初探威脅情資

Threat Intelligence 101: What, Why, How

Still Hsu



Persistent Cyber Threat Hunters

AGENDA







_ 報告 SOP

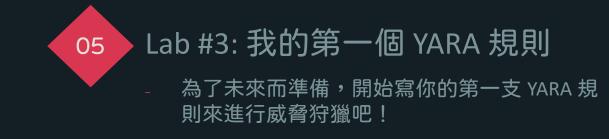
02

- 找尋敵人以及相關研究
- 研究事件樣本並拼湊出事件發生的由來

D3 Lab #1: 工欲善其事,必先利其器 - 調查事件總不能每次都空手硬幹或遇到心 樣本幹掉重練,學會用點工具吧!

04 Lab #2: 從蛛絲馬跡到破案關鍵

- 學會如何透過惡意樣本分析,一步一腳印 蒐集各式各樣的資訊,並找出背後的藏鏡 人。



whoami

Still Hsu

- ◆ BEL, English Dep. @ NPTU (屏東大學)
 - Pingtung Hacker TA
- Threat Intelligence Researcher @ TeamT5
- Interested in...
 - Windows internals
 - ♦ .NET
 - Anything and everything!
- Participated in...
 - AIS3 2019/2020
 - ◆ 第四屆臺灣好厲駭
- Spoken @...
 - ◆ FCU (逢甲大學)
 - ◆ NSYSU (中山大學)
 - HITCON 2020 Lightning Talk





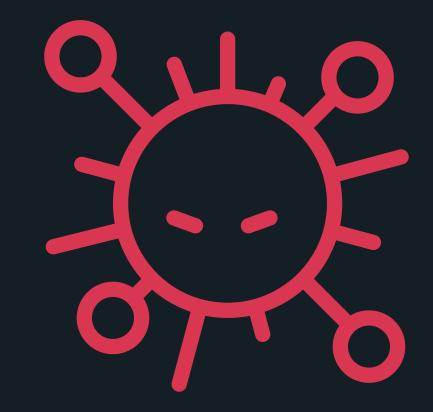
Disclaimer

This lab session assumes you have...

- Basic reverse engineering skills
- A disassembler/decompiler installed
 - Preferably IDA Pro, though any other ones are fine
- A **CONTAINED ENVIRONMENT** for testing (e.g., VM) that should be **OFFLINE**

The lab session WILL require interaction with a real malware.

If you are not confident enough, don't risk it. Feel free to watch others do it instead.





Introduction to CTI



"Threat intelligence is data that is collected, processed, and analyzed to understand a threat actor's motives, targets, and attack behaviors."

- (CrowdStrike, 2021)





- Past
- Present
- Future

Records and collects indicators of cyberthreats Provides additional insights into these threats

- Tactics
- Operations
- Strategies



- Past
- Present
- Future

Records and collects indicators of cyberthreats Provides additional insights into these threats

- Tactics
- Operations
- Strategies

• Reduce impact

- Improve IR efficiency
- Gain better knowledge in the cyberthreat landscape

Helps businesses plan for mitigations



Why CTI?

A decade ago...

Typical incident responses

Process and provide feedback as cases come along

Number of cases were few and far between

Most of them were trivial

Relatively easy to handle



Why CTI?

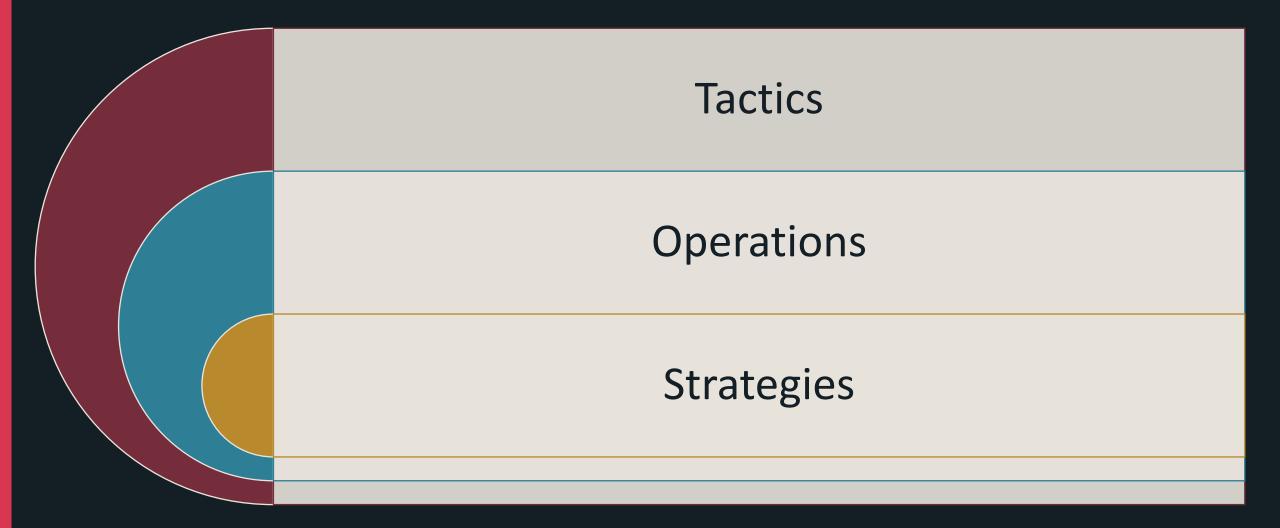
A decade _ ago	Typical incident responses
	Process and provide feedback as cases come along
	Number of cases were few and far between
-	Most of them were trivial
_	Relatively easy to handle
Now	Complex attacks are now being carried out by APTs worldwide
	Number of sophisticated social engineering attacks have been increasing yearly

Signature-based defenses are being defeated left and right





Who uses CTI?





Who uses CTI?





Sounds scary...

I don't know anything about threat research.



Neither did I!

Lifecycle



Ask Yourself These...

What is the most significant threat?

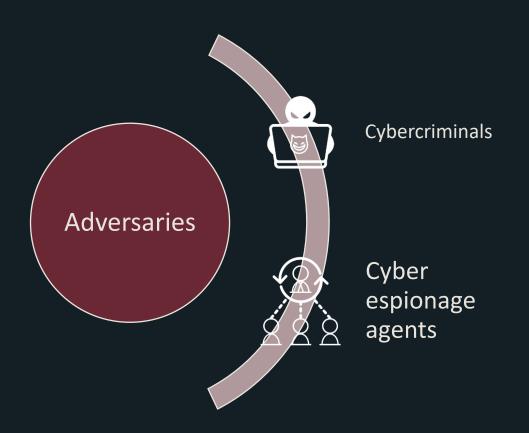
How to prioritize the threats?

Who will consume and benefit from the finished product?

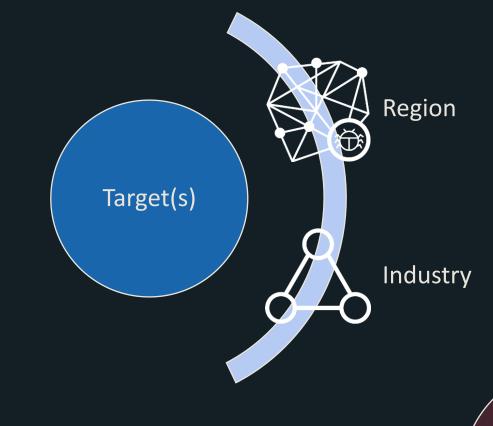
Planning & Direction

Cyber Attacks

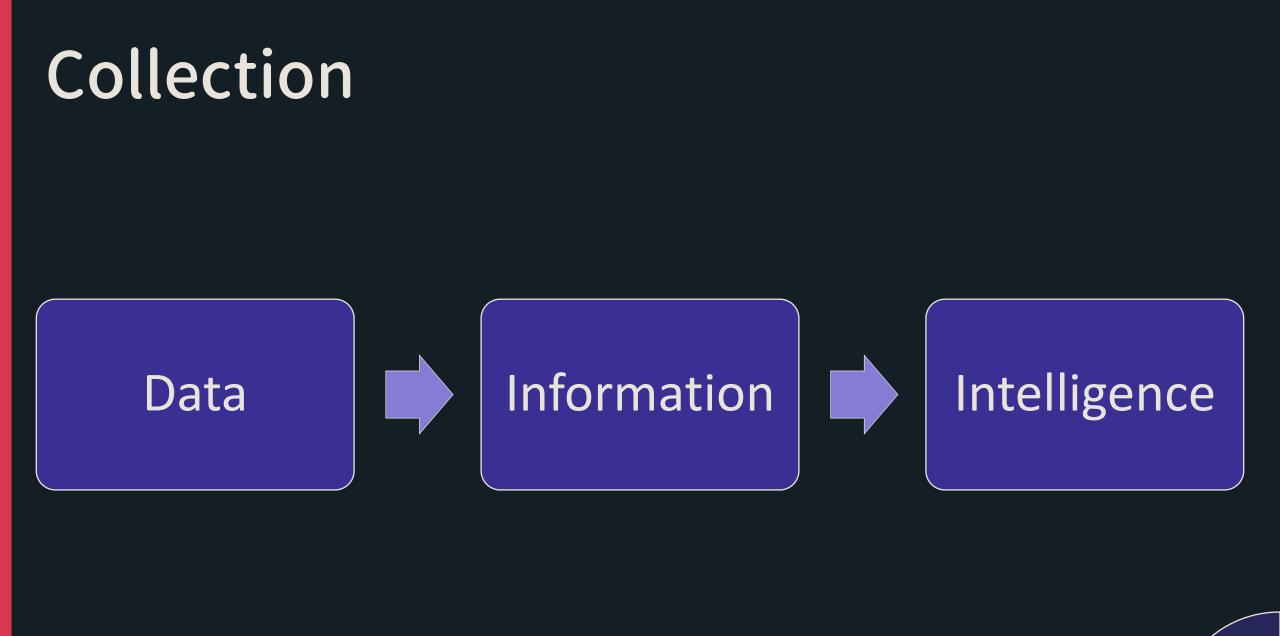
Type of adversaries



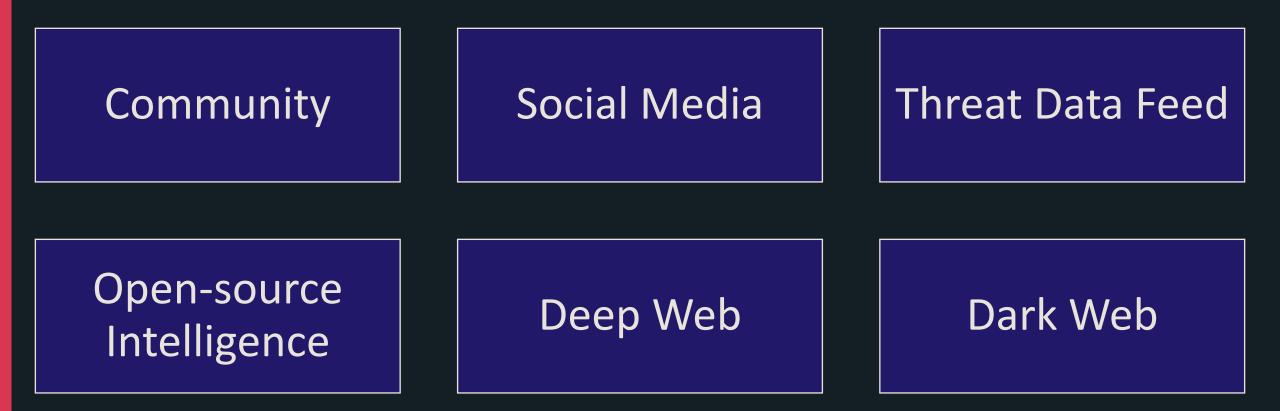
Information about the target(s)



Planning & Direction



External Source



Internal Source



Diamond Model

- Reconnaissance techniques
- Delivery methods
- Attacking exploit / vulnerability
- Remote control malware / backdoor
- Lateral movement skills and tools
- Data stealing techniques

CAPABILITY 🎢

- Purpose
- Target countries / regions
- Target sectors
- Target individuals
- Target data

ADVERSARY

TARGET

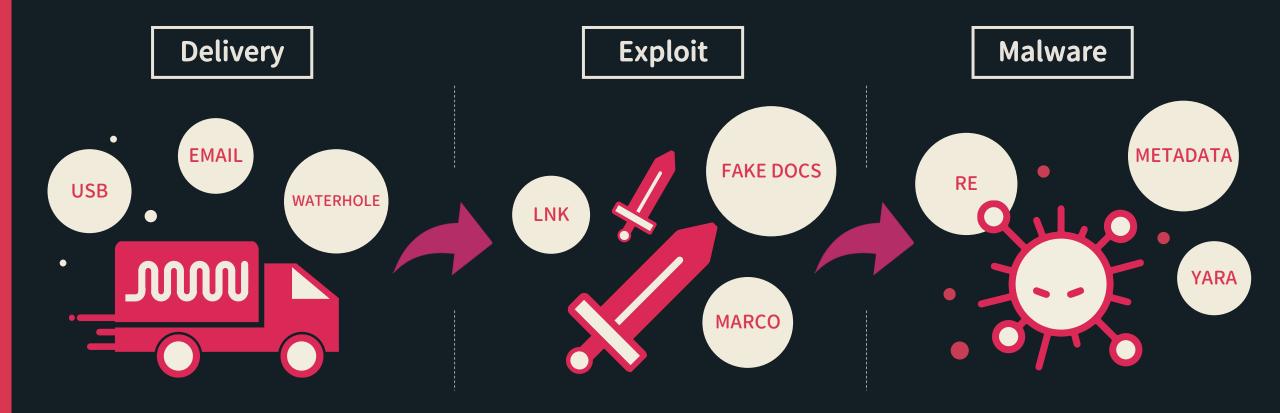
- Where are they from?
- Who are they?
- Who is sponsoring them?
- Why do they attack?
- Campaign timeline and plan

کر اا (ک

INFRASTRUCTURE

- C2 Domain names
- Location of C2 servers
- Type of C2 servers
- Compromised machines
- C2 management mechanism and structure
- Path of Control and data leakage

Capability Analysis



Infrastructure Analysis

Domain

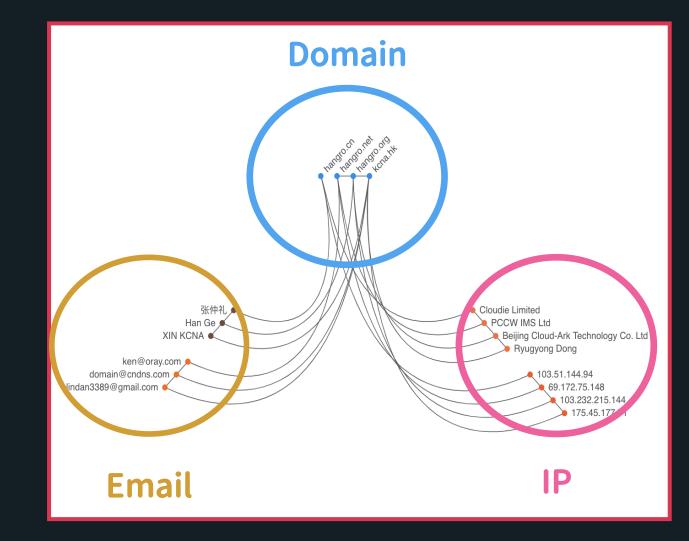
- WHOIS -> Email
- Passive DNS -> IP

♦ IP

Passive DNS -> Domain

Email

Reverse WHOIS - > Domain



Adversary Analysis



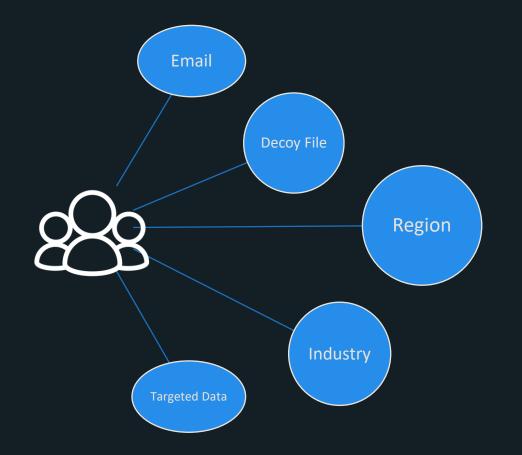
Actors

- ♦ Language
- Tools
- Infrastructure
- Time zone
- Motivations, intentions
- Cooperation relationship between different groups
 - Sharing Tool
 - Sharing C2

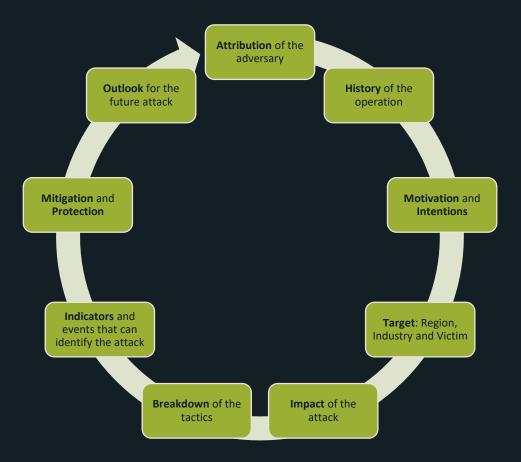


Target Analysis

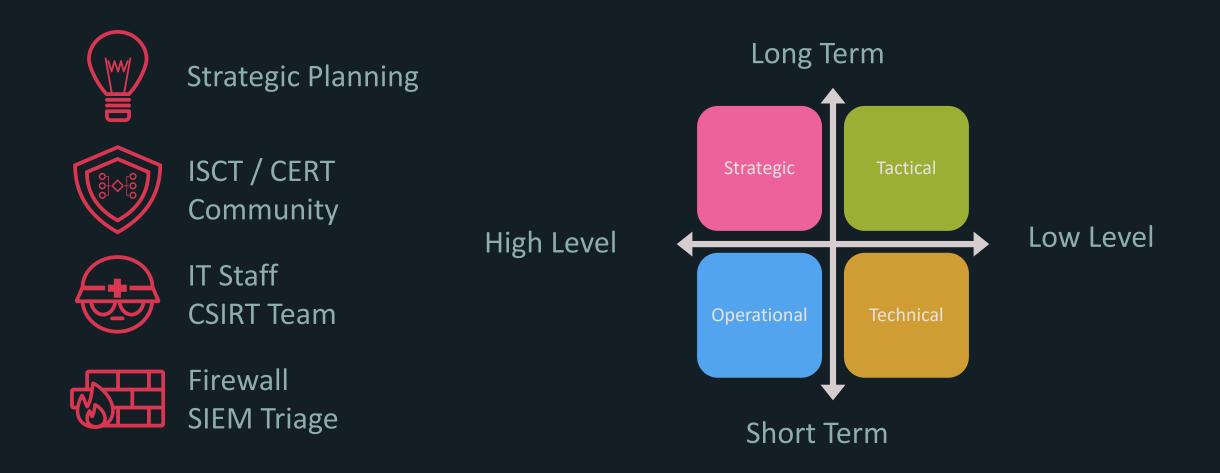
Victim Analysis



Threat Analysis Report



Dissemination & Integration

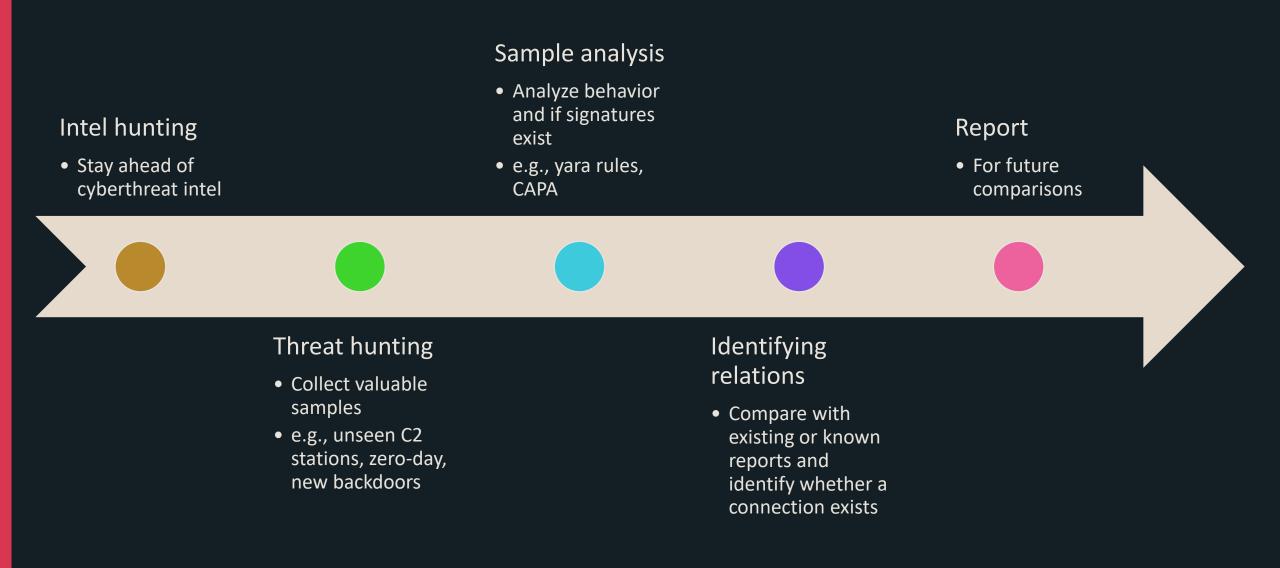


Writing a CTI Report



Standard Operating Procedures





Adversary & Intel Research



Sanyo



TEAMT5

Targets

- ♦ IN, JP, MN, RU, US, KR
- Government, Defense, Telecom, Aerospace, Heavy Industry
- Aliases
 - Tonto Team, Karma Panda

Description

- Sanyo Team has been active for at least 10 years.
- Their malwares are very simple and lack all kinds of intricate skills, such as rootkit, injection or selfprotection techniques.
- The Sanyo actors have so far shown their capability by successfully breaching several defense contractors and heavy industries in the world.

GuDiao

Targets

- ♦ HK, MY, PH, VN
- Dissident, Military, Government
- Description
 - Related to other Chinese APT groups
 - The group mainly aims at governments and military units in South East Asia, such as Vietnam and Malaysia.
 - In recent years, it has developed its own malwares and adopted the RoyalRoad exploit, which is popular among Chinese APT groups.





Lapis





- Targets
 - ♦ AF, IN
 - Military, Government, Foreign, Affairs
- Aliases
 - C-Major, Transparent Tribe

Description

- Lapis is an APT group active in South Asia. More specifically, almost all attacks were against India or Indian related organizations.
- The group has been well-known to the public around 2016, as many security vendors revealed its attacks against the Indian government.

OceanLotus

Targets

- ◆ KH, CN, JP, SG, TH, VN
- Government, Automobile, Financial
- Aliases
 - ♦ APT32
- Description
 - OceanLotus is a Vietnam APT group active since at least 2012.
 - Found targeting private sector companies in Southeast Asia.
 - Believed to be a state-sponsored APT group.





CloudDragon



- Targets
 - ♦ JP, US, KR
- Aliases
 - Kimsuky, Thallium
- Description
 - Two groups were created, named CloudDragon and KimDragon, as we observed different TTP in the recent years.
 - Main target is South Korea.
 - Recently began to attack United States and Japan as well.



Threat/Intel Hunting Resources



- Twitter
 - ♦ #APT
 - @cyberwar_15
 - ♦ @Timele9527
 - Oblackorbird
 - @Rmy_Reserve
 - @_re_fox

- Curated Resources
 - https://start.me/p/rxRbpo/ti

Threat/Intel Hunting Resources



Yara rules

- ♦ Yara-Rules/rules @ GitHub
- ◆ InQuest/awesome-yara @ GitHub
- ♦ <u>Neo23x0/signature-base</u> @ GitHub

CAPA

◆ <u>FireEye/CAPA</u> @ GitHub

Manual analysis

- Behavior analysis via sandboxes
 - e.g., cuckoo, CAPEv2, etc.
- Static analysis via disassemblers
 - e.g., IDA Pro, Ghidra, etc.
- Dynamic analysis via contained environments
 - e.g., virtual machines, physical bare-bones

Threat/Intel Hunting Resources



Open Sandbox Platforms

- Any.Run
 - Requires registration
- VirusTotal
 - Requires enterprise license to download sample
- CAPEv2
- Hybrid-Analysis
 - Requires approval by filling out the vetting form

MITRE ATT&CK

- CTI news outlets/blogs
 - FireEye Threat Research Blog
 - ♦ JPCERT Blog
 - Kaspersky Lab Resource Center
 - Check Point Software Blog
 - ...many more.

Content of a Report



How did the incident occur?

- Delivery method(s)
- Phishing method(s)/theme(s)
- Exploitation method(s)

Content of a Report



How did the incident occur?

- Delivery method(s)
- Phishing method(s)/theme(s)
- Exploitation method(s)

What did it cause?

- Summary of the malicious behaviors
- IOC (Indicator of Compromise)

Content of a Report



Who did it?

- Source infrastructure analysis
- Piece everything together with existing reports

How did the incident occur?

- Delivery method(s)
- Phishing method(s)/theme(s)
- Exploitation method(s)

What did it cause?

- Summary of the malicious behaviors
- IOC (Indicator of Compromise)

How did the incident occur?



Deli	Delivery Methods								
	Spear-phishing email								
0—III 0—III 0—III 0—III	Watering hole attack								
	Supply chain attack								

Delivery Methods

Spear-phishing email



Targeted attack

- Typically used against high-profile individuals or company head
- e.g., CEO, head of a division, activists

Social engineering

- Sensitive subject matter
- e.g., something that involves sense of urgency
- Disguised as legitimate corporate email
 - Potentially contains malicious attachments or links

ing hole attack

Delivery Methods

phishing email



Watering hole attack

Compromise sites that victim frequents

EAMT5

- Drive-by via malvertisements or domain redirection
- Example
 - Holy Water campaign in 2020
 - Targeted religious and charity websites



Delivery Methods



ing hole attack



Supply chain attack

- Compromise components from supply chains
 - e.g., software update hosts
- Easily wide-spread as these software components may be mass distributed (i.e., from part of a supply chain)
- Example
 - ASUS ShadowHammer in 2019

Exploit Methods





Fake documents

• Executables or shortcuts (LNK) with document icons

Exploit Methods



Fake documents

• Executables or shortcuts (LNK) with document icons



Malicious documents

• Macro

- Object Linking and Embedding (OLE)
- Unpatched RCE CVEs (CVE-2018-0798)

Exploit Methods



Fake documents

• Executables or shortcuts (LNK) with document icons



Malicious documents

- Macro
- Object Linking and Embedding (OLE)
- Unpatched RCE CVEs (CVE-2018-0798)



Software vulnerabilities

- CVE-2018-20250 (WinRAR ACE)
- CVE-2018-15982 (Flash Player use-after-free)
- Other CVEs or zero-days

What did it cause?



Malware Analysis





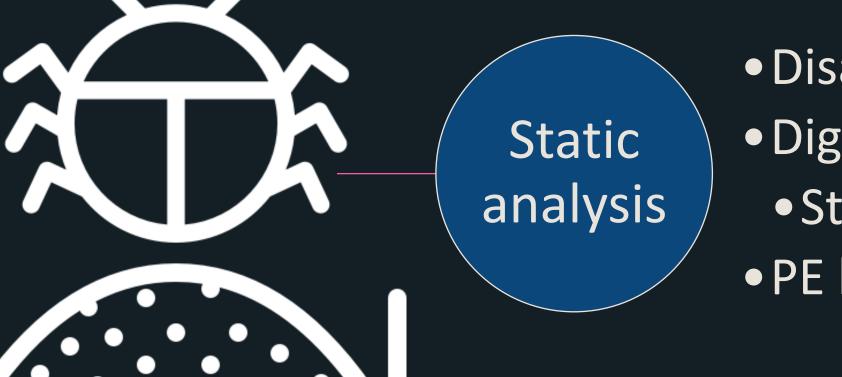


• Preferably offline

 Otherwise, connect to a VPN/TOR at host-level

Malware Analysis

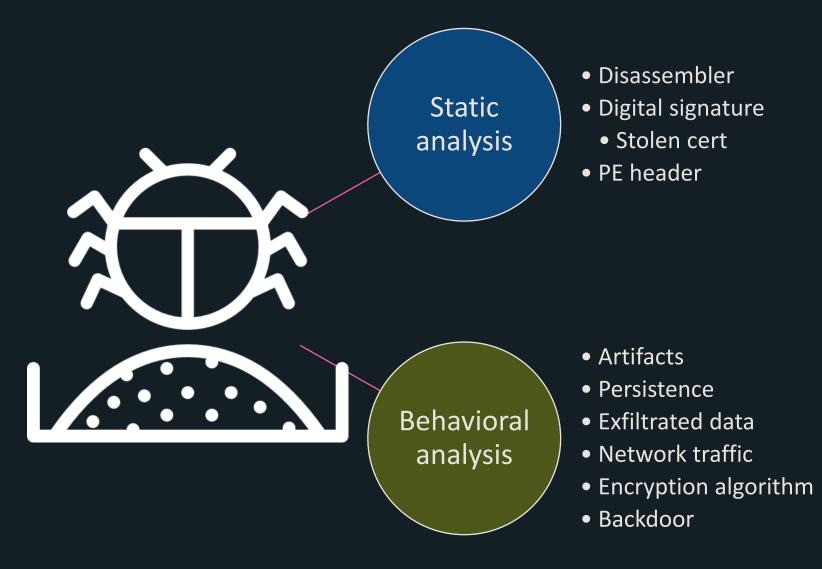




Disassembler
Digital signature
Stolen cert
PE header

Malware Analysis





Who did it?



Infrastructure Analysis



 Virtual Private Server
 (VPS)
 e.g., Linode, Digital Ocean, Aliyun, AWS, GCP

Web hosting

 e.g., hostinger, Bluehost, SiteGround



Compromised server

i.e., privately owned by an individual, overtaken by threat actor

Virtual Private Server (VPS)



Rented or bought by the threat actor

- Usually assigned a fixed and unique IP
- Threat actor has complete control over the server
 - Open certain ports or services for backdoor connection
 - Connect via SSH/RDP

Web Hosting



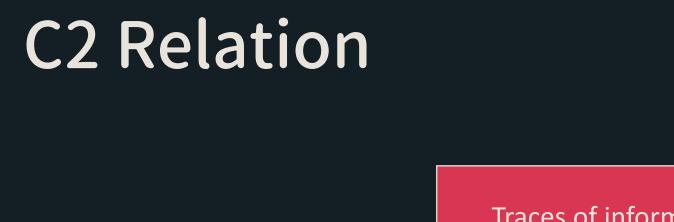
Free/paid

- Two or more users may share the same machine
 - More than one domain may resolve to the same IP address or set of addresses
 - Threat actors could only access the frontend
 - Implemented alongside simple backdoors or only used to serve malicious files

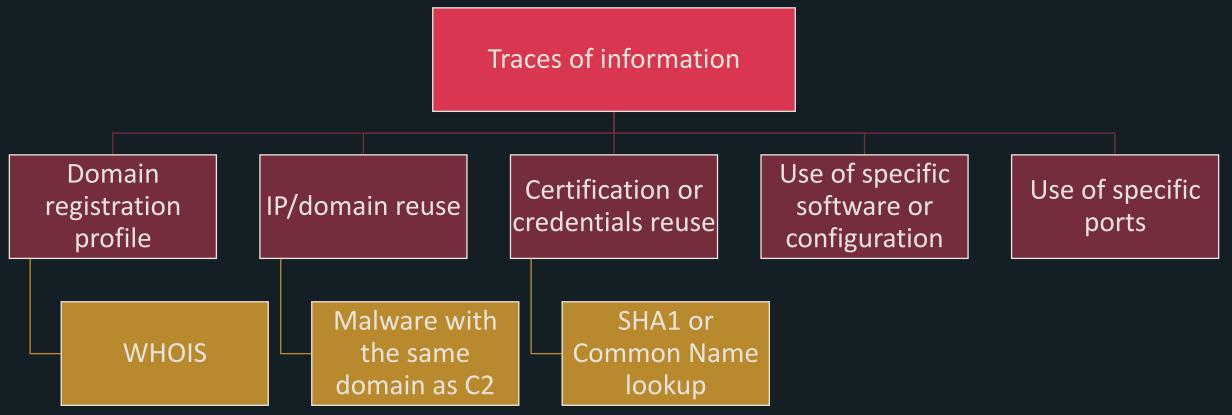
Compromised Server



- Unauthorized access via...
 - Web application vulnerabilities
 - Software vulnerabilities
 - Compromised credentials
- Access level highly depends on the method of intrusion
- Backdoors are generally well-hidden to avoid raising suspicion

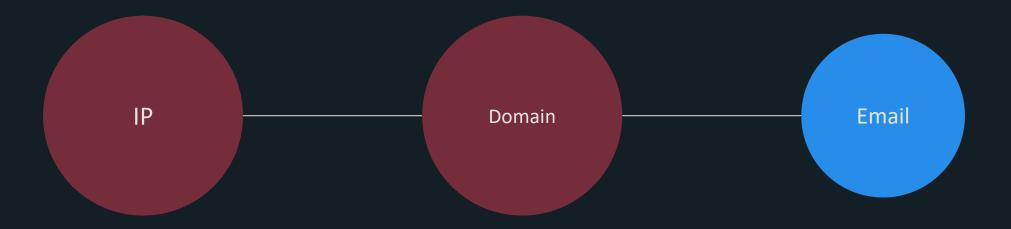






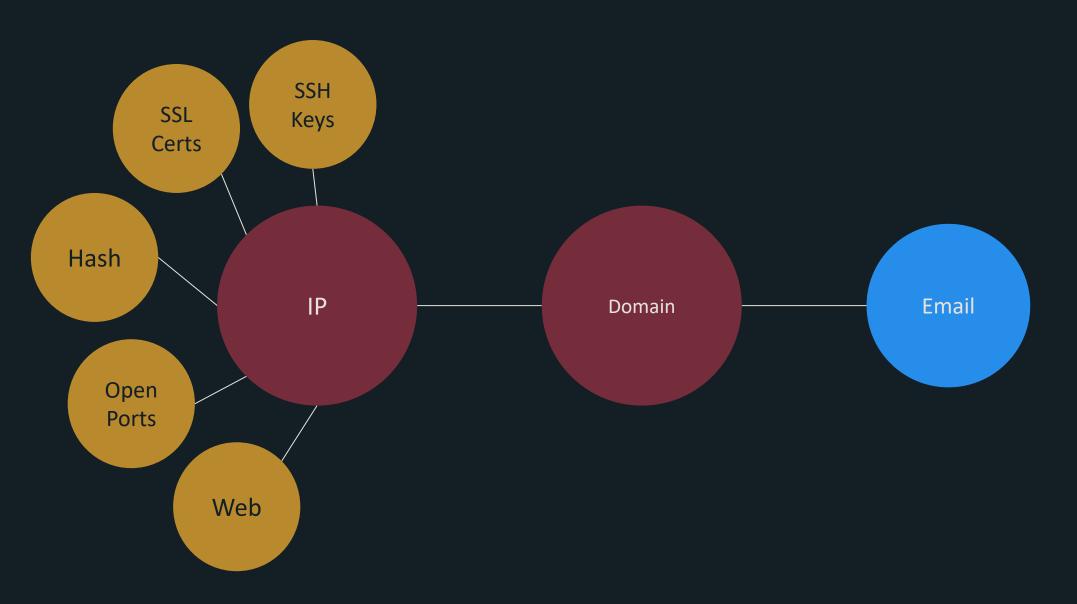
C2 Relation





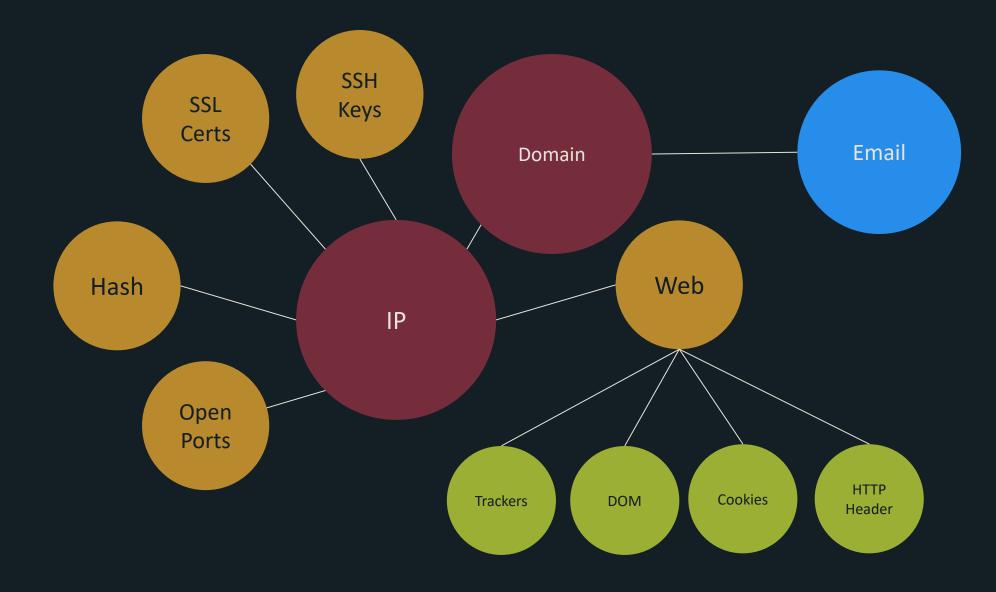
C2 Relation





C2 Relation





Compare Findings

Collect OSINT resources

- Other analysists' view or thoughts
 - Twitter, Medium, blogs, etc.
- Existing reports on the sample published by another security firm or researcher
 - FireEye, Kaspersky, CrowdStrike, Malwarebytes, etc.
- Personal or internal documents
 - Look for past records in the archive, if any
 - Cross-compare C2 used, behaviors exhibited, peculiar strings, etc.



EAMT5

Lab #1: Getting the Hang of Tools





Windows Sysinternals 03/23/2021 • 4 minutes to read • 🕐 🎱 🌗 🌘 😚 😝

The Sysinternals web site was created in 1996 by Mark Russinovich 🖒 to host his advanced system utilities and technical information. Whether you're an IT Pro or a developer, you'll find Sysinternals utilities to help you manage, troubleshoot and diagnose your Windows systems and applications.

- Read the official guide to the Sysinternals tools, Troubleshooting with the Windows Sysinternals Tools
- $\bullet\,$ Read the Sysinternals $\operatorname{Blog}{\tt L}^{\!2}\,$ for a detailed change feed of tool updates
- Watch Mark's Sysinternals Update videos on YouTube $\ensuremath{ \ensuremath{ \e$
- Watch Mark's top-rated Case-of-the-Unexplained troubleshooting presentations and other webcasts
- Read Mark's Blog 🖉 which highlight use of the tools to solve real problems
- Check out the Sysinternals Learning Resources page
- $\bullet\,$ Post your questions in the Sysinternals Forum ${\bf e}^{\!\scriptscriptstyle 2}$

(Microsoft Corp., 2021)

Description

- Originally third-party, now acquired by Microsoft
- Contains a series of tools for system management and Windows debugging



Procmon

- Process Monitor is an advanced monitoring tool for Windows that shows real-time file system, Registry and process/thread activity.
- Helps give a quick idea of what the malware will do upon startup.

🔊 Process Monitor - Sys	internals	: www.sysinternals	.com — 🗆 🗙	<
File Edit Event Filter	Tools	Options Help		
😅 🖬 💸 🕮 🖾	ቅ ₽	🌚 🗉 🖊 🦻		
Time Process Name	PID	Operation	Path	\mathbf{h}
5:04:0 svchost.exe	3168	KegCloseKey	HKLM	
5:04:0 💽 svchost.exe		RegClose Key	HKLM	
5:04:0 💽 svchost.exe	3168	RegQueryValue	HKLM\System\CurrentControlSet\Services\NetBT\Linkage\Export	
5:04:0 💽 svchost.exe		RegQuery Value	HKLM\System\CurrentControlSet\Services\NetBT\Linkage\Export	
5:04:0 💽 svchost.exe	3168	RegQuery Value	HKLM\System\CurrentControlSet\Services\NetBT\Linkage\Export	
5:04:0 💽 svchost.exe	3168	RegQuery Value	HKLM\System\CurrentControlSet\Services\NetBT\Linkage\Export	
5:04:0 💽 svchost.exe	3168	RegQuery Value	HKLM\System\CurrentControlSet\Services\NetBT\Linkage\Export	
5:04:0 💽 svchost.exe	3168	KegQueryValue	HKLM\System\CurrentControlSet\Services\NetBT\Linkage\Export	
5:04:0 💽 svchost.exe	3168	式 RegClose Key	HKLM\System\CurrentControlSet\Services\NetBT\Linkage	
5:04:0 💽 svchost.exe	3168	🕰 RegClose Key	HKLM\System\CurrentControlSet\Services\NetBT\Linkage	
5:04:0 Fontdrvhost.exe	1644	🛃 Read File	C:\Windows\System32\fontdrvhost.exe	
5:04:0 Fontdrvhost.exe	1644	🛃 Read File	C:\Windows\System32\fontdrvhost.exe	
5:04:0 Fontdrvhost.exe	1644	🛃 Read File	C:\Windows\System32\fontdrvhost.exe	
5:04:0 Fontdrvhost.exe	1644	🛃 Read File	C:\Windows\System32\fontdrvhost.exe	
5:04:0 Interventional 5:04:0	1644	🛃 Read File	C:\Windows\Fonts\micross.ttf	
5:04:0 Fontdrvhost.exe	1644	🛃 Read File	C:\Windows\Fonts\micross.ttf	
5:04:0 💷 dwm.exe	1720	RegQueryValue	HKCU\SOFTWARE\Microsoft\Windows\DWM\ColorPrevalence	
5:04:0 💶 dwm.exe	1720	RegQueryValue	HKCU\SOFTWARE\Microsoft\Windows\DWM\ColorPrevalence	
5:04:0 💶 dwm.exe	1720	RegQueryValue	HKCU\SOFTWARE\Microsoft\Windows\DWM\ColorPrevalence	
5:04:0 💶 dwm.exe	1720	RegQuery Value	HKCU\SOFTWARE\Microsoft\Windows\DWM\ColorPrevalence	
5:04:0 💶 dwm.exe	1720	RegQueryValue	HKCU\SOFTWARE\Microsoft\Windows\DWM\ColorPrevalence	
5:04:0 💶 dwm.exe	1720	RegQueryValue	HKCU\SOFTWARE\Microsoft\Windows\DWM\ColorPrevalence	
5:04:0 💷 dwm.exe		RegQueryValue	HKCU\SOFTWARE\Microsoft\Windows\DWM\ColorPrevalence	
5:04:0 💶 dwm.exe	1720	RegQuervValue	HKCU\SOFTWARE\Microsoft\Windows\DWM\ColorPrevalence	~
<			>	
Showing 4,919 of 77,527 ev	ents (6.3	%) Ba	icked by virtual memory	



🖅 Autoruns - Sysinternal	s: www.sysinternals.com		
File Entry Options He	elp		
📓 🖹 🏦 🖸 🗙 🗾	Filter:		
🚳 KnownDLLs	🚇 Winlogon	🛸 Winsock Providers	🍓 Print Mor
🖾 Everything 🏾 🔬	Logon 🛛 💆 Explorer	🥭 Internet Explorer	🔯 Scheduled Tasks
Autorun Entry	Description	Publisher	
HKLM\SYSTEM\Current	tControlSet\Control\SafeBoot\	AlternateShell	
🗹 📷 cmd.exe	Windows Command	Processor (Verified) N	Nicrosoft Windows
HKLM\SOFTWARE\Mid	crosoft\Windows\CurrentVersio	on\Run	
🗹 🔎 Everything	Everything	(Verified) v	oidtools
HKLM\SOFTWARE\Wo	w6432Node\Microsoft\Windo	ows\CurrentVersion\Run	
vmware-tray.exe	VMware Tray Proce	ess (Verified) V	/Mware, Inc.
HKCU\SOFTWARE\Mid	crosoft\Windows\CurrentVersio	on\Run	
🗹 👸 Google Update	Google Update Con	e (Verified) G	ioogle LLC

AutoRuns

- AutoRuns shows you what programs are configured to run during system bootup or login, and when you start various built-in Windows applications like Internet Explorer, Explorer and media players.
- Quick overview of the existing persistence entries on the machine.



Process Explorer

- Process Explorer shows you information about which handles, and DLLs processes have opened or loaded.
- Buffed up Task Manager, useful for dynamic analysis (e.g., memory dump, handle listing, etc.)

🎝 Process Explorer - Sysinternals: www.sysinternals.com — 🗆 🗙										
File Options View Process Find Users Help										
	* 4	•						,		
Process	CPU	Private Bytes	Working Set	PID	Description	Company Name		^		
Secure System	Susp	184 K	39,940 K	72						
Registry		18,684 K	53,284 K	132						
System Idle Process	42.44	60 K	8 K	0						
🖃 🔳 System	4.49	208 K	2,692 K	4						
Interrupts	1.75	0 K	0 K	n/a	Hardware Interrupts and DPCs					
smss.exe		1,076 K	940 K	584						
Memory Compression	8.11	2,884 K	291,332 K	3440						
Csrss.exe	< 0.01	2,036 K	3,520 K	900						
🖃 📰 wininit.exe		1,412 K	3,592 K	996						
services.exe	0.95	8,660 K	10,976 K	660						
svchost.exe	0.01	15,896 K	27,320 K	1164	Host Process for Windows S	Microsoft Corporation				
unsecapp.exe		1,752 K	5,204 K	4068						
WmiPrvSE.exe		16,440 K	21,832 K	3896						
WmiPrvSE.exe		2,960 K	8,284 K	4160						
dlhost.exe	< 0.01	3.452 K	7,616 K	5764						
iqfxext.exe		2,880 K	4,988 K	8196	iqfxext Module	Intel Corporation				
RuntimeBroker.exe		7.052 K	22.884 K	9492	Runtime Broker	Microsoft Corporation				
RuntimeBroker.exe	0.01	14,520 K	41,128 K	9836	Runtime Broker	Microsoft Corporation				
SettingSyncHost.exe	< 0.01	9,776 K	7,588 K	10088	Host Process for Setting Syn	Microsoft Corporation				
RuntimeBroker.exe		8,200 K	22,672 K	10284	Runtime Broker	Microsoft Corporation				
unsecapp.exe		1,932 K	6,480 K	13428	Sink to receive asynchronou	Microsoft Corporation				
GCC.exe		24,208 K		15256		Intel Corporation				
Runtime Broker.exe		2,168 K	5,432 K	15196	Runtime Broker	Microsoft Corporation				
Application Frame Host	< 0.01	25,744 K	22,660 K	16296	Application Frame Host	Microsoft Corporation				
Win Store.App.exe	Susp	53,420 K	2,940 K			Microsoft Corporation				
RuntimeBroker.exe		5,392 K	8,036 K	13520	Runtime Broker	Microsoft Corporation				
SystemSettings.exe	Susp	22,460 K	2,600 K	6032	Settings	Microsoft Corporation				
Calculator.exe	Susp	24,216 K	2,388 K	15660						
RuntimeBroker.exe		1,328 K	4,236 K	14252	Runtime Broker	Microsoft Corporation		~		
	57 100/		DI	74.7		10 0.0		-		

CPU Usage: 57.56% Commit Charge: 57.19% Processes: 284 Physical Usage: 74.38%

Other Third-party Tools



📮 Process Hacker	(Administrator)												
Hacker View Tools Users H	lelp												
🤹 Refresh 🛛 🎲 Options 🕴 📸 I													
Processes Services Network Dis	k Firewall												
Name	PID CPU I/O to	otal rate Private bytes User name Description											
✓ ■ System Idle Process	DAX3APLexe (552	DAX3API.exe (5524) Properties - X											
💙 📧 System													
📧 Secure System	Services .NE	ET assemblies .NET performance GPU Disk and Network Comment Windows											
Registry	General Statistic	s Performance Threads Token Modules Memory Environment Handles Job											
📧 smss.exe	File												
Memory Compr	N/A												
Interrupts	(Verified) Dolby Laboratories, Inc.												
csrss.exe	Version: 1.1.9.33												
✓ ■ wininit.exe	Image file name (Win32):												
✓ ■ services.exe	C:\Program Files\	Polby\Polby DAX3\API\DAX3API.exe											
✓ ■ svchost.exe	Image file name:	Image file name:											
unsecapp.exe	\Device\HarddiskVolume3\Program Files\Dolby\Dolby DAX3\API\DAX3API.exe												
WmiPrvSE.exe	Process												
WmiPrvSE.exe	Command line:	nd line: "C: \Program Files\Dolby\Dolby DAX3\API\DAX3API.exe"											
iafxext.exe													
RuntimeBroke	Current directory:	Current directory: C:\Windows\system32\											
RuntimeBroke	Started:	Started: a day and 8 hours ago (8:57:18 AM 3/30/2021)											
SettingSyncH	Parent console:	services.exe (660) Image type: 64-bit											
RuntimeBroke	Parent process:	services.exe (660)											
unsecapp.exe													
IGCC.exe	Mitigation policies:	DEP (permanent); ASLR (high entropy) Details											
RuntimeBroke	Protection: None	Permissions Terminate											
ApplicationFr													

Process Hacker

- Community-maintained procmon clone; actively maintained (v3 nightly branch)
- Provides even more details for each processes where possible (e.g., detailed .NET assembly view, service management, looks nicer, etc.)

Other Third-party Tools



Wireshark

- Wireshark is the world's foremost and widely-used network protocol analyzer.
- Provides an overview of the incoming and outgoing packets; useful for traffic analysis.

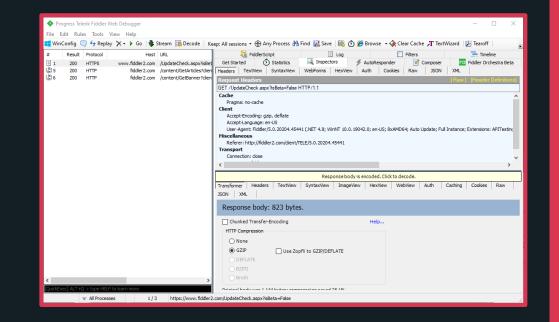
، 🖊	:lump.p	cap																
File	Edit	View	Go	Captu	ire	Analyz	e	Statis	stics	Te	lepho	ony	Wir	eless		Tools	He	elp
		۲	010	🗙 【		۹ 👳	⇒	2	Ŷ	ۍ			Ð,	Q	Q			
Apply a display filter <ctrl-></ctrl->																		
No.		Time		So	urce					Desti	nation	ı				Protoc	ol	Ler
	17	11.7478	389	10	.52	.20.5				211.	233.	.50.	229			TCP		
	18	11.7479	938	21	1.2	33.50.	229)		10.5	52.20	0.5				TCP		
	19	12.5824	158	10	.52	.20.5				10.5	52.1	.1				TCP		
	20	12.5825	598	10	.52	.1.1				10.5	52.20	0.5				TCP		
	21	12.5827	/31	10	.52	.20.5				10.5	52.1	.1				TCP		
	22	12.5838	355	10	.52	.20.5				10.5	52.1	.1				DCERP	PC .	
	23	12.5839	001	10	.52	.1.1				10.5	52.20	0.5				TCP		
	24	12.5840	38	10	.52	.1.1				10.5	52.20	0.5				DCERP	PC .	
	25	12.5841	80	10	.52	.20.5				10.5	52.1	.1				DCERP	PC .	
	26	12.5842	241	10	.52	.1.1				10.5	52.20	0.5				TCP		
	27	12.5843	31	10	.52	.1.1				10.5	52.20	0.5				DCERP	PC .	

Other Third-party Tools



Fiddler

- Proxy debugger for HTTP(s)-based traffic
- Useful for dissecting HTTP(s)-based malware traffic
- AutoResponder
 - Sends forged responses based on the incoming request



Other Third-party Tools



Detect It Easy v3.01				1 <u>00</u> 0	
File name C: \Utils\upx-3.95-win32'	\upx.exe				
File type	Entry point		Base address	Margar Ale	MIME
PE32 -	005f6550	> Disasm	00400000	Memory map	Hash
PE	Export	ort Resources	,NET TLS	Overlay	Strings
Sections	TimeDateStamp	SizeOfImage	Resources		Entropy
0003 >	1970-01-01 01:00:00	001f9000	Manifes	st Version	Hex
Scan	Endianr	ness Mode	Architecture	Туре	
Detect It Easy(DiE)	▼ LE	32	1386	Console	
packer	ι	JPX(3.95)[LZMA,brute]]		
compiler		MinGW(-)[-]			
linker	GNU linker Id (G	NU Binutils)(2.25)[Con	nsole32,console]		
					Options
Signatures			Deep scan		About
	100%		Log 97 msec	Scan	Exit

Detect it Easy

- Swiss-army knife of examining PEs
- Quick overview of any specified file (incl. compiler, packer, linker, etc.) based on community-submitted signatures.
- Examine import tables, exports, hashes of the file, strings, and more!

Other Third-party Tools



NTCore Explorer Suite - CFF Explorer

- Another PE viewer
- Header overview
 - Ability to make quick edits to the header
- Dependency walker
- Imports/exports view
- and more!
- R/W by default; easy ASLR toggle

🛩 CFF Explorer VIII - [HelloWorldCpp.e	exe]					-	\times
File Settings ?							
🔌 😃 🖍	HelloWorldCpp.exe	1					×
	Member	Offset	Size	Value	Meaning		^
File: HelloWorldCpp.exe Dos Header	SizeOfCode	00000124	Dword	00001200			
	SizeOfInitializedData	00000128	Dword	00002600			
File Header Optional Header	SizeOfUninitializedData	0000012C	Dword	0000000			
Data Directories [x]	AddressOf DIICharacteristics			× ⁴	.text		
Bection Headers [x] Directory	BaseOfCoc			00			
🗀 Resource Directory	ImageBase 🗌 Code Integrity			0140000000			
Exception Directory Control Contro Control Control Control Control Control Control Control Control Co	SectionAlic Image is NX con	mpatible ands isolation and (doesn't want it	00			
Debug Directory	FileAlignm Do not bind this			oc			
	MajorOpet Driver uses WDM model						
	MinorOpe						
	Majorimag						
	MinorImag						
	MajorSubs						
	MinorSubs						
	Win32Versi	ок	Cancel	00			
	SizeOfIma			00			
	SizeOfHeaders	0000015C	Dword	00000400			
	CheckSum	00000160	Dword	00000000			
	Subsystem	00000164	Word	0003	Windows Console		
	DIICharacteristics	00000166	Word	8160	Click here		
	SizeOfStackReserve	00000168	Qword	000000000100000			
	SizeOfStackCommit	00000170	Qword	000000000001000			
	SizeOfHeapReserve	00000178	Qword	000000000100000			
	SizeOfHeapCommit	00000180	Qword	000000000001000			
	LoaderFlags	00000188	Dword	00000000			
	NumberOfRvaAndSizes	0000018C	Dword	00000010			~

Other Third-party Tools



🗹 pestudio 9.09 - Malware Initial Assessment - www.winitor.com [c:\programdata\microsoft\windows defender\platform\4.18.2102.3-0				
file settings about				
c:\programdata\microsoft\windows defender\pl	property	value		
virustotal (0/70)	md5	C7C81E2467FA619D5B4A03B0FA51519C		
dos-header (64 bytes)	sha1	39C7EC92E51D72626D4F5D29D80BDF3EB46E8F7C		
dos-stub (176 bytes)	sha256	E3F1ECA768F304512804A54489E8A658E9906E7E09		
	md5-without-overlay	n/a		
	sha1-without-overlay	n/a		
	sha256-without-overlay	n/a		
directories (time-stamp)	first-bytes-hex	4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 B8 0		
···· > sections (wait)	first-bytes-text	M Z @		
libraries (14)	file-size	318432 (bytes)		
	size-without-overlay	n/a		
exports (n/a)	entropy	5.629		
🖉 exceptions (339)	imphash	n/a		
	signature	Microsoft Visual C++ 8.0		
→ ŷ relocations (118)	entry-point	48 83 EC 28 E8 DB 05 00 00 48 83 C4 28 E9 5A FE FF		
	file-version	4.18.2102.3 (WinBuild.160101.0800)		
abc strings (size)	description	Microsoft Malware Protection DLP Command Line		
	file-type	executable		
	cpu	64-bit		
	subsystem	console		
🔄 certificate (time-stamp)	compiler-stamp	0xCFCCDDE6 (Sun Jun 23 03:24:22 2080)		
overlay (n/a)	debugger-stamp	0xCFCCDDE6 (Sun Jun 23 03:24:22 2080)		
	resources-stamp	0x0F00DDE0 (30H 30H 23 03.24.22 2080)		
	exports-stamp	n/a		
	version-stamp	empty		
	certificate-stamp	0x2CA1C000 (Fri Sep 25 03:16:16 2020)		

pestudio

- Yet another PE viewer
- Useful for initial malware assessment
- Provides a quick overview of...
 - File type
 - Target architecture
 - Hashes
 - Compiled date
 - DLL characteristics
 - Strings
 - Imports/exports
 - and more!

Try using these tools while installing your favorite software and see what happens!



Lab #2: Connecting the Dots



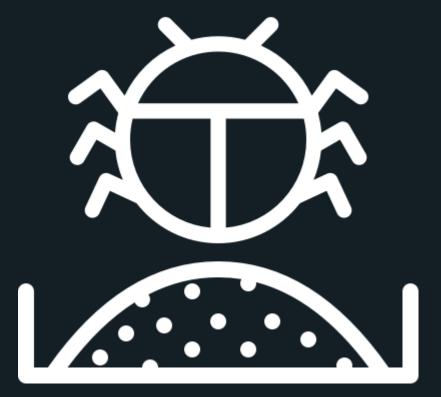
Start by looking up the hash fd866f6e1b997c31bdb6ba24361663e5

Don't skip to the next page until you've found something!



Putting it Together: Example





 You stumbled upon a zipped sample 86950b81df2003d08ae4a7869ecf8 8fe on an online sandbox platform.

What behavior does the sample exhibit? Is there any embedded data?

Tip: Try not to rely on sandbox reports; they can often be misleading or do not provide a bigger picture!

Don't skip to the next page until you've found something!



Putting it Together: Example



- After an extensive research, you've concluded the following characteristics from the sample,
 - Contacts 103.192.226.100
 - Loads AvastAuth.dat and decodes it using XOR key " DFtokTybRE"
 - The decoded file is a PE file that was compiled on 2020-02-15 20:35:46
 - Contains an encoded configuration file using XOR key "123456789"
 - The config has a hardcoded name of "AvastSvcyHA"



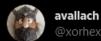
Given the clues thus far, what's the next logical step?

Don't skip to the next page until you've found something!



Putting it Together: Example





#MustangPanda #PlugX

Encrypted Version: virustotal.com/gui/file/6097c...

Decryption Key: 0x55, 0x43, 0x57, 0x46, 0x58, 0x69, 0x79, 0x6e, 0x48, 0x50

103.192.226[.]100:80 103.192.226[.]100:110 103.192.226[.]100:8080 103.192.226[.]100:5938

5:03 AM · Feb 26, 2021 · Twitter Web App



 By looking up the features of the sample, you've discovered that...

- The IP address points to HK.
- The IP address was recently documented on Twitter.
- They referenced a group called MustangPanda and something called PlugX.

(Avallach (@xorhex) / Twitter, 2021)

Who are MustangPanda and what is PlugX?

Don't skip to the next page until you've found something!



Putting it Together: Example

TEAMT5

- By looking up these two mysterious terms, you've discovered...
 - Malpedia is a malware/APT encyclopedia.
 - PlugX is a malware family, specifically, it is used as a RAT backdoor.
 - MustangPanda is a China-based APT group that targets Mongolians.

win.plugx (Back to overview)

aka: Destroy RAT, Kaba, Korplug, Sogu, TIGERPLUG Actor(s): APT 22, APT 26, APT31, APT41, Aurora Panda, Calypso group, DragonOK, Emissary Par Stone Panda, UPS, Violin Panda

RSA describes PlugX as a RAT (Remote Access Trojan) malware family that is around since 2008 remotely execute several kinds of commands on the affected system.

Notable features of this malware family are the ability to execute commands on the affected ma machine information capture the screen send keyboard and mouse events keylogging reboot the system manage processes (create, kill and enumerate) manage services (create, start, stop, etc.); and manage Windows registry entries, open a shell, etc.

Putting it Together: Example



Through nothing but FREE resources, you've learned that...

- There is an APT group called MustangPanda in China.
 - Malpedia
- Mongolians may be a target of interest for China.
 - Malpedia
- PlugX is a RAT, and now you've learned what it may look like internally.
 - Through disassemblers and extensive debugging
- PlugX may disguise itself as an anti-virus component.
 - Twitter

Putting it Together: Next Step? ITEAMT5

Write a YARA rule to threat hunt

- VirusTotal (Paid)
- Hybrid Analysis (Free)
- Abuse.ch MalwareBazaar (Free)
- Publish your finding to help other researchers
 - …and that might help you land a job if you don't have one already.
- Continue digging down the rabbit hole for other findings

Lab #3: Your First YARA Rule (hopefully)





"YARA is a tool aimed at (but not limited to) helping malware researchers to identify and classify malware samples."

- (VirusTotal/Yara, 2012/2021)

```
meta:
    description = "Detect obfuscated .dat file containing shellcode and core
REDLEAVES RAT"
    author = "USG"
    true_positive = "fb0c714cd2ebdcc6f33817abe7813c36" // handkerchief.dat
    reference = "https://www.us-cert.gov/ncas/alerts/TA17-117A"
    strings:
        $RedleavesStringObfu = {73 64 65 5e 60 74 75 74 6c 6f 60 6d 5e 6d 64 60
77 64 72 5e 65 6d 6d 6c 60 68 6f 2f 65 6d 6d} // This is 'red_autumnal_leaves_dl
lmain.dll' XOR'd with 0x01
    condition:
        any of them
```



Install the latest YARA standalone scanner via VirusTotal/Yara

@ GitHub

eleases	Tags		
(Latest release)	YARA v4.0.5 Plusvic released this on Feb 5 · 0 commits to 92077e73786e6ca4f5c915689b2148309c29c787 since this release • BUGFIX: Fix bug in "macho" module introduced in v4.0.4.		
		✓ Assets 4	
			1.35 M
			1.98 M
		Source code (zip)	
		Source code (tar.gz)	



Let's start with the syntax:

- Similar to YAML
- Similar to Python naming conventions snake_case for variables
- Each rule begins with...
 - rule RuleName
- Each rule block requires at least one...
 strings block
 condition block

!		
<pre>rule My_First_Rule</pre>		
1		
strings:		
struigs.		
condition:		
}		



strings block

- Each string is declared with the \$ prefix.
- Case sensitive by default.
- A simple string can be declared using a set of quotation marks.

```
• e.g., $my_variable = "asdf"
```

```
rule My_First_Rule
{
    strings:
        $vegetal = "vegetal"
    condition:
}
```



strings block

- A block of bytes can be declared using a set of braces.
 - e.g., \$dead_beef = {DE AD BE EF}
- Unknown bytes can be replaced with ??.
- A known range of bytes can be replaced with
 [i] or [i-j].

```
rule My_First_Rule
```

strings:

{

```
$vegetal = "vegetal"
$dead_beef = {DE A? ?? EF}
$face_booc = {FA CE B0 0C}
$dead_babe = {DE AD [1-9] BA BE}
condition:
```



strings block

- A string can have additional modifiers:
 - ascii (match ASCII chars; used with wide) fullword
 - wide (UTF-16 chars)
 - xor (search for strings with byte XOR applied) base64
 - base64wide
 - private (never match)
 - nocase (case insensitive)

rule My_First_Rule

strings:

{

}

```
$vegetal = "vegetal"
$utf16_beef = "beef" wide
$cheese = "cheesecake" xor(0x01-0x05)
$dead_beef = {DE A? ?? EF}
$face_booc = {FA CE B0 0C}
$dead_babe = {DE AD [1-9] BA BE}
condition:
```



strings block

- Regex can also be used.
 - Perl-like syntax
- e.g., /hello{1,3}world/ matches
 "helloworld", "hellooworld", "helloooworld".

```
rule My_First_Rule
```

```
strings:
```

{

}

```
$vegetal = "vegetal"
$utf16_beef = "beef" wide
$cheese = "cheesecake" xor(0x01-0x05)
$dead_beef = {DE A? ?? EF}
$face_booc = {FA CE B0 0C}
$dead_babe = {DE AD [1-9] BA BE}
$pastry = /slice of (cake|pie|bread)/ nocase
condition:
```



condition block

- Defines when the scanner should mark the target file as positive.
- All defined strings MUST be referenced in this block.
- Loosest condition is any of them, which returns true on any string match.

```
rule My_First_Rule
```

strings:

{

\$vegetal = "vegetal"

\$utf16_beef = "beef" wide

\$cheese = "cheesecake" xor(0x01-0x05)

\$dead_beef = {DE A? ?? EF}

\$face_booc = {FA CE B0 0C}

\$dead_babe = {DE AD [1-9] BA BE}

\$pastry = /slice of (cake|pie|bread)/ nocase
condition:

any of them



rule My_First_Rule

{

condition block

- Conditions can be chained using or.
- Conditions can be limited using and.

```
strings:
    $vegetal = "vegetal"
    $utf16_beef = "beef" wide
    $cheese = "cheesecake" xor(0x01-0x05)
    $dead_beef = {DE A? ?? EF}
    $face_booc = {FA CE B0 0C}
    $dead_babe = {DE AD [1-9] BA BE}
    $pastry = /slice of (cake|pie|bread)/ nocase
condition:
    ($dead_beef and $face_booc) or
        any of them
```



rule My_First_Rule

strings:

{

condition block

- any can be substituted with any number of integer.
- A set of strings with a common variable name can be referenced using wildcard with parentheses around the variable.
 - e.g., 3 of (\$bad_*)

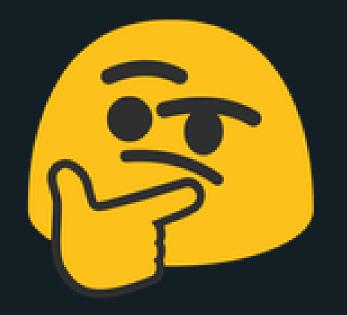
```
$evil_vegetal = "vegetal"
$evil_pastry = /slice of (cake|pie|bread)/ nocase
$bad_utf16_beef = "beef" wide
$bad_cheese = "cheesecake" xor(0x01-0x05)
$bad_dead_beef = {DE A? ?? EF}
$face_booc = {FA CE B0 0C}
$dead_babe = {DE AD [1-9] BA BE}
condition:
2 of ($bad_*) or
any of ($evil_*) or
($face_booc and $dead_babe)
```



condition block

- Many more conditions can be defined.
- See YARA docs for a list of valid syntaxes.

```
rule Complex_Yara
{
    strings:
        $a = "Aaa"
        b = BbBb''
        $c = "ccc"
    condition:
        for any of ($a,$b,$c) : ( $ at pe.entry_point ) or
        for any section in pe.sections : ( section.name == ".text" )
}
```



Isn't this just the strings command with extra steps?



Yesn't



strings block

- A block of bytes can be declared using a set of braces.
 - e.g., \$dead_beef = {DE AD BE EF}
- Unknown bytes can be replaced with ??.
- A known range of bytes can be replaced with
 [i] or [i-j].

```
rule My_First_Rule
```

```
strings:
```

{

```
$vegetal = "vegetal"
$dead_beef = {DE A? ?? EF}
$face_booc = {FA CE B0 0C}
$dead_babe = {DE AD [1-9] BA BE}
condition:
```

strings: \$vegetal = "vegetal" $dead beef = \{DE A? ?? EF\}$ $face_booc = {FA CE B0 0C}$ $dead_babe = \{DE AD [1-9] BA BE\}$



000093d0	55	push ebp
000093d1	8bec	mov ebp, esp
000093d3	6aff	push -1
000093d5	alf8e50210	mov eax, dword ptr [0x1002e5f8]
000093da	50	push eax
000093db	e8d0c4ffff	call 0x58b0
000093e0	85c0	test eax, eax
000093e2	7407	je 0x93eb
000093e4	b84f050000	mov eax, 0x54f
000093e9	eb21	jmp 0x940c
000093eb	833df8e5021000	cmp dword ptr [0x1002e5f8], 0
000093f2	7416	je 0x940a
000093£4	8b0df8e50210	mov ecx, dword ptr [0x1002e5f8]
000093fa	51	push ecx
000093fb	e80070ffff	call 0x400
00009400	c705f8e5021000000000	mov dword ptr [0x1002e5f8], 0
0000940a	33c0	xor eax, eax
0000940c	5d	pop ebp
0000940d	_c3	ret

ule My_First_Rule
strings:
<pre>\$vegetal = "vegetal"</pre>
<pre>\$dead_beef = {DE A? ?? EF}</pre>
<pre>\$face_booc = {FA CE B0 0C}</pre>
<pre>\$dead_babe = {DE AD [1-9] BA BE}</pre>
condition:



- Of course, you can write YARA rule to match for specific sets of instructions!
 - Instructions in a binary are just a series of bytes.
- IDA Pro plugins for writing YARA rules
 - hyuunnn/Hyara @ GitHub
 - fox-it/mkYara @ GitHub

ি mkYARA :: Generated Yara Rule		 ?	×
Generated Yara rule from 0x140008634 to 0x14000866d			
Generated Yararule from 0x140008644 fo 0x140008649 0x140008642 48FF15AFAB0000 0x140008659 0F1F440000 0x140008659 4CBD45D0 0x140008655 8D5328 0x140008656 8D5328 0x140008666 750C 0x140008666 750C 0x140008666 750C 0x140008666 750C 0x140008668 E8831C0000 */ strings: \$chunk_1 = { 80 48 ?? F3 AA 48 8D 7D ?? 80 48 ?? F3 AA 48 FF 15 ?? ?? ?? ?? 0F IF 44 00 ?? 80 55 ?? 80 55 ?? 80 55 ?? 85 C0	<pre>call qword ptr [rip + 0xabaf] nop dword ptr [rax + rax] mov rcx, rax lea r8, [rbp - 0x30] lea edx, [rbx + 0x28] call qword ptr [rip + 0xab11] nop dword ptr [rax + rax] test eax, eax jn 00x140008674 call 0x14000a2f0</pre>		
75 ?? E8 ?? ?? ?? ?? }			
condition:			
any of them			
}			

DO

- Target unique characteristics common in the same malware family
 - e.g., certain PDB paths or project folder names
- Compare code from the same malware family

DO NOT

- × Rely on imports as an indicator
- × Match for common strings
- Match for instructions that may be part of a library
 - e.g., OpenSSL, json-parser, etc.
- × Write YARA rules for .NET modules without using the .NET YARA module
 - Difficult & high false positive

Now give it a try!

Try writing a YARA rule for FD866F6E1B997C31BDB6BA24361663E5.



THANK YOU!



@StillAzureH



still@teamt5.org



Persistent Cyber Threat Hunters

References

- Avallach (@xorhex) / Twitter. (2021, February 26). Twitter. https://twitter.com/xorhex
- Jelen, S. (2019, July 16). SecurityTrails | Cyber Threat Intelligence. https://securitytrails.com/blog/cyber-threat-intelligence
- Microsoft Corp. (2021, March 23). Windows Sysinternals—Windows Sysinternals. https://docs.microsoft.com/en-us/sysinternals/
- VirusTotal. (2021). VirusTotal/yara [C]. VirusTotal. <u>https://github.com/VirusTotal/yara</u> (Original work published 2012)

