АНТИВИРУСНАЯ ЗАЩИТА БИЗНЕС-КЛАССА

Winnti: атака на цепочки поставок – под прицелом азиатские разработчики игр

12 марта 2019 года

Не впервые злоумышленники атакуют игровую индустрию, компрометируют разработчиков, добавляют бэкдоры в среду сборки игр, а затем распространяют вредоносное ПО под видом легитимного. В апреле 2013 года «Лаборатория Касперского» сообщала о подобном инциденте. Эта атака приписана кибергруппе, названной Winnti.

Недавно внимание специалистов ESET привлекли новые атаки на цепочки поставок. Две игры и одна игровая платформа были скомпрометированы для внедрения бэкдора. Эти атаки нацелены на Азию и игровую индустрию, за ними снова стоит группа Winnti.

Три кейса, один бэкдор

Несмотря на разные конфигурации вредоносного ПО, три скомпрометированных программных продукта включали один и тот же код бэкдора и запускались с использованием идентичного механизма. Сейчас в двух продуктах бэкдоров нет, но один все еще распространяется в троянизированной версии – по иронии, эта игра называется Infestation («заражение»), ее выпускает тайская компания Electronics Extreme. Мы пытаемся связаться с разработчиком с начала февраля, но пока безрезультатно.

Разберем, как внедряется вредоносная полезная нагрузка, а затем рассмотрим детально бэкдор.

Внедрение полезной нагрузки

Код полезной нагрузки запускается во время выполнения троянизированного исполняемого файла. Сразу после PE entry point стандартный вызов инициализации C Runtime (`__scrt_common_main_seh` на рисунке ниже) перехватывается для запуска вредоносной полезной нагрузки раньше чего-либо еще (рисунок 2). Это может указывать на то, что злоумышленники изменили конфигурацию сборки, а не сам исходный код.

```
.text:0054BCE5
.text:0054BCE5                  public start
.text:0054BCE5 start            proc near
.text:0054BCE5                  call    ___security_init_cookie
.text:0054BCEA                  jmp     ?__scrt_common_main_seh@@YAHXZ ; __scrt_common_main_seh(void)
.text:0054BCEA start            endp
.text:0054BCEA
.text:0054BCEF ; [00000006 BYTES: COLLAPSED FUNCTION j_nullsub_1. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:0054BCF5 ; [00000015 BYTES: COLLAPSED FUNCTION __EH_epilog3. PRESS CTRL-NUMPAD+ TO EXPAND]
.text:0054BD0A ; [00000011 BYTES: COLLAPSED FUNCTION __EH_epilog3_GS. PRESS CTRL-NUMPAD+ TO EXPAND]
```

Рисунок 1. Чистая точка входа в исполняемый файл

```
.text:00558D85                  public start
.text:00558D85 start            proc near
.text:00558D85                  call    ___security_init_cookie
.text:00558D8A                  jmp     sub_401000
.text:00558D8A start            endp
.text:00558D8A
.text:00558D8F ; [00000006 BYTES: COLLAPSED
.text:00558D95 ; [00000015 BYTES: COLLAPSED
.text:00558DAA ; [00000011 BYTES: COLLAPSED
.text:00558DBB ; [00000011 BYTES: COLLAPSED
.text:00558DCC ; [00000034 BYTES: COLLAPSED
.text:00558E00 ; [00000037 BYTES: COLLAPSED
.text:00558E37 ; [00000037 BYTES: COLLAPSED
.text:00558E6E ; [0000003A BYTES: COLLAPSED
.text:00558EA8                  align 10h
.text:00558EB0 ; [00000046 BYTES: COLLAPSED
.text:00558EF6 ; [00000015 BYTES: COLLAPSED
.text:00558F0B                  align 10h
.text:00558F10 ; [00000049 BYTES: COLLAPSED
.text:00558F59 ; [00000011 BYTES: COLLAPSED
.text:00558F6A                  align 10h

                        assume es:nothing, ss:nothing, ds:_data, fs:nothing, gs:nothing
; =============== S U B R O U T I N E =======================================
; Attributes: library function
sub_401000      proc near               ; CODE XREF: start+5↓j
                                        ; DATA XREF: decrypt_and_launch_payload+16C↓o
                call    decrypt_and_launch_payload
                jmp     ?__scrt_common_main_seh@@YAHXZ ; __scrt_common_main_seh(void)
sub_401000      endp
; ---------------------------------------------------------------------------
                retn
```

Рисунок 2. Точка входа скомпрометированного исполняемого файла

Код, добавленный к исполняемому файлу, расшифровывает и запускает бэкдор в оперативной памяти, прежде чем возобновить нормальное выполнение кода инициализации C Runtime и последующего кода хост-приложения. Встроенная полезная нагрузка имеет особую структуру, показанную на рисунке ниже, которую интерпретирует добавленный код распаковки.

```
bin             db 6, 0, 0Fh, 2, 0Fh, 2, 3, 5, 8 dup(0); rc4_key ; xor(0x37) -> '17858542'
                dd 0C06h                ; added_code_size
                dd 8                    ; rc4_key_size
                dd 0Ch                  ; crypted_filename_size
                dd 18h                  ; crypted_filename_wide_size
                dd 0C600h               ; payload_size
                dd 2                    ; launch_type
filename        db 76h, 12h, 0A7h, 9Ah, 1Eh, 8, 0B3h, 37h, 70h, 0D7h, 26h ; '111.bin.tmp\x00'
                db 0C8h
filename_wide   db 76h, 23h, 0A7h, 0B4h, 4Dh, 61h, 0F3h, 19h, 66h, 0BAh ; u'111.bin.tmp\x00'
                db 3Fh, 0C8h, 0D5h, 2Bh, 31h, 8Ah, 7Ah, 62h, 8Ah, 43h
                db 2Bh, 0E0h, 35h, 79h
pe              db 0Ah, 79h, 6, 0B4h, 7Fh, 61h, 0DDh, 19h, 0, 0BAh, 56h
                db 0C8h, 44h, 0D4h, 1Fh, 8Ah, 0B6h, 62h, 0E7h, 43h, 5Bh
```

Рисунок 3. Структура встроенной полезной нагрузки

В ее составе ключ RC4 (зашифрованный XOR с 0x37), который используется для расшифровки имени файла и встроенного файла DLL.

Вредоносная полезная нагрузка

Фактическая вредоносная полезная нагрузка содержит всего 17 Кбайт кода и данных.

Конфигурация

Данные конфигурации, представленные на рисунке ниже, представляют собой список строк, разделенных пробелами.

```
config          db 'https://nw.infestexe.com/version/last.php 240000 warz wireshark.e'
                                        ; DATA XREF: sub_402460+14↑o
                db 'xe;perfmon.exe;procmon64.exe;procmon.exe;procexp.exe;procexp64.ex'
                db 'e;netmon.exe',0
                db '                                                                  '
                db '                          ',0
```

Рисунок 4. Данные конфигурации полезной нагрузки

Конфигурация состоит из четырех полей:

1. URL управляющего C&C-сервера.
2. Переменная (t), используемая для определения времени ожидания в миллисекундах перед продолжением выполнения. Время ожидания выбирается в диапазоне от 2/3 до 5/3 t случайным образом.
3. Строка, идентифицирующая кампанию.
4. Список имен исполняемых файлов, разделенный точкой с запятой. Если какой-либо из них работает, бэкдор останавливает его выполнение.

Мы идентифицировали пять версий полезной нагрузки:

Усеченный SHA-1	PE время компиляции (UTC)	URL C&C-сервера
a045939f	2018-07-11 15:45:57	https://bugcheck.xigncodeservice[.]com/Common/Lib/Common_bsod.php
a260dcf1	2018-07-11 15:45:57	https://bugcheck.xigncodeservice[.]com/Common/Lib/Common_Include.php
dde82093	2018-07-11 15:45:57	https://bugcheck.xigncodeservice[.]com/Common/Lib/common.php
44260a1d	2018-08-15 10:59:09	https://dump.gxxservice[.]com/common/up/up_base.php
8272c1f4	2018-11-01 13:16:24	https://nw.infestexe[.]com/version/last.php

В первых трех вариантах код не перекомпилирован, но данные конфигурации были отредактированы в самом DLL-файле. Остальная часть содержимого является побайтовой копией.

Инфраструктура C&C

Имена доменов выбраны таким образом, чтобы иметь сходство с сайтами разработчиков

скомпрометированных приложений. Домен верхнего уровня настроен на перенаправление на соответствующий легитимный сайт с помощью сервиса Namecheap, в то время как субдомен указывает на вредоносный C&C-сервер.

Имя домена	Дата регистрации	Цель перенаправления
xigncodeservice.com	2018-07-10 09:18:17	https://namu.wiki/w/XIGNCODE
gxxservice.com	2018-08-14 13:53:41	Нет или неизвестна
infestexe.com	2018-11-07 08:46:44	https://www.facebook.com/infest.in.th

Имя субдомена	IP-адрес	Провайдер
bugcheck.xigncodeservice.com	167.99.106[.]49, 178.128.180[.]206	DigitalOcean
dump.gxxservice.com	142.93.204[.]230	DigitalOcean
nw.infestexe.com	138.68.14[.]195	DigitalOcean

На момент написания поста ни один из доменов не доступен, C&C-серверы не отвечают.

Отчет об исследовании

Идентификатор бота генерируется из MAC-адреса машины. Бэкдор передает на C&C-сервер информацию о машине, включая имя пользователя, имя компьютера, версию Windows и язык системы, а затем ожидает команды. Данные шифруются XOR с помощью ключа “`*&b0i0rong2Y7un1`” и кодируются base64. Данные, полученные с C&C-сервера, шифруются с использованием того же ключа.

Команды

Простой бэкдор поддерживает только четыре команды, которые могут использоваться атакующими:

— `DownUrlFile`
— `DownRunUrlFile`
— `RunUrlBinInMem`
— `UnInstall`

Названия команд говорят сами за себя. Они позволяют атакующим запускать дополнительные исполняемые файлы с заданного URL.

Возможно, последняя команда менее очевидна. `UnInstall` не удаляет вредоносную программу из системы. В конце концов, она встроена в легитимный исполняемый файл, который еще будет запускаться. Вместо удаления чего-либо, команда отключает вредоносный код, задавая значение 1

для ключа реестра:

```
HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\ImageFlag
```

После запуска полезной нагрузки запрашивается значение из реестра и, если оно задано, выполнение отменяется. Возможно, атакующие пытаются снизить нагрузку на свои C&C-серверы, избегая обратных вызовов от жертв, не представляющих интереса.

Второй этап

По данным телеметрии, одна из полезных нагрузок второго этапа, отправляемых жертвам, Win64/Winnti.BN. Насколько мы можем судить, дроппер этой вредоносной программы загружается через HTTPS с `api.goallbandungtravel[.]com`. Мы видели, что он устанавливался как служба Windows и как DLL в `C:\Windows\System32`, используя следующие имена файла:

— cscsrv.dll
— dwmsvc.dll
— iassrv.dll
— mprsvc.dll
— nlasrv.dll
— powfsvc.dll
— racsvc.dll
— slcsvc.dll
— snmpsvc.dll
— sspisvc.dll

Образцы, которые мы проанализировали, были достаточно большими – около 60 Мбайт. Однако это только видимость, поскольку реальный размер или PE файл – от 63 до 72 Кбайт, в зависимости от версии. К вредоносным файлам просто добавляется множество чистых. Вероятно, это делает компонент, сбрасывающий и устанавливающий вредоносную службу.

После запуска службы она добавляет к своему пути DLL расширение `.mui` и расшифровывает его с помощью RC5. Расшифрованный файл MUI содержит независимый от позиции код со смещением 0. Ключ RC5 получен из серийного номера жесткого диска и строки “`f@Ukd!rCto R$.`”. Мы не смогли получить ни файлы MUI, ни код, который их устанавливает в первую очередь. Таким образом, точное назначение вредоносного сервиса нам неизвестно.

Последние версии вредоносного ПО включают механизм автообновления с использованием C&C-сервера `http://checkin.travelsanignacio[.]com`. Этот сервер обслуживал последнюю версию файлов MUI, зашифрованных статическим ключом RC5. В ходе нашего исследования этот C&C-сервер не отвечал.

Цели

Начнем с тех, на кого точно **не** нацелена кампания. В начале полезной нагрузки вредоносная программа проверяет, является ли язык системы русским или китайским (см. на рисунке ниже). В случае положительного ответа программа прекращает работу. Обойти это исключение невозможно – атакующим принципиально не интересны компьютеры с этими языковыми настройками.

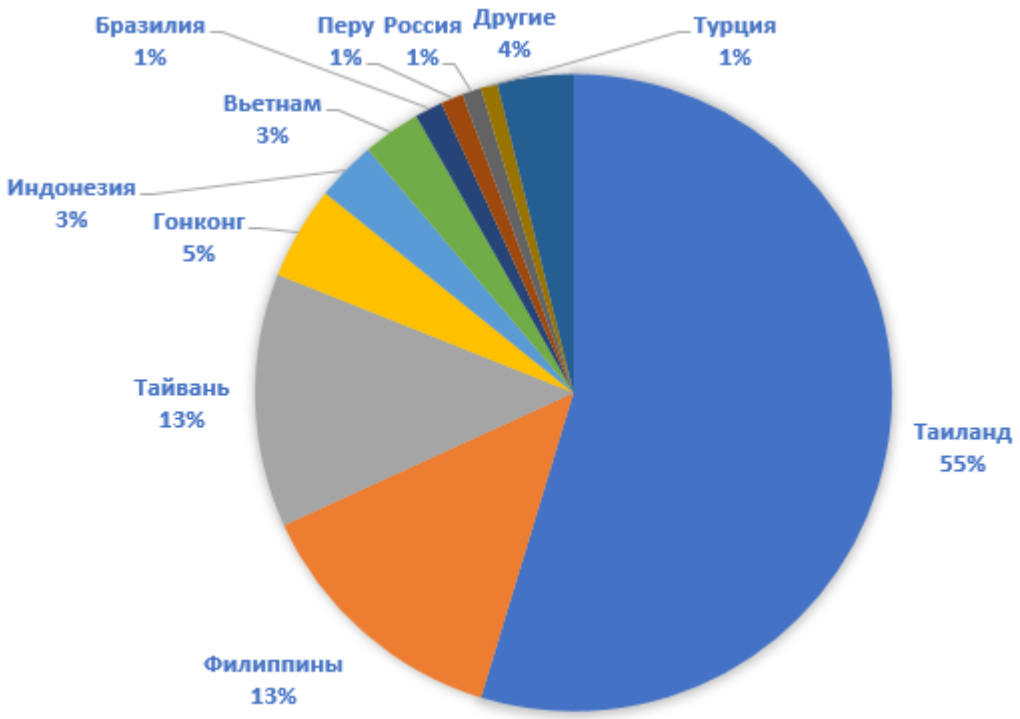
```
BOOL __stdcall DllMain(HINSTANCE hinstDLL, DWORD fdwReason, LPVOID lpvReserved)
{
  LANGID lang; // ax
  HANDLE v4; // eax
  struct WSAData WSAData; // [esp+0h] [ebp-190h]

  if ( fdwReason == DLL_PROCESS_ATTACH )
  {
    Enable_SeDebugPrivilege();
    WSAStartup(0x202u, &WSAData);
    lang = GetSystemDefaultLangID();
    if ( lang != LANG_RUSSIAN && lang != 0x804 && !is_uninstalled() )
                                                // 0x804 = MAKELANGID(LANG_CHINESE, SUBLANG_CHINESE_SIMPLIFIED)
    {
      v4 = CreateThread(0, 0, start_payload, 0, 0, 0);
      CloseHandle(v4);
    }
  }
  return 1;
}
```

Рисунок 5. Проверка языка до запуска полезной нагрузки

Статистика распространения

Согласно телеметрии, большинство заражений пришлось на Азию, прежде всего, Таиланд. С учетом популярности скомпрометированного приложения, которое все еще распространяется разработчиком, неудивительно, если число жертв исчисляется десятками и сотнями тысяч.



Вывод

Атаки на цепочки поставок сложно обнаружить на стороне пользователя. Невозможно проанализировать весь запускаемый софт, а также все рекомендуемые обновления. Пользователь по умолчанию доверяет разработчикам и предполагает, что их файлы не содержат вредоносного кода. Вероятно, именно поэтому несколько кибергрупп нацеливают атаки на поставщиков ПО – компрометация позволит создать ботнет, размер которого сопоставим с популярностью троянизированного софта. У этой тактики есть и обратная сторона – когда схема будет раскрыта, атакующие потеряют контроль над ботнетом, и пользователи смогут очистить систему, установив очередное обновление.

Мотивы кибергруппы Winnti в настоящий момент неизвестны. Возможно, атакующие ищут финансовую выгоду или же планируют использовать ботнет как часть более масштабной операции.

Продукты ESET детектируют угрозу как Win32/HackedApp.Winnti.A, Win32/HackedApp.Winnti.B, полезную нагрузку – как Win32/Winnti.AG, второй этап – как Win64/Winnti.BN.

Индикаторы компрометации доступны по ссылке.