, что этот файл уже сканировался на данном ресурсе 2013-09-09. Также обращается внимание на имя файла – MSSECMGR и его хэш-сумму.

Ниже приведены показания антивирусов при сканировании данного файла:

ESET-NOD32	Win32/Flamer.A	20130908
F-Prot	W32/Flamer.A	20130909
F-Secure	Trojan.Generic.8473796	20130909
Fortinet	W32/Flame.A!worm	20130909
GData	Trojan.Generic.8473796	20130909
Ikarus	Worm.Win32.Flame	20130909
Jiangmin	Worm/Flame.g	20130903
K7AntiVirus	EmailWorm	20130906
K7GW	EmailWorm	20130906
Kaspersky	Worm.Win32.Flame.a	20130909

Основной модуль Flame – это DLL-файл под названием mssecmgr.ocx. Были обнаружили две модификации этого модуля. На большинстве зараженных машин содержалась «большая» версия – 6 Мбайт, которая содержит в себе и развертывает дополнительные модули. «Малая» версия составляет всего 900 Кбайт и не содержит дополнительных модулей. После установки малый модуль подсоединяется к одному из C&C-серверов и пытается загрузить оттуда и установить оставшиеся компоненты.

Mssecmgr может по-разному называться на каждой конкретной зараженной машине в зависимости от метода заражения и текущего состояния ВП (установка, распространение, обновление). Например, wavesup3.drv, ~zff042.ocx, msdclr64.ocx, etc.

Перед нами основной модуль ВП Flame. Егл размер:

Идентификация файла – «большая» версия данной ВП.

Дата создания ВП Flame? Авторы Flame специально изменили даты создания файлов таким образом, чтобы исследователи не смогли узнать, когда ВП была создана на самом деле. Файлы датированы годами 1992, 1994, 1995 и т.д. – очевидно, что эти даты не имеют отношения к действительности.

Проверка электронной подписи данного файла и хэш-суммы с помощью утилиты SIGCheck из пакета SYSINTERNALS:

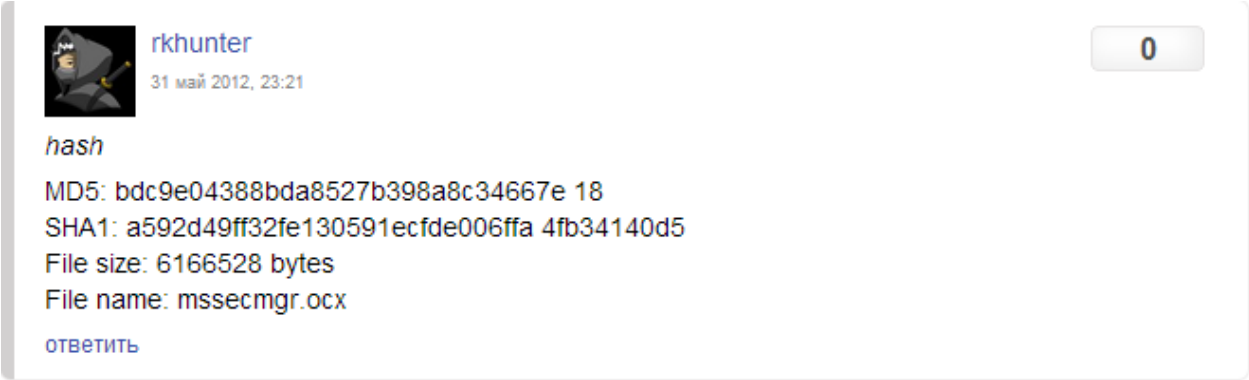
```
c:\soft\ref\virus.win32.flame.sample.mssecmgr.ocx\MSSECMGR:
        Verified:       Unsigned
        Link date:      4:36 20.02.1992
        Publisher:      Microsoft Corporation
        Description:    Windows Authentication Client Manager
        Product:        Microsofto Windowso Operating System
        Prod version:   5.1.2600.0
        File version:   5.1.2600.0
        MachineType:    32-bit
        MD5:    BDC9E04388BDA8527B398A8C34667E18
        SHA1:   A592D49FF32FE130591ECFDE006FFA4FB34140D5
        PESHA1: 52F8B55A46525180E93CFEF55724F4E5DBB6B98D
        PE256:  1D4EFDF47325F531B30086174B759DAACB985FD05D7AF5B1B2A81F5BDBCA10B5
        SHA256: 295B089792D00870DB938F2107772E0B58B23E5E8C6C4465C23AFFE87E2E67AC
```

Как видно, данное приложение не подписано, идентифицирует себя как Windows Authentication Client Manager от Microsoft Corporation.

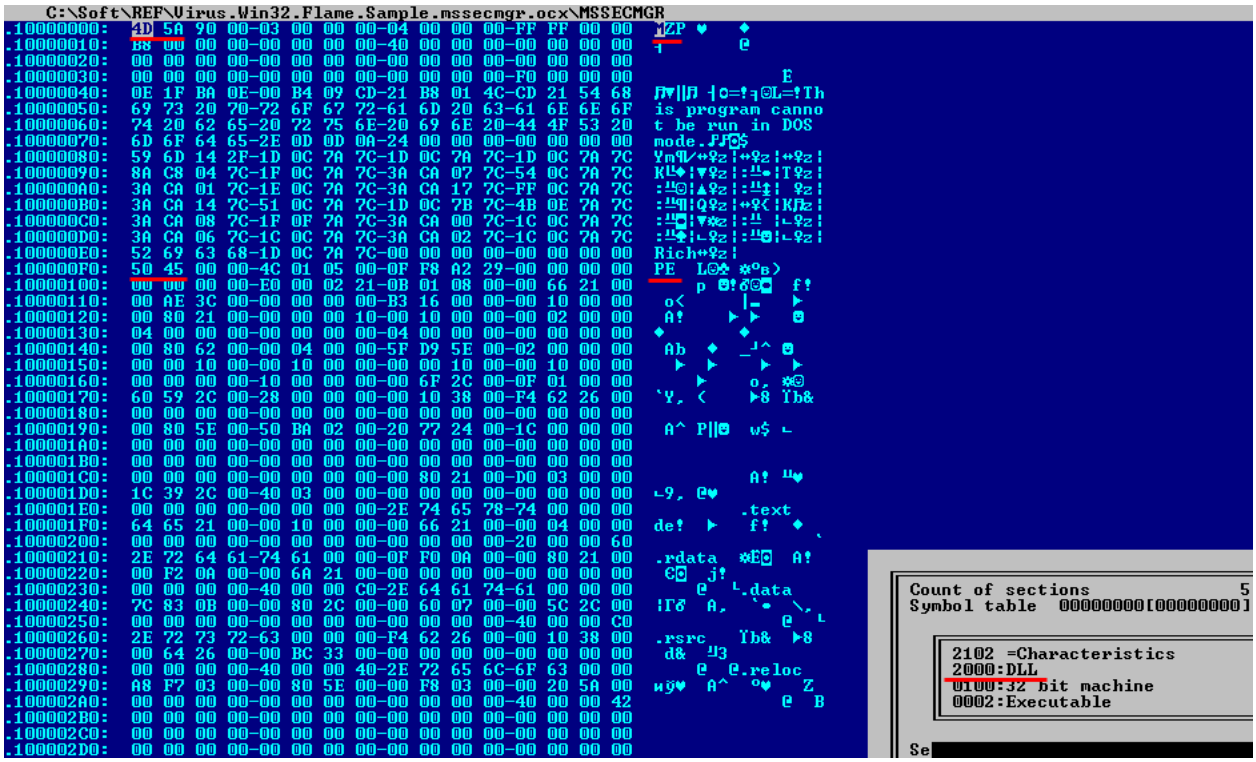
Обратим внимание на поле Link date, как видно, приложение датирует себя 1992 годом, что соответствует словам экспертов.

Хэш-суммы соответствуют указанным в комментарии к посту по следующей ссылке:

http://www.securelist.com/ru/blog/207764003/Flame_Bunny_Frog_Munch_i_BeetleJuice



Откроем данную программу в шестнадцатиричном редакторе – Hiew.



Убедимся, что это действительно переносимый исполняемый файл, причем именно DLL.

Первичная активация файла запускается одним из способов извне: либо через инструменты WMI, с использованием MOF-файла, если задействован эксплойт к уязвимости MS10-061, либо с использованием BAT-файла:

s1 = new ActiveXObject("Wscript.Shell"); s1.Run("%SYSTEMROOT%\\system32\\rundll32.exe msdclr64.ocx,DDEnumCallback");

После активации mssecmgr регистрирует себя в реестре Windows:

HKLM\SYSTEM\CurrentControlSet\Control\Lsa

Authentication Packages = mssecmgr.ocx [добавляется к существующим записям]

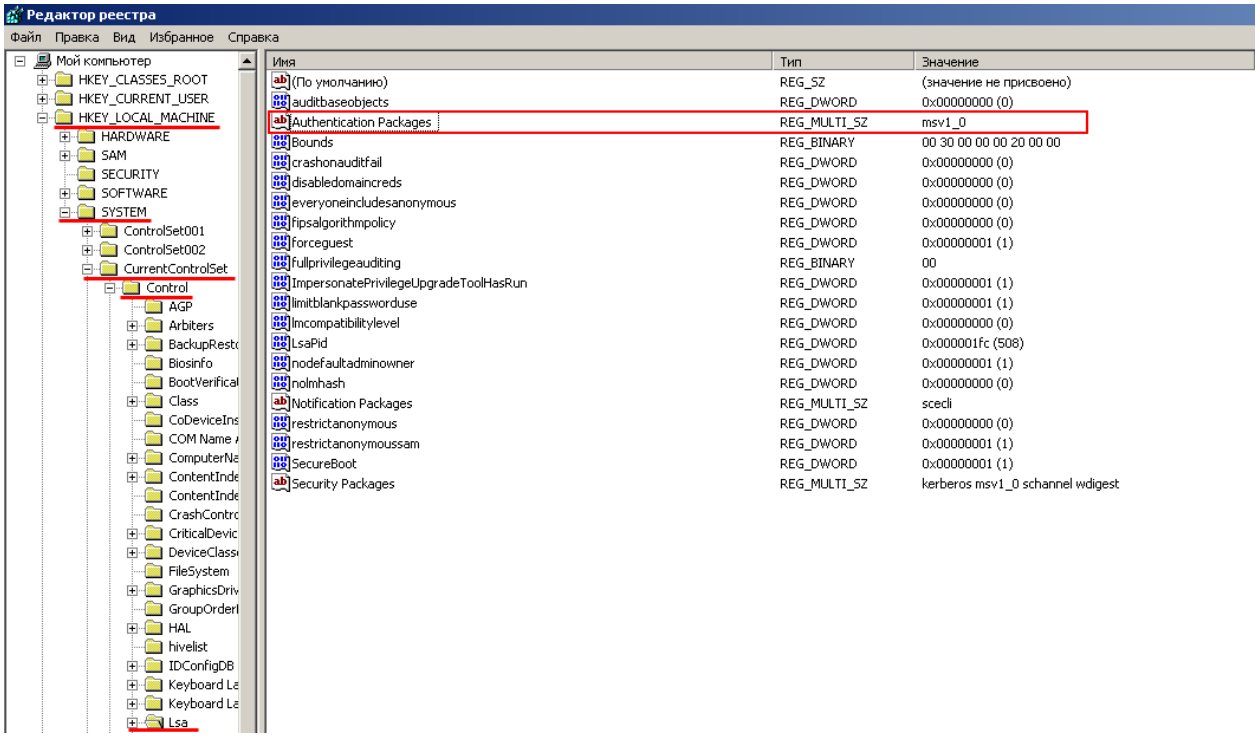
При следующей загрузке системы модуль автоматически загружается операционной системой.

Приступим к динамическому анализу данного файла. Запускать файл будем на виртуальной машине, но для начала переименуем исходный файл в MSSECMGR.OCX, и создадим bat-файл со следующим содержимым:

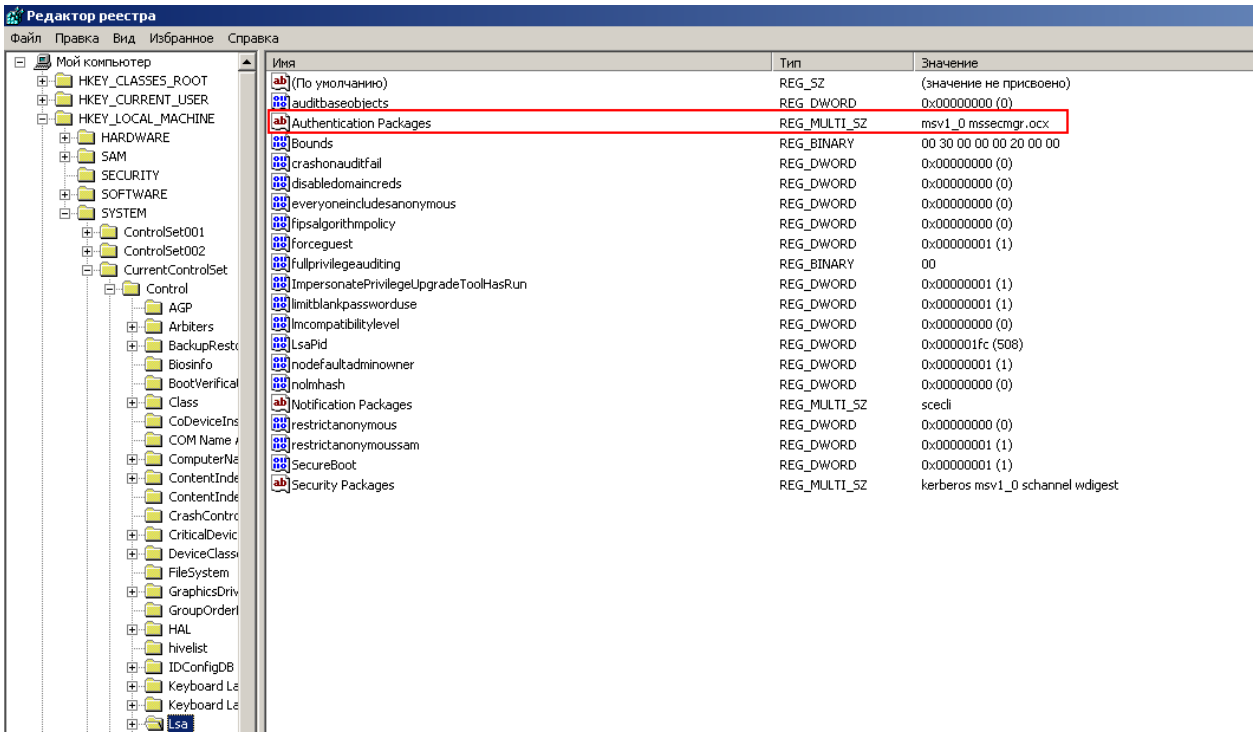
rundll32.exe MSSECMGR.OCX,DDEnumCallback

Перед запуском ВП проверим содержимое реестра по указанному пути:

HKLM\SYSTEM\CurrentControlSet\Control\Lsa



После запуска проверим эту же ветку реестра:



Данная запись в этом ключе реестра приводит к загрузке mssecmmgr.ocx как часть процесса lsass.exe, при запуске системы.

Чтобы было легче отслеживать изменения в системе, которые делает эта ВП, будем использовать утилиту TotalUninstall.

Данная утилита делает слепок системы до запуска приложения, и после запуска. Потом сверяет слепки и выдает разницу между ними.

Дополнительные модули устанавливаются в директорию %windir%\system32\:

mssecmgr.ocx

advnetcfg.ocx

msglu32.ocx

nteps32.ocx

soapr32.ocx

ccalc32.sys

boot32drv.sys

Кроме того, в папке %windir%\ может присутствовать следующий файл:

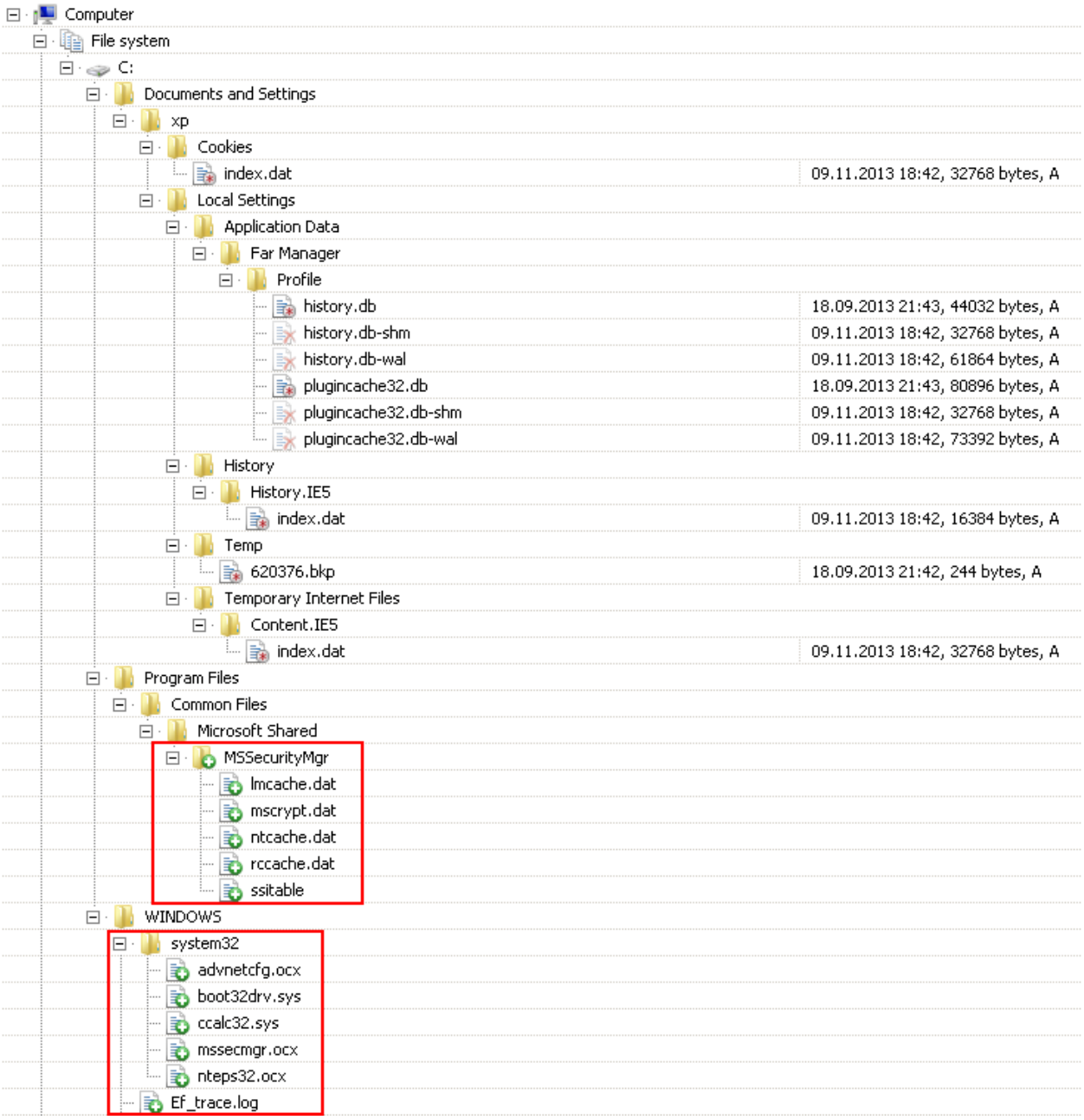
Ef_trace.log

Имена директории, используемые дополнительными компонентами Flame, могут немного различаться в зависимости от типа установки и конфигурационных опций, содержащихся в ресурсе 146:

C:\Program Files\Common Files\Microsoft Shared\MSSecurityMgr

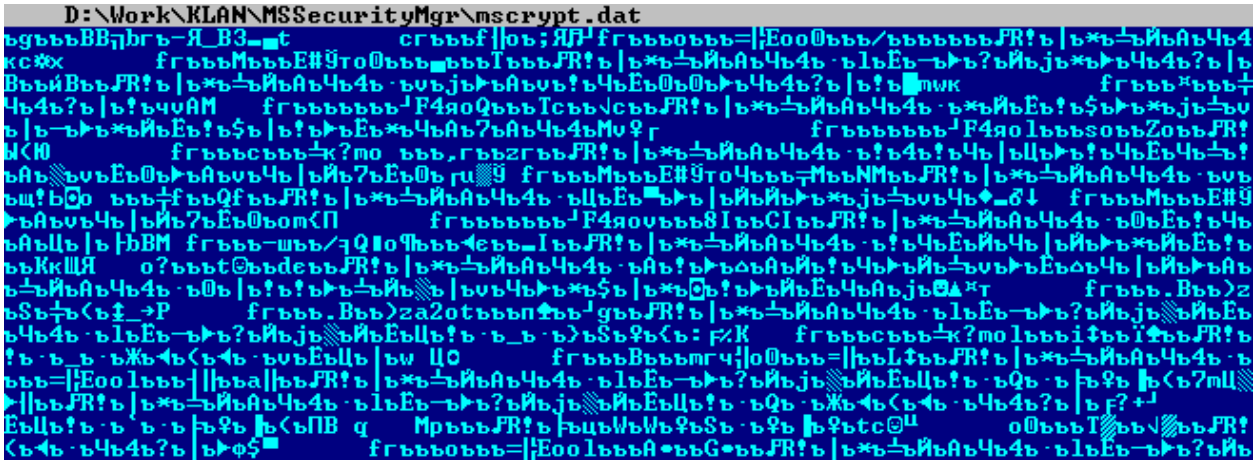
Этот каталог может содержать следующие файлы:

dstrlog.dat lmcache.dat mscrypt.dat (or wpgfilter.dat) ntcache.dat rccache.dat (or audfilter.dat) ssitable (or audache) secindex.dat wavesup3.drv (a copy of the main module, mssecmgr.ocx, in the MSAudio directory)



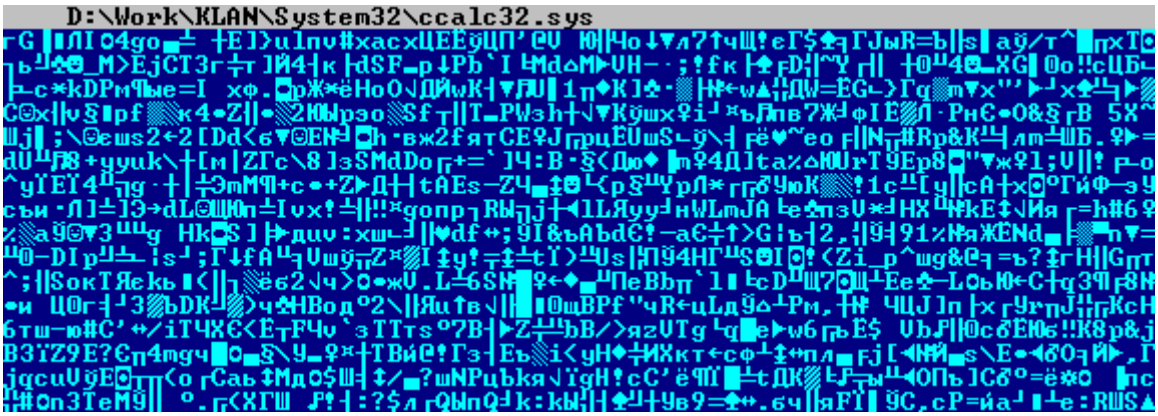
Как видно из изображения, информация, предоставленная экспертами, нашла подтверждение.

Файлы из папки MSSecurityMgr представляют собой зашифрованные данные, пример представлен ниже на скриншоте:



После анализа файлов из папки system32 было установлено:

- mssecmgr.ocx – представляет собой полную копию исходного файла
- advnetcfg.ocx – переносимый исполняемый файл, DLL
- nteps32.ocx – переносимый исполняемый файл, DLL
- boot32drv.sys и ccalc32.sys – зашифрованные данные:



Перечисление основных модулей Flame:

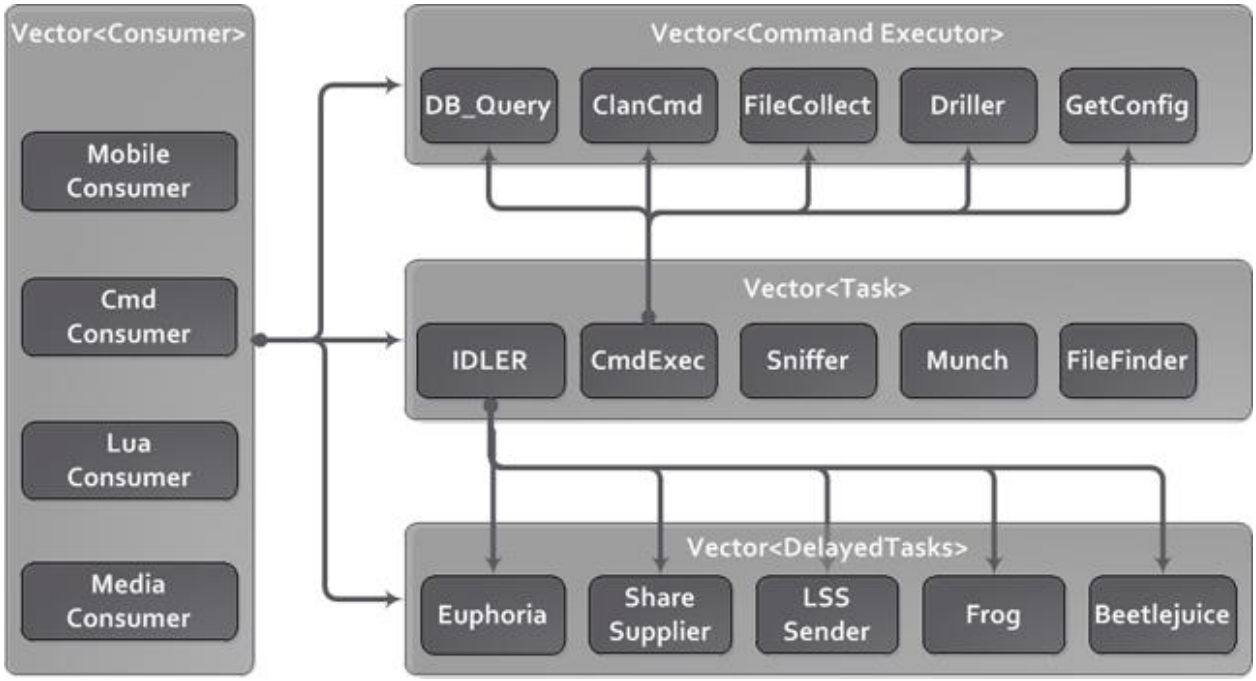
Beetlejuice	Bluetooth: составляет список устройств, находящихся вблизи зараженной машины. Может выполнять функцию «радиомаяка» – включает возможность обнаружения компьютера в настройках

	Bluetooth и шифрует с помощью base64 статус вредоносной программы для его отображения в информации об устройстве.
Microbe	Записывает аудио сигнал с имеющихся аппаратных источников. Составляет перечень всех устройств мультимедиа, записывает полную информацию о конфигурации устройств, пытается выбрать подходящее устройство для записи звука.
Infectmedia	Выбирает один из методов заражения внешних устройств, т.е. USB-носителей. Доступные методы: Autorun_infector, Euphoria.
Autorun_infector	Создает файл autorun.inf, содержащий вредоносное ПО, и запускает его с помощью специально сформированной команды open. Этот же метод использовался Stuxnet, прежде чем был задействован LNK-эксплойт.
Euphoria	Создает каталог с файлами desktop.ini и target.lnk, взятыми из записей LINK1 и LINK2 ресурса146 (отсутствовали в ресурсном файле). Директория используется как ярлык для запуска Flame.
Limbo	Создает на других машинах в сетевом домене backdoor-аккаунты с именем пользователя HelpAssistant, если имеются необходимые права.
Frog	Заражает другие компьютеры, используя предварительно созданные учетные записи пользователей. Единственная учетная запись, указанная в

	конфигурационном ресурсе, – HelpAssistant. Этот аккаунт создается атакой Limbo.
Munch	HTTP-сервер, отвечающий на запросы /view.php и /wpad.dat.
Snack	Прослушивает сетевые интерфейсы, получает пакеты NBNS и сохраняет их в лог-файле. Имеет опцию, при включении которой стартует только при старте блока Munch. Собранные данные затем используются для распространения вредоносной программы по сети.
Boot_dll_loader	Конфигурационный раздел, содержащий список всех дополнительных модулей, которые должны быть загружены и запущены.
Weasel	Создает список папок на зараженном компьютере.
Boost	Создает список «интересных» файлов по нескольким маскам имен файлов.
Telemetry	Функции ведения журнала.
Gator	При появлении доступа в Интернете устанавливает соединение с командными серверами, загружает новые модули и выгружает на сервер собранные данные.
Security	Выявляет программы, которые могут помешать работе Flame, т.е. антивирусы и сетевые экраны.
Bunny Dbquery	Назначение этих модулей на данный момент

Driller Headache Gadget	неизвестно.

Схема взаимодействия модулей между собой:



Насколько сложен Flame?

Прежде всего, Flame – это огромный пакет, состоящий из программных модулей, общий размер которых при полном развертывании составляет почти 20 МБ. Вследствие этого анализ данной ВП представляет огромную сложность. Причина столь большого размера Flame в том, что в него входит множество разных библиотек, в том числе для сжатия кода (zlib, libbz2, ppmd) и манипуляции базами данных (sqlite3), а также виртуальная машина Lua.

Lua – это скриптовый язык, т.е. язык программирования, легко поддающийся расширению и интеграции с кодом, написанным на языке C. Для многих компонентов Flame логика верхнего уровня написана на Lua – при

этом подпрограммы и библиотеки, непосредственно реализующие заражение, компилируются с C++.

Отметим немного об алгоритмах шифрования, которые используются во flame:

Для того чтобы получить доступ к файлам конфигурации, которые находятся в ресурсах, их нужно расшифровать и распаковать. Довольно простой алгоритм предоставлен на рисунке ниже:

```
if ( ResourceSize )
{
  Encrypted = Resource;
  Decrypted = (_Decrypted - Resource);
  do
  {
    v7 = *Encrypted;
    if ( *Encrypted && v7 != 0xA9 )
      v8 = 0xA9u;
    else
      v8 = 0;
    v4 ^= v8 ^ v7;
    Decrypted[Encrypted++] = v4;
    --ResourceSize;
  }
  while ( ResourceSize );
}
```

Когда данные расшифрованы, их распаковывают. Flame распаковывает данные небольшими порциями (менее килобайта каждая). Для обхода антивирусных технологий между распакованными частями вызывается Sleep API продолжительностью 10 миллисекунд. Это делает процесс распаковки более устойчивым:

```
do
{
  v7 = StreamBuffer_DeriveNew(Compressed, &v14, offset, _CompressSize - offset);
  LOBYTE(v21) = 3;
  StreamBuffer_CmpAndFree(v7, &tmp_buffer);
  LOBYTE(v21) = 2;
  BufferStream_Destructor(&v14);
  _zib = zlib;
  v9 = zlib->vTable;
  decompressed_size = 0;
  (v9->inflate)(zlib, &decompressed, &tmp_buffer, 1024, &decompressed_size);// call inflate from zlib
  LOBYTE(v21) = 4;
  if ( decompressed.BuffSize <= 0u && decompressed_size <= 0 )
  {
    v12 = GetExceptionInfo(&offset, 1, 0);
    LOBYTE(v21) = 5;
    ThrowExc(v12);
  }
  offset = (offset + decompressed_size);
  Decompress_CopyBuffer(&v13, &decompressed);
  if ( LOBYTE(_zib->bSleep) )
    Sleep(0xAu);                          // sleep 10 ms
  LOBYTE(v21) = 2;
  BufferStream_Destructor(&decompressed);
}
while ( offset < _CompressSize );
```

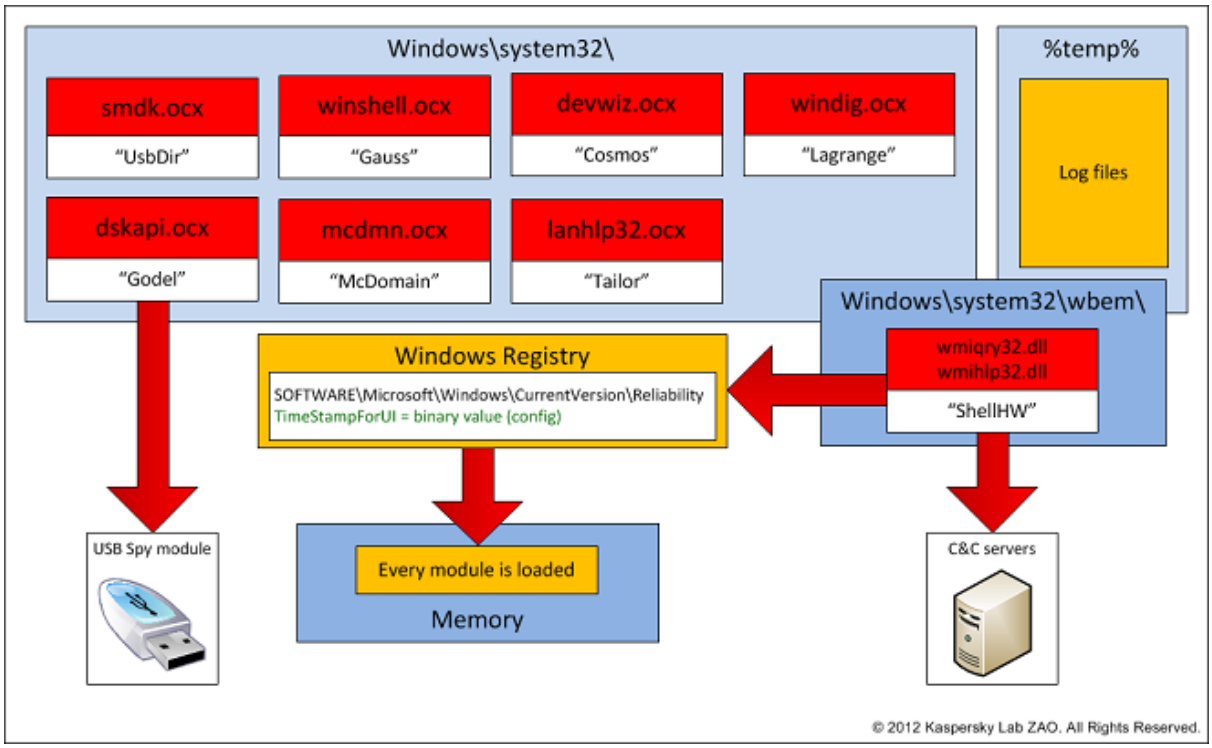
2.4 Вредоносная программа Gauss

ВП Gauss – это сложный комплекс инструментов для осуществления кибершпионажа, реализованный той же группой специалистов, что создала вредоносную платформу Flame. Комплекс имеет модульную структуру и поддерживает удаленное развертывание операторами нового функционала, который реализуется в виде дополнительных модулей. Известные на сегодняшний день модули выполняют следующие функции:

- перехват cookie-файлов и паролей в браузере;
- сбор и отправка злоумышленникам данных по конфигурации системы;
- заражение USB-носителей модулем, предназначенным для кражи данных;
- создание списков содержимого системных накопителей и папок;
- кража данных, необходимых для доступа к учетным записям различных банковских систем, действующих на Ближнем Востоке;
- перехват данных по учетным записям в социальных сетях, почтовым сервисам и системам мгновенного обмена сообщениями.

Модули имеют внутренние имена, которые, очевидно, отдают дань уважения знаменитым математикам и философам, таким как Курт Гёдель, Иоганн Карл Фридрих Гаусс и Жозеф Луи Лагранж.

Модуль под названием Gauss – наиболее важный элемент вредоносной программы, поскольку в нем реализованы возможности, связанные с кражей банковских данных. Таким образом, весь вредоносный комплекс назван по имени этого компонента.



Кроме того, из некоторых образцов ВП Gauss авторы забыли удалить отладочную информацию, в состав которой входят пути к файлам проекта. Известны следующие пути:

Вариант	Путь к файлам проекта
Август 2011 г.	d:\projects\gauss
Октябрь 2011 г.	d:\projects\gauss_for_macis_2
Дек. 2011– янв.	c:\documents and set-

2012 гг.	tings\flamer\desktop\gauss_white_1

Развитая модульная архитектура ВП Gauss напоминает ВП Flame: программа использует зашифрованный ключ реестра для хранения информации о том, какие модули должны быть загружены, она рассчитана на скрытную работу в системе, избегает средств защиты и мониторинга, и при этом сама осуществляет детальный мониторинг системы. Кроме того, ВП Gauss содержит 64-разрядный вредоносный функционал, а также совместимые с браузером Firefox плагины, предназначенные для кражи и мониторинга данных, пересылаемых пользователями нескольких ливанских банков – Bank of Beirut, EBLF, BlomBank, ByblosBank, FransaBank и Credit Libanais. Вдобавок, программа нацелена, в частности, на пользователей Citibank и PayPal.

2.5 Вредоносная программа Duqu

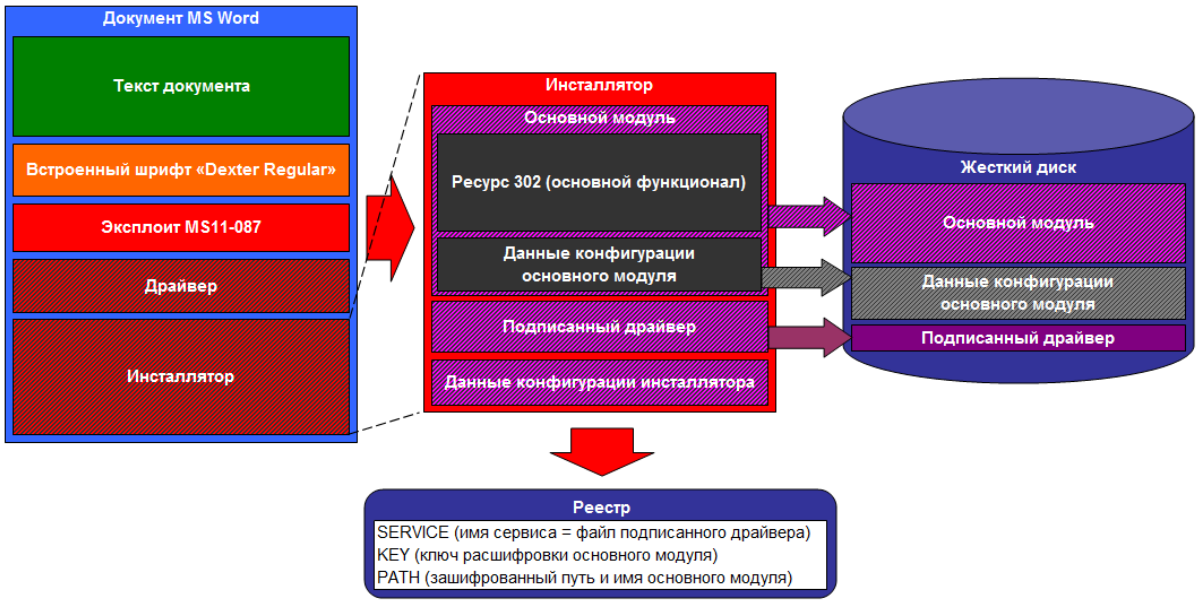
Расследование, проводимое специалистами Венгерской организации CrySyS (Венгерская лаборатория криптографии и системной безопасности Будапештского университета технологии и экономики), привело к обнаружению установщика (дроппера), посредством которого происходило заражение системы ВП Duqu. Он представлял собой файл формата Microsoft Word с эксплойтом уязвимости драйвера win32k.sys (MS11-087, описана Microsoft 13 ноября 2011), отвечающего за механизм рендеринга TTF шрифтов.

В документе Word, таким образом, находились следующие компоненты:

- текстовое содержимое;
- встроенный шрифт;
- шелкод эксплойта;
- драйвер;
- инсталлятор (библиотека DLL).

Благодаря тому, что win32k.sys выполняется от имени привилегированного пользователя 'System', разработчиками Duqu была элегантно решена задача как несанкционированного запуска, так и повышения прав (запуск из-под аккаунта пользователя с ограниченными правами). Инсталлятор после получения управления расшифровывал в памяти находящиеся в нем три блока данных, содержащих:

- подписанный драйвер (sys);
- основной модуль (dll);
- данные конфигурации инсталлятора (pnf).



Компанией Symantec было обнаружено, по меньшей мере, четыре вида «полезной нагрузки», загруженной по команде от управляющего центра Duqu. При этом только одна из них была резидентной и скомпилированной в виде исполняемого файла (.exe), который сохранялся на диск. Остальные три были выполнены в виде dll-библиотеки. Они загружались динамически и выполнялись в памяти без сохранения на диск.

Шпионский модуль собирает следующую информацию:

- список запущенных процессов, информацию о текущем пользователе и домене;
- перечень логических дисков, включая сетевые;
- снимки экрана;
- адреса сетевых интерфейсов, таблицы маршрутизации;
- лог-файл нажатий клавиш клавиатуры;
- имена открытых окон приложений;
- перечень доступных ресурсов сети (sharing resources);
- полный список файлов на всех дисках, включая съемные;
- список компьютеров в сетевом окружении.

Проанализировав отчеты антивирусных компаний, можно сделать вывод, что атаки, проводимые при помощи Duqu, являются таргетированными, нацеленными на конкретных лиц. Об этом свидетельствует число зарегистрированных обнаружений данной ВП:

организации A – Франция, Нидерланды, Швейцария, Украина;

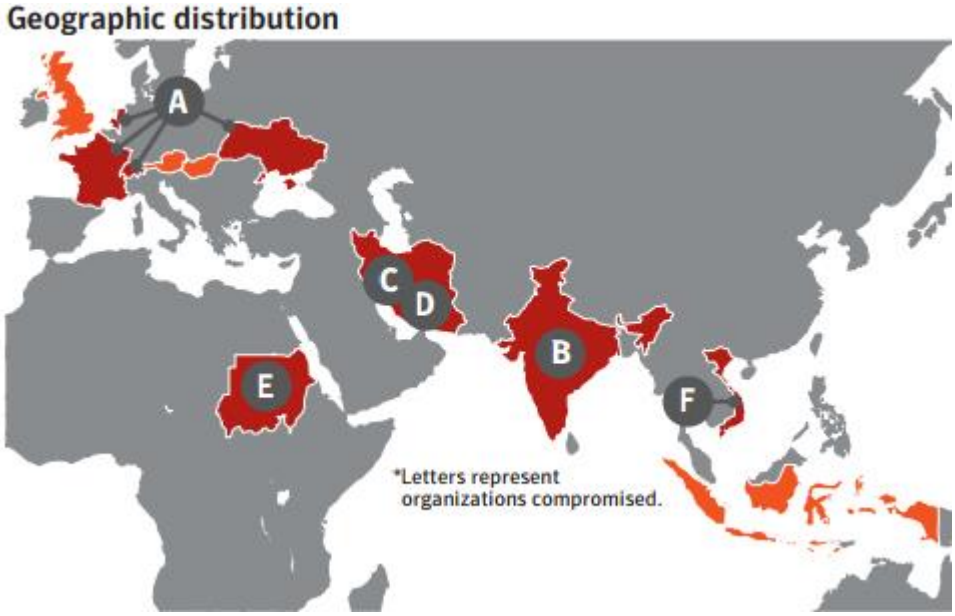
организация B – Индия;

организация C – Иран;

организация D – Иран;

организация E – Судан;

организация F – Вьетнам.



К сожалению, в связи с малым числом обнаружений, авторам работы не удалось найти исходных копий данной ВП, поэтому остается полагаться на те данные, которые предоставлены антивирусными компаниями.

2.6 Вредоносная программа Icefog

С 2011 года специалистами в сфере информационной безопасности был отслежен целый ряд атак, которые связывают с группой киберпреступников, под названием Icefog. По мнению экспертов, это относительно небольшая группа злоумышленников, которая на заказ проводит кибератаки на госучреждения, компании, выполняющие заказы военных ведомств, судостроительные фирмы, телекоммуникационные компании и операторов спутниковой связи, промышленные и высокотехнологичные компании.

Атаки Icefog проводят с использованием специально созданных инструментов кибершпионажа, направленных на операционные системы Microsoft Windows и Apple Mac OS X. Во время атаки злоумышленники напрямую управляют зараженными компьютерами. Помимо Icefog та же группа злоумышленников использует и другие вредоносные инструменты и «бэкдоры» для распространения ВП по локальным сетям и организации утечки данных с зараженных компьютеров.

2.6.1 Основные результаты анализа атак ВП Icefog

Основные методы, которыми пользуются Icefog, – это целевой фишинг (spear-phishing) и использование известных уязвимостей. В качестве «наживки» использовались документы, связанные с интересами потенциальной жертвы. Злоумышленники крадут секретные документы, бизнес-документы, логины и пароли к электронной почте, пароли к различным ресурсам в локальных сетях «жертв» и за их пределами.

В ходе большинства других атак компьютеры остаются зараженными в течение нескольких месяцев и даже лет и все это время происходит незаметная утечка информации. Сама атака проводится быстро и с «хирургической точностью»: цель каждой операции – получить конкретную информацию на конкретных компьютерах. Сразу после этого киберпреступники оставляют зараженный компьютер и атакуют следующие цели. В большинстве случаев группа Icefog, по всей видимости, имеет очень четкое представление о том, что им нужно от конкретных жертв. На взломанных компьютерах осуществляется поиск файлов с конкретными названиями, которые затем пересылаются на командный сервер.

2.6.2 Содержание атак ВП Icefog

По сути, одноименный Icefog – это «бэкдор», выступающий в роли интерактивного шпионского инструмента, который контролируется непосредственно злоумышленниками. Он не похищает данные автоматически – управляется вручную, выполняя действия прямо на работающих зараженных компьютерах. В ходе Icefog-атаки на машины пользователей загружаются другие вредоносные инструменты и «бэкдоры» для последовательного распространения в локальной сети «жертвы» и кражи данных.

После загрузки «бэкдора» на компьютер, он действует как троянская программа с удаленным управлением, который имеет пять основных функции для ведения кибершпионажа:

- кража основной информации о системе и ее загрузка на командные серверы, которыми владеет и которые контролирует группа Icefog;
- прием команд Icefog и выполнение их на зараженном компьютере;
- кража паролей пользователя и их пересылка на командные серверы Icefog;
- загрузка файлов (инструментов) с командных серверов на зараженные компьютеры;
- прямое выполнение SQL-команд, передаваемых Icefog, на любых MSSQL-серверах в сети.

2.6.3 Особенности ВП Icefog

Вообще говоря, любая APT-атака особенна и по-своему уникальна. Если говорить об Icefog, у него есть характерные черты, которые отличают его от всех остальных ВП подобного рода:

- кража файлов происходит не автоматически. Вместо этого злоумышленники обрабатывают «жертвы» одну за другой – ищут и копируют только определенную информацию;
- командные серверы реализованы как веб-приложение на платформе Microsoft .NET;
- командные серверы ведут полную запись производимых атак, где фиксируется каждая команда и каждое действие, производимое на компьютере-жертве;
- используются HWP-документы с эксплойтами;
- известно несколько сотен случаев заражения Mac OS X компьютеров.

2.6.4 Последействие ВП Icefog

Атаки Icefog могут быть нацелены на любую организацию или компанию, владеющую ценными данными; задачи атак могут быть различными – от кибершпионажа или разведопераций, финансируемых отдельными госу-

дарствами, до «финансовой» киберпреступной операции. Судя по результатам анализа и географии атакованных компьютеров, члены группы Icefog могут зарабатывать деньги на краденных данных или использовать их в целях кибершпионажа.

Необычен стиль проведения таких операций – быстрое удаление с «места преступления». В других случаях, компьютеры обычно, остаются зараженными в течение скольких месяцев, а то и лет, и все это время данные с них «потихоньку сливаются на сторону». В противоположность этому субъекты, стоящие за Icefog, как уже говорилось выше, заранее имеют очень четкое представление о том, что им нужно от конкретной жертвы. Как только операция проведена, и необходимая информация с конкретного компьютера получена, вредоносная активность на нем прекращается.

3 АНАЛИЗ СРЕДСТВ ДОСТАВКИ ВРЕДОНОСНЫХ ПРОГРАММ ДО ОБЪЕКТОВ ИХ АТАКИ

Ниже представлены[8] некоторые из достаточно большого списка потенциальных устройств средств доставки вредоносных программ (СДВП)[9] до объекта атаки для получения скрытого доступа к КС «противника».

IRATEMONK позволяет обеспечить присутствие ВП для слежки на настольных и портативных компьютерах с помощью закладки в прошивке жесткого диска, которая позволяет получить возможность исполнения своего кода путем замещения MBR. Метод работает на различных дисках Western Digital, Seagate, Maxtor и Samsung. Из файловых систем поддерживаются FAT, NTFS, EXT3 и UFS. Системы с RAID не поддерживаются. После внедрения IRATEMONK будет запускать свою функциональную часть при каждом включении целевого компьютера.

SWAP позволяет обеспечить присутствие ВП для шпионажа за счет использования BIOS материнской платы и HPA области жесткого диска путем исполнения кода до запуска операционной системы. Данная ВП позволяет получить удаленный доступ к различным операционным системам (Windows, FreeBSD, Linux, Solaris) с различными файловыми системами (FAT32, NTFS, EXT2, EXT3, UFS 1.0). Для установки используются две утилиты: ARKSTREAM перепрошивает BIOS, TWISTEDKILT записывает в HPA область диска SWAP и его функциональная часть.

COTTONMOUTH-I аппаратная закладка на USB, предоставляющая беспроводной мост к целевой сети, а также загрузки эксплойтов на ресурсы целевой системы. Может создавать скрытый канал связи для передачи команд и данных между аппаратными и программными закладками. При помощи встроенного радиопередатчика может взаимодействовать с другими COTTONMOUTH. В основе лежит элементная база TRINITY, в качестве ра-

---

[8] По данным сайта http://www.habrahabr.ru
[9] Доставки не через глобальные сети.

диопередатчика используется HOWLERMONKEY. Существует версия под названием MOCCASIN, представляющая собой закладку в коннекторе USB-клавиатуры.

FIREWALK аппаратная сетевая закладка, способная пассивно собирать трафик сети Gigabit Ethernet, а также осуществлять активные инъекции в Ethernet пакеты целевой сети. Позволяет создавать VPN-туннель между целевой сетью и центром. Возможно установление беспроводной коммуникации с другими HOWLERMONKEY-совместимыми устройствами. Исполнение данной закладки аналогично COTTONMOUTH-III, такой же блок разъемов (RJ45 и два USB) на шасси. В основе лежит элементная база TRINITY, в качестве радиопередатчика используется HOWLERMONKEY.

NIGHTSTAND мобильный комплекс для проведения активных атак на Wi-Fi-сети, целями являются машины под управлением Windows (от Win2k до WinXP SP2). Обычно используется в операциях, в которых доступ к цели невозможен. Комплекс реализован на базе ноутбука под управлением Linux и радиооборудования. Вместе с внешними усилителями и антеннами дальность действия может достигать 13 км.

DEITYBOUNCE предоставляет программный доступ к серверам Dell PowerEdge при помощи BIOS материнской платы и использования SMM-режима для получения возможности запуска перед загрузкой системы. Установка может быть произведена при помощи ARKSTREAM, либо при помощи USB-флеш-накопителя. После установки будет выполняться каждый раз при включении системы. Целями могут являться Dell PowerEdge 1850/2850/1950/2950 с версиями BIOS A02, A05, A06, 1.1.0, 1.2.0 или 1.3.7.

FEEDTROUGH представляет собой технику установки двух ВП – BANANAGLEE и ZESTYLEAK, используемых «против» межсетевого экрана Juniper Netscreen. Подвержены угрозе следующие модели Juniper: ns5xt, ns25, ns50, ns200, ns500 и ISG 1000. Метод отрабатывается при старте межсетевого экрана, если операционная система есть в базе данных, то устанавливаются

ВП, в противном случае устройство загружается в обычном режиме. FEEDTROUGH сохраняется при обновлении операционной системы межсетевого экрана.

CTX4000 представляет собой портативный излучатель непрерывного действия, предназначен для подсвета целевых систем для получения данных от установленных там закладок. Предшественник PHOTOANGLO.

NIGTHWATCH – система на базе портативного компьютера предназначенная для обработки сигналов от монитора цели. Сигнал может поступать как от систем сбора информации (например, от CTX4000 или PHOTOANGLO при подсвете закладки в видеокабеле RAGEMASTER), так и от приемника общего назначения.

HOWLERMONKEY представляет собой радиопередатчик малого и среднего радиуса. Является специальном радиомодулем для других аппаратных закладок. Используется для получения данных от закладок и предоставления удаленного доступа к ним.

Существует и способы внедрения ВП через трансиверы, встраиваемые в USB-кабели или USB-устройства, через WiFi-, Bluetooth-, GSM-устройства и соединения, подключаемые к атакуемому компьютеру.

Таким образом, СДВП до объекта атаки могут быть самыми разнообразными и «изощренными», использовать самые разнообразные физические принципы обработки и передачи информации, самые разнообразные среды ее распространения, представлять собой вполне «безобидные», на первый взгляд, устройства и гаджеты. Противодействие им – задача ничем не легче, чем противодействие самим БВП, а создание средств противодействия – сложная ресурсоемкая научно-техническая задача, комплексное решение которой, скорее всего, под силу только государству.

4 ОСНОВНЫЕ ВЫВОДЫ ПО РЕЗУЛЬТАТАМ АНАЛИЗА И ЗАКЛЮЧЕНИЕ

Исследуемые ВП имеют такую большую сложность, требуют таких ресурсов для разработки, что немногие злоумышленники и группы злоумышленников будут способны создать их в будущем. Поэтому не стоит ожидать внезапного появления массы подобных по изощренности ВП.

Следует отметить, например, что ВП Stuxnet, например, выделяется попытками прямого поражения критически важных объектов инфраструктуры. Воздействие Stuxnet на объекты реального мира – выделяют его «особняком» от любых угроз, которые наблюдались в прошлом. Экспертный инженерный анализ Stuxnet позволяет говорить о том, что это, пожалуй, первый характерный пример применения разрушительного кибероружия, в качестве которого выступает эта БВП.

Хотя здесь возникает целый ряд вопросов, на некоторые из которых пока нет ответов:

- насколько политический и военный эффект от проведения подобных кибероперациий соответствует их замыслу;
- может все же это – демонстрация возможностей, устрашение, «проба пера» или что-то подобное;
- насколько затраты (временные, финансовые, «людские», интеллектуальные) сопоставимы с наносимым ущербом для объекта кибератаки и следует ли вообще ожидать результатов от такого сопоставления[10].

Ведь, если бы Stuxnet нанес заводу в Натанзе значительный ущерб, то обогащение урана замедлилось бы. Однако отчеты МАГАТЭ говорят об обратном: в период с 2007 по 2013 г.г. количество урана обогащенного на нем

[10] Может здесь стоит ожидать только военного и/или политического эффекта (но не экономического).

равномерно росло. Обогащение до 20% началось как раз в тот период, когда часть центрифуг была выведена из строя.

Существенным отличием БВП от конвенциального оружия является невозможность повторного или кратного их применения. БВП или, точнее их сигнатуры уже внесены во все антивирусные базы данных, атакованные информационные объекты, как, впрочем, и другие объекты аналогичного назначения уже давно «привиты», агентурные каналы проникновения к ним перекрыты. Надо писать принципиально новые ВП, искать принципиально новые каналы доставки их кодов. В то время как в конвенциальном оружии (при повторном применении) и средствах его доставки никаких принципиальных доработок или уж тем более коренной модификации проводить не надо.

В ноябре 2013 года во многих отечественных и зарубежных СМИ появилась информации[11] о том, что большим количеством ВП, разработанных АНБ США, были «заражены» более 50 000 компьютеров по всему миру. Если это верно, то хотелось бы узнать об этих программах как можно больше. И это является одним из направлений дальнейшего развития настоящего проекта.

[11] Со ссылкой на Э.Сноудена.

Список использованных источников

1. http://www.kaspersky.ru/news?id=207733835
2. http://www.habrahabr.ru
3. http://www.securelist.com/ru/
4. http://www.infosecurity-magazine.com
5. http://www.securitylab.ru/blog/personal/tsarev/28372.php
6. http://www.securitylab.ru/blog/personal/tsarev/30693.php
7. http://www.securelist.com/ru/analysis/208050779/Kaspersky_Security_bulletin_2012_Kiberoruzhie
8. http://www.securelist.com/ru/blog/207764148
9. http://compsmir.ru/
10. *Казарин О.В.* Методология защиты программного обеспечения. – М.: МЦНМО, 2009. – 464 с.
11. *Казарин О.В., Тарасов А.А.* Современные концепции кибербезопасности ведущих зарубежных государств// Вестник РГГУ. Серия «Информатика. Защита информации. Математика». – 2013. – №15. – С.58-74.
12. Программно-аппаратные средства информационной безопасности. Защита программ и данных / *П.Ю. Белкин, О.О. Михальский, А.С. Першаков и др*. – М.: Радио и связь, 2000. – 168 с.