BROWSER ATTACK POINTS STILL ABUSED BY BANKING TROJANS

Peter Kálnai
Malware Researcher

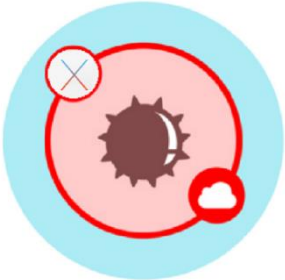
peter.kalnai@eset.cz

Michal Poslušný
Malware Analyst

michal.poslusny@eset.cz

Outline

Man-in-the-browser attacks



Banking Trojans

browserhooks for Volatility Framework

MitB Attacks

“Web browsers are not able to defend against the attacks by their own means.”

Web Browsers

Web Browsers -

2005
MitB PoC

2007
The name by P. Gühring

2010
Firefox attacked in-the-wild

2015
Opera/Chrome switch to BoringSSL

2011
Chrome attacked in-the-wild

2015
HTTP/2 introduced

eset

FSOCIETY

features

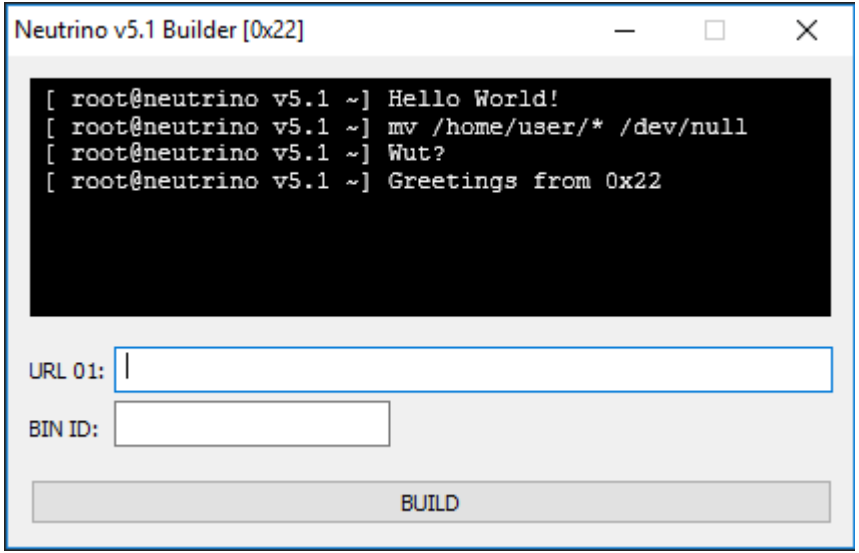
- ✔ Retained Startup (UAC Privilege Escalation Retention)
- ✔ HTTP / 2 Huffman Decoder
- ✔ SPDY Support
- ✔ QUIC Support (for google services)
- ✔ PE Injection
- ✔ Thread Safe

```
(Buf,4)
ameLength -= 1;
FrameLength -= (INT)(HTTP2READUC
Offset += 1;

if ((HTTP2READUCHAR(Buf,4) & 32))

ength -= 5;
```

prices

$250 /bin

or $2500 for builder + reseller rights



- Supported browsers
  - Chrome (x86), x64 Coming next update!
  - FireFox (x86), x64 Coming next update!
  - Microsoft Edge coming next update! x64 & x86
- Supported Protocols (All Browsers Specified Above)
  - SPDY
  - HTTP / 2
  - SSL
  - HTTP / 1.1
  - QUIC
- HUFFMAN Decoder for HTTP / 2 (Plaintext headers!)

Attackers' goals

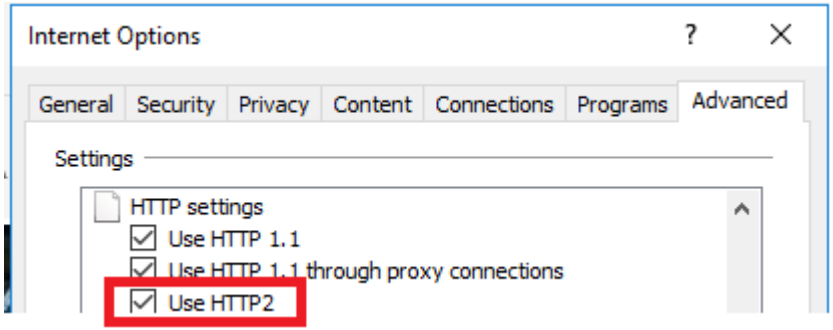
1) Locating a browser's process memory
2) Injecting a payload
3) Locating the attack points
   - Chromium-based projects only (SSL VMT)
4) Installing hooks
   - Inline hooks pointing to the payload

eset

Attack points

HttpOpenRequest
InternetReadFile
HttpSendRequest
InternetWriteFile
…

Disable HTTP2:



my.eset.com

login.eset.com/lo

my eset

Sign in

Enter your email ad

Email address: michal.poslusny@eset.cz

Password:

Forgot password?

You have entered an incorrect email/password combination.

Sign in or Create new account

C:\Users\IEUser\Desktop\EdgeInjector.exe

```
Initiailzing socket server...
Looking for edge processes...
3 processes found
PID[6872]: Injected!
PID[5988]: Injected!
PID[6872]: HttpSendRequestWHook[dataLen=9671]

PID[6872]: LoginEmail=michal.poslusny%40eset.cz&ctl00%24body%24LoginCtr%24
txtLoginPassword=MySecurePassword&ctl00%24body%24LoginCtr%24btnLogin=Sign+in
```

MicrosoftEdgeCP.exe (6872) Properties

Memory | Environment | Handles | Job
General | Statistics | Performance | Threads

File

Microsoft Edge Content Process

(Verified) Microsoft Corporation

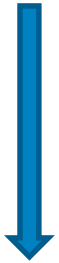
Mitigation Policies

Policy
ASLR (high entropy, force relocate)
CF Guard
DEP (permanent)
Dynamic code prohibited
Images restricted (remote images)
Signatures restricted (Store only)
Strict handle checks

Attack points

nspr4dll!PR_Read
nspr4dll!PR_Write
nss3!PR_Read
nss3!PR_Write

Disable HTTP2:



```
firefoxPref network.http.spdy.enabled false
firefoxPref network.http.spdy.enabled.http2 false
firefoxPref network.http.spdy.enabled.v3-1 false
```

SSL VMT: Attack points

[chromium] // src / third_party / boringssl / src / ssl / tls_method.cc

tls_method.cc

```
109 }
110
111 static const SSL_PROTOCOL_METHOD kTLSProtocolMethod = {
112     0 /* is_dtls */,
113     ssl3_new,
114     ssl3_free,
115     ssl3_get_message,
116     ssl3_read_message,
117     ssl3_next_message,
118     ssl3_read_app_data,
119     ssl3_read_change_cipher_spec,
120     ssl3_read_close_notify,
121     ssl3_write_app_data,
122     ssl3_dispatch_alert,
123     ssl3_supports_cipher,
```

Disable HTTP2:

↓

`--disable-http2`
`(--use-spdy=off)`

Banking Trojans

“Gangs behind banking bots are persistent in their implementations of MiTBs”

Win/PSW.Papras (2013)

```
if ( result )
{
  switch ( browserType )
  {
    case INTERNET_EXPLORER:
      return HookIE();
    case FIREFOX:
      return HookFirefox();
    case CHROME:
      WSACleanup();
      dword_1001C00C = CreateThread(0, 0, ChromeThreadProc, 0, 0, 0);
      result = dword_1001C00C != 0;
      break;
  }
  return result;
}
```

```
1 void __stdcall __noreturn ChromeThreadProc(LPVOID lpThreadParameter)
2 {
3   while ( 1 )
4   {
5     WSACleanup();
6     Sleep(1000u);
7   }
8 }
```

Win/Dridex

1) Pointer in the .text section (pattern-based search)
2) Left: patterns; Right: version checks

```
.rdata:6EEA7A0C    qword_6EEA7A0C  dq 0E85353535353h
.rdata:6EEA7A14    qword_6EEA7A14  dq 0E806890000h
.rdata:6EEA7A1C    byte_6EEA7A1C   db 0
.rdata:6EEA7A1D    qword_6EEA7A1D  dq 0E85757575757h
.rdata:6EEA7A25    qword_6EEA7A25  dq 0E806890000h
.rdata:6EEA7A2D    byte_6EEA7A2D   db 0
.rdata:6EEA7A2E    qword_6EEA7A2E  dq 0E850505050C033h
.rdata:6EEA7A36    dword_6EEA7A36  dd 0
.rdata:6EEA7A3A    word_6EEA7A3A   dw 789h
.rdata:6EEA7A3C    byte_6EEA7A3C   db 0E8h
.rdata:6EEA7A3D    qword_6EEA7A3D  dq 0E850505050C033h
.rdata:6EEA7A45    dword_6EEA7A45  dd 0
.rdata:6EEA7A49    word_6EEA7A49   dw 689h
.rdata:6EEA7A4B    byte_6EEA7A4B   db 0E8h
.rdata:6EEA7A4C    qword_6EEA7A4C  dq 0E850505050h
.rdata:6EEA7A54    qword_6EEA7A54  dq 0E80189E04D8B0000h
.rdata:6EEA7A5C    qword_6EEA7A5C  dq 0E850505050C033h
.rdata:6EEA7A64    dword_6EEA7A64  dd 0
.rdata:6EEA7A68    word_6EEA7A68   dw 789h
.rdata:6EEA7A6A    byte_6EEA7A6A   db 0E8h
.rdata:6EEA7A6B    dword_6EEA7A6B  dd 0A0D0920h
```

```
if ( chromeVersion >= 0x907005A )          // 42.0.2311.90
{
  if ( chromeVersion >= 0x9350041 )        // 43.0.2357.65
  {
    if ( chromeVersion >= 0x9960055 )      // 45.0.2454.85
    {
      if ( chromeVersion >= 0x9DE0049 )    // 47.0.2526.73
      {
        if ( chromeVersion >= 0xA040061 )  // 48.0.2564.97
        {
          sslClose = sslClass->vTable;
          if ( chromeVersion >= 0xA3F0057 ) // 49.0.2623.87
          {
            if ( chromeVersion >= 0xAE10074 )// 53.0.2785.116
            {
              sslWrite = &sslClass->vTable[5];
              sslRead = &sslClass->vTable[2];
            }
            else
            {
              sslWrite = &sslClass->vTable[7];
              sslRead = &sslClass->vTable[4];
```

Win/Dridex – Reaction times

Chrome version	Release date	Dridex version	Timestamp
40.0.2214.115	19.2.2015	2.093	11.3.2015
42.0.2311.90	14.4.2015	2.108	17.4.2015
43.0.2357.65	19.5.2015	3.011	26.5.2015
44.0.2403.89	21.7.2015	3.073	6.8.2015
45.0.2454.85	1.9.2015	3.102	25.9.2015
47.0.2526.73	1.12.2015	3.154	7.12.2015
48.0.2564.97	27.1.2016	3.167	29.1.2016
49.0.2623.87	8.3.2016	3.188	10.3.2016
51.0.2704.106	23.6.2016	3.225	24.6.2016
53.0.2785.116	14.9.2016	3.258	26.9.2016
54.0.2840.71	20.10.2016	3.269	17.11.2016
58.0.3029.81	19.4.2017	4.048	16.5.2017

Win/Spy.Ursnif

```
chrome.dll:   dd SSL3_VERSION
chrome.dll:   dd TLS1_3_VERSION
chrome.dll:   dd offset ssl3_version_from_wire
chrome.dll:   dd offset ssl3_version_to_wire
chrome.dll:   dd offset ssl3_new
chrome.dll:   dd offset ssl3_free
chrome.dll:   dd offset ssl3_get_message
chrome.dll:   dd offset ssl3_get_current_message
chrome.dll:   dd offset ssl3_release_current_message
chrome.dll:   dd offset ssl3_read_app_data
chrome.dll:   dd offset ssl3_read_change_cipher_spec
chrome.dll:   dd offset ssl3_read_close_notify
chrome.dll:   dd offset ssl3_write_app_data
chrome.dll:   dd offset ssl3_dispatch_alert
chrome.dll:   dd offset ssl3_supports_cipher
chrome.dll:   dd offset ssl3_init_message
chrome.dll:   dd offset ssl3_finish_message
chrome.dll:   dd offset ssl3_add_message
chrome.dll:   dd offset ssl3_add_change_cipher_spec
chrome.dll:   dd offset ssl3_add_alert
chrome.dll:   dd offset ssl3_flush_flight
chrome.dll:   dd offset nullsub
chrome.dll:   dd offset nullsub
chrome.dll:   dd offset ssl3_set_read_state
chrome.dll:   dd offset ssl3_set_write_state
```

```
dd SSL3_VERSION
dd TLS1_3_VERSION
dd offset ssl3_version_from_wire
dd offset ssl3_version_to_wire
dd offset ssl3_new
dd offset loc_CED1713
dd offset loc_CED1734
dd offset loc_CED1755
dd offset loc_CED1776
dd offset loc_CED1797
dd offset loc_CED17B8
dd offset loc_CED17D9
dd offset loc_CED17FA
dd offset loc_CED181B
dd offset ssl3_supports_cipher
dd offset loc_CED183C
dd offset loc_CED185D
dd offset loc_CED187E
dd offset loc_CED189F
dd offset loc_CED18C0
dd offset loc_CED18E1
dd offset nullsub
dd offset nullsub
dd offset loc_CED1902
dd offset loc_CED1923
```

```
30  if ( !a3 )
31    goto LABEL_42;
32  if ( a3 <= 4 || (v5 = *a2, *a2 != ' TEG') && v5 != ' TUP' && v5 != 'TSOP' && v5 != 'ITPO' )
33  {
34    v4 = sub_10007CCB(*a1);
35 LABEL_42:
36    v25 = (*(a1[1] + 8))(*a1, a2, a3, a1[2]);
37    goto LABEL_43;
38  }
39  v6 = lstrlenA("\r\n\r\n");
40  v26 = 0;
41  v7 = a3 - v6 + 1;
42  nChar = v6;
43  if ( v7 <= 0 )
44    goto LABEL_42;
45  while ( *(v26 + a2) != Str2[0] || StrCmpNA((v26 + a2), "\r\n\r\n", nChar) )
46  {
47    if ( ++v26 >= v7 )
48      goto LABEL_42;
49  }
```

- Attack points lookup
  - “instrumentation” of the browser process
- Registry storage:
  - Checksum(chrome)
  - Offset (SSL VMT)

Win/Spy.Ursnif

```
chrome.dll:63BE4A88 dd SSL3_VERSION
chrome.dll:63BE4A88 dd TLS1_3_VERSION
chrome.dll:63BE4A88 dd offset ssl3_version_from_wire
chrome.dll:63BE4A88 dd offset ssl3_version_to_wire
chrome.dll:63BE4A88 dd offset ssl3_new
chrome.dll:63BE4A88 dd offset loc_CED1713
chrome.dll:63BE4A88 dd offset loc_CED1734
chrome.dll:63BE4A88 dd offset loc_CED1755
chrome.dll:63BE4A88 dd offset loc_CED1776
chrome.dll:63BE4A88 dd offset hook_read_app_data
chrome.dll:63BE4A88 dd offset loc_CED17B8
chrome.dll:63BE4A88 dd offset hook_read_close_notify
chrome.dll:63BE4A88 dd offset hook_write_app_data
chrome.dll:63BE4A88 dd offset loc_CED181B
chrome.dll:63BE4A88 dd offset ssl3_supports_cipher
chrome.dll:63BE4A88 dd offset loc_CED183C
chrome.dll:63BE4A88 dd offset loc_CED185D
chrome.dll:63BE4A88 dd offset loc_CED187E
chrome.dll:63BE4A88 dd offset loc_CED189F
chrome.dll:63BE4A88 dd offset loc_CED18C0
chrome.dll:63BE4A88 dd offset loc_CED18E1
chrome.dll:63BE4A88 dd offset nullsub
chrome.dll:63BE4A88 dd offset nullsub
chrome.dll:63BE4A88 dd offset loc_CED1902
chrome.dll:63BE4A88 dd offset loc_CED1923
```

- Attack points is SSL VMT replaced

Win/Spy.Ursnif.AX (St.Nicholas Case)

```
ll    IsWow64Process  chrome.exe  firefox.exe iexplore.exe    mi
    [INJECT] inject_via_remotethread_wow64: pExecuteX64=0x%08p,
-(    ReflectiveLoader        H
C:\Users\W7\Downloads\ModificationSourceCode_16_12_6\Bin\Loader.
```

- Loose conditions to locate SSL VMT
- However, support for new Chrome releases lost easily
- Strict opcode condition left the recent 64-bit Chrome unsupported

Win/Spy.Ursnif.AX (St.Nicholas Case)

```
1 int HookBrowser()
2 {
3   int result; // eax@4
4   HMODULE v1; // eax@5
5
6   switch ( browserType_ )
7   {
8     case INTERNET_EXPLORER:
9       if ( LoadLibraryA("WININET.DLL") )
10      {
11        dllName[0] = "WININET.DLL";
12        result = HookFunctions(&wininetHooks, 13);
13      }
14      else
15      {
16        result = 126;
17      }
18      break;
19    case FIREFOX:
20      result = HookFirefox();
21      break;
22    case CHROME:
23      v1 = GetModuleHandleA("CHROME.DLL");
24      if ( v1 )
25        result = HookChrome(v1);
26      else
27        result = HookFunctions(&loadlibraryExHook, 1);
28      break;
```

Locating attack points starting from legacy to recent variants ☺

```
v3 = FindSSLVTable_Legacy(this, &v13);
v4 = v13;
if ( v3 )
{
  v2 = 1;
  if ( FindSSLVTable_v49_to_v52(v1, &v14) )
  {
    v2 = 2;
    if ( FindSSLVTable_v53(v1, &v15) )
    {
      v2 = 3;
      result = FindSSLVTable_v54_to_v58(v1, &v16) != 0;
      if ( result )
        return result;
      off_5FD85250 = CrSSLReadHook;
```

Win/Qbot

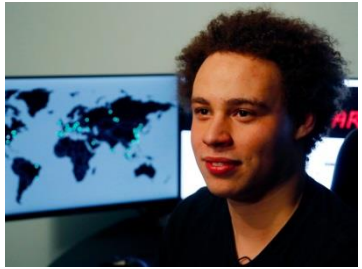
```
 1 BOOL __stdcall DllEntryPoint(HINSTANCE hinstDLL, DWORD fdwReason, LPVOID lpReserved)
 2 {
 3   CHAR OutputString; // [esp+10h] [ebp-80h]@4
 4
 5   hModule = hinstDLL;
 6   if ( fdwReason == 1 )
 7   {
 8     Heap::Init();
 9     if ( DecryptStrings_ResolveAPIs(1) < 0 )
10       return 0;
11     sub_1000D31C(hinstDLL, lpReserved, 0, 1);
12     sub_1000B826(&OutputString, 128, "%s: %#p", dword_10022AD0);
13     OutputDebugStringA(&OutputString);
...
24     MH_Initialize();
...
42       PrepareHooks_IE();
43       PrepareHooks_Firefox();
44       PrepareHooks_Chrome(0);
45       if ( sub_10006E45() )
46         sub_10002C19();
47     }
48     MH_EnableAllHooks();
```

```
.1001B000:  10 03 00 00.DE 02 00 00 52 00 55 8B.EC 56 57 8B
.1001B010:  FA 8B F1 E8.AA AA AA AA 83 7E 18 00.75 1F 68 F0
.1001B020:  02 00 00 68.AA AA AA AA.68 E2 00 00.00 6A 10 59
.1001B030:  E8 AA AA AA.AA 83 C4 0C.83 C8 FF EB.5A 8B 46 28
.1001B040:  83 B8 88 00.00 00 00 74.11 68 F5 02.00 00 68 AA
.1001B050:  AA AA AA 68.C2 00 00 00.EB D3 8B CE 00 00 00 00
.1001B060:  00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00
.1001B070:  00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00
```

Win/Tinukebot

- Original author — 18-year-old Augustin Inzirillo (from France)
- His project shared with his contacts
- These guys tried to profit off
- Augustin got sad → the sources released on Github for free for grabs

"I am very worried for him, because some technology company told him they wanted to fly him to the U.S. for a job interview as a result of him posting that online," Daniel Inzirillo, Augustin's father



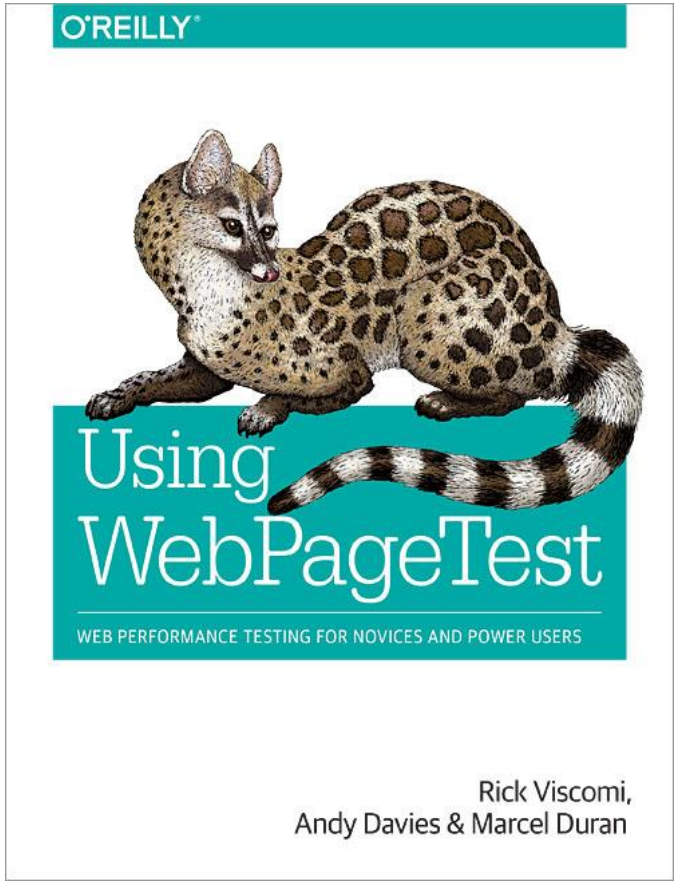
?

Win/Tinukebot

- Code borrowed from the WebPageTest project supported by Google

https://github.com/WPO-Foundation/webpagetest/

```
static SSL_METHODS_SIGNATURE methods_signatures[] = {
  // August 2016 - hhlen is            // Chrome 53            // Nov 2015
  { 0,      switched for ssl max DWORD ,{53,                   ,{52,          // Ended in
    54,                                  53,                     0,           // No start version
    21,                                  14,                     13,          // count
    8,                                   4,                      4,           // Signature len
    "\x0\x0\x0\x3\x4\x3\x0\x0",          "\x0\x0\x0\x0",         "\x0\x0\x0\x0", // signature
    0,                                   (const void **)4,       (const void **)4,// hhlen value
    -1,                                  9,                      11, // hhlen_index
    2,                                   0,                      0,  // ssl_new_index
    3,                                   1,                      1,  // ssl_free_index
    -1,                                  -1,                     3,  // ssl_connect_index
    -1,                                  -1,                     -1, // ssl_begin_handshake_index
    -1,                                  3,                      5,  // ssl_read_app_data_old_index
    7,                                   -1,                     -1, // ssl_read_app_data_index
    10}                                  6}                      8}  // ssl_write_app_data_index
                                                               };
```



Summary of Hooking types

Hooking type	Banking Trojan
Replacement of a function in SSL VMT	Win/Spy.Ursnif-based, Win/Qadars, Win/Trickbot, Win/Zbot-based
Inline hook in SSL VMT	Win/Dridex, Win/Tinukebot
Custom method	Win/Qbot

Summary of Targets

Banking trojan	Latest version	IE	Edge	Firefox	Chrome 32-bit	Chrome 64-bit
Win/Dridex	4.057	✓	⚠	✓	**48-59**	**48-59**
Win/TrickBot	1025	✓	✓	✓	**54-59**	**54-59**
Win/Spy.Ursnif * Gozi/ISFB	2.16 b. 943	✓	✓	✓	**44-59**	**44-59**
Win/Spy.Ursnif.AX	26.05.2017	✓	✓	✓	**49-58**	**49-57**
Win/Qbot	25.05.2017	✓	⚠	✓	**48-58**	**54-58**
Win/Qadars	04.04.2017	✓	⚠	✓	**48-57**	**49-57**
Win/Tinukebot	06.06.2017	✓	⚠	✓	**52-59**	**52-59**

*Attacks also Opera

Other active banking trojans

Panda Banker

Win32/Spy.Zbot.{ACM,ACY,ACZ}

FormBook

Win32/Agent.YIJ

Neutrino Bot

Win/Kasidet

Kronos

Win32/Agent.QMH

GozNym

Win32/TrojanDownloader.Nymaim

Rovnix

Win/Rovnix

Remarks

- Malware authors do not copy from each other
- Banking modules usually separated from the distributed binaries
- SSL_VERSION is dropped Chromium 61 → may affect many current implementations
- Versioning available → good to track changes
- Support for browsers: good indicator if the family is active

browserhooks

“When the plugin prints some findings, then it’s a little bit too late.”

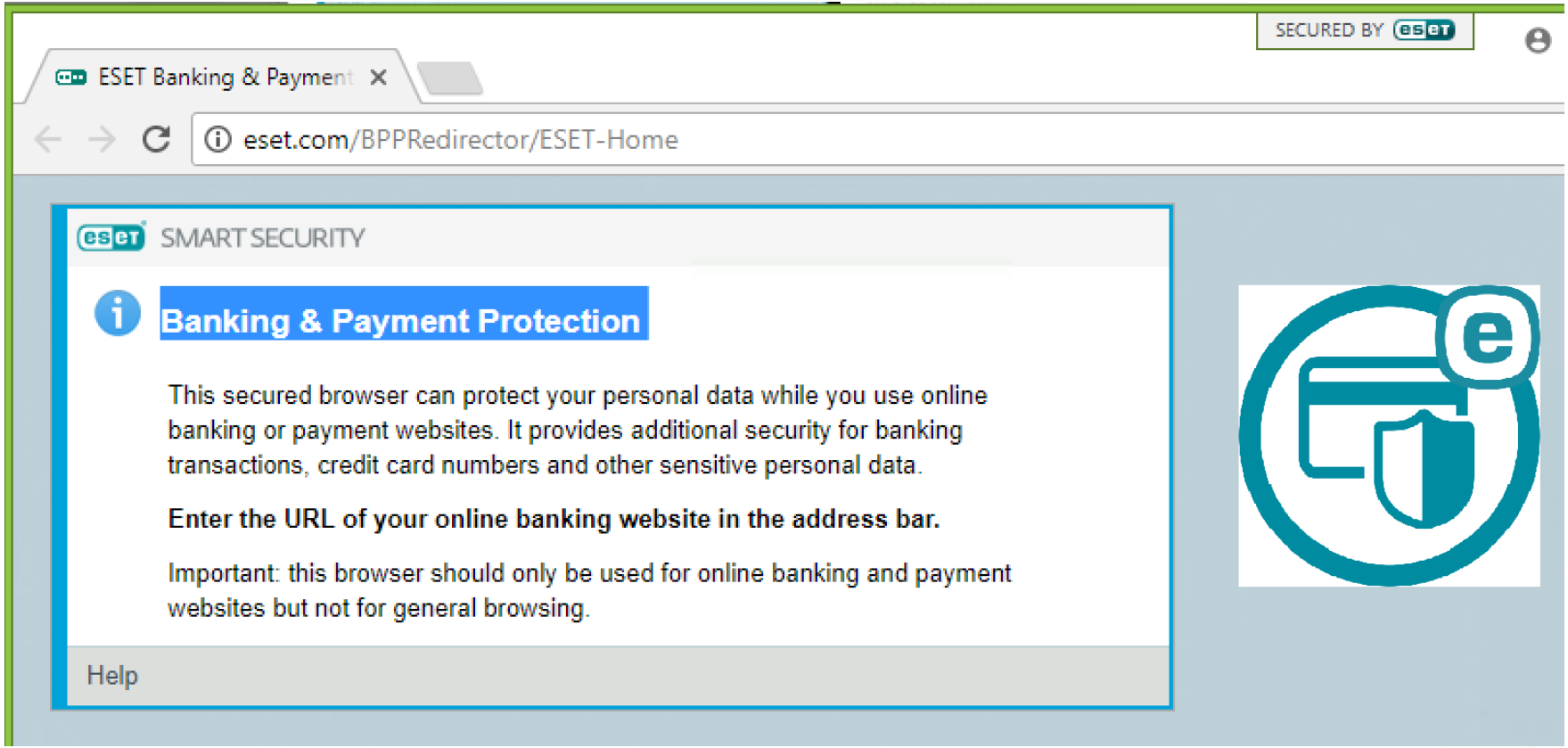
browserhooks

https://github.com/eset/volatility-browserhooks

- Extending **apihooks** with 3 new hooking types
- 32-bit modules in WoW64 processes supported
- Integration with VolUtility GUI (Kevin Breen, 2016)

- Some limitations discovered

Protecting browsers



Questions & Answers

VF – browserhooks

SSL Hooks for Chrome implemented by Qbot	chrome.exe	2220	chrome.dll	0x7feeab790b8L	0x180001f34L	0x180000000L
Inline/Trampoline	chrome.exe	2220	WS2_32.dll	WS2_32.dll!WSAConnect at 0x7fefdbee0f0	0x18000c2a8L	0x7fefdbc0000
Inline/Trampoline	chrome.exe	2220	WS2_32.dll	WS2_32.dll!WSAConnect at 0x7fefdbee0f0	0x18000c2a8L	0x180000000L
Inline/Trampoline	chrome.exe	2220	WS2_32.dll	WS2_32.dll!WSASend at 0x7fefdbc13b0	0x18000bf70L	0x7fefdbc0000
Inline/Trampoline	chrome.exe	2220	WS2_32.dll	WS2_32.dll!WSASend at 0x7fefdbc13b0	0x18000bf70L	0x180000000L
Inline/Trampoline	chrome.exe	2220	WS2_32.dll	WS2_32.dll!send at 0x7fefdbc8000	0x18000c078L	0x7fefdbc0000
Inline/Trampoline	chrome.exe	2220	WS2_32.dll	WS2_32.dll!send at 0x7fefdbc8000	0x18000c078L	0x180000000L

Showing 1 to 10 of 10 entries

BookMark Row
Search cell value
Export Row
Export Table
Store Hooking Module

Next

k to Top

VF – browserhooks

Filter Plugins

Plugin Command	Plugin Type	Date Completed	Actions
dlldump	Processes and DLLs	09 Sep 17 20:45:16	
browserhooks	Other	09 Sep 17 20:35:52	
amcache	Registry		
apihooks	Processes and DLLs		

Dlldump

Show 25 entries

#	Process	ImageBase	Name	StoredFile
1	0xfffffa80023d1380L	6442450944	chrome.exe	File Details

Showing 1 to 1 of 1 entries

0x18000000

PID 2220 = chrome.exe

File Details - module.2220.3e9d1380.180000000.dll

- Details
- HexViewer
- ExifData
- VirusTotalSearch
- SqliteViewer
- ExtractStrings

Details

FileName	module.2220.3e9d1380.180000000.dll
FileSize	137728 bytes
MD5	c346f3d3082163927e2da9e834b52e3d
SHA256	1a62c4c0fd09a91ff47ec58411614b41c2ae9
Download	Download
Delete	

VirusTotal - complete

PermaLink	Link to Report
ScanDate	2017-09-29 18:48:13
Results	7 / 63

Engine	Version	Result
ESET-NOD32	16161	a variant of Win64/Qbot.B