Dr.WEB

Исследование троянца Belonard, использующего уязвимости нулевого дня в Counter-Strike 1.6

«Доктор Веб» предлагает эффективные антивирусные и антиспам-решения как для государственных организаций и крупных компаний, так и для частных пользователей.

Антивирусные решения семейства Dr.Web разрабатываются с 1992 года и неизменно демонстрируют превосходные результаты детектирования вредоносных программ, соответствуют мировым стандартам безопасности.
Сертификаты и награды, а также обширная география пользователей свидетельствуют об исключительном доверии к продуктам компании.

Исследование троянца Belonard, использующего уязвимости нулевого дня в Counter-Strike 1.6
3/11/2019

ООО «Доктор Веб», Центральный офис в России
125040
Россия, Москва
3-я улица Ямского поля, вл.2, корп.12А

Сайт: https://www.drweb.com/
Телефон: +7 (495) 789-45-87

Информацию о региональных представительствах и офисах Вы можете найти на официальном сайте компании.

Содержание

Введение 4

Инфицирование 7

Установка в клиенте 8

Установка в системе 10

Полезная нагрузка и распространение 13

Шифрование 16

Закрытие ботнета 18

Индикаторы компрометации 19

Введение

Игра Counter-Strike была выпущена компанией Valve еще в 2000 году. Несмотря на возраст игры, она имеет большую фанатскую базу — количество игроков с официальными клиентами CS 1.6 в среднем достигает 20 000 человек онлайн, а общее число зарегистрированных в Steam игровых серверов превышает 5000. Продажа, аренда и раскрутка серверов стали настоящим бизнесом в виде услуг на различных сайтах. К примеру, поднятие сервера в рейтинге на неделю стоит примерно 200 рублей, но большое количество покупателей при сравнительно небольших расходах делают эту стратегию довольно успешной бизнес-моделью.

Многие владельцы популярных игровых серверов тоже зарабатывают за счет игроков, продавая различные привилегии: защита от бана, доступ к оружию и многое другое. Если одни держатели серверов рекламируются самостоятельно, то некоторые платят за раскрутку сервера поставщикам. Покупая такую услугу, заказчики часто не знают, какие методы используются для продвижения их серверов. Как выяснилось, разработчик под ником Belonard прибегал к нелегальным средствам раскрутки: его сервер заражал устройства игроков троянцем и использовал их для продвижения других игровых серверов.

Держатель вредоносного сервера использует уязвимости клиента игры и созданного им троянца как техническое обеспечение для своего бизнеса. Задача троянца заключается в том, чтобы проникнуть на устройство игрока и скачать вредоносное ПО, которое обеспечит автозапуск троянца в системе и его распространение на устройства других

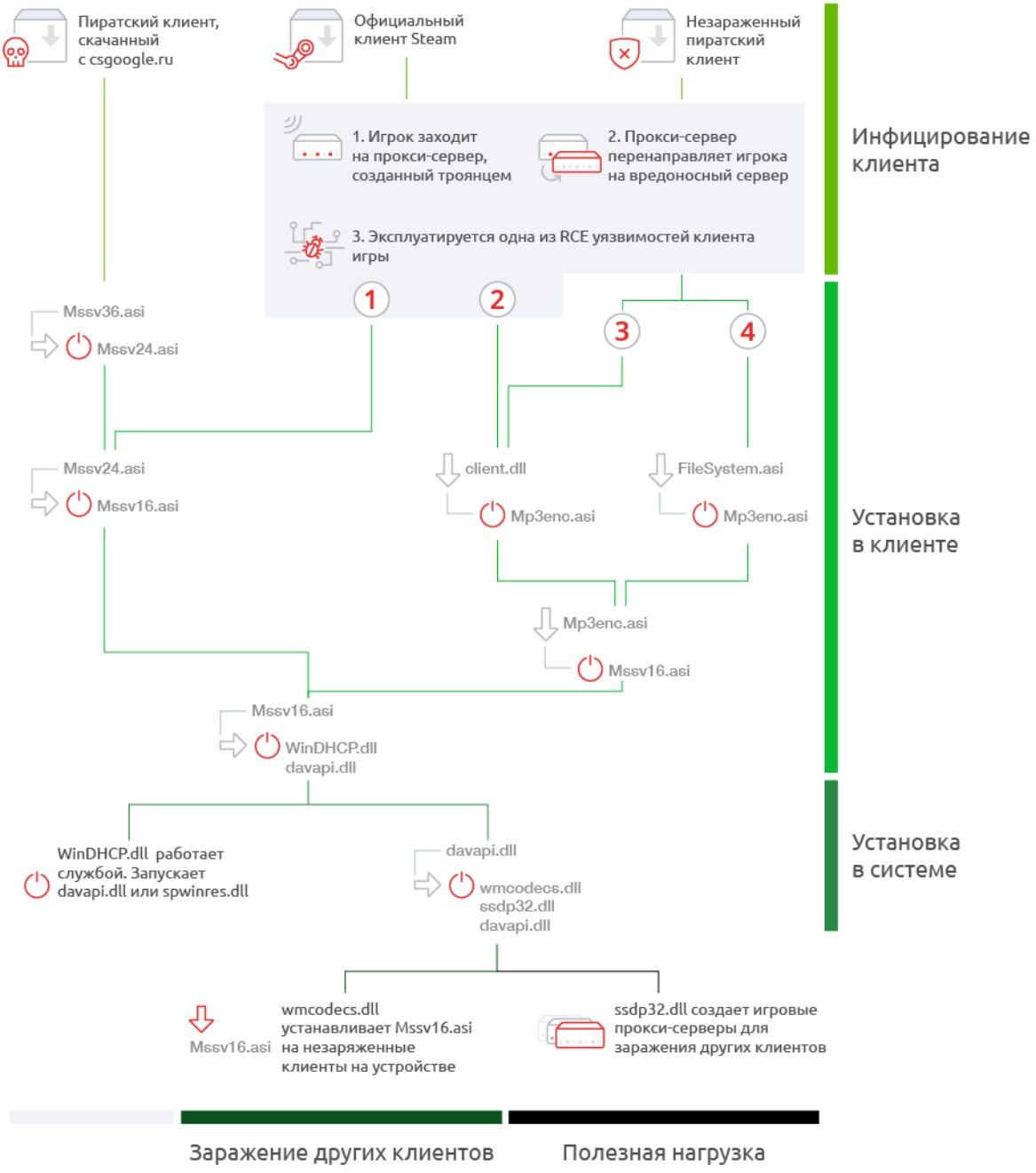
игроков. Для этого используются уязвимости Remote Code Execution (RCE): две такие уязвимости найдены в официальном клиенте игры и четыре в пиратском.

Установившись в системе, Trojan.Belonard меняет список доступных игровых серверов в клиенте игры, а также создает на зараженном компьютере игровые прокси-серверы для распространения троянца. Как правило, на прокси-серверах – невысокий пинг, поэтому другие игроки видят их вверху списка серверов. Выбрав один из них, игрок попадает на вредоносный сервер и заражается Trojan.Belonard.

Благодаря такой схеме разработчику троянца удалось создать ботнет, занимающий значительную часть игровых серверов CS 1.6. По данным наших аналитиков, из порядка 5000 серверов, доступных из официального клиента Steam, 1951 оказались созданными троянцем Belonard. Это составляет 39% процентов игровых серверов. Сеть такого размера позволила разработчику троянца продвигать другие серверы за деньги, добавляя их в списки доступных серверов зараженных игровых клиентов.

Ранее мы уже описывали похожий случай атаки через CS 1.6, где троянец попадал на устройство игрока через вредоносный сервер. Но если в том случае пользователь должен был подтвердить загрузку вредоносных файлов, то на этот раз троянец попадает на устройство незаметно для жертвы. «Доктор Веб» уведомил об этих и других уязвимостях разработчика игры — компанию Valve. Ее представители сообщили о том, что они работают над этой проблемой, но на данный момент нет информации о сроках, в которые уязвимости будут устранены.

Dr.WEB® Схема работы Trojan.Belonard



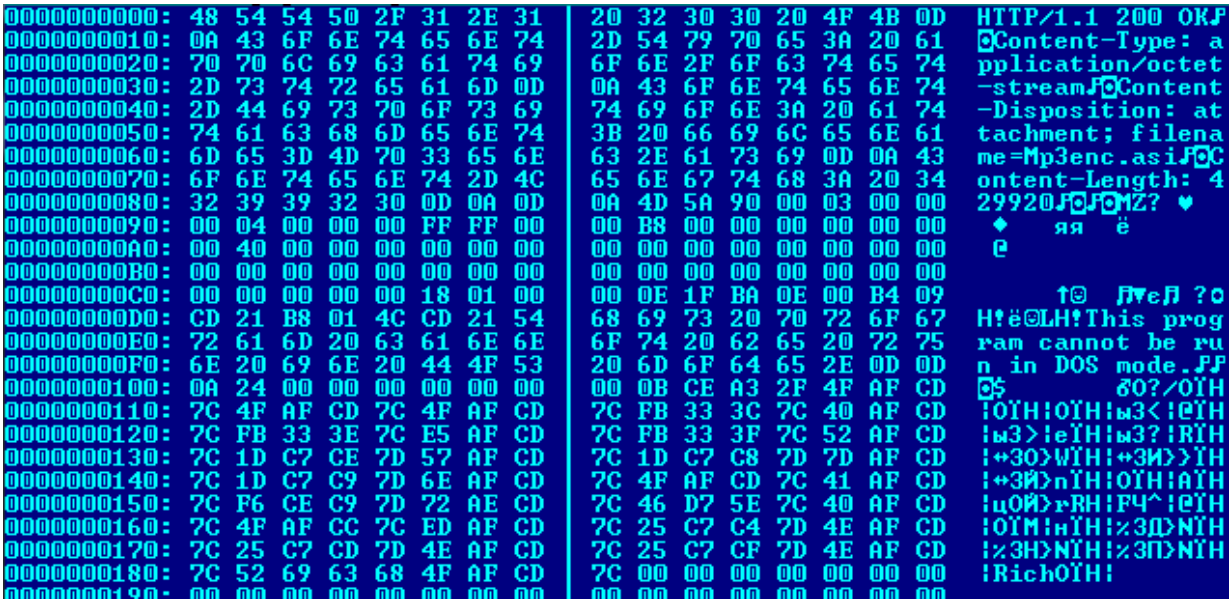
Инфицирование

Trojan.Belonard состоит из 11 компонентов и действует по разным сценариям в зависимости от клиента игры. Если клиент игры лицензионный, троянец попадает на устройство пользователя через RCE-уязвимость, эксплуатируемую вредоносным сервером, после чего обеспечивает себе установку в системе. По такому же сценарию происходит заражение чистого пиратского клиента. Если пользователь скачивает зараженный клиент с сайта владельца вредоносного сервера, установка троянца в системе происходит после первого запуска игры.

Рассмотрим подробнее процесс заражения клиента. Игрок запускает официальный клиент Steam и выбирает сервер для игры. При подключении к вредоносному серверу задействуется одна из RCE-уязвимостей, в результате чего на компьютер игрока будет загружена, а затем выполнена троянская библиотека client.dll (Trojan.Belonard.1) или файл Mssv24.asi (Trojan.Belonard.5).

Попав на устройство, Trojan.Belonard.1 первым делом удаляет любые .dat файлы, находящиеся в одном каталоге с файлом процесса библиотеки. После чего она подключается к управляющему серверу fuztxhus.valve-ms[.]ru:28445 и отправляет ему шифрованный запрос на скачивание файла Mp3enc.asi (Trojan.Belonard.2). Сервер передает запрошенный файл в зашифрованном формате.

Скриншот расшифрованного пакета данных, полученного от сервера:



Установка в клиенте

Автозапуск на официальном или пиратском клиенте происходит за счет особенности клиента Counter-Strike. При запуске игра автоматически загружает любые файлы с расширением .asi из корня игры.

Клиент, скачанный с сайта разработчика троянца, уже заражен Trojan.Belonard.10 (имя файла — Mssv36.asi), но обеспечение автозапуска в клиенте у него происходит иначе, чем на чистом. После установки зараженного клиента, Trojan.Belonard.10 проверяет наличие одного из своих компонентов в ОС пользователя. Если их нет, распаковывает из своего тела и загружает в память своего процесса Trojan.Belonard.5 (имя файла — Mssv24.asi). Как и многие другие модули троянца, Trojan.Belonard.10 подменяет дату и время создания, модификации, а также доступа к файлу. Это делается для того, чтобы нельзя было найти файлы троянца, отсортировав содержимое папки по дате создания.

После установки нового компонента Trojan.Belonard.10 остается в системе и выполняет роль протектора клиента. Его задача — фильтровать запросы, файлы и консольные команды, полученные от других игровых серверов, а также передавать информацию о попытках внесения изменения в клиент на сервер разработчика троянца.

Trojan.Belonard.5 получает в DllMain информацию о запущенном процессе и путях до модуля. Если имя процесса отлично от rundll32.exe, запускает отдельный поток для последующих действий. В запущенном потоке Trojan.Belonard.5 создает ключ [HKCU\SOFTWARE\Microsoft\Windows NT\CurrentVersion\AppCompatFlags\Layers] '<путь до исполняемого файла процесса>', присваивает ему значение «RUNASADMIN» и проверяет имя модуля. Если это не «Mssv24.asi», копирует себя в «Mssv24.asi», удаляет версию с другим названием, скачивает и запускает Trojan.Belonard.3 (имя файла — Mssv16.asi). Если имя совпадает, сразу переходит к скачиванию и запуску троянца.

Автозапуск на чистом клиенте происходит через Trojan.Belonard.2. После загрузки на компьютер он проверяет в DllMain имя процесса, куда загружена client.dll (Trojan.Belonard.1). Если это не rundll32.exe, он создает поток с ключом [HKCU\SOFTWARE\Microsoft\Windows NT\CurrentVersion\AppCompatFlags\Layers] '<путь до исполняемого файла процесса>' и присваивает ему значение «RUNASADMIN». Выполнив эти действия, он собирает данные об устройстве пользователя и извлекает информацию из файла DialogGamePage.res. После чего в зашифрованном виде отправляет полученные данные на сервер разработчика троянца.

Структура собранных данных о системе:

```
#pragma pack(push, 1)
struct st_info
{
  _BYTE byte0; // 0x01
  _BYTE bIsWow64Process;
  _BYTE DialogGameNameResData[0x10];
  _DWORD dwProductVersionMS;
  _WORD ProductVersionLS; // (dwProductVersionLS & 0xffff0000) >> 16
  _WORD DefaultUILang;
  _DWORD TotalMemory; // in Mb
  _DWORD dwNumberOfProcessors;
  _WORD wProcessorArchitecture;
  _WORD wProcessorLevel;
  _WORD wProcessorRevision;
  _QWORD ticks;
  _BYTE IsTokenElevated;
  _BYTE IsWinDHCPServiceRunning;
  _BYTE IsWinDHCPDllExists;
  _BYTE IsDavapiDllExists;
  _BYTE IsSpwinresDllExists;
  _BYTE IsWmcodecsDllExists;
  _BYTE IsSsdp32DllExists;
  char szSystemDir[];
  char szSysWow64Dir[]; // absent for x86 OS
  char szMp3encAsiPath[];
  char szProcessExePath[];
  char szCurrentDir[];
  _BYTE IsMssv24AsiExists;
  _BYTE IsDialogServerNameResExists;
  _BYTE DialogServerNameResData[0x0e];
  _BYTE IsCstrikeSaveFolderExists;
  _DWORD dwCstrikeSaveFilesCount;
  _BYTE IsSteamClient;
  _BYTE ModSHA256[32];
}
#pragma pack(pop)
```

В ответ сервер отправляет файл Mssv16.asi (Trojan.Belonard.3). Метаинформация о новом модуле сохраняется в файле DialogGamePage.res, а Trojan.Belonard.5 удаляется с устройства пользователя.

Установка в системе

Обеспечение автозапуска в системе происходит за счет Trojan.Belonard.3. Оказавшись на устройстве, он удаляет Trojan.Belonard.5 и проверяет процесс, в контексте которого работает. Если это не rundll32.exe, он сохраняет в %WINDIR%\System32\ двух других троянцев: Trojan.Belonard.7 (имя файла — WinDHCP.dll) и Trojan.Belonard.6 (davapi.dll). При этом, в отличие от Trojan.Belonard.5, седьмой и шестой хранятся внутри троянца в «разобранном» виде. Тела этих двух троянцев разбиты на блоки по 0xFFFC байт (последний блок может иметь меньший размер). При сохранении на диск троянец склеивает блоки нужным образом для получения рабочих файлов.

Собрав троянцев, Trojan.Belonard.3 создает сервис WinDHCP для запуска WinDHCP.dll (Trojan.Belonard.7) в контексте svchost.exe. В зависимости от языковых настроек ОС, использует тексты на русском или английском для задания параметров службы.

Параметры службы WinDHCP:

- Название службы: «Windows DHCP Service» или «Служба Windows DHCP»;
- Описание: «Windows Dynamic Host Configuration Protocol Service» или «Служба протокола динамической настройки узла Windows»;
- В параметре ImagePath указано: «%SystemRoot%\System32\svchost.exe -k netsvcs», а в ServiceDll указан путь до троянской библиотеки.

После чего Trojan.Belonard.3 регулярно проверяет, запущен ли сервис WinDHCP. Если не запущен, то заново устанавливает его.

Trojan.Belonard.7 представляет собой WinDHCP.dll с экспортом ServiceMain и устанавливается на зараженном устройстве службой с автоматическим стартом. Его задача состоит в том, чтобы проверять в реестре ключа «HKLM\SYSTEM\CurrentControlSet\Services\WinDHCP» параметр «Tag». Если в нем выставлено значение 0, Trojan.Belonard.7 загружает библиотеку davapi.dll (Trojan.Belonard.6) и вызывает ее экспорт, передавая аргументом указатель на структуру SERVICE_STATUS, характеризующую состояние сервиса WinDHCP. Затем ждет 1 секунду и снова проверяет параметр «Tag». Если значение не равно 0, Trojan.Belonard.7 загружает библиотеку spwinres.dll (Trojan.Belonard.4), которая является более старой версией Trojan.Belonard.6. После чего вызывает ее экспорт, передавая аргументом указатель на структуру SERVICE_STATUS, характеризующую состояние сервиса WinDHCP.

Троянец повторяет эти действия каждую секунду.

Параметры службы WinDHCP, взятые из отчета нашего клиента:

```
<RegistryKey Name="WinDHCP" Subkeys="1" Values="11">
<RegistryKey Name="Parameters" Subkeys="0" Values="1">
```

```
<RegistryValue Name="ServiceDll" Type="REG_EXPAND_SZ" SizeInBytes="68"
Value="%SystemRoot%\system32\WinDHCP.dll" />

</RegistryKey>

<RegistryValue Name="Type" Type="REG_DWORD" Value="32" />

<RegistryValue Name="Start" Type="REG_DWORD" Value="2" />

<RegistryValue Name="ErrorControl" Type="REG_DWORD" Value="0" />

<RegistryValue Name="ImagePath" Type="REG_EXPAND_SZ" SizeInBytes="90"
Value="%SystemRoot%\System32\svchost.exe -k netsvcs" />

<RegistryValue Name="DisplayName" Type="REG_SZ" Value="Служба Windows
DHCP" />

<RegistryValue Name="ObjectName" Type="REG_SZ" Value="LocalSystem" />

<RegistryValue Name="Description" Type="REG_SZ" Value="Служба протокола
динамической настройки узла Windows" />

<RegistryValue Name="Tag" Type="REG_DWORD" Value="0" />

<RegistryValue Name="Data" Type="REG_BINARY" SizeInBytes="32"
Value="f0dd5c3aeda155767042fa9f58ade24681af5fbd45d5df9f55a759bd65bc0b7e"
/>

<RegistryValue Name="Scheme" Type="REG_BINARY" SizeInBytes="16"
Value="dcef62f71f8564291226d1628278239e" />

<RegistryValue Name="Info" Type="REG_BINARY" SizeInBytes="32"
Value="55926164986c6020c60ad81b887c616db85f191fda743d470f392bb45975dfeb"
/>

</RegistryKey>
```

Перед запуском всех функций Trojan.Belonard.6 проверяет в реестре службы WinDHCP параметры «Tag» и «Data». В параметре «Data» должен находиться массив байтов, из которых генерируется AES ключ. Если его нет, троянец с помощью библиотеки openssl генерирует 32 случайных байта, которые позднее будут использоваться для генерации ключа шифрования. После этого читает параметры «Info» и «Scheme» службы WinDHCP. В «Scheme» троянец хранит 4 параметра в зашифрованном AES-ом виде. В «Info» хранится SHA256 хэш от списка установленных программ.

Собрав информацию, Trojan.Belonard.6 расшифровывает адрес управляющего сервера — oihcyenw.valve-ms[.]ru — и пытается установить с ним соединение. Если это не удается, троянец использует DGA для генерации доменов в зоне .ru. Однако в коде для генерации доменов допущена ошибка, из-за которой алгоритм создает не те домены, которые хотел получить разработчик троянца.

После отправки зашифрованной информации троянец получает ответ от сервера, расшифровывает его и сохраняет переданные файлы в %WINDIR%\System32\. Среди полученных данных содержатся троянцы wmcodecs.dll (Trojan.Belonard.8) и ssdp32.dll (Trojan.Belonard.9).

Кроме описанных выше действий Trojan.Belonard.6 также вызывает со случайной периодичностью следующие функции:

- поиск запущенных клиентов Counter-Strike 1.6;
- запуск Trojan.Belonard.9;
- обращение к серверу разработчика.

Диапазоны периодов могут быть изменены при получении соответствующей команды от управляющего сервера.

Полезная нагрузка и распространение

Belonard также устанавливается на новых клиентах игры, установленных на устройстве. Эту функцию выполняют Trojan.Belonard.8 и Trojan.Belonard.6.

Trojan.Belonard.8 инициализирует контейнер с информацией об именах файлов клиента Counter-Strike 1.6 и их SHA256-хэшах. Trojan.Belonard.6 начинает искать установленные клиенты игры. Если троянец находит запущенный клиент, он сверяет наличие файлов и их SHA256-хэши с информацией, полученной от Trojan.Belonard.8. Если найдено несоответствие, Trojan.Belonard.8 завершает процесс чистого клиента, после чего добавляет файл hl.exe в директорию игры. Этот файл необходим только для того, чтобы вывести сообщение об ошибочной загрузке игры «Could not load game. Please try again at a later time». Это позволяет троянцу выиграть время, чтобы заменить файлы клиента. Когда это сделано, троянец заменяет hl.exe файл на рабочий, и игра запускается без ошибки.

Троянец удаляет следующие файлы клиента:

```
<path>\\valve\\dlls\\*
<path>\\cstrike\\dlls\\*
<path>\\valve\\cl_dlls\\*
<path>\\cstrike\\cl_dlls\\*
<path>\\cstrike\\resource\\*.res
<path>\\valve\\resource\\*.res
<path>\\valve\\motd.txt
<path>\\cstrike\\resource\\gameui_english.txt
<path>\\cstrike\\resource\\icon_steam.tga
<path>\\valve\\resource\\icon_steam.tga
<path>\\cstrike\\resource\\icon_steam_disabled.tga
<path>\\valve\\resource\\icon_steam_disabled.tga
<path>\\cstrike\\sound\\weapons\\fiveseven_reload_clipin_sliderelease.dll
<path>\\cstrike_russian\\sound\\weapons\
\fiveseven_reload_clipin_sliderelease.dll
<path>\\cstrike_romanian\\sound\\weapons\
\fiveseven_reload_clipin_sliderelease.dll
```

В зависимости от языка ОС загружает файлы русскоязычного или англоязычного игрового меню.

Измененный клиент игры содержит файл троянца Trojan.Belonard.10, а также рекламу ресурсов его разработчика. При входе в игру «ник» игрока изменится на адрес сайта, где можно скачать зараженный клиент игры, а в меню игры появится ссылка на группу

сообщества «ВКонтакте», посвященную CS 1.6 и насчитывающую более 11 500 подписчиков.

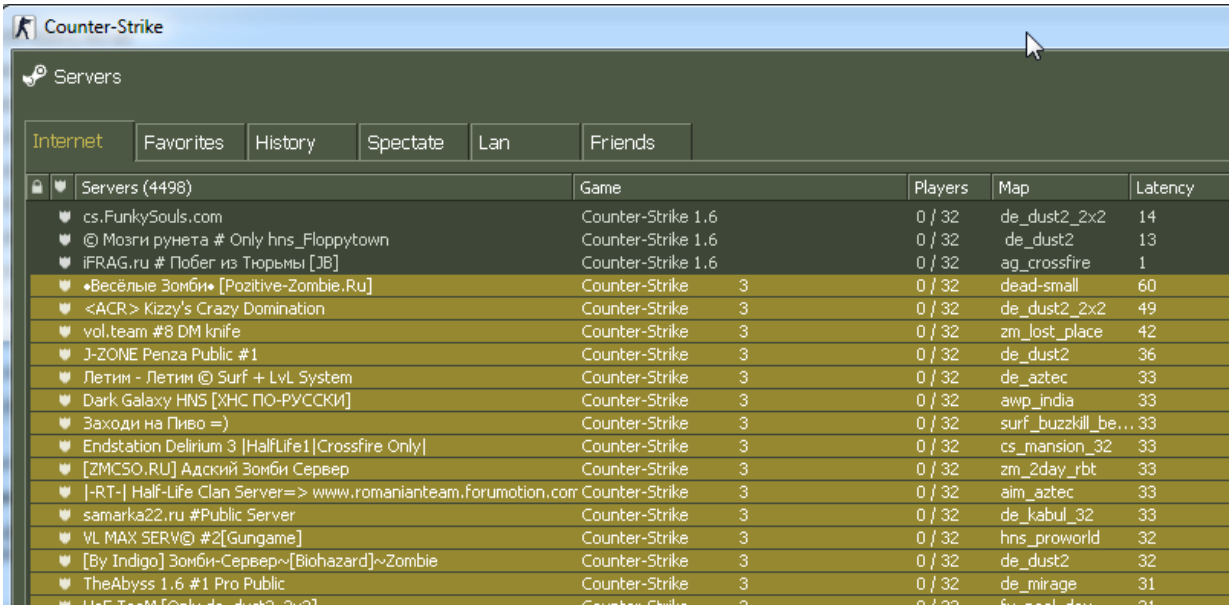
Полезная нагрузка троянца заключается в эмуляции на устройстве пользователя некоторого количества поддельных игровых серверов. Для этого троянец отправляет на сервер разработчика информацию о клиенте игры и получает в ответ в зашифрованном виде параметры для создания поддельных серверов.

```
#pragma pack(push,1)
struct fake_srv_params
{
  _DWORD steamappid;
  _DWORD stamapi_param;
  unsigned __int16 num_of_fake_servers;
  unsigned __int16 game_srv_low_port;
  _DWORD sleep_delay;
  unsigned __int16 fakesrvbatch;
  _DWORD SrvQueryAnsDelay;
  _DWORD rnd_data_update_interval;
  _DWORD min_param_value;
  _DWORD max_param_value;
  unsigned __int8 min_players_on_server;
  unsigned __int8 max_players_on_server;
  unsigned __int8 min_players_on_server_for_naming;
  unsigned __int8 max_players_on_server_for_naming;
  _DWORD min_player_kills;
  _DWORD max_player_kills;
  _DWORD min_player_uptime;
  _DWORD max_player_uptime;
  _DWORD uptimemul;
  _DWORD check_period;
  char szGameName[];
  char szProtocolVersion[];
  char szServerName[];
};
#pragma pack(pop)
```

Trojan.Belonard.9 создает игровые прокси-серверы и регистрирует их через Steam API. Порты игровых серверов берутся последовательно от нижнего значения game_srv_low_port, указанного сервером. Сервер также задает fakesrvbatch, от значения которого зависит количество потоков эмулятора протокола. Эмулятор поддерживает базовые запросы к игровому серверу на движке Goldsource - A2S_INFO, A2S_PLAYER, A2A_PING, получение challenge steam/non-steam клиента, а также команду клиента Counter-Strike «connect». После формирования ответа на команду «connect» троянец отслеживает первый и второй пакеты от клиента.

После обмена пакетами троянец отправляет последний пакет - svc_director с сообщением типа DRC_CMD_STUFFTEXT, которое позволяет выполнить произвольные команды клиента Counter-Strike. Эта проблема известна компании Valve с 2014 года и до сих пор не исправлена. Таким образом, при попытке соединиться с игровым прокси-сервером игрок будет перенаправлен на вредоносный сервер. Это позволит разработчику троянца пытаться эксплуатировать уязвимости игрового клиента пользователя для установки Trojan.Belonard.

Стоит отметить, что в Trojan.Belonard.9 существует баг, позволяющий обнаружить созданные троянцем прокси-серверы. Кроме того, часть прокси-серверов можно определить по названию: в графе «Game» у ненастоящего сервера будет строка вида «Counter-Strike n», где n может являться числом от 1 до 3.



Шифрование

Для хранения данных в троянце и общения с сервером в Belonard используется шифрование. В зашифрованном виде хранится имя управляющего сервера, а также некоторые строки кода и имена библиотек. Для этого используется один алгоритм шифрования с разными константами для отдельных модулей троянца. При этом в старых версиях вредоносной программы использовался другой алгоритм для шифрования строк кода.

Алгоритм дешифровки в Trojan.Belonard.2:

```
def decrypt(d):
  s = ''
  c = ord(d[0])
  for i in range(len(d)-1):
    c = (ord(d[i+1]) + 0xe2*c - 0x2f*ord(d[i]) - 0x58) & 0xff
  s += chr(c)
  return s
```

Старый алгоритм дешифровки:

```
def decrypt(data):
  s = 'f'
  for i in range(0,len(data)-1):
    s += chr((ord(s[i]) + ord(data[i]))&0xff)
  print s
```

Для обмена информацией с управляющем сервером Belonard использует более сложное шифрование. Перед отправкой на сервер информация оборачивается в структуру, уникальную для каждого модуля. Полученные данные шифруются RSA с использованием имеющегося в троянце публичного ключа. Важно заметить, что RSA шифруются только первые 342 байта данных. Если модуль отправляет объем данных, превышающий 342 байта, только первая часть данных будет шифроваться с использованием RSA, а остальная часть шифруется AES. Данные для получения AES-ключа содержатся в блоке, зашифрованном RSA-ключом. В нем же содержатся и данные для генерации AES-ключа, с помощью которого управляющий сервер зашифровывает ответ клиенту.

В начало зашифрованных данных дописывается нулевой байт, после чего они отправляются на управляющий сервер. В ответ сервер отдает зашифрованные данные с указанием размера данных и хэша, что необходимо для сверки с AES-ключом.

Пример данных, полученных от управляющего сервера:

```
#pragma pack(push,1)
struct st_payload
{
_BYTE hash1[32];
_DWORD totalsize;
_BYTE hash2[32];
_DWORD dword44;
_DWORD dword48;
_DWORD dword4c;
_WORD word50;
char payload_name[];
_BYTE payload_sha256[32];
_DWORD payload_size;
_BYTE payload_data[payload_size];
}
#pragma pack(pop)
```

Для расшифровки используется AES в режиме CFB с размером блока 128 бит и ключ, отправленный раннее на сервер. Сначала расшифровываются первые 36 байт данных, из них последний DWORD – это реальный размер полезной нагрузки с заголовком. К AES-ключу добавляется DWORD и хэшируется SHA256. Полученный хэш должен совпасть с первыми 32 расшифрованными байтами. Только после этого расшифровываются остальные принятые данные.

Закрытие ботнета

Для обезвреживания троянца и прекращения работы ботнета наши специалисты приняли ряд мер. При содействии регистратора REG.ru используемые разработчиком троянца домены были сняты с делегирования. Поскольку перенаправление с игровых прокси-серверов происходило по доменному имени, игроки CS 1.6 больше не будут попадать на вредоносный сервер и заражаться троянцем Belonard. Это также нарушило работу практически всех его модулей.

Кроме того, в вирусную базу Dr.Web были добавлены записи для детектирования всех компонентов троянца, а также ведется мониторинг ботов, переключившихся на использование DGA. После принятия всех мер по обезвреживанию ботнета синкхол-сервер зарегистрировал 127 зараженных клиентов. А по данным нашей телеметрии, антивирусом Dr.Web были обнаружены модули троянца Belonard на устройствах 1004 пользователей.

На данный момент ботнет можно считать обезвреженным, но для обеспечения безопасности клиентов Counter-Strike необходимо закрытие существующих уязвимостей со стороны разработчика игры.

Индикаторы компрометации

Хэши файлов

```
8bbc0ebc85648bafdba19369dff39dfbd88bc297 - Backdoored Counter-Strike 1.6
client

200f80df85b7c9b47809b83a4a2f2459cae0dd01 - Backdoored Counter-Strike 1.6
client

8579e4efe29cb999aaedad9122e2c10a50154afb - Backdoored Counter-Strike 1.6
client

ce9f0450dafda6c48580970b7f4e8aea23a7512a - client.dll - Trojan.Belonard.1

75ec1a47404193c1a6a0b1fb61a414b7a2269d08 - Mp3enc.asi - Trojan.Belonard.2

4bdb31d4d410fbbc56bd8dd3308e20a05a5fce45 - Mp3enc.asi - Trojan.Belonard.2

a0ea9b06f4cb548b7b2ea88713bd4316c5e89f32 - Mssv36.asi -
Trojan.Belonard.10

e6f2f408c8d90cd9ed9446b65f4b74f945ead41b - FileSystem.asi -
Trojan.Belonard.11

15879cfa3e5e4463ef15df477ba1717015652497 - Mssv24.asi - Trojan.Belonard.5

4b4da2c0a992d5f7884df6ea9cc0094976c1b4b3 - Mssv24.asi - Trojan.Belonard.5

6813cca586ea1c26cd7e7310985b4b570b920803 - Mssv24.asi - Trojan.Belonard.5

6b03e0dd379965ba76b1c3d2c0a97465329364f2 - Mssv16.asi - Trojan.Belonard.3

2bf76c89467cb7c1b8c0a655609c038ae99368e9 - Mssv16.asi - Trojan.Belonard.3

d37b21fe222237e57bc589542de420fbdaa45804 - Mssv16.asi - Trojan.Belonard.3

72a311bcca1611cf8f5d4d9b4650bc8fead263f1 - Mssv16.asi - Trojan.Belonard.3

73ba54f9272468fbec8b1d0920b3284a197b3915 - davapi.dll - Trojan.Belonard.6

d6f2a7f09d406b4f239efb2d9334551f16b4de16 - davapi.dll - Trojan.Belonard.6

a77d43993ba690fda5c35ebe4ea2770e749de373 - spwinres.dll -
Trojan.Belonard.4

8165872f1dbbb04a2eedf7818e16d8e40c17ce5e - WinDHCP.dll -
Trojan.Belonard.7

027340983694446b0312abcac72585470bf362da - WinDHCP.dll -
Trojan.Belonard.7

93fe587a5a60a380d9a2d5f335d3e17a86c2c0d8 - wmcodecs.dll -
Trojan.Belonard.8

89dfc713cdfd4a8cd958f5f744ca7c6af219e4a4 - wmcodecs.dll -
Trojan.Belonard.8

2420d5ad17b21bedd55309b6d7ff9e30be1a2de1 - ssdp32.dll - Trojan.Belonard.9
```

Имена файлов

```
client.dll - Trojan.Belonard.1

Mp3enc.asi - Trojan.Belonard.2
```

```
Mssv16.asi - Trojan.Belonard.3
spwinres.dll - Trojan.Belonard.4
Mssv24.asi - Trojan.Belonard.5
davapi.dll - Trojan.Belonard.6
WinDHCP.dll - Trojan.Belonard.7
wmcodecs.dll - Trojan.Belonard.8
ssdp32.dll - Trojan.Belonard.9
Mssv36.asi - Trojan.Belonard.10
FileSystem.asi - Trojan.Belonard.11
```

Используемые домены

```
csgoogle.ru
etmpyuuo.csgoogle.ru
jgutdnqn.csgoogle.ru
hl.csgoogle.ru
half-life.su
play.half-life.su
valve-ms.ru
bmeadaut.valve-ms.ru
fuztxhus.valve-ms.ru
ixtzhunk.valve-ms.ru
oihcyenw.valve-ms.ru
suysfvtm.valve-ms.ru
wcnclfbi.valve-ms.ru
reborn.valve-ms.ru
```

IP-адреса

```
37.143.12.3
46.254.17.165
```