

FlokiBot 銀行木馬詳細分析（二）

























2017-03-28 由 看雪學院 發表



✿ 用 IDAPython 完全靜態去混淆

→ 鑑別函數

首先，我們注意到 dropper 重用了一些 payload 的重要函數。創造該 dropper 的一個 Rizzo 簽名並在 payload 中加載它能夠 IDA 讓識別並重命名少部分函數。

Name	Address
 allocate_mem	0040FA0A
 free_heap	0040FA6C
 qmemcpy	0040FB03
 cstm_memset	0040FB77
 get_length	004105EF
 get_filename_len	00410601
 crc	00410F68
 security_descr	004131C1
 check_pe_header	00414302
 unhook_api	00414320
 open_ntdll	004144C5
 map_ntdll_unhook	004144FC
 extract_syscall	00414567
 get_modules_peb	004167E7
 unhook	00416AD2
 is_64b_process	00417475
 resolve_syscall	00417497
 int0x2e	004174A7
 NtCreateSection_wrap	004174B4
 NtMapViewOfSection_wrap	004174C3
 syscall_4	004174E1
 inj_CreateSection	0041760D
 inj_MapViewOfSection	00417697
 change_perm	0041775D

找不到想看的？搜尋看看！

✿ API 調用和鉤子的靜態去混淆

思路是用 Python 重新實現哈希過程，哈希所有被 FlokiBot 加載的所有 API，然後將他們和我們用代碼收集到的哈希值進行比較。如果匹配，我們就用 IDAPython 重命名該函數，使得反彙編更具可讀性。因為 payload 用的是同樣的 CRC 函數和同樣的異或密鑰，所以這個腳本對它們都管用。



→ 字符串去混淆

跟 ZeuS 和 Fobber (Tinaba 的進化版) 一樣，很多字符串都用它們自己的一字節的密鑰異或加密了。惡意軟體將所有的 ENCRYPTED_STRING 存儲在一個數組中，並將在傳輸過程中通過下標去混淆。加密過的字符串將以下面的數據結構展現：

```
typedef struct {
    char xor_key;

    WORD size;

    void* strEncrypted;
} ENCRYPTED_STRING;
```

首先，為弄明白如何沒有錯誤的檢索出它們，我會運行一段代碼羅列 decrypt_string 的參數是如何入棧的。

運行完我們的腳本後，這裡有一個在 IDA 中反彙編後的樣本：

```
call    decrypt_string ; Start Page
push    17h
lea     esi, [ebp+78h+var_A4]
pop     eax
call    decrypt_string ; Software\Microsoft\Internet Explorer\Main
mov     ecx, edi
call    cstm_strlen
lea     eax, [eax+eax+2]
push    eax
push    edi
push    1
lea     eax, [ebp+78h+var_2C]
push    eax
mov     eax, esi
push    eax
push    ebx
call    cstm_RegSetValueExW

loc_40E03C:
push    19h
lea     esi, [ebp+78h+var_B8]
pop     eax
call    decrypt_string ; Software\Microsoft\Internet Explorer\PhishingFilter
push    1Ah
lea     esi, [ebp+78h+var_28]
pop     eax
call    decrypt_string ; Enabled
push    1Bh
lea     esi, [ebp+78h+var_4C]
pop     eax
call    decrypt_string ; EnabledV8
```

找不到想看的？搜尋看看！



→ 完整的 IDAPython 腳本

這是我用來去混淆該 payload 的完整的 Python 腳本：

<https://gist.github.com/adelfmas/8c864315648a21ddabbd6bc7e0b64119>.

它基於 IDAPython 和 PeFile。它專為靜態分析設計，你不用開啟任何 debugger 來讓這段程序工作。它將完成以下的工作：

- 明確bot引入的所有函數並以[API name]_wrap 的格式重命名它們。
- 解析WINAPIHOOK 結構並以hook_[API name] 的格式重命名鉤子函數。
- 解密字符串並將解密後的值放在解密字符串函數調用處的注釋中。

找不到想看的？搜尋看看！

```
# coding: utf-8

# ===== #

# #

# FLOKIBOT BOT32 DEOBFUSCATION IDA SCRIPT #

# #

# http://adelmas.com/blog/flokibot.php #

# #

# ===== #

# IDAPython script to deobfuscate statically the bot32 payload of the
banking malware FlokiBot.

# Imports are fully resolved, hooks are identified and named and strings
are decrypted and added in comments, without using any debugger.

# May take a few minutes to resolve imports.

# Works with FlokiBot dropper with some small changes.

import sys

# sys.path.append("/usr/local/lib/python2.7/dist-packages")

# idaapi.enable_extlang_python(True)

import pefile

# RunPlugin("python", 3)

CRC_POLY = 0xEDB88320 # Depending on sample

XOR_KEY = 0x34ED # Depending on sample

ARRAY_ADDR = 0x41B350 # Depending on sample

ARRAY_ITER = 12 # Size of a triplet (3*sizeof(DWORD))

i = 0

# -----
..... (代碼省略)
```

找不到想看的？搜尋看看！



→ 持久性

bot 用一個偽隨機名字把自己複製到 C:\Documents and Settings\
[username]\Application Data 並通過在 Windows 的啟動文件夾創建一個 .lnk
來獲得持久性。

找不到想看的？搜尋看看！

```

int startup_Ink() {

int v0; // edi@1

_WORD *v1; // ecx@1

int v2; // eax@2

_WORD *v3; // ecx@2

const void *v4; // eax@2

const void *v5; // esi@3

int strStartupFolder; // [sp+8h] [bp-20Ch]@1

int v8; // [sp+210h] [bp-4h]@6

v0 = 0;

SHGetFolderPathW_wrap(0, 7, 0, 0, &strStartupFolder); // 7 =
CSIDL_STARTUP

v1 = (_WORD *)PathFindFileNameW_wrap(&pFilename);

if ( v1 && (v2 = cstm_strlen(v1), sub_40FECB(v2 - 4, v3), v4) )

v5 = v4;

else

v5 = 0;

if ( v5 ) {

v8 = 0;

if ( build_Ink((int)&v8, (const char *)L"%s\\%s.Ink", &strStartupFolder, v5) >
0 )

v0 = v8;

cstm_FreeHeap(v5);

}

return v0;

}

```

找不到想看的？搜尋看看！

🌸 掛鉤API

→ 概述

基於Zeus，FlokiBot 用了同一種但又有些許不同的結構數組來存儲它的鉤子：

```
typedef struct
{
    void *functionForHook;
    void *hookerFunction;
    void *originalFunction;
    DWORD originalFunctionSize;
    DWORD dllHash;
    DWORD apiHash;
} HOOKWINAPI;
```

在我們運行完前面用來去混淆 API 調用的腳本，以及定位好鉤子結構數組之後，我們就可以很輕易的用其他的 IDA 腳本來解析它，以確定和命名鉤子函數（hook_*）。我們最後得到下面的表格：

```
Parsing hook table @ 0x41B000... Original Function Hooked Hooker
Function DLL Hash API Hash -----
----- NtProtectVirtualMemory_wrap
hook_NtProtectVirtualMemory_wrap 84C06AAD (ntdll) 5C2D2E7A
NtResumeThread_wrap hook_NtResumeThread_wrap 84C06AAD (ntdll)
6273819F LdrLoadDll_wrap hook_LdrLoadDll_wrap 84C06AAD (ntdll)
18364D1F NtQueryVirtualMemory_wrap
hook_NtQueryVirtualMemory_wrap 84C06AAD (ntdll) 03F6C761
NtFreeVirtualMemory_wrap hook_NtFreeVirtualMemory_wrap 84C06AAD
(ntdll) E9D6FAB3 NtAllocateVirtualMemory_wrap
hook_NtAllocateVirtualMemory_wrap 84C06AAD
.....代碼省略
```

找不到想看的？搜尋看看！

它們中的大多數都有安裝在 ZeuS 和其他銀行惡意軟體中。儘管如此，我們還是能夠注意到 NtFreeVirtualMemory 和 NtProtectVirtualMemory 的一些有趣的、新的鉤子。我們將在下一部分看到它們的用途。

→ 瀏覽器中間人 (Man-in-the-Browser)

Floki 通過把自己注入到 Firefox 和 Chrome 進程中並攔截 LdrLoadDll 來實現瀏覽器中間人攻擊。如果瀏覽器加載的 DLL 的哈希值和 nss3.dll, nspr4.dll 或 chrome.dll 任一個的哈希值匹配，API 鉤子就會自動安裝，讓惡意軟體可以實現表單抓取和網站注入。

```
int __stdcall hook_LdrLoadDll_wrap(int PathToFile, int Flags, int
ModuleFileName, int *ModuleHandle)

{

int result; // eax@2

int filename_len; // eax@8

int dll_hash; // eax@8

[...]

if ( cstm_WaitForSingleObject() ) {

v5 = LdrGetDllHandle_wrap(PathToFile, 0, ModuleFileName,
ModuleHandle);

v6 = LdrLoadDll_wrap(PathToFile, Flags, ModuleFileName,
ModuleHandle);

v12 = v6;

if ( v5 < 0 && v6 >= 0 && ModuleHandle && *ModuleHandle &&
ModuleFileName )

{

RtlEnterCriticalSection_wrap(&unk_41D9F4);

filename_len = cstm_strlen((_WORD **)(ModuleFileName + 4));

dll_hash = hash_filename(filename_len, v8);

if ( !(dword_41DA0C & 1) ) {

if ( dll_hash == 0x2C2B3C88 || dll_hash == 0x948B9CAB ) { // hash
nss3.dll & nspr4.dll

sub_416DBD(*ModuleHandle, dll_hash);

if ( dword_41DC2C )

v11 = setNspr4Hooks(v10, dword_41DC2C);
```

找不到想看的？搜尋看看！


```
}

else if ( dll_hash == 0xCAAD3C25 ) { // hash chrome.dll

if ( byte_41B2CC ) {

if ( setChromeHooks() )

dword_41DA0C |= 2u;

}

[... ]

}

else

{

result = LdrLoadDll_wrap(PathToFile, Flags, ModuleFileName,
ModuleHandle);

}

return result;

}
```

→ 證書竊取

通過掛鉤 `PFXImportCertStore`，FlokiBot 可以竊取數字證書。此法 Zeus 和 Carberp 也有用到。

→ 保護鉤子

FlokiBot 通過放置一個鉤子和過濾 `NtProtectVirtualMemory` 調用來保護它的鉤子，以防止它們被累死殺毒軟體復位到原函數中。無論何時，當一個程序想要改變Floki已經注入的進程的內存保護機制的時候，Floki會阻斷該調用並返回 `STATUS_ACCESS_DENIED`。

找不到想看的？搜尋看看！

```

unsigned int __stdcall hook_NtProtectVirtualMemory_wrap(void
*ProcessHandle, int *BaseAddress, int NumberOfBytesToProtect, int
NewAccessProtection, int OldAccessProtection)

{

int retBaseAddress; // [sp+18h] [bp+Ch]@7

[...]

v11 = 0;

v5 = BaseAddress;

if ( cstm_WaitForSingleObject() && BaseAddress && ProcessHandle ==
GetCurrentProcess() )

{

if ( check_base_addr(*BaseAddress) )

return 0xC0000022; // STATUS_ACCESS_DENIED

RtlEnterCriticalSection_wrap(&unk_41E6E8);

v11 = 1;

}

retBaseAddress = NtProtectVirtualMemory_wrap(

ProcessHandle,

BaseAddress,

NumberOfBytesToProtect,

NewAccessProtection,

OldAccessProtection);

[...]

LABEL_18:

if ( v11 )

RtlLeaveCriticalSection_wrap(&unk_41E6E8);

return retBaseAddress;

}

```

找不到想看的？搜尋看看！

→ **PoS**惡意軟體特徵：內存截取

在我的前一篇文章中，我逆向了一款非常基礎的叫做 TreasureHunter 的 PoS 惡意軟體。它主要用內存截取為主要手段來竊取主帳號（PAN）。

像大多數PoS惡意軟體，FlokiBot 通過定期讀取進程內存來搜索 track2 PAN 。顯然，這並不是很有效，因為你不能時刻監測內存，這樣就會漏掉很多潛在的 PAN。為克服這個問題，在 Floki 把自己注入到某一個進程後，它會放置一個鉤子到 NtFreeVirtualMemory 中，這樣當該進程想要釋放一大塊內存的時候它就可以提前搜尋 track2 PAN 。用這種方法，它就不太可能會錯失PAN。

```
int __stdcall hook_NtFreeVirtualMemory_wrap(HANDLE ProcessHandle,
PVOID *BaseAddress, PSIZE_T RegionSize, ULONG FreeType)

{

PVOID v4; // ebx@1

int v5; // edi@3

RtlEnterCriticalSection_wrap(&unk_41E6E8);

v4 = 0;

if ( BaseAddress )

v4 = *BaseAddress;

v5 = NtFreeVirtualMemory_wrap(ProcessHandle, BaseAddress,
RegionSize, FreeType);

if ( v5 >= 0 && !dword_41E6A8 && ProcessHandle == (HANDLE)-1 &&
cstm_WaitForSingleObject() )

trigger_ram_scraping((int)v4);

RtlLeaveCriticalSection_wrap(&unk_41E6E8);

return v5;

}
```

當 Floki 發現 track2 數據，它就會通過查看 PAN 的開頭來確定發行方。在這個飽含信息量的網頁，你可以找到一系列發行方的識別號：

找不到想看的？搜尋看看！

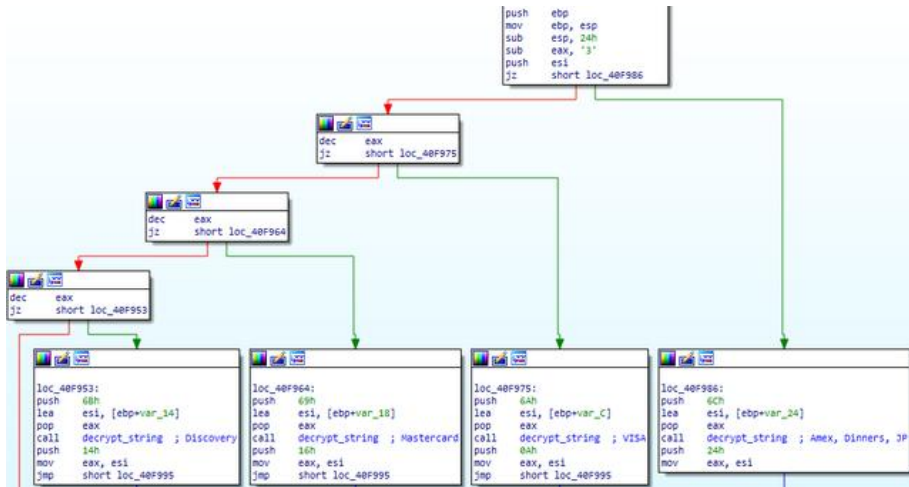
http://www.stevemorse.org/ssn/List_of_Bank_Identification_Numbers.html.

Floki 並沒有查看整個IIN (6 位)，而是只檢查了第一位看它是否符合下面的發行方：

- 3： Amex / Dinners / JP
- 4： VISA
- 5： Mastercard

■ 6 : Discover

FlokiBot identify_mii 流程：



然後，它根據 Luhn 算法查看 PAN 是否有效：

找不到想看的？搜尋看看！

```

char __usercall check_mii_luhn@<al>(void *a1@<ecx>, _BYTE
*a2@<esi>)

{

char result; // al@1

[...]

result = identify_mii(*a2, a1);

if ( result )

{

v7 = 0; v3 = 1; v8 = 2;

v9 = 4; v10 = 6; v11 = 8;

v12 = 1; v13 = 3; v14 = 5;

v15 = 7; v16 = 9; v4 = 0; v5 = 16;

do // Luhn Algorithm

{

v6 = a2[--v5] - '0';

if ( !v3 )

v6 = *(&v7 + v6);

v4 += v6;

v3 = v3 == 0;

}

while ( v5 );

result = v4 % 10 == 0;

}

return result;

}

```

找不到想看的？搜尋看看！

→ 通訊

通訊是用 RC4 和異或混合加密的。我們用來去混淆字符串的代碼可以幫我們識別下面這些明確命名的命令行：

```

user_flashplayer_remove user_flashplayer_get
user_homepage_setuser_url_unblock user_url_block user_certs_remove
user_certs_get user_cookies_remove user_cookies_get user_execute
user_logoff user_destroy fs_search_remove fs_search_add fs_path_get
bot_ddos_stop bot_ddos_start
bot_httpinject_enablebot_httpinject_disablebot_bc_remove bot_bc_add
bot_update_exe bot_update bot_uninstall os_reboot os_shutdown

```

現在 FlokiBot 還沒有只是 TOR，但你可以從代碼中找到這個特徵的一些痕跡。

→ 激活遠程桌面協議 (RDP)

這個 payload 想要通過寄存器來手動激活遠程 Windows 桌面，然後執行控制台命令添加一個隱形的管理員帳號 `test_account:test_password`。

```

push esi
push edi
call enable_remote_desktop
push 6
pop ecx
mov esi, offset aTest_account ; "test_account"
lea edi, [ebp+74h+login]
rep movsd
push 7
pop ecx
lea eax, [ebp+74h+passwd]
movsw
push eax ; strPassword
lea eax, [ebp+74h+login]
mov esi, offset aTest_password ; "test_password"
lea edi, [ebp+74h+passwd]
push eax ; strLogin
rep movsd
call cmd_add_user
pop edi
pop esi
test al, al
jz short loc_409D7D

```

`enable_remote_desktop` 函數的偽碼：

```

void enable_remote_desktop()
{
    signed int v0; // eax@3

    int v1; // [sp+0h] [bp-Ch]@2

    int v2; // [sp+4h] [bp-8h]@2

    int v3; // [sp+8h] [bp-4h]@2

    if ( byte_41E43C ) {

        v2 = 0;

        v1 = 4;

        v3 = 0x80000002;

        if ( RegOpenKeyExW_wrap(0x80000002,

```

找不到想看的？搜尋看看！

```

L"SYSTEM\\CurrentControlSet\\Control\\Terminal Server", 0, 1, &v3) )

v0 = -1;

else

v0 = cstm_RegQueryValueExW(&v3, (int)L"fDenyTSConnections",
(int)&v1, (int)&v2, 4);

if ( v0 != -1 ) {

if ( v2 ) {

v3 = 0; // 0 = Enables remote desktop connections

cstm_RegSetValueExW(

0x80000002,

(int)L"SYSTEM\\CurrentControlSet\\Control\\Terminal Server",

(int)L"fDenyTSConnections",

4,

(int)&v3,

4);

}

}

}

}

```

自從 ATS 這種方式因為太複雜而不能編程以及太難部署後，使用遠程桌面進行網絡犯罪成為了新的方式。通過這種方式，它們可以獲取被感染的電腦的所有權限，從而獲得目標的信息，並執行欺詐任務，例如手動轉移錢財。

✿ 最後需要注意的和哈希值

找不到想看的？搜尋看看！

FlokiBot 是又一基於 Zeus 的惡意軟體，有些代碼甚至是直接從 Carberp 拿來的。雖然如此，它的解除掛鉤操作和 PoS 惡意軟體特徵都很有趣，值得分析。而且，它的混淆技術很簡單，可以不用 AppCall，只用 IDA 腳本就可以進行靜態分析。

針對最近的 FlokiBot 樣本，@v0id_hunter 上傳了下面這些 SHA256。

```

23E8B7D0F9C7391825677C3F13FD2642885F6134636E475A3924BA5BDD·
997841515222dbfa65d1aea79e9e6a89a0142819eaec3467c31fa169e57076·
f778ca5942d3b762367be1fd85cf7add557d26794fad187c4511b3318aff5cfd....

```

感謝閱讀。

閱讀原文<http://bbs.pediy.com/thread-216639.htm>，查看完整代碼~

本文由 看雪翻譯小組 lumou 編譯，來源 Arnaud Delmas

♥ 往期熱門內容推薦

- 滲透測試 Node.js 應用
- TI（德州儀器）TMS320C674x 逆向分析方法
- Firefox 中一個 Cross-mmap 溢出的利用
- UPDATE 查詢中的 SQL 注入
- 繞過補丁實現欺騙地址欄和惡意軟體警告
- FlokiBot 銀行木馬詳細分析

看雪論壇：<http://bbs.pediy.com/>

微信公眾號 ID：ikanxue

微博：看雪安全

投稿、合作：www.kanxue.com



相關文章



這個網址。

主頁被全改成hao123後的處理辦法

2017-02-20

拿谷歌瀏覽器來舉例！Chrome主頁被全改成hao123後的處理辦法最近不知何故瀏覽器的Chrome的主頁被篡改為了hao123。每次第一次打開，都自動跳轉到https://www.hao123.com/?tn=97175858_hao_pg

找不到想看的？搜尋看看！

PHP開發程序應該注意的42個優化準則

2016-11-15

PHP 獨特的語法混合了 C、Java、Perl 以及 PHP 自創新的語法。它可以比 CGI或者Perl更快速的執行動態網頁。用PHP做出的動態頁面與其他的程式語言相比，PHP是將程序嵌入到HTML文檔中去執行，執行效率比完全生成HTML標記的CGI要高許多。

30個技巧提高你的PHP網站程序執行效率

2016-08-05

PHP（外文名: Hypertext Preprocessor，中文名：「超文本預處理器」）是一種通用開源腳本語言。語法吸收了C語言、Java和Perl的特點，易於學習，使用廣泛，主要適用於web開發領域。

1、\$row['id'] 的速度是\$row的7倍。



FlokiBot 銀行木馬詳細分析

2017-03-27

原文首發：<http://bbs.pediy.com/thread-216639.htm> 介紹FlokiBot是最近一款針對於歐洲和巴西聯邦共和國的銀行木馬，作為一款惡意軟體工具集，它在一些黑客論壇上被賣到\$1000。它通過垃圾郵件和

滲透代碼工具包來傳播。

修改源碼實現全局(無需root)注入躲開注入檢測

2016-12-15

看這篇文章需要的技能1.會編譯android源碼(如果你不願意編譯源碼，還有另外一種辦法,下面我會提供)2.

PHP高手必須要掌握的40個重點

2016-08-09

1、儘量採用大量的PHP內置函數。2、如果能將類的方法定義成static，就儘量定義成static，它的速度會提升將近4倍。3、\$row['id'] 的速度是\$row[id]的7倍。

絕對精華，Python學習筆記之Python執行環境

2016-07-24

1、可調用對象 許多Python對象都是可調用的，即任何能通過函數操作符「()」來調用的對象。Python有4種可調用對象：函數、方法、類以及一些類實例，這些對象的任何引用或者別名都是可調用的。



關於JavaScript你可能不知道的7個功能

2016-10-14

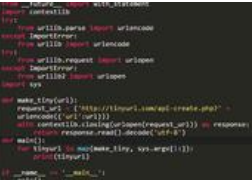
技術領域總是充滿著神秘的未知和挑戰，有趣又令人不能自拔。就像JavaScript，即使是每天使用它進行開發交互的開發人員，而語言的某些部分仍然未被開發。本篇文章中我列舉了關於JavaScript的7件事，可能都是你所不了解的，一起看下吧。



解密一個反殺毒惡意驅動

2017-01-17

翻譯：胖胖秦預估稿費：200RMB投稿方式：發送郵件至linwei#360.cn，或登陸網頁版在線投稿前言IBM X-Force安全研究團隊在調查一起針對巴西銀行的遠程惡意軟體攻擊事件中，發現了一個惡意的反殺毒驅動，它作為惡意金融軟體的一部分。



用python製作url短鏈

2016-09-03

Python部落(python.freelycode.com)組織翻譯，禁止轉載，歡迎轉發。Hi，夥計們！今天這篇文章中我要向你展示我們如何利用python來使複雜的url變得苗條。