Talos

When The Screen Goes Dark

Protecting Broadcasts in the Modern Age

Edmund Brumaghin / Threat Researcher

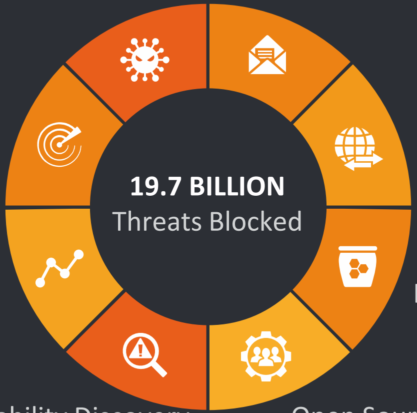
Who Am I?

Edmund Brumaghin

- Threat Researcher at Cisco Talos.
- Spent the past decade defending critical infrastructure.
- I <3 Malware.

Talos Intel Breadown

THREAT INTEL



19.7 BILLION
Threats Blocked

1.5 MILLION
Daily Malware Samples

600 BILLION
Daily Email Messages

16 BILLION
Daily Web Requests

Honeypots

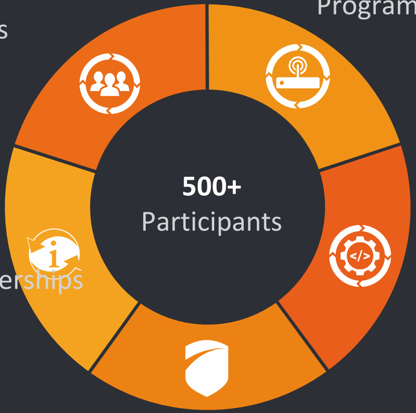
Open Source Communities

Vulnerability Discovery (Internal)

Product Telemetry

Internet-Wide Scanning

INTEL SHARING



500+
Participants

Customer Data Sharing Programs

Service Provider Coordination Program

Open Source Intel Sharing

3rd Party Programs (MAPP)

Industry Sharing Partnerships (ISACs)

250+
Full Time Threat Intel Researchers

MILLIONS
Of Telemetry Agents

4
Global Data Centers

100+
Threat Intelligence Partners

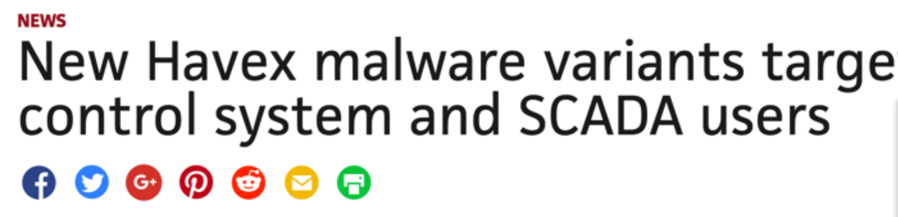
1100+
Threat Traps

Emerging Threats

Supply Chain Attacks

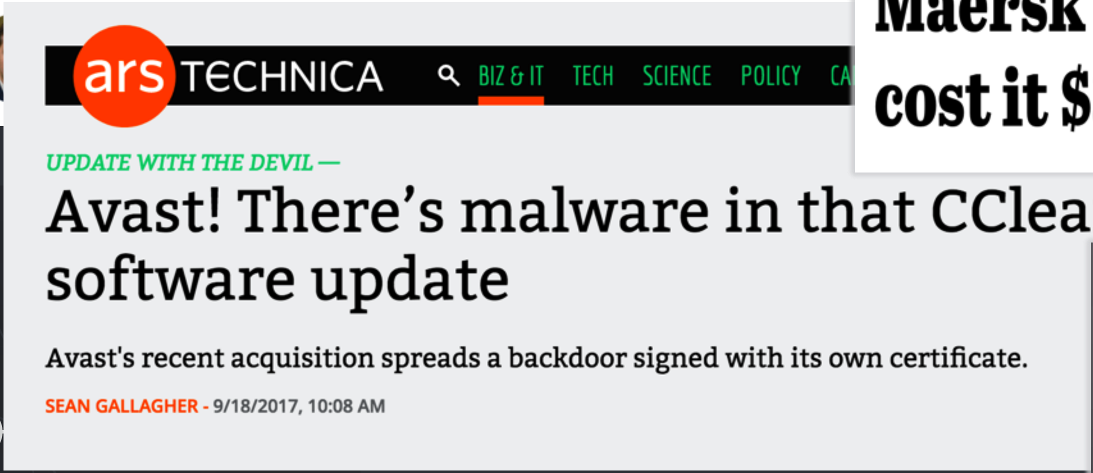
Supply Chain Attacks

Exploiting Trust Relationships



NEWS

New Havex malware variants target industrial control system and SCADA users

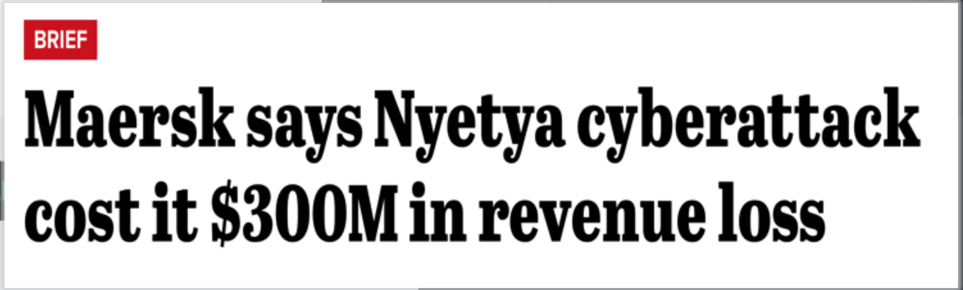


UPDATE WITH THE DEVIL —

Avast! There’s malware in that CCleaner software update

Avast's recent acquisition spreads a backdoor signed with its own certificate.

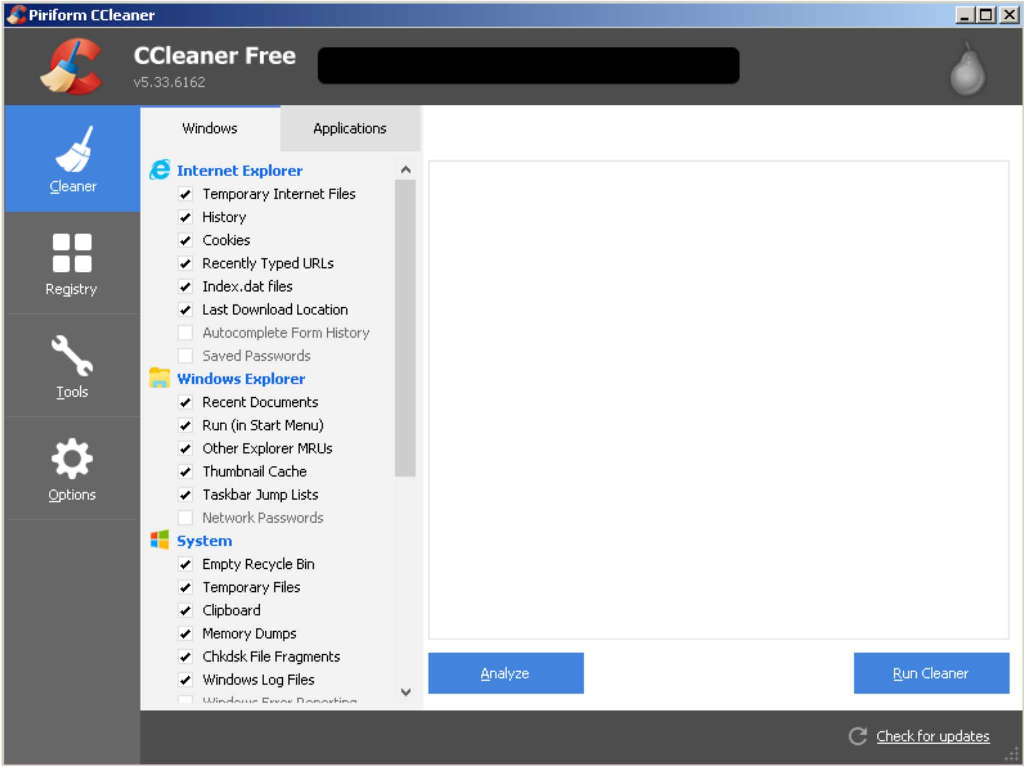
SEAN GALLAGHER - 9/18/2017, 10:08 AM



BRIEF

Maersk says Nyetya cyberattack cost it $300M in revenue loss

Beta Testing New Engine in AMP Leads to Discovery – CCleaner Serving Malware



[redacted]

Internet Explorer

- ☑ Temporary Internet Files
- ☑ History
- ☑ Cookies
- ☑ Recently Typed URLs
- ☑ Index.dat files
- ☑ Last Download Location
- □ Autocomplete Form History
- □ Saved Passwords

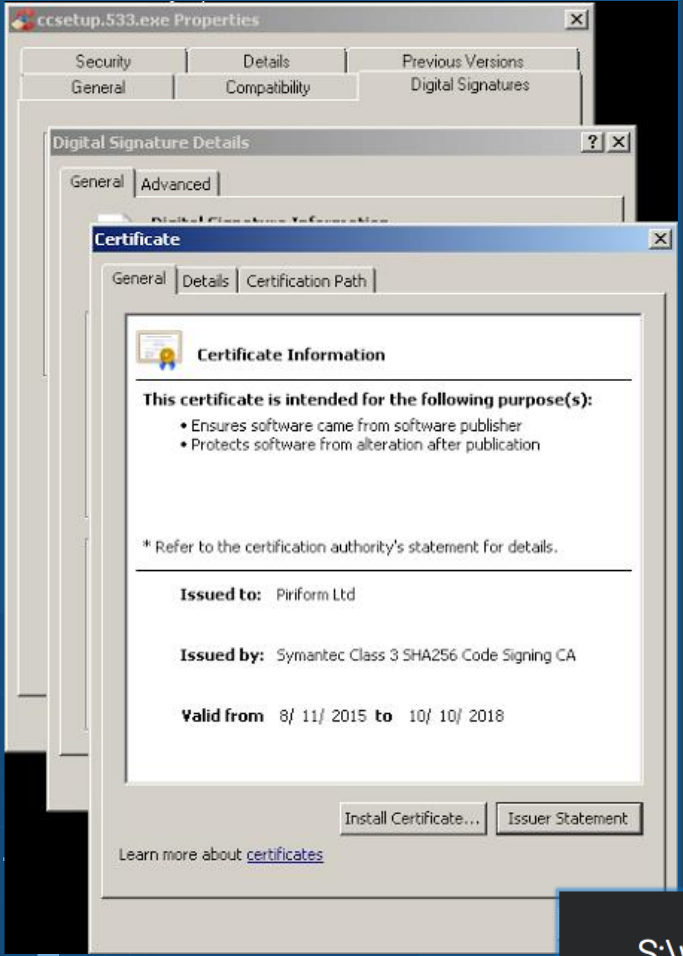
Windows Explorer

- ☑ Recent Documents
- ☑ Run (in Start Menu)
- ☑ Other Explorer MRUs
- ☑ Thumbnail Cache
- ☑ Taskbar Jump Lists
- □ Network Passwords

System

- ☑ Empty Recycle Bin
- ☑ Temporary Files
- ☑ Clipboard
- ☑ Memory Dumps
- ☑ Chkdsk File Fragments
- ☑ Windows Log Files
- □ Windows Error Reporting

- new exploit detection technology identified an executable triggering our advanced malware protection systems
- malicious payload featured a Domain Generation Algorithm (DGA) as well as hardcoded Command and Control (C2) functionality



Certificate Information

This certificate is intended for the following purpose(s):

- Ensures software came from software publisher
- Protects software from alteration after publication

* Refer to the certification authority's statement for details.

Issued to: Piriform Ltd

Issued by: Symantec Class 3 SHA256 Code Signing CA

Valid from 8/ 11/ 2015 **to** 10/ 10/ 2018

Digital Signature of CCleaner 5.33

- presence of a valid digital may be indicative of a larger issue that resulted in portions of the development or signing process being compromised
- this certificate should be revoked and untrusted moving forward

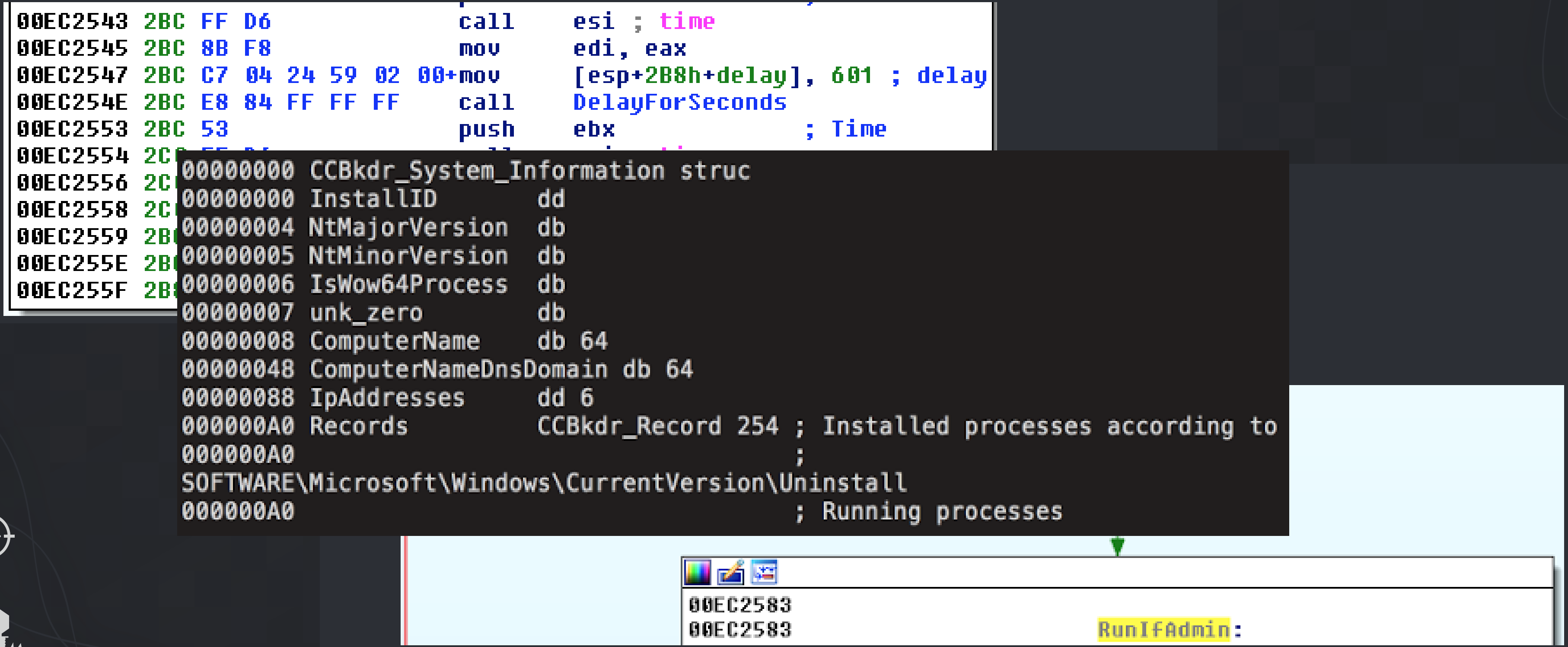
Compilation Artifact

- 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

```
S:\workspace\ccleaner\branches\v5.33\bin\CCleaner\Release\CCleaner.pdb
```

Malware Installation and Operation

Delay Routine – Admin Check – Backdoor SysInfo



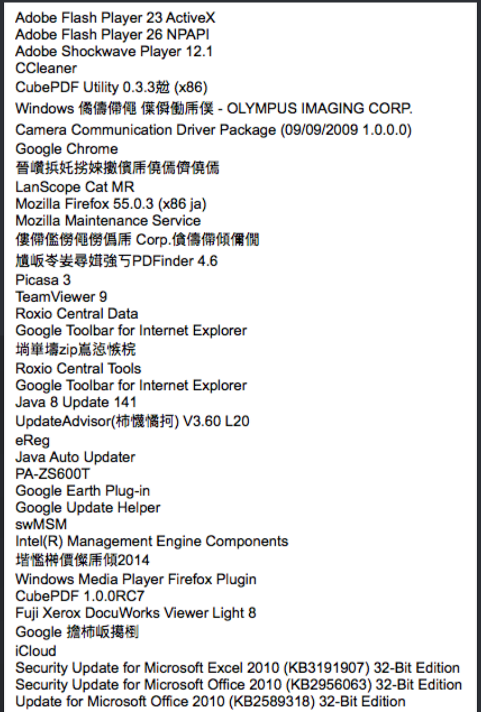
```
00EC2543 2BC FF D6                  call    esi ; time
00EC2545 2BC 8B F8                  mov     edi, eax
00EC2547 2BC C7 04 24 59 02 00+mov     [esp+2B8h+delay], 601 ; delay
00EC254E 2BC E8 84 FF FF FF         call    DelayForSeconds
00EC2553 2BC 53                     push    ebx                 ; Time
00EC2554 2C
00EC2556 2C
00EC2558 2C
00EC2559 2B
00EC255E 2B
00EC255F 2B
```

```
00000000 CCBkdr_System_Information struc
00000000 InstallID           dd
00000004 NtMajorVersion      db
00000005 NtMinorVersion      db
00000006 IsWow64Process      db
00000007 unk_zero            db
00000008 ComputerName        db 64
00000048 ComputerNameDnsDomain db 64
00000088 IpAddresses         dd 6
000000A0 Records             CCBkdr_Record 254 ; Installed processes according to
000000A0                                       ;
SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall
000000A0                                       ; Running processes
```

```
00EC2583
00EC2583                                   RunIfAdmin:
```

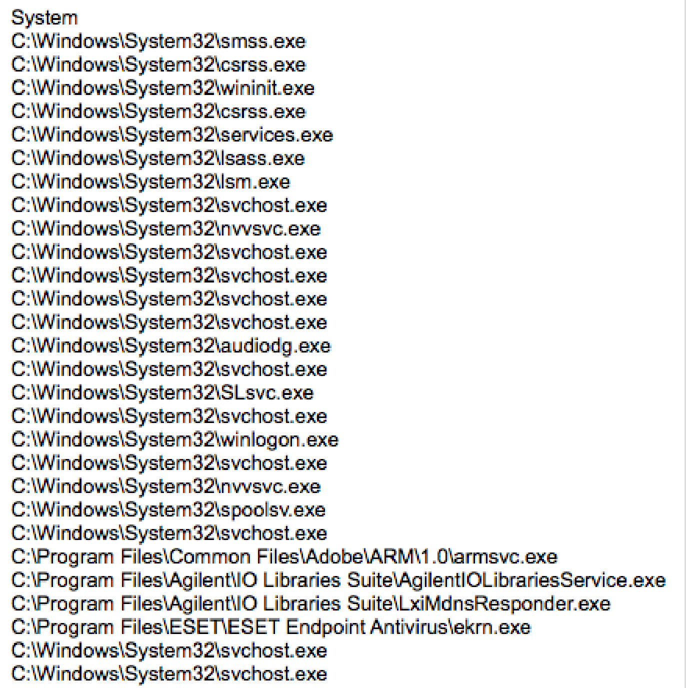
Data Collected on Infected Systems

Installed Programs



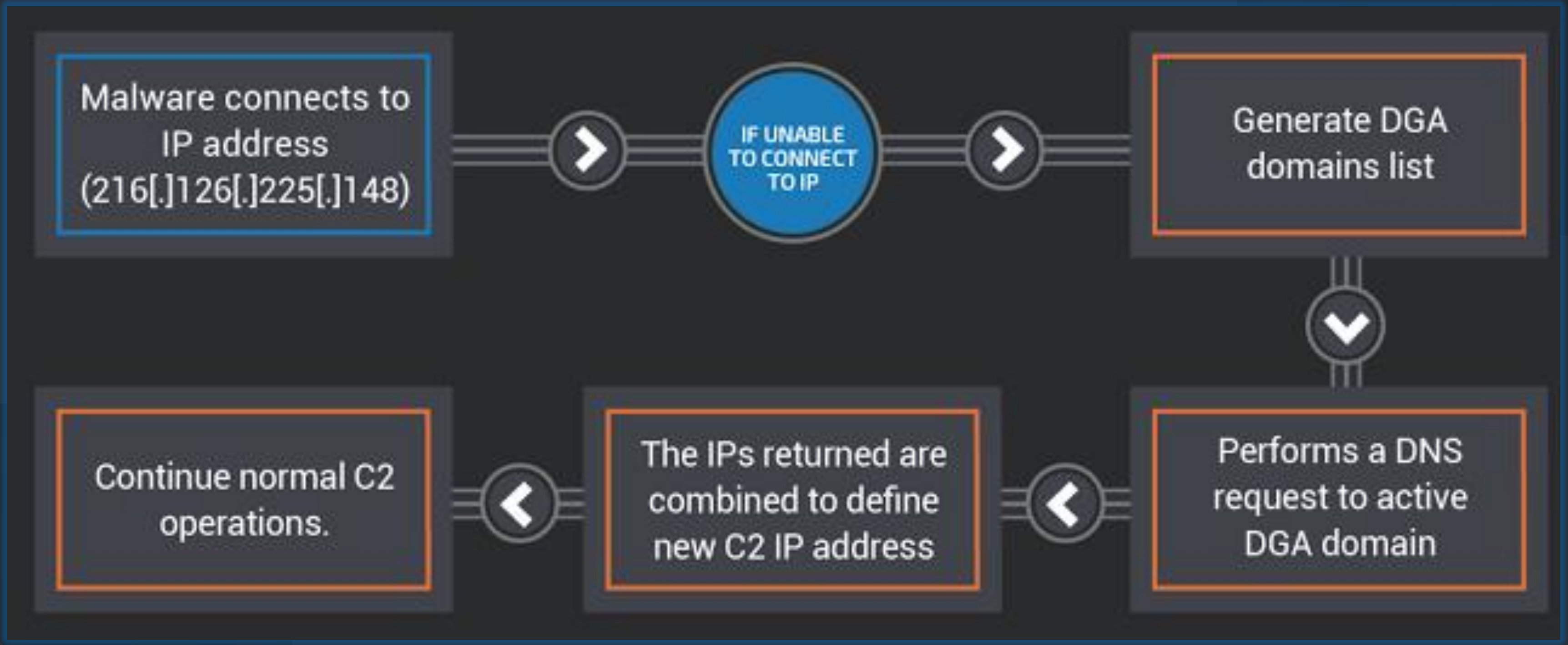
```
Adobe Flash Player 23 ActiveX
Adobe Flash Player 26 NPAPI
Adobe Shockwave Player 12.1
CCleaner
CubePDF Utility 0.3.3尯 (x86)
Windows 僪傷僀僶 僷僢働乕僕 - OLYMPUS IMAGING CORP.
Camera Communication Driver Package (09/09/2009 1.0.0.0)
Google Chrome
晢嬻拡妊拶妺擻僀乕僕僃僨僢僕僃
LanScope Cat MR
Mozilla Firefox 55.0.3 (x86 ja)
Mozilla Maintenance Service
僂僀儖僗僶僗僞乕 Corp.僪儔僀僶傿僗
擴岅夋婇尋媈強丂PDFinder 4.6
Picasa 3
TeamViewer 9
Roxio Central Data
Google Toolbar for Internet Explorer
埆筆墸zip嵩惉帹椃
Roxio Central Tools
Google Toolbar for Internet Explorer
Java 8 Update 141
UpdateAdvisor(柿懡曏抲) V3.60 L20
eReg
Java Auto Updater
PA-ZS600T
Google Earth Plug-in
Google Update Helper
swMSM
Intel(R) Management Engine Components
堦慛榊價傑乕傎2014
Windows Media Player Firefox Plugin
CubePDF 1.0.0RC7
Fuji Xerox DocuWorks Viewer Light 8
Google 擔柿岅擖椡
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
```

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
```

C2 Process



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

IF UNABLE TO CONNECT TO IP

Generate DGA domains list

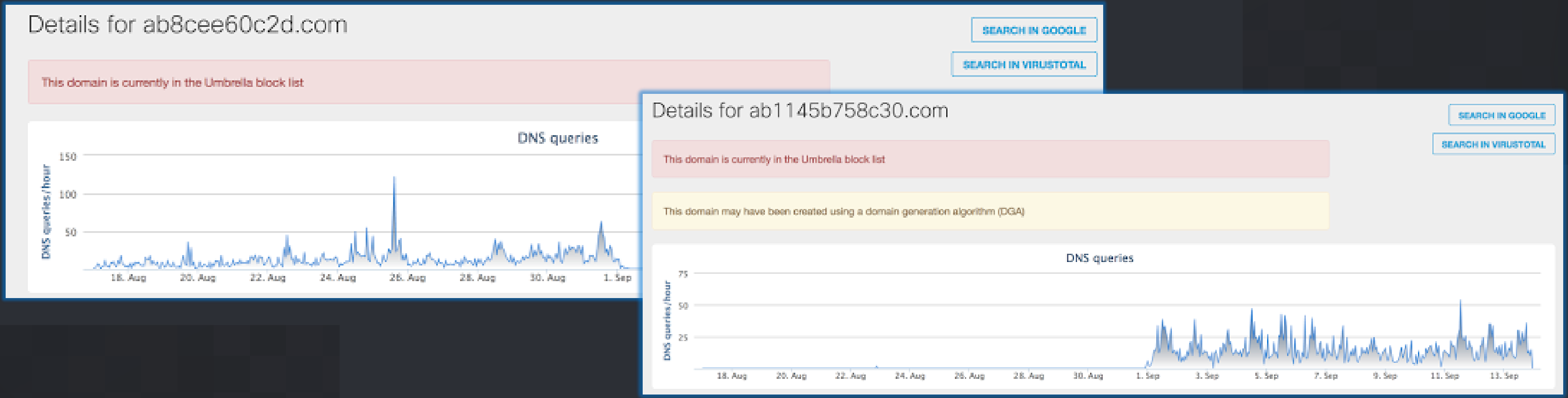
Performs a DNS request to active DGA domain

The IPs returned are combined to define new C2 IP address

Continue normal C2 operations.

DNS Activity for the DGA Domain

July – August – September – Following Takedown



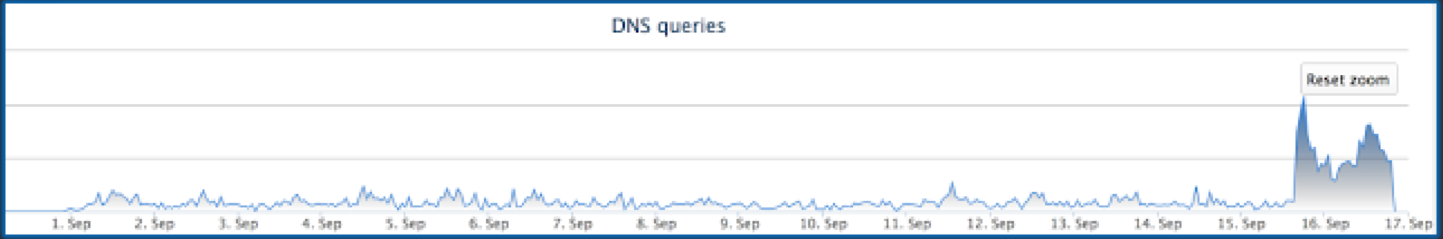
Details for ab8cee60c2d.com

This domain is currently in the Umbrella block list

Details for ab1145b758c30.com

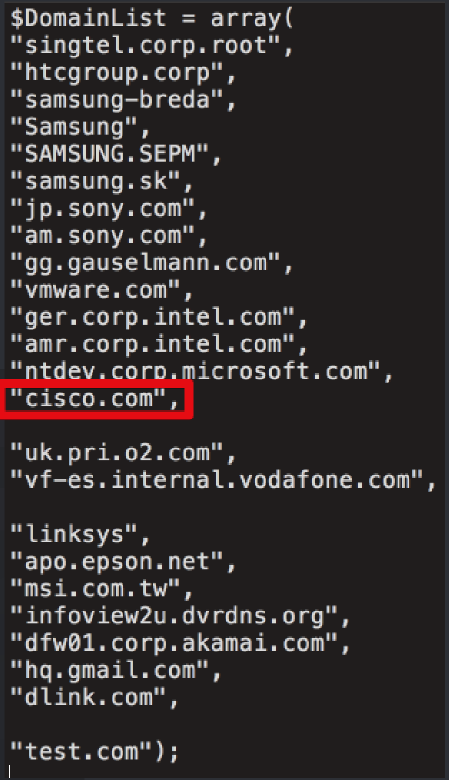
This domain is currently in the Umbrella block list

This domain may have been created using a domain generation algorithm (DGA)



Targeted to Tech Companies

2nd Stage only delivered to 23 specific domains



```
$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");
```

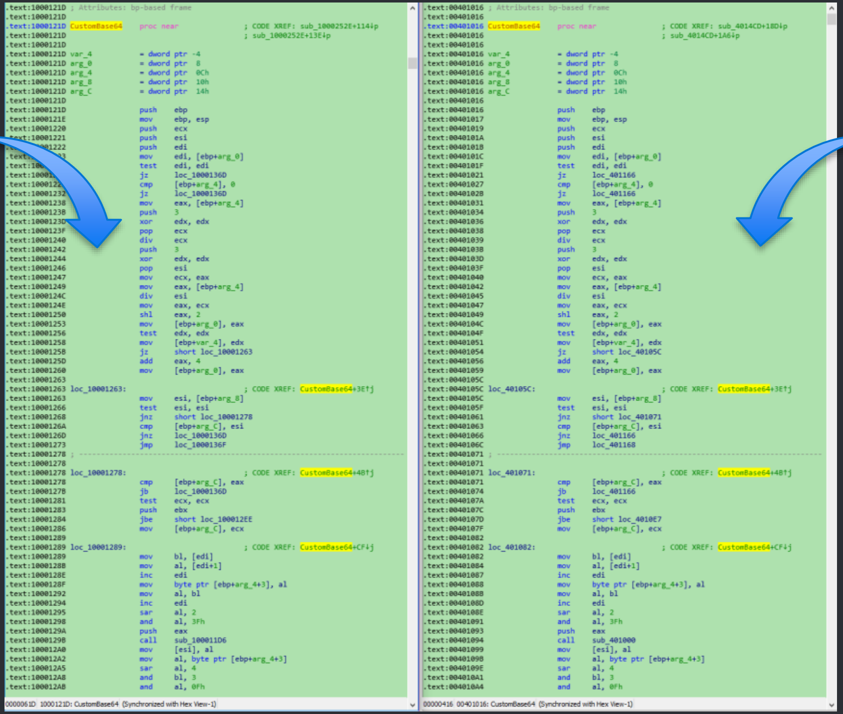
- Database Tracked 2nd Stage Delivery
- No Cisco Devices Delivered 2nd Stage

Code Reuse with Group 72

The 2nd stage payload shows similarities to code used by Group 72

CCleaner Malware

Group 72 Malware



```
.text:1000121D ; Attributes: bp-based frame
.text:1000121D
.text:1000121D CustomBase64    proc near               ; CODE XREF: sub_1000252E+114↓p
.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
```

0000061D 1000121D: CustomBase64 (Synchronized with Hex View-1)

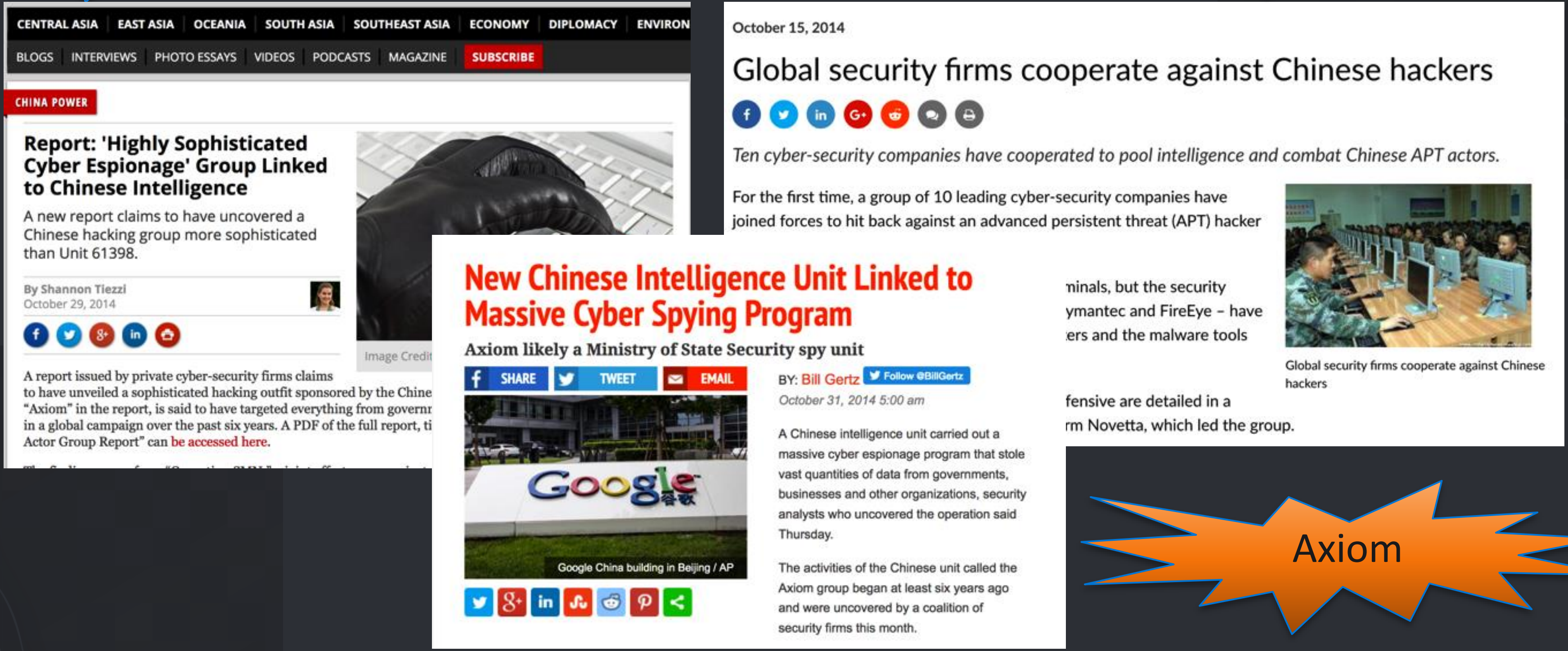
```
.text:00401016 ; Attributes: bp-based frame
.text:00401016
.text:00401016 CustomBase64    proc near               ; CODE XREF: sub_4014CD+18D↓p
.text:00401016                                         ; sub_4014CD+1A6↓p
.text:00401016
.text:00401016 var_4           = dword ptr -4
.text:00401016 arg_0           = 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)

What is Group 72

Operation SMN

APT 17



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

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

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

Global security firms cooperate against Chinese hackers

New Chinese Intelligence Unit Linked to Massive Cyber Spying Program

Axiom likely a Ministry of State Security spy unit

SHARE | TWEET | EMAIL

Google China building in Beijing / AP

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

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.

Axiom

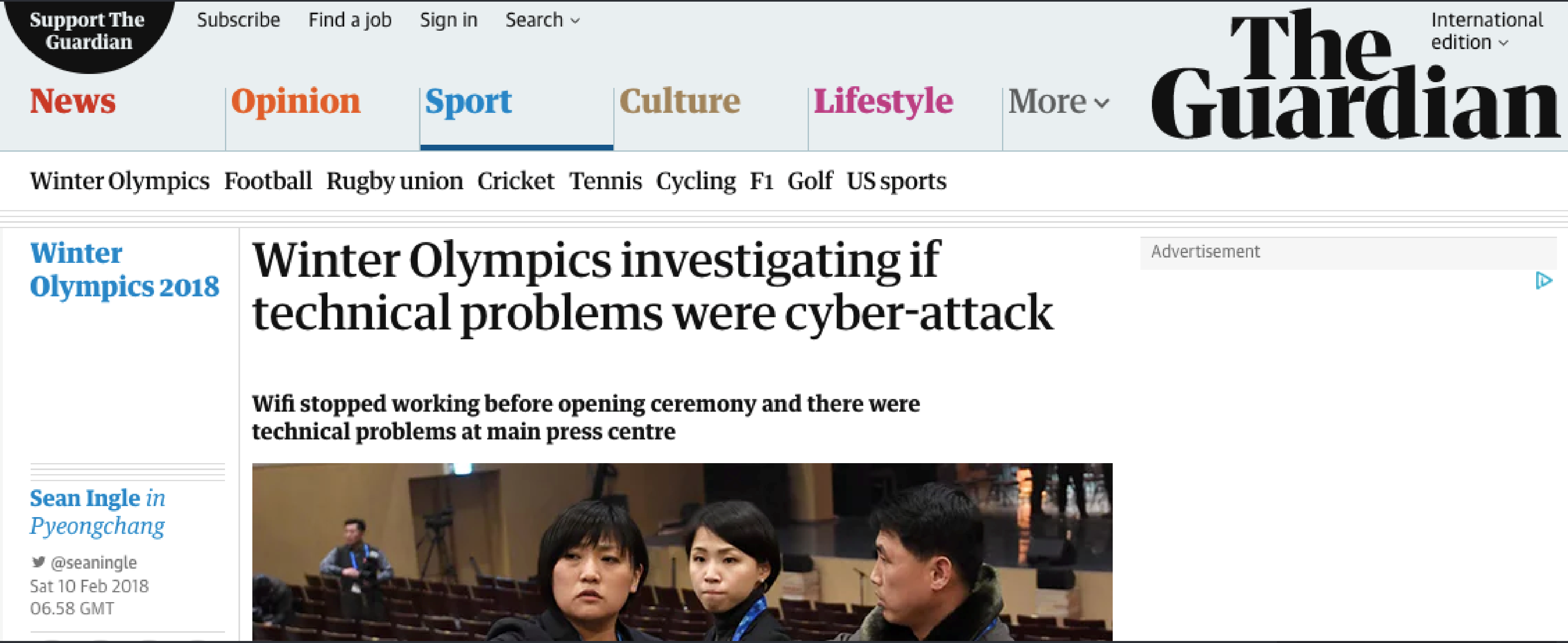
https://blogs.cisco.com/security/talos/threat-spotlight-group-72

Emerging Threats

Destructive Worms

Olympic Destroyer
Takes Aim At Winter Olympics

The Guardian Publication



Winter Olympics 2018

Winter Olympics investigating if technical problems were cyber-attack

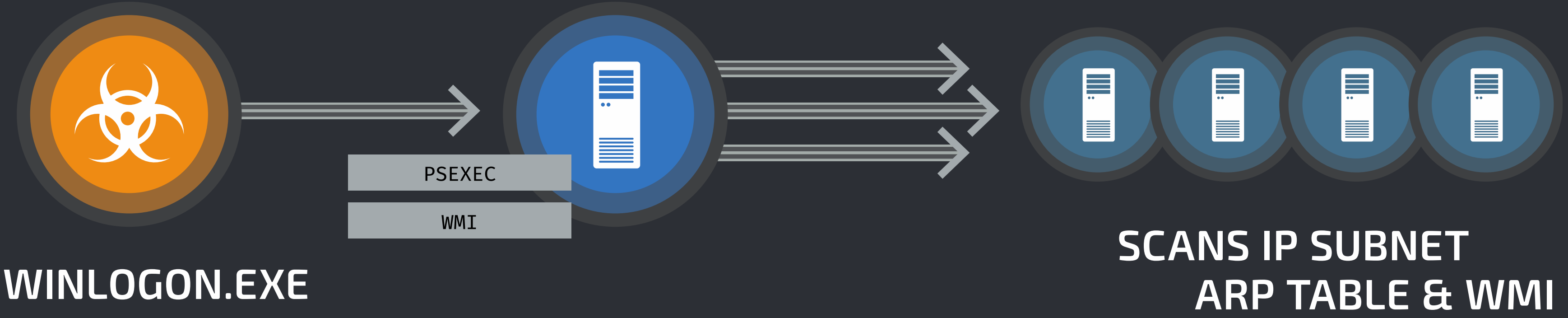
Wifi stopped working before opening ceremony and there were technical problems at main press centre

Sean Ingle *in Pyeongchang*

@seaningle

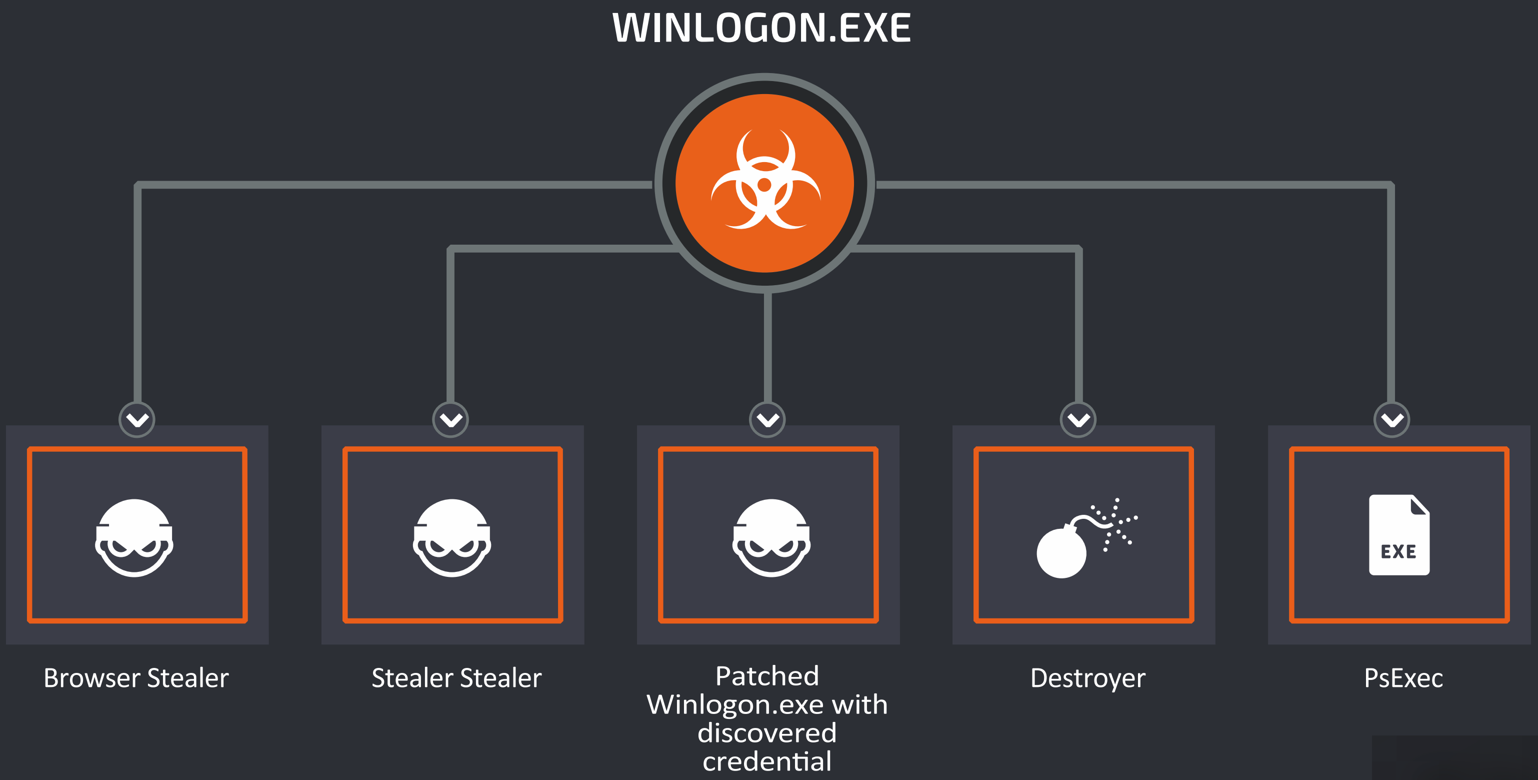
Sat 10 Feb 2018
06.58 GMT

Olympic Destroyer Propagation



Talos

Olympic Destroyer Workflow



WINLOGON.EXE

Browser Stealer

Stealer Stealer

Patched Winlogon.exe with discovered credential

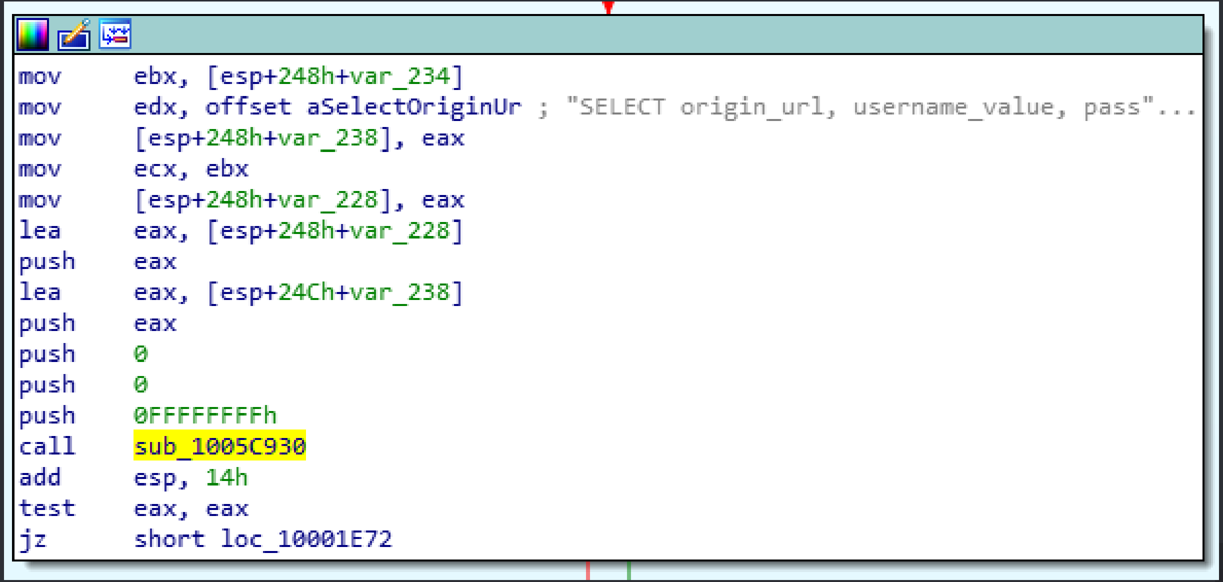
Destroyer

PsExec

TALOS

Password Stealer

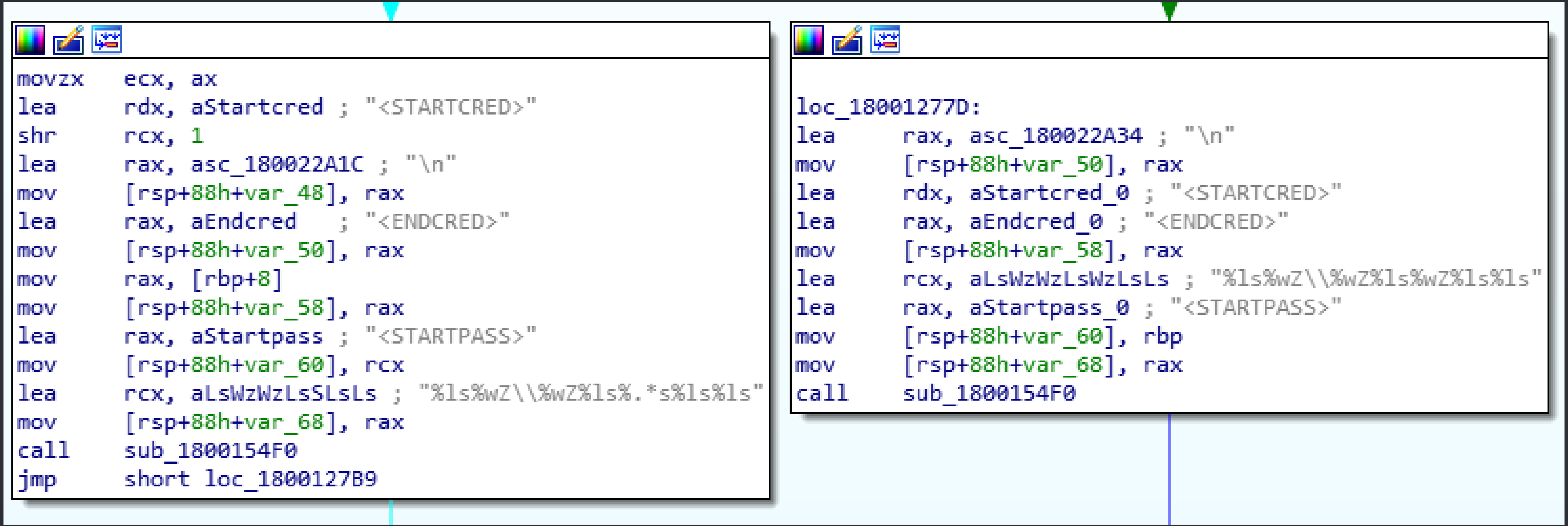
- Browsers: IE, Firefox, Chrome (communication to the main module via named pipe)



```
mov     ebx, [esp+248h+var_234]
mov     edx, offset aSelectOriginUr ; "SELECT origin_url, username_value, pass"...
mov     [esp+248h+var_238], eax
mov     ecx, ebx
mov     [esp+248h+var_228], eax
lea     eax, [esp+248h+var_228]
push    eax
lea     eax, [esp+24Ch+var_238]
push    eax
push    0
push    0
push    0FFFFFFFFh
call    sub_1005C930
add     esp, 14h
test    eax, eax
jz      short loc_10001E72
```

System Stealer

- Mimikatz (communication to the main module via named pipe)

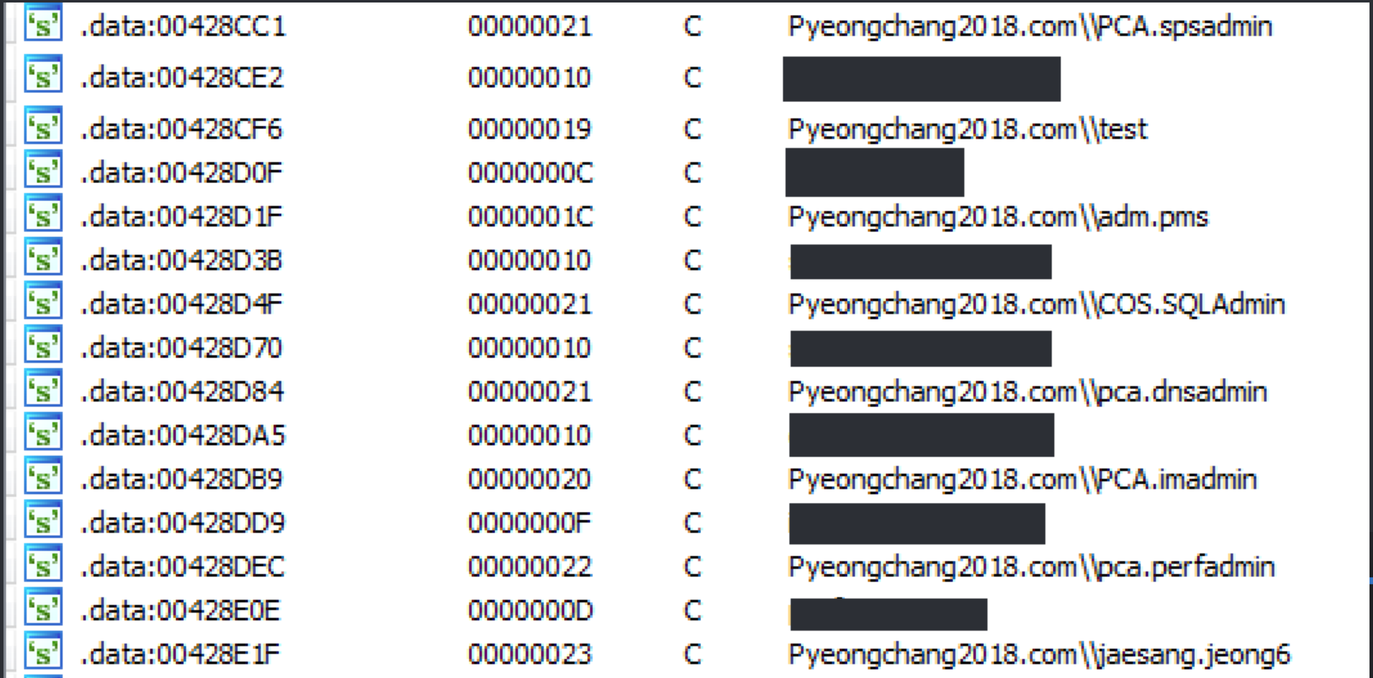


```
movzx   ecx, ax
lea     rdx, aStartcred ; "<STARTCRED>"
shr     rcx, 1
lea     rax, asc_180022A1C ; "\n"
mov     [rsp+88h+var_48], rax
lea     rax, aEndcred   ; "<ENDCRED>"
mov     [rsp+88h+var_50], rax
mov     rax, [rbp+8]
mov     [rsp+88h+var_58], rax
lea     rax, aStartpass ; "<STARTPASS>"
mov     [rsp+88h+var_60], rcx
lea     rcx, aLsWzWzLsSLsLs ; "%ls%wZ\\%wZ%ls%.*s%ls%ls"
mov     [rsp+88h+var_68], rax
call    sub_1800154F0
jmp     short loc_1800127B9
```

```
loc_18001277D:
lea     rax, asc_180022A34 ; "\n"
mov     [rsp+88h+var_50], rax
lea     rdx, aStartcred_0 ; "<STARTCRED>"
lea     rax, aEndcred_0 ; "<ENDCRED>"
mov     [rsp+88h+var_58], rax
lea     rcx, aLsWzWzLsWzLsLs ; "%ls%wZ\\%wZ%ls%wZ%ls%ls"
lea     rax, aStartpass_0 ; "<STARTPASS>"
mov     [rsp+88h+var_60], rbp
mov     [rsp+88h+var_68], rax
call    sub_1800154F0
```

System Stealer

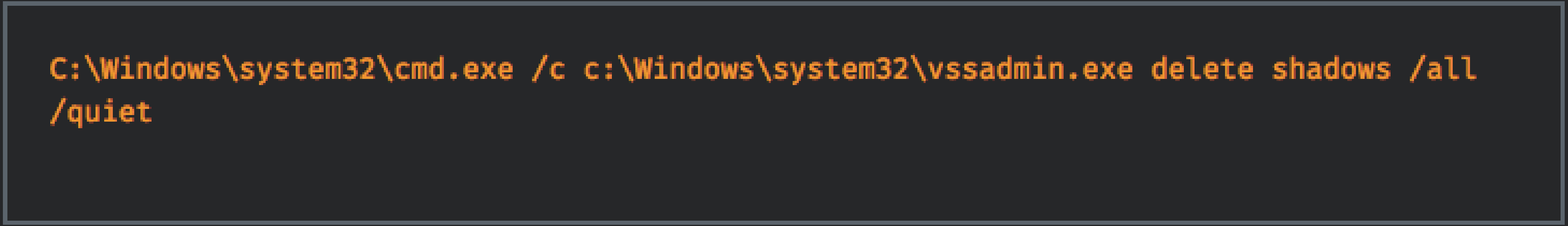
- The stolen credentials are used to patch the main binary
- The patched binary will be used for the propagation



```
.data:00428CC1    00000021    C    Pyeongchang2018.com\\PCA.spsadmin
.data:00428CE2    00000010    C    [redacted]
.data:00428CF6    00000019    C    Pyeongchang2018.com\\test
.data:00428D0F    0000000C    C    [redacted]
.data:00428D1F    0000001C    C    Pyeongchang2018.com\\adm.pms
.data:00428D3B    00000010    C    [redacted]
.data:00428D4F    00000021    C    Pyeongchang2018.com\\COS.SQLAdmin
.data:00428D70    00000010    C    [redacted]
.data:00428D84    00000021    C    Pyeongchang2018.com\\pca.dnsadmin
.data:00428DA5    00000010    C    [redacted]
.data:00428DB9    00000020    C    Pyeongchang2018.com\\PCA.imadmin
.data:00428DD9    0000000F    C    [redacted]
.data:00428DEC    00000022    C    Pyeongchang2018.com\\pca.perfadmin
.data:00428E0E    0000000D    C    [redacted]
.data:00428E1F    00000023    C    Pyeongchang2018.com\\jaesang.jeong6
```

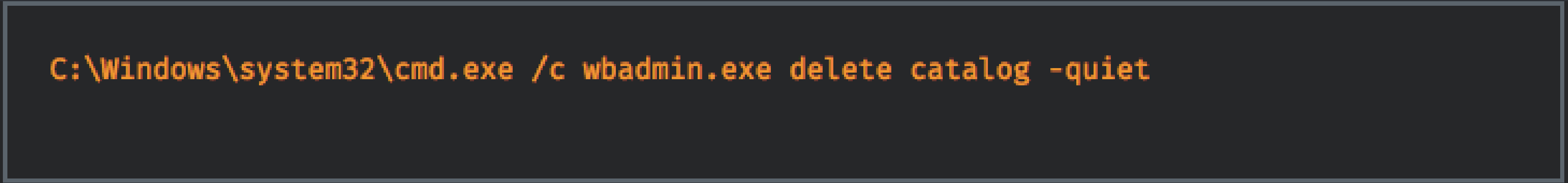
Destroyer

- Shadow copy destruction



```
C:\Windows\system32\cmd.exe /c c:\Windows\system32\vssadmin.exe delete shadows /all
/quiet
```

- Backup destruction

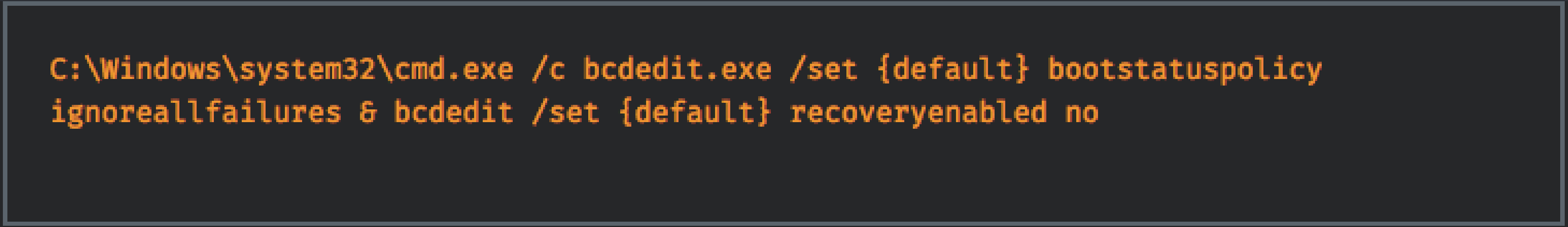


```
C:\Windows\system32\cmd.exe /c wbadmin.exe delete catalog -quiet
```

- Wipe files located on a mapped share folder

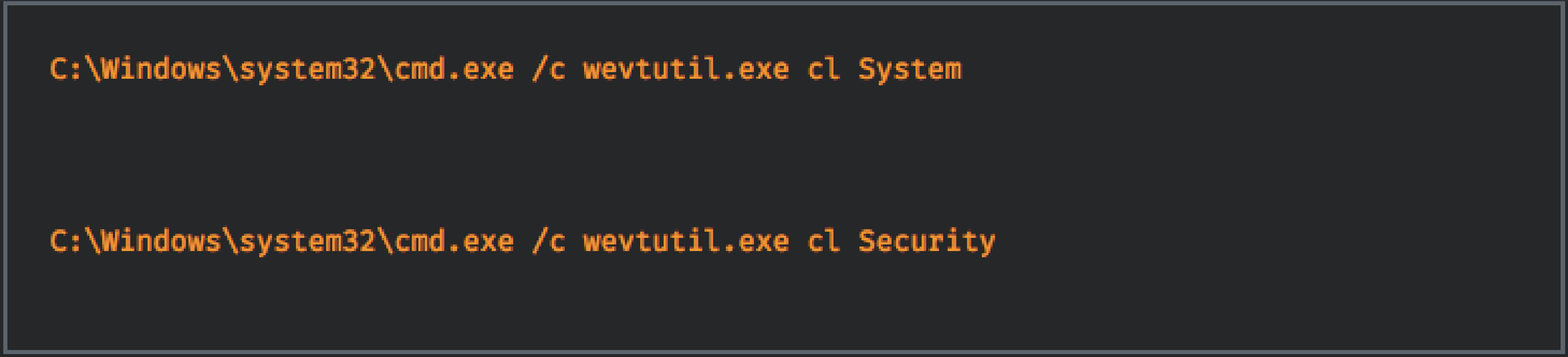
Destroyer

- Disable boot recovery



```
C:\Windows\system32\cmd.exe /c bcdedit.exe /set {default} bootstatuspolicy
ignoreallfailures & bcdedit /set {default} recoveryenabled no
```

- Event logs destruction

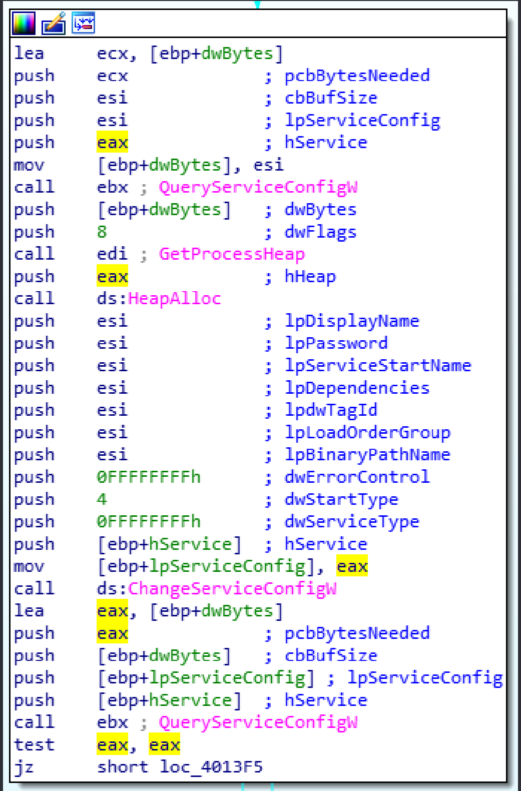


```
C:\Windows\system32\cmd.exe /c wevtutil.exe cl System


C:\Windows\system32\cmd.exe /c wevtutil.exe cl Security
```

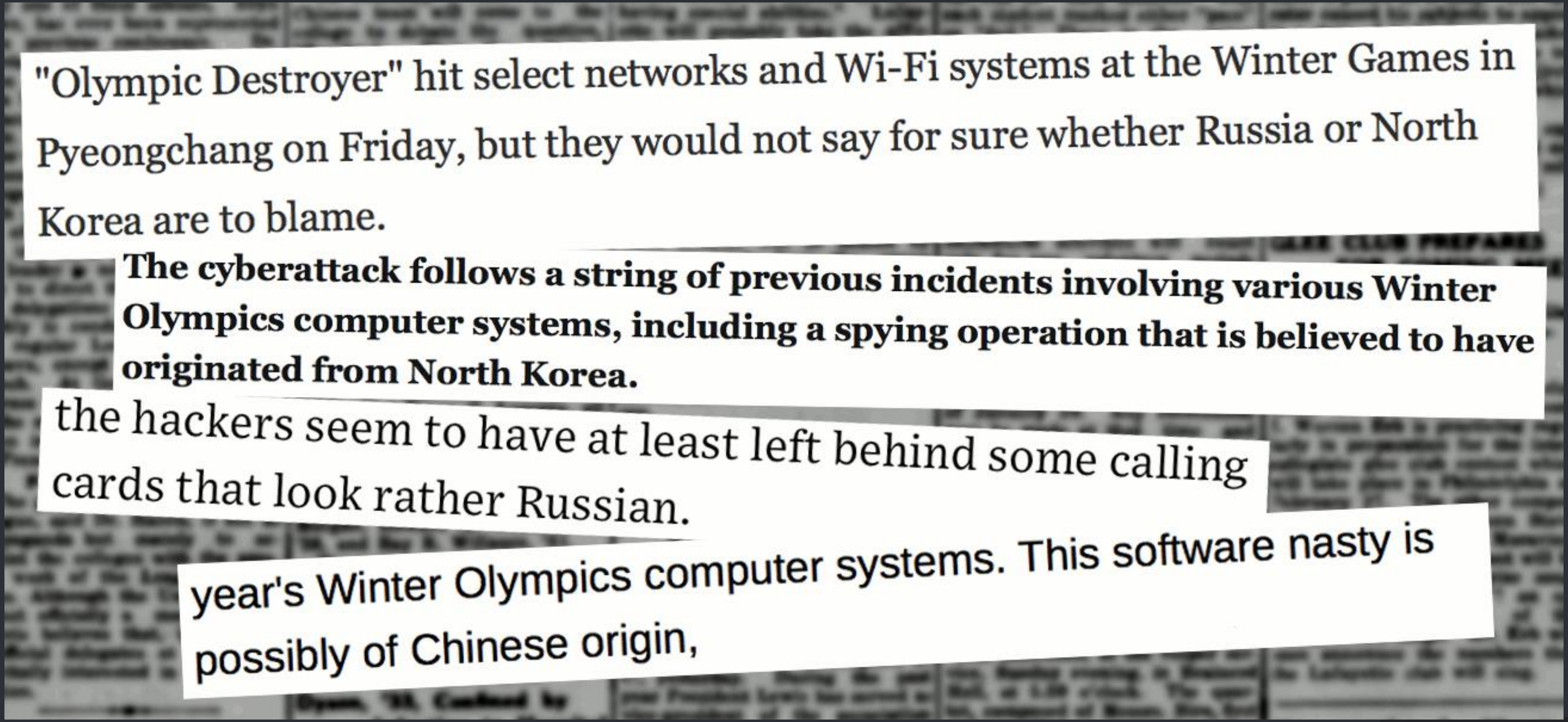
Destroyer

- Disable all Windows services



```
lea     ecx, [ebp+dwBytes]
push    ecx                 ; pcbBytesNeeded
push    esi                 ; cbBufSize
push    esi                 ; lpServiceConfig
push    eax                 ; hService
mov     [ebp+dwBytes], esi
call    ebx ; QueryServiceConfigW
push    [ebp+dwBytes]       ; dwBytes
push    8                   ; dwFlags
call    edi ; GetProcessHeap
push    eax                 ; hHeap
call    ds:HeapAlloc
push    esi                 ; lpDisplayName
push    esi                 ; lpPassword
push    esi                 ; lpServiceStartName
push    esi                 ; lpDependencies
push    esi                 ; lpdwTagId
push    esi                 ; lpLoadOrderGroup
push    esi                 ; lpBinaryPathName
push    0FFFFFFFFh          ; dwErrorControl
push    4                   ; dwStartType
push    0FFFFFFFFh          ; dwServiceType
push    [ebp+hService]      ; hService
mov     [ebp+lpServiceConfig], eax
call    ds:ChangeServiceConfigW
lea     eax, [ebp+dwBytes]
push    eax                 ; pcbBytesNeeded
push    [ebp+dwBytes]       ; cbBufSize
push    [ebp+lpServiceConfig] ; lpServiceConfig
push    [ebp+hService]      ; hService
call    ebx ; QueryServiceConfigW
test    eax, eax
jz      short loc_4013F5
```

Who Wasn’t Responsible?



"Olympic Destroyer" hit select networks and Wi-Fi systems at the Winter Games in Pyeongchang on Friday, but they would not say for sure whether Russia or North Korea are to blame.

The cyberattack follows a string of previous incidents involving various Winter Olympics computer systems, including a spying operation that is believed to have originated from North Korea.

the hackers seem to have at least left behind some calling cards that look rather Russian.

year's Winter Olympics computer systems. This software nasty is possibly of Chinese origin,

Who Wasn’t Responsible?

- Lazarus Group?
  - Same filename pattern than Bluenoroff group against the SWIFT infrastructure in a Bank in Bangladesh
  - Same wiper code: wiping the only first 0x1000 bytes of larges file

Who Wasn’t Responsible?

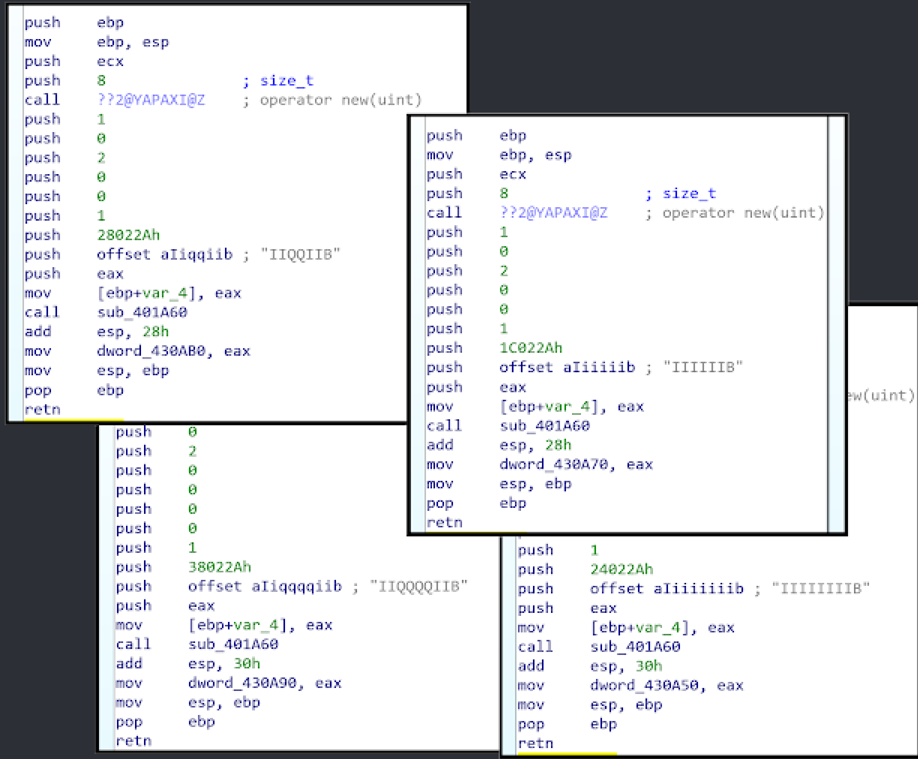
- APT 3 / APT 10 ?
  - Code sharing based on Intezer Labs analysis
    - Similarities in the credential stealer (based on Open Source code)
    - Similarities in the AES functions

Who Wasn't Responsible?

- Nyetya?
  - Same propagation technical (PsExec/WMI)
  - Same way to transer stolen credentials to the main module (named pipe)

Who Wasn’t Responsible?

- Nyetya?
  - ETERNALROMANCE trace
  - But no usage of the exploit…



```
push    ebp
mov     ebp, esp
push    ecx
push    8               ; size_t
call    ??2@YAPAXI@Z    ; operator new(uint)
push    1
push    0
push    2
push    0
push    0
push    1
push    28022Ah
push    offset aIiqqiib ; "IIQQIIB"
push    eax
mov     [ebp+var_4], eax
call    sub_401A60
add     esp, 28h
mov     dword_430AB0, eax
mov     esp, ebp
pop     ebp
retn
```

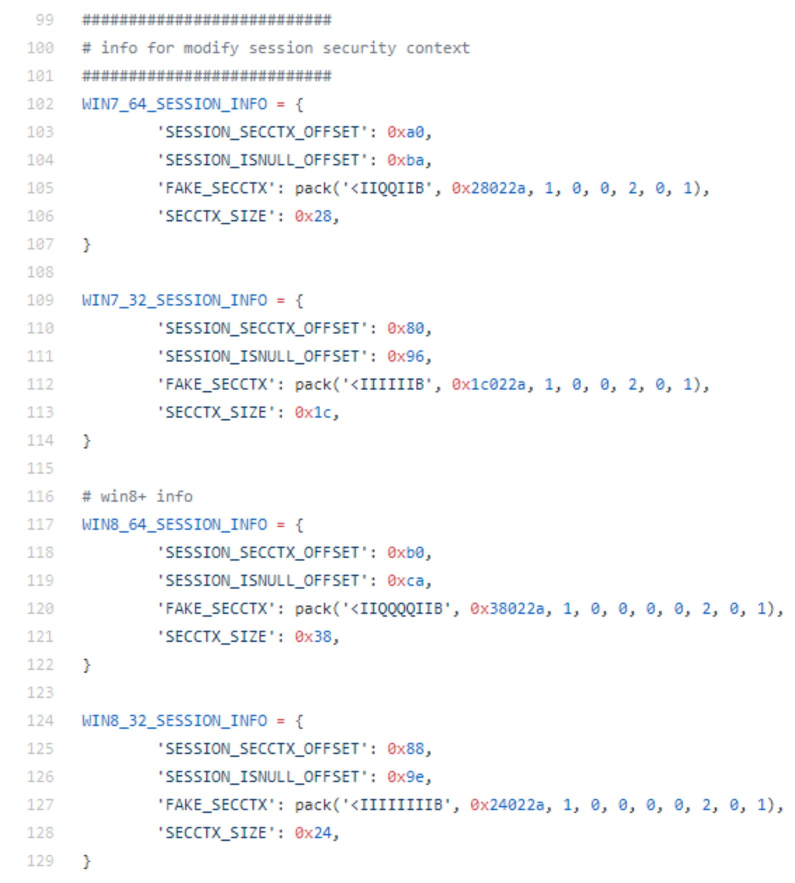
```
push    ebp
mov     ebp, esp
push    ecx
push    8               ; size_t
call    ??2@YAPAXI@Z    ; operator new(uint)
push    1
push    0
push    2
push    0
push    0
push    1
push    1C022Ah
push    offset aIiiiiib ; "IIIIIIB"
push    eax
mov     [ebp+var_4], eax
call    sub_401A60
add     esp, 28h
mov     dword_430A70, eax
mov     esp, ebp
pop     ebp
retn
```

```
push    0
push    2
push    0
push    0
push    0
push    0
push    1
push    38022Ah
push    offset aIiqqqiib ; "IIQQQIIB"
push    eax
mov     [ebp+var_4], eax
call    sub_401A60
add     esp, 30h
mov     dword_430A90, eax
mov     esp, ebp
pop     ebp
retn
```

```
ew(uint)
```

```
push    1
push    24022Ah
push    offset aIiiiiiib ; "IIIIIIIB"
push    eax
mov     [ebp+var_4], eax
call    sub_401A60
add     esp, 30h
mov     dword_430A50, eax
mov     esp, ebp
pop     ebp
retn
```

Who Wasn’t Responsible?



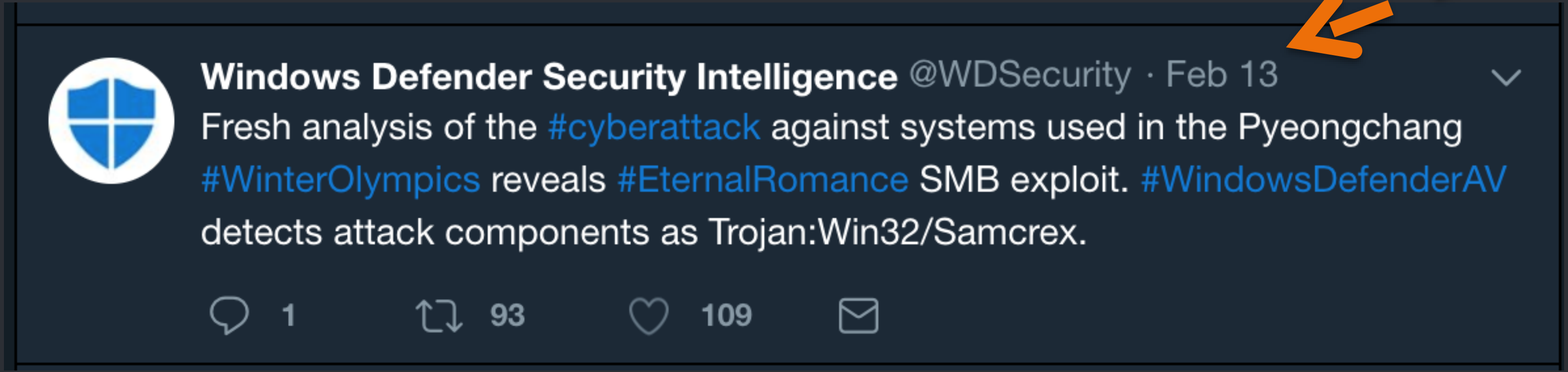
```
###########################
# info for modify session security context
###########################
WIN7_64_SESSION_INFO = {
        'SESSION_SECCTX_OFFSET': 0xa0,
        'SESSION_ISNULL_OFFSET': 0xba,
        'FAKE_SECCTX': pack('<IIQQIIB', 0x28022a, 1, 0, 0, 2, 0, 1),
        'SECCTX_SIZE': 0x28,
}

WIN7_32_SESSION_INFO = {
        'SESSION_SECCTX_OFFSET': 0x80,
        'SESSION_ISNULL_OFFSET': 0x96,
        'FAKE_SECCTX': pack('<IIIIIIB', 0x1c022a, 1, 0, 0, 2, 0, 1),
        'SECCTX_SIZE': 0x1c,
}

# win8+ info
WIN8_64_SESSION_INFO = {
        'SESSION_SECCTX_OFFSET': 0xb0,
        'SESSION_ISNULL_OFFSET': 0xca,
        'FAKE_SECCTX': pack('<IIQQQQIIB', 0x38022a, 1, 0, 0, 0, 0, 2, 0, 1),
        'SECCTX_SIZE': 0x38,
}

WIN8_32_SESSION_INFO = {
        'SESSION_SECCTX_OFFSET': 0x88,
        'SESSION_ISNULL_OFFSET': 0x9e,
        'FAKE_SECCTX': pack('<IIIIIIIIB', 0x24022a, 1, 0, 0, 0, 0, 2, 0, 1),
        'SECCTX_SIZE': 0x24,
}
```

Who Wasn’t Responsible?

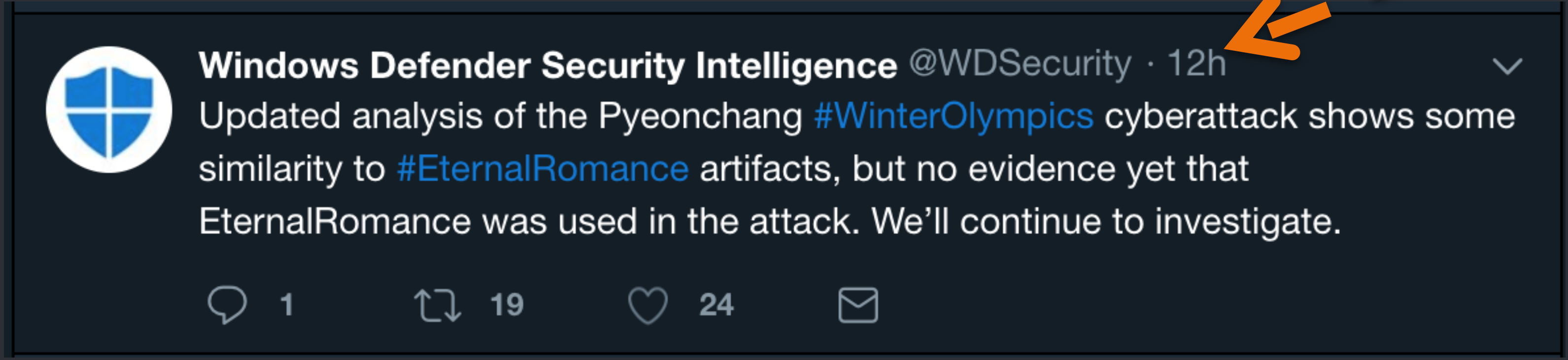


Windows Defender Security Intelligence @WDSecurity · Feb 13

Fresh analysis of the #cyberattack against systems used in the Pyeongchang #WinterOlympics reveals #EternalRomance SMB exploit. #WindowsDefenderAV detects attack components as Trojan:Win32/Samcrex.

- Tweet from Microsoft – February 13 2018

Who Wasn’t Responsible?



Windows Defender Security Intelligence @WDSecurity · 12h

Updated analysis of the Pyeonchang #WinterOlympics cyberattack shows some similarity to #EternalRomance artifacts, but no evidence yet that EternalRomance was used in the attack. We’ll continue to investigate.

- Tweet from Microsoft – February 14th 2018 showing that they now do not believe ETERNALROMANCE was used

Final Thoughts

- The author has purposefully included false attribution flags
- This could be taken to the extreme of a country denying an attack based on third party false attribution
- Attackers will continue to evolve & copy each other.
- Attribution based solely on information from malware samples is not accurate.

Stay Informed

Spreading security news, updates, and other information to the public

White papers, articles, & other information
talosintelligence.com

ThreatSource Newsletter
cs.co/TalosUpdate

Talos Blog
blog.talosintelligence.com

Social Media Posts
Facebook: TalosGroupatCisco
Twitter: @talossecurity

Instructional Videos
cs.co/talostube

Talos publically shares security information through numerous channels to help make the internet safer for everyone.

Q&A

TALOSINTELLIGENCE.COM

@talossecurity

blog.talosintelligence.com

@infosec_nick