初探威脅情資

Threat Intelligence 101: What, Why, How

Still Hsu

TEAMT5

Persistent Cyber Threat Hunters

AGENDA

TEAMT5

01 走入 CTI 的第一步

- CTI 到底是啥？
- CTI 產品生命週期

02 我的第一篇情資報告

- 報告 SOP
- 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

03 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。

05 Lab #3: 我的第一個 YARA 規則

- 為了未來而準備，開始寫你的第一支 YARA 規則來進行威脅狩獵吧！

whoami

TEAMT5

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

TEAMT5

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

What is CTI, anyways?

What is CTI, anyways?

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

- (CrowdStrike, 2021)

What is CTI, anyways?

- Past
- Present
- Future

Records and collects indicators of cyberthreats

TEAMT5

What is CTI, anyways?

- Past
- Present
- Future

Records and collects indicators of cyberthreats

Provides additional insights into these threats

- Tactics
- Operations
- Strategies

TEAMT5

What is CTI, anyways?

- 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

TEAMT5

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

TEAMT5

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

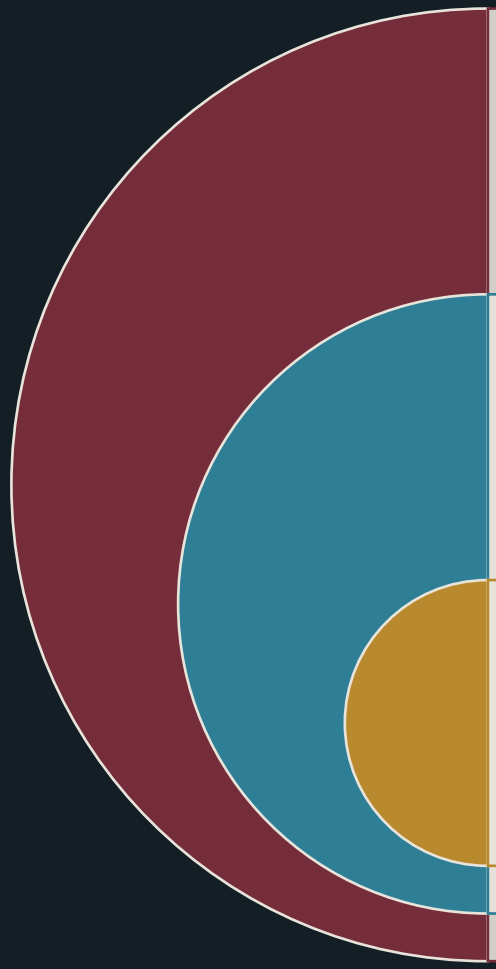
(Jelen, 2019)

TEAMT5

Who uses CTI?

TEAMT5

Who uses CTI?



Tactics	• Valid malware and traffic signatures • Critical vulnerabilities
Operations	• Detailed context around alerts and events
Strategies	• Better understanding of trends • Better decision-making regarding security budgets, improvements, technological requirements

TEAMT5

Sounds scary…

I don’t know anything about threat research.

Neither did I!

Lifecycle


Planning &
Direction
Collection
Dissemination
& Integration
Analysis &
Production

Ask Yourself These…

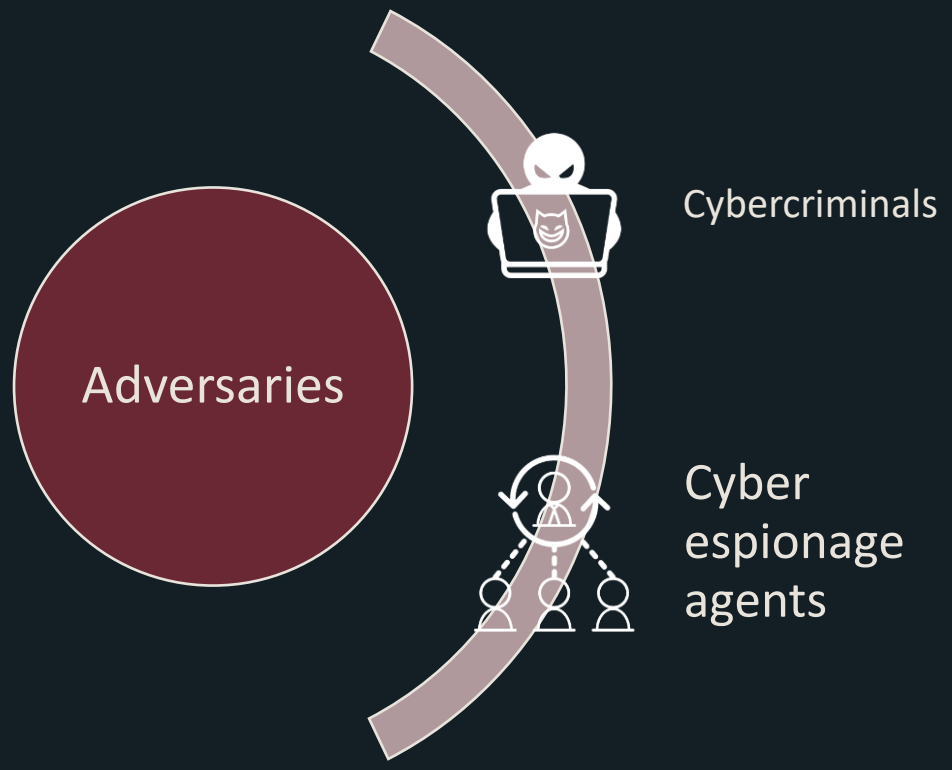
What is the most significant threat?

How to prioritize the threats?

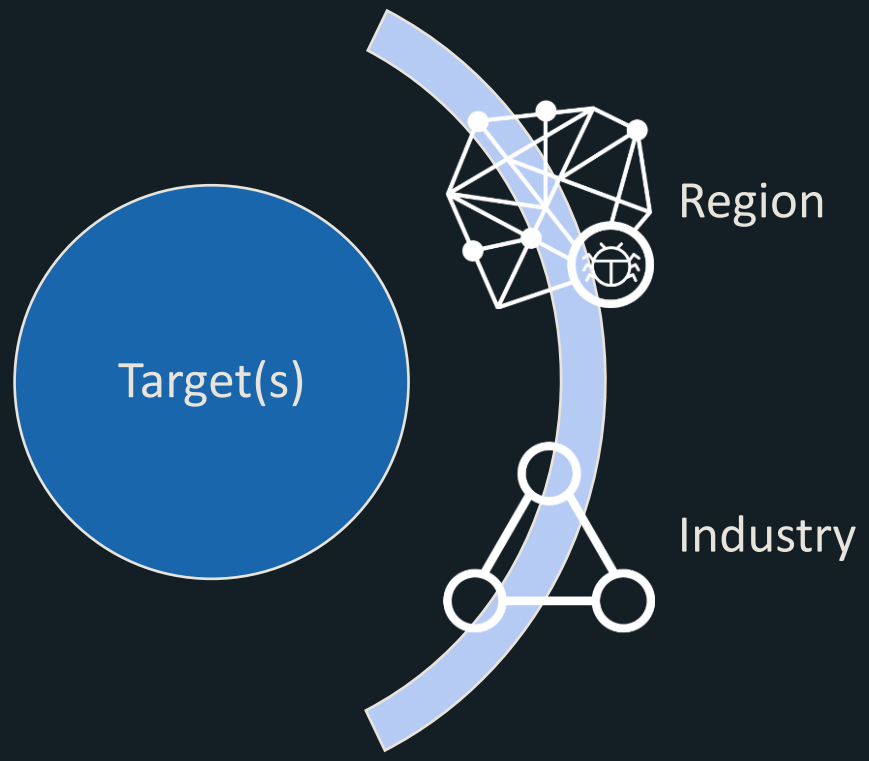
Who will consume and benefit from the finished product?

Cyber Attacks

Type of adversaries



Information about the target(s)



Collection

External Source

Community

Social Media

Threat Data Feed

Open-source Intelligence

Deep Web

Dark Web

Internal Source

SIEM / Sensors

Incident Response

Network Visibility

Endpoint Visibility

Malware Analysis

Research Lab

Diamond Model

ADVERSARY

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

CAPABILITY

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

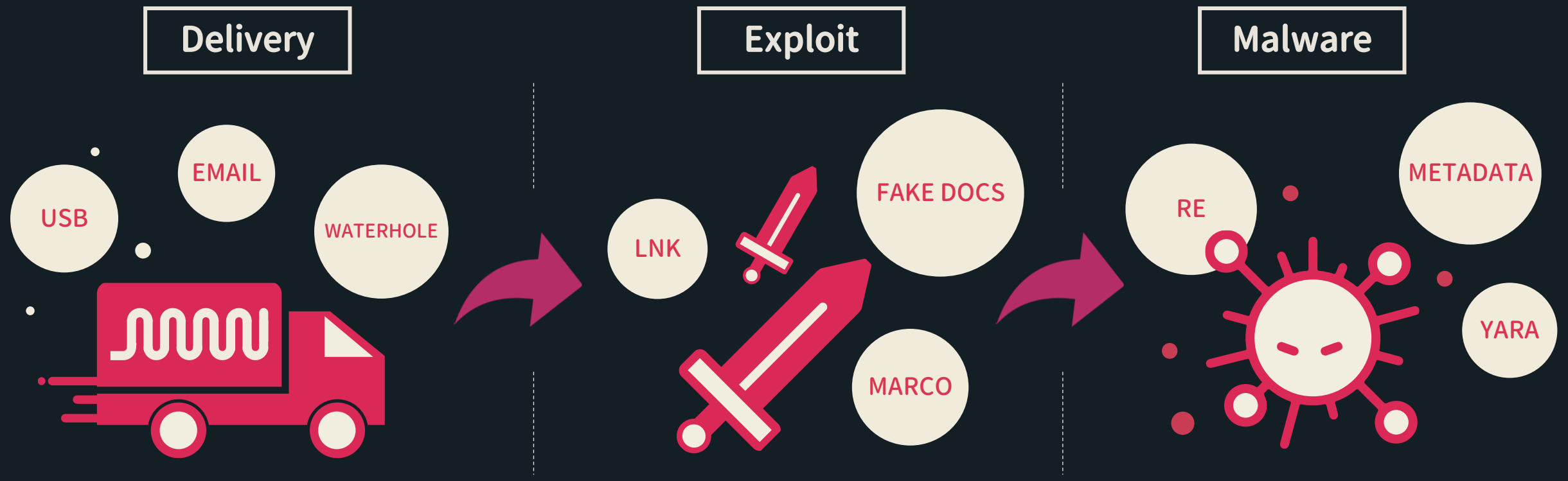
INFRASTRUCTURE

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

TARGET

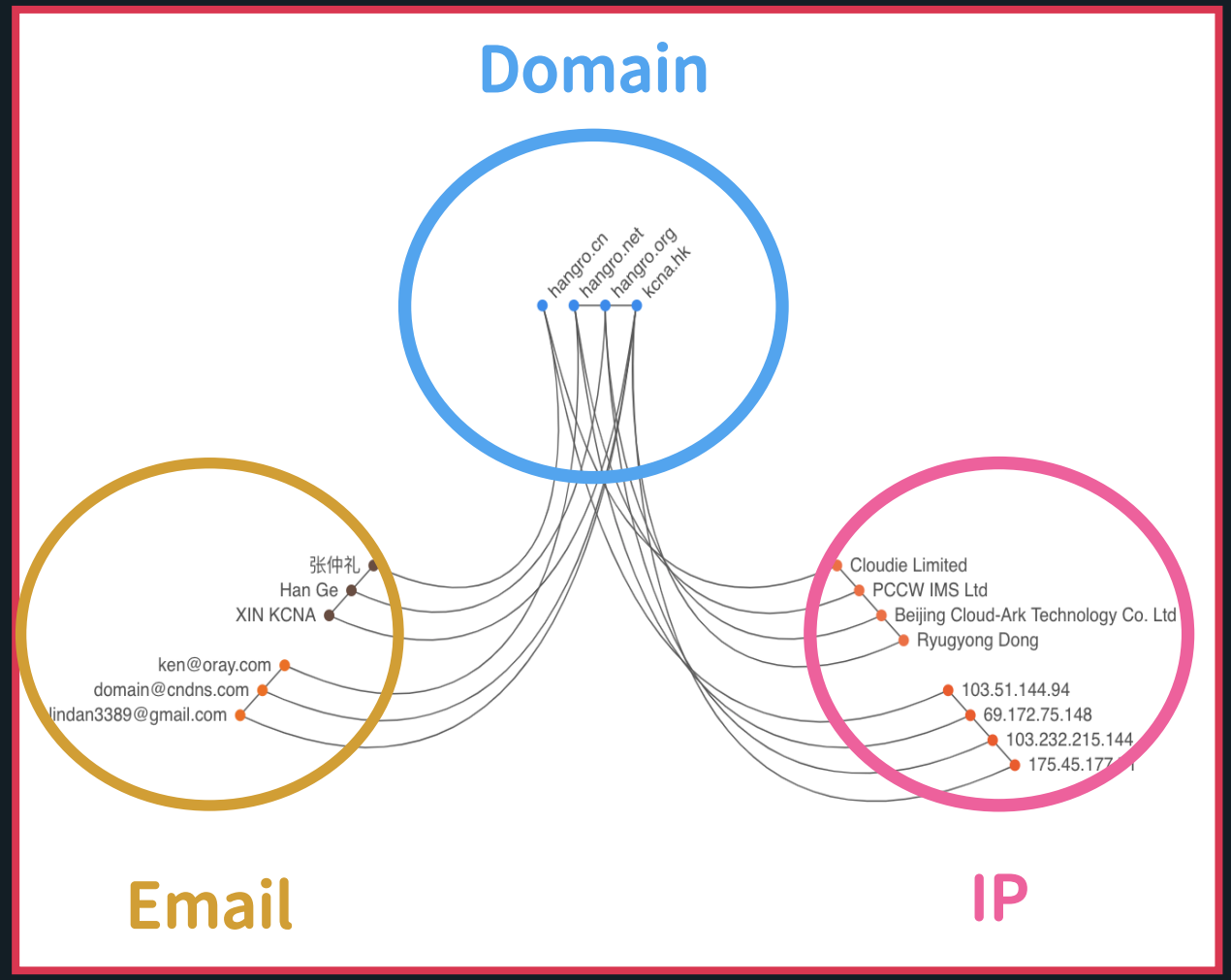
- Purpose
- Target countries / regions
- Target sectors
- Target individuals
- Target data

Capability Analysis



Infrastructure Analysis

- Domain
  - WHOIS -> Email
  - Passive DNS -> IP
- IP
  - Passive DNS -> Domain
- Email
  - Reverse WHOIS - > Domain



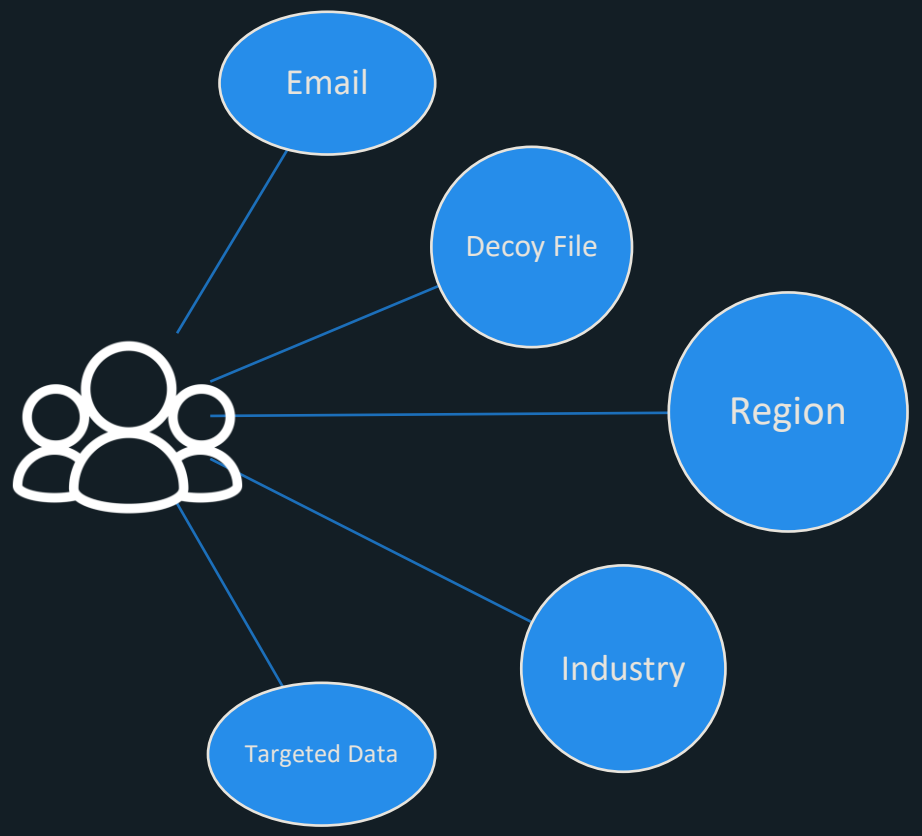
Adversary Analysis

TEAMT5

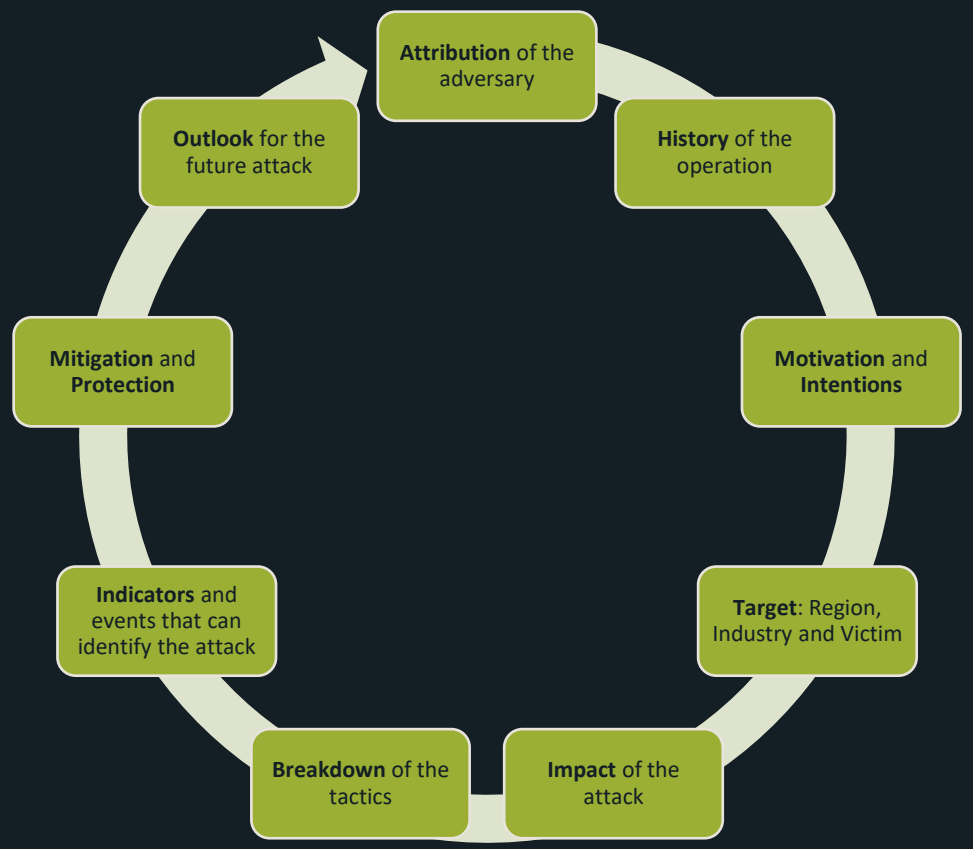
- Actors
  - Language
  - Tools
  - Infrastructure
  - Time zone
- Motivations, intentions
- Cooperation relationship between different groups
  - Sharing Tool
  - Sharing C2

Target Analysis

Victim Analysis



Threat Analysis Report



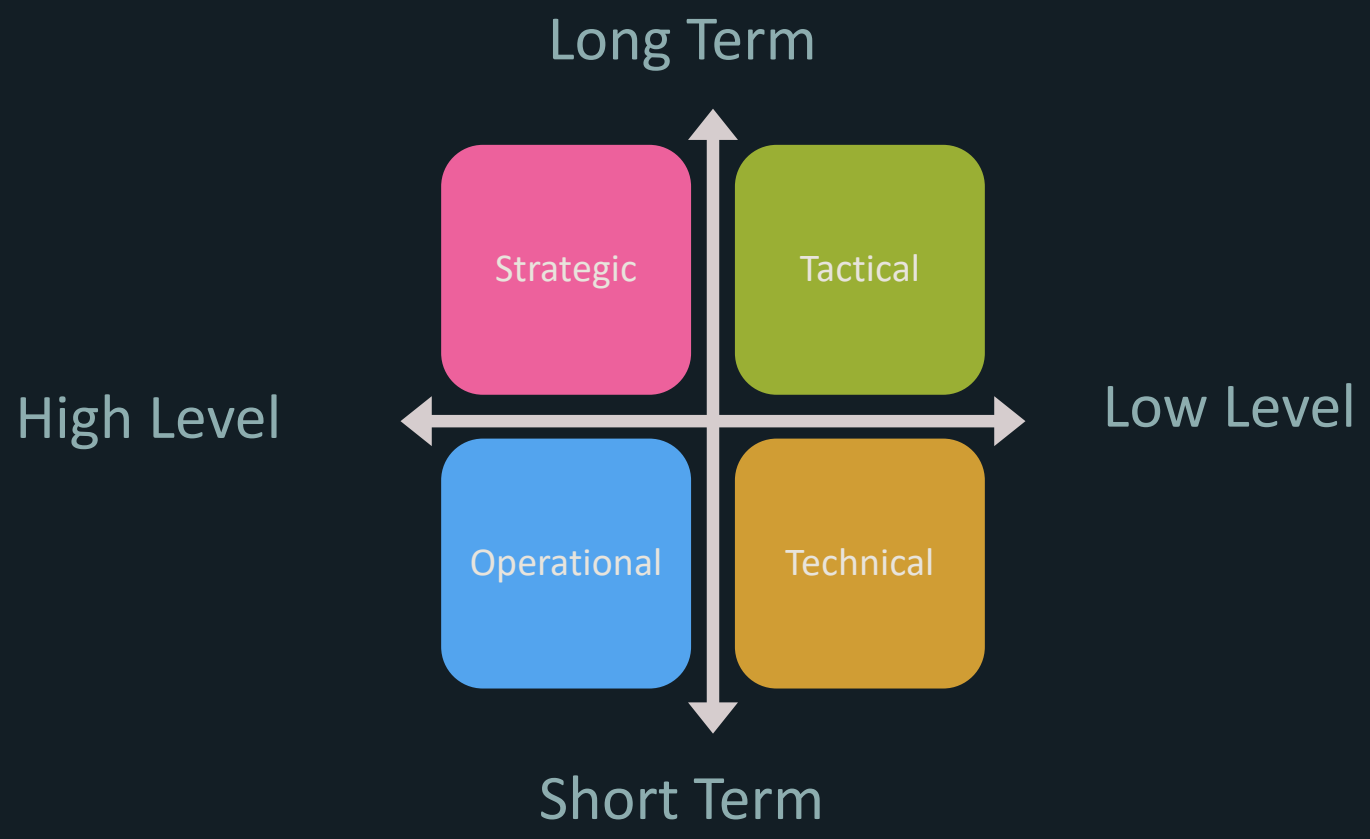
Dissemination & Integration

Strategic Planning

ISCT / CERT
Community

IT Staff
CSIRT Team

Firewall
SIEM Triage



Writing a CTI Report

Standard Operating Procedures

TEAMT5

Intel hunting

- Stay ahead of cyberthreat intel

Threat hunting

- Collect valuable samples
- e.g., unseen C2 stations, zero-day, new backdoors

Sample analysis

- Analyze behavior and if signatures exist
- e.g., yara rules, CAPA

Identifying relations

- Compare with existing or known reports and identify whether a connection exists

Report

- For future comparisons

Adversary & Intel Research

TEAMT5

Sanyo

- 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 self-protection techniques.
  - The Sanyo actors have so far shown their capability by successfully breaching several defense contractors and heavy industries in the world.

GuDiao

TEAMT5

- 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

TEAMT5

- 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

TEAMT5

- Twitter
  - #APT
  - @cyberwar_15
  - @Timele9527
  - @blackorbird
  - @Rmy_Reserve
  - @_re_fox
- Curated Resources
  - https://start.me/p/rxRbpo/ti

Threat/Intel Hunting Resources

TEAMT5

- Yara rules
  - Yara-Rules/rules @ GitHub
  - InQuest/awesome-yara @ GitHub
  - Neo23x0/signature-base @ GitHub
- CAPA
  - FireEye/CAPA @ 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

TEAMT5

- 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

TEAMT5

How did the incident occur?

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

Content of a Report

TEAMT5

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

TEAMT5

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)

Who did it?

- Source infrastructure analysis
- Piece everything together with existing reports

How did the incident occur?

Delivery Methods

TEAMT5

- Spear-phishing email
- Watering hole attack
- Supply chain attack

Delivery Methods

TEAMT5

Spear-phishing email

ing hole attack

- 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

Delivery Methods

TEAMT5

phishing email

Watering hole attack

chain attack

- Compromise sites that victim frequents
- Drive-by via malvertisements or domain redirection
- Example
  - Holy Water campaign in 2020
    - Targeted religious and charity websites

Delivery Methods

TEAMT5

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

TEAMT5

Fake documents

- Executables or shortcuts (LNK) with document icons

Exploit Methods

TEAMT5

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

TEAMT5

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

Containerized environment

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

Malware Analysis

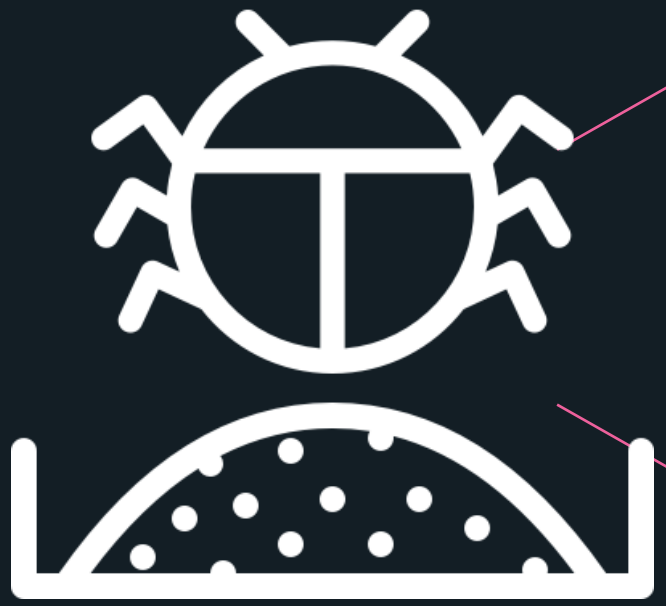
TEAMT5

Static analysis

- Disassembler
- Digital signature
  - Stolen cert
- PE header

Malware Analysis

TEAMT5



Static analysis

- Disassembler
- Digital signature
  - Stolen cert
- PE header

Behavioral analysis

- Artifacts
- Persistence
- Exfiltrated data
- Network traffic
- Encryption algorithm
- Backdoor

Who did it?

Infrastructure Analysis

TEAMT5

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

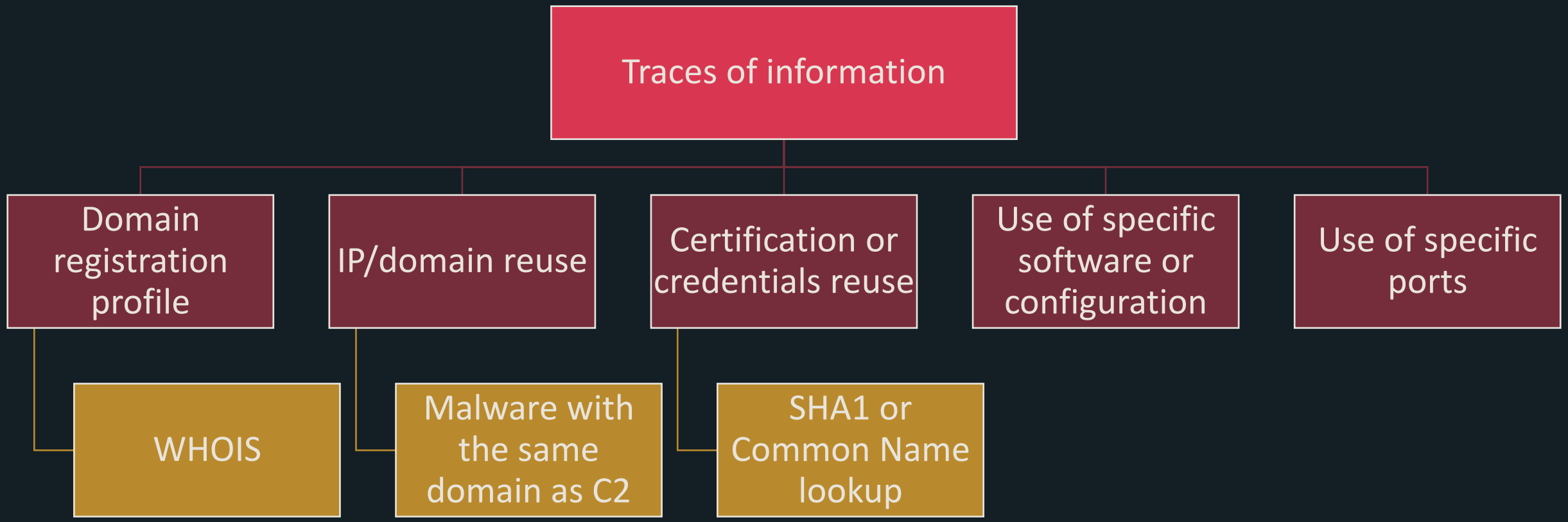
- Free/paid
- Two or more users may share the same machine
  - More than one domain may resolve to the same IP address or set of addresses
  - Threat actors could only access the frontend
  - Implemented alongside simple backdoors or only used to serve malicious files

Compromised Server

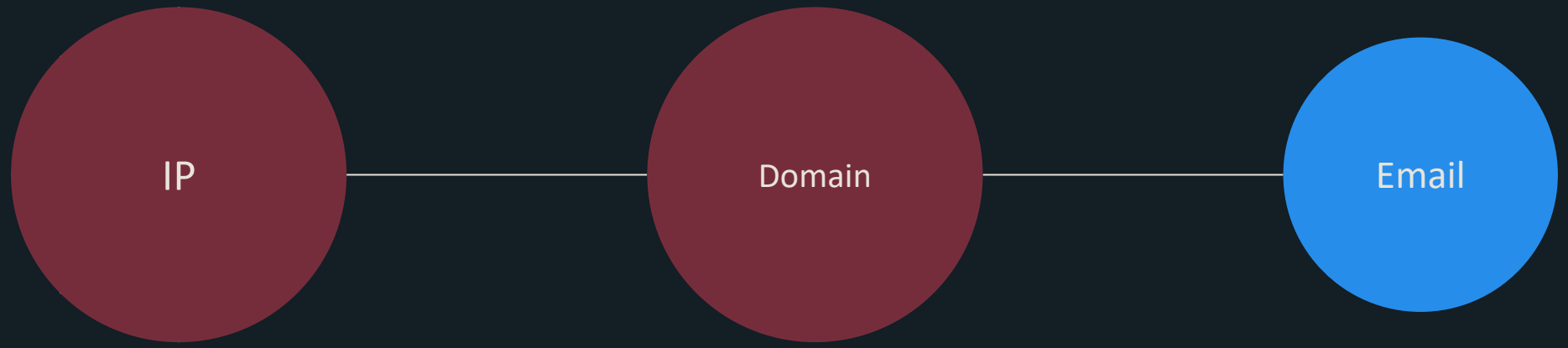
- Unauthorized access via…
  - Web application vulnerabilities
  - Software vulnerabilities
  - Compromised credentials
- Access level highly depends on the method of intrusion
- Backdoors are generally well-hidden to avoid raising suspicion

C2 Relation

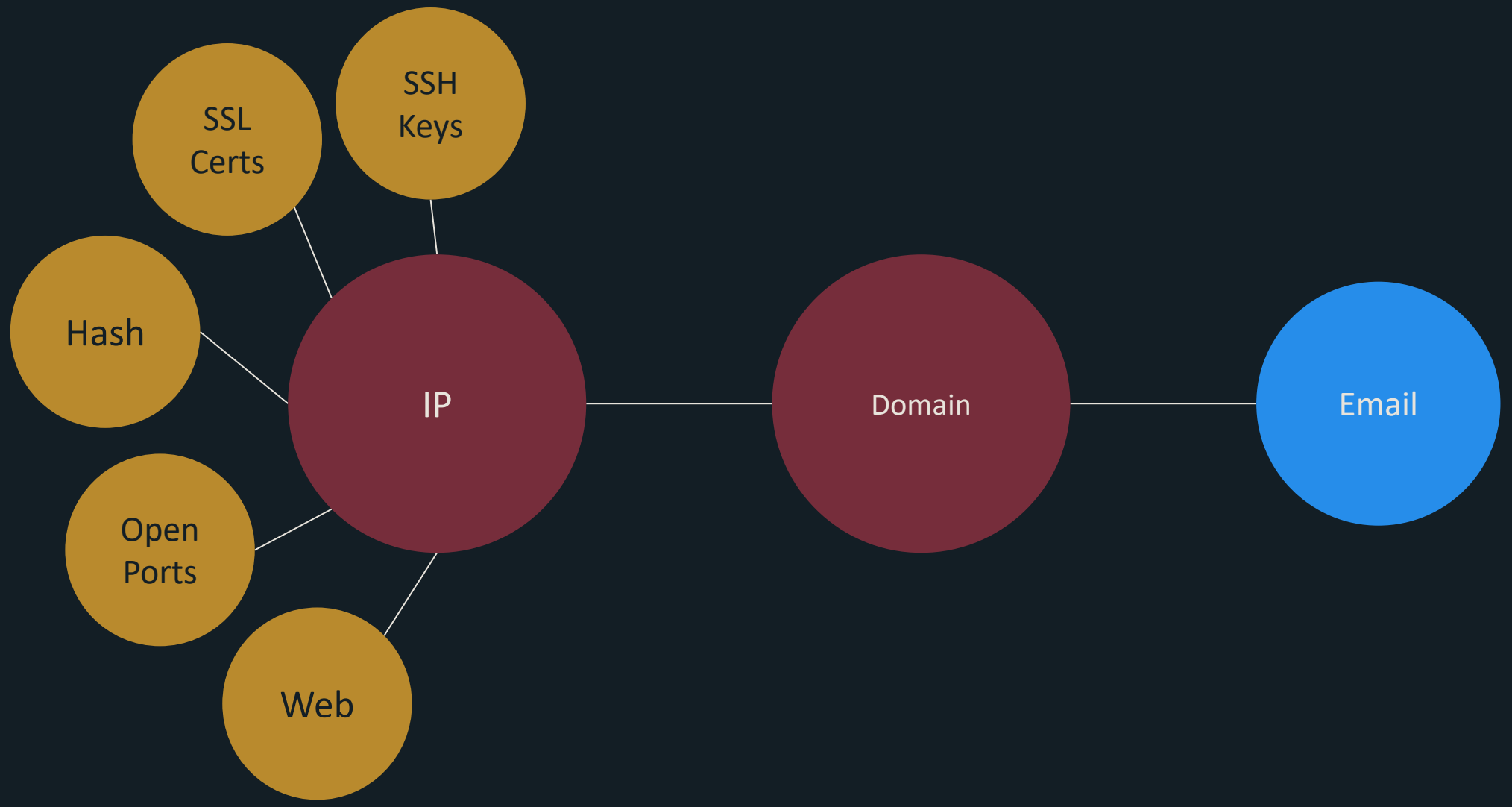
TEAMT5



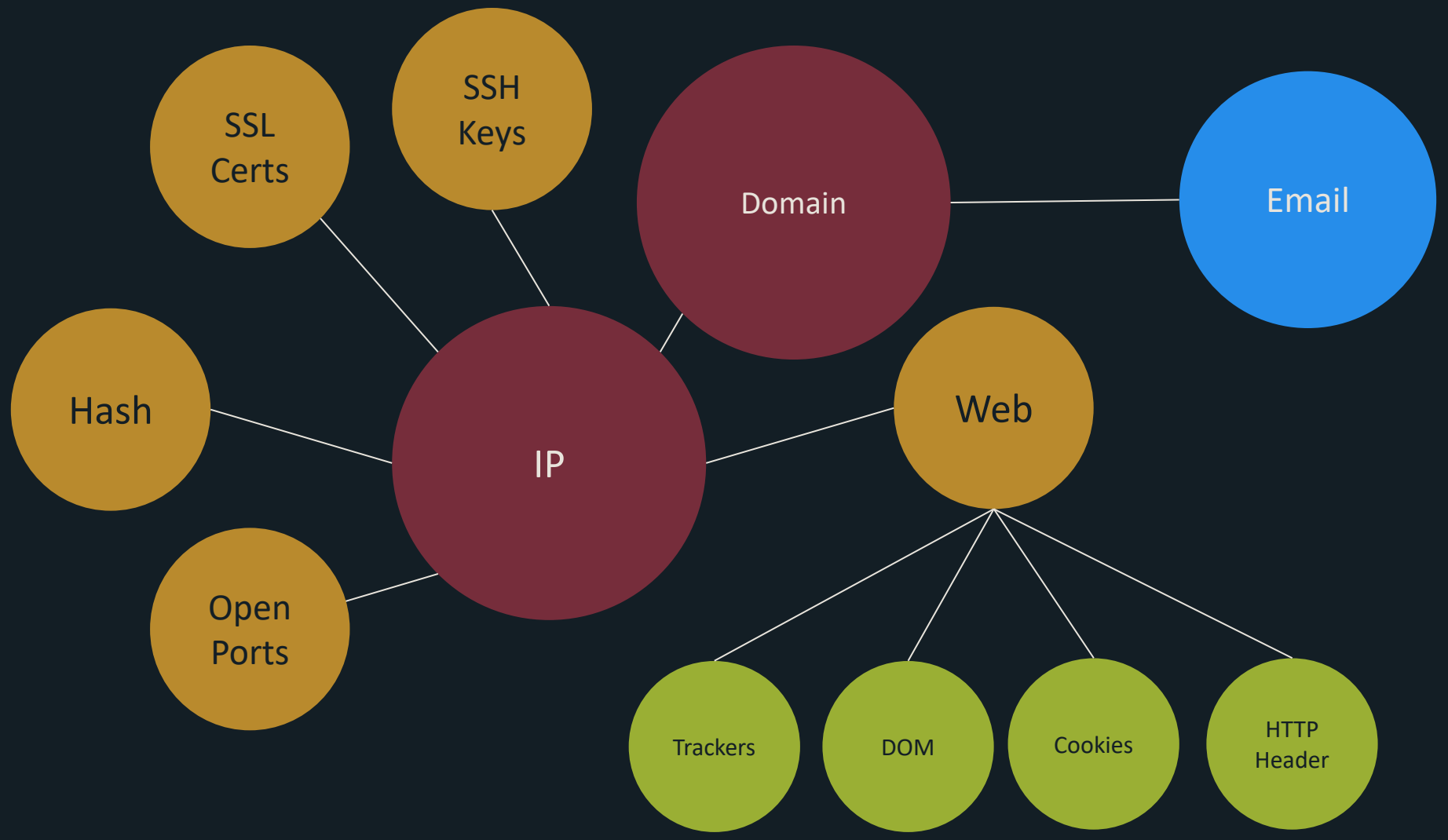
C2 Relation



C2 Relation



C2 Relation


SSL Certs
SSH Keys
Domain
Email
Hash
IP
Web
Open Ports
Trackers
DOM
Cookies
HTTP Header

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.

Lab #1: Getting the Hang of Tools

Sysinternals Suite

TEAMT5

Windows Sysinternals

03/23/2021 • 4 minutes to read • +3

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.

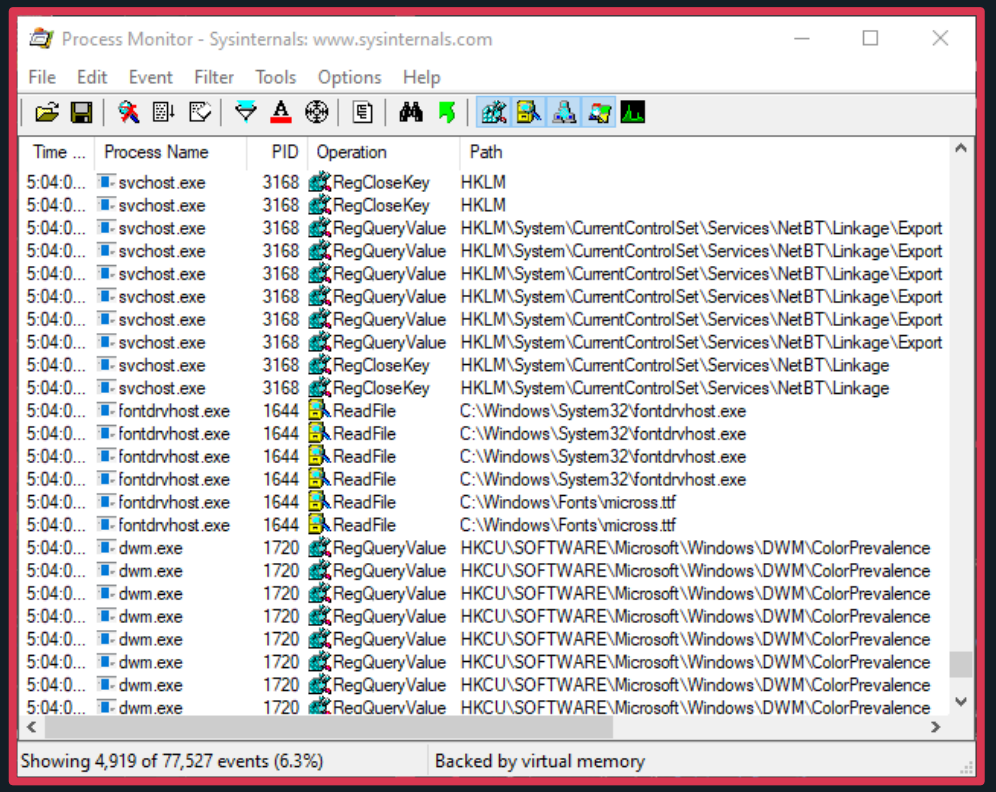
- Read the official guide to the Sysinternals tools, Troubleshooting with the Windows Sysinternals Tools
- Read the Sysinternals Blog for a detailed change feed of tool updates
- Watch Mark's Sysinternals Update videos on YouTube
- Watch Mark's top-rated Case-of-the-Unexplained troubleshooting presentations and other webcasts
- Read Mark's Blog which highlight use of the tools to solve real problems
- Check out the Sysinternals Learning Resources page
- Post your questions in the Sysinternals Forum

(Microsoft Corp., 2021)

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

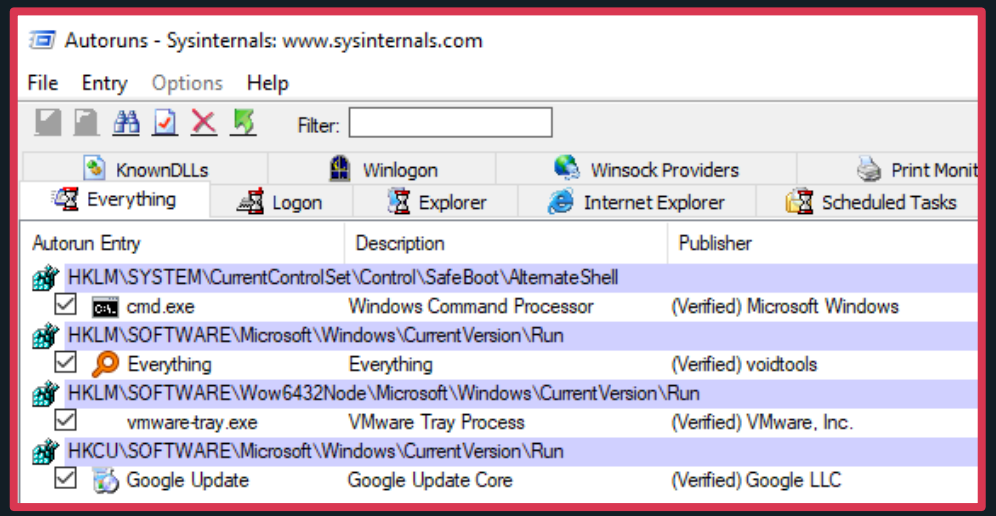
Sysinternals Suite

- 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.



```
Time ...  Process Name     PID   Operation      Path
5:04:0... svchost.exe      3168  RegCloseKey    HKLM
5:04:0... svchost.exe      3168  RegCloseKey    HKLM
5:04:0... svchost.exe      3168  RegQueryValue  HKLM\System\CurrentControlSet\Services\NetBT\Linkage\Export
5:04:0... svchost.exe      3168  RegQueryValue  HKLM\System\CurrentControlSet\Services\NetBT\Linkage\Export
5:04:0... svchost.exe      3168  RegQueryValue  HKLM\System\CurrentControlSet\Services\NetBT\Linkage\Export
5:04:0... svchost.exe      3168  RegQueryValue  HKLM\System\CurrentControlSet\Services\NetBT\Linkage\Export
5:04:0... svchost.exe      3168  RegQueryValue  HKLM\System\CurrentControlSet\Services\NetBT\Linkage\Export
5:04:0... svchost.exe      3168  RegQueryValue  HKLM\System\CurrentControlSet\Services\NetBT\Linkage\Export
5:04:0... svchost.exe      3168  RegCloseKey    HKLM\System\CurrentControlSet\Services\NetBT\Linkage
5:04:0... svchost.exe      3168  RegCloseKey    HKLM\System\CurrentControlSet\Services\NetBT\Linkage
5:04:0... fontdrvhost.exe  1644  ReadFile       C:\Windows\System32\fontdrvhost.exe
5:04:0... fontdrvhost.exe  1644  ReadFile       C:\Windows\System32\fontdrvhost.exe
5:04:0... fontdrvhost.exe  1644  ReadFile       C:\Windows\System32\fontdrvhost.exe
5:04:0... fontdrvhost.exe  1644  ReadFile       C:\Windows\System32\fontdrvhost.exe
5:04:0... fontdrvhost.exe  1644  ReadFile       C:\Windows\Fonts\micross.ttf
5:04:0... fontdrvhost.exe  1644  ReadFile       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  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  RegQueryValue  HKCU\SOFTWARE\Microsoft\Windows\DWM\ColorPrevalence
5:04:0... dwm.exe          1720  RegQueryValue  HKCU\SOFTWARE\Microsoft\Windows\DWM\ColorPrevalence
Showing 4,919 of 77,527 events (6.3%)    Backed by virtual memory
```

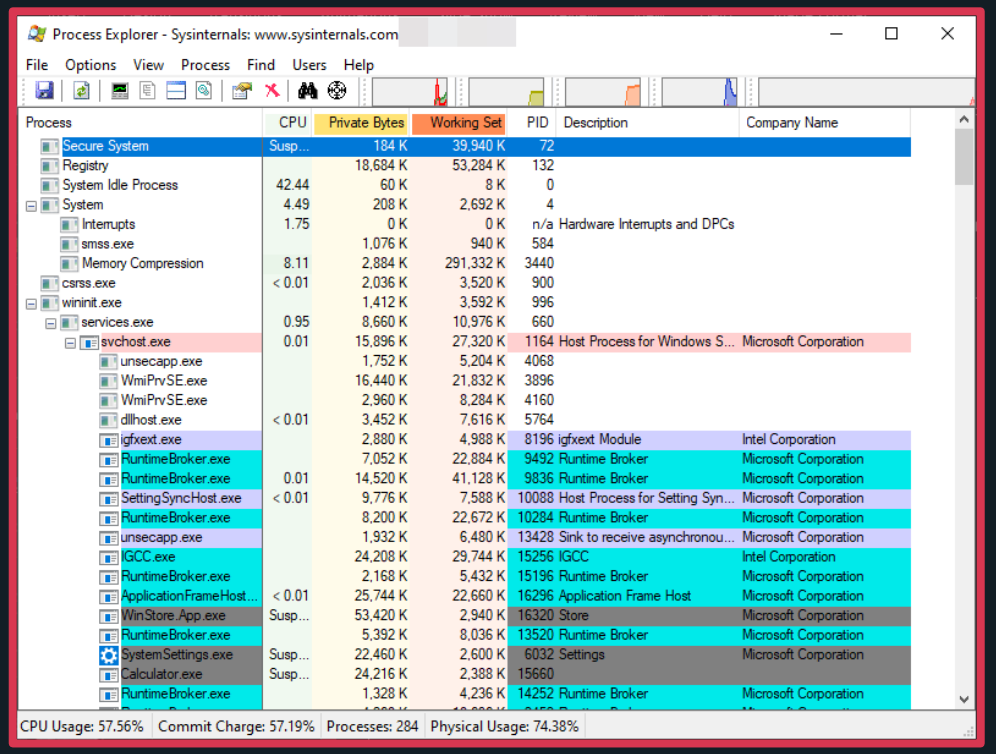
Sysinternals Suite



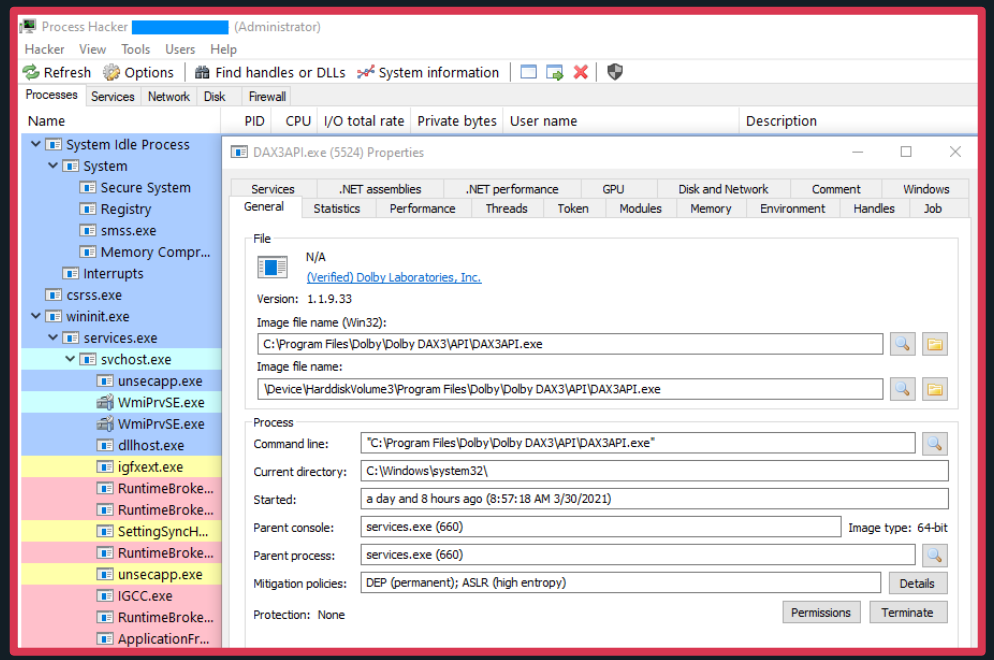
- AutoRuns
  - *AutoRuns* shows you what programs are configured to run during system bootup or login, and when you start various built-in Windows applications like Internet Explorer, Explorer and media players.
  - Quick overview of the existing persistence entries on the machine.

Sysinternals Suite

- 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.)



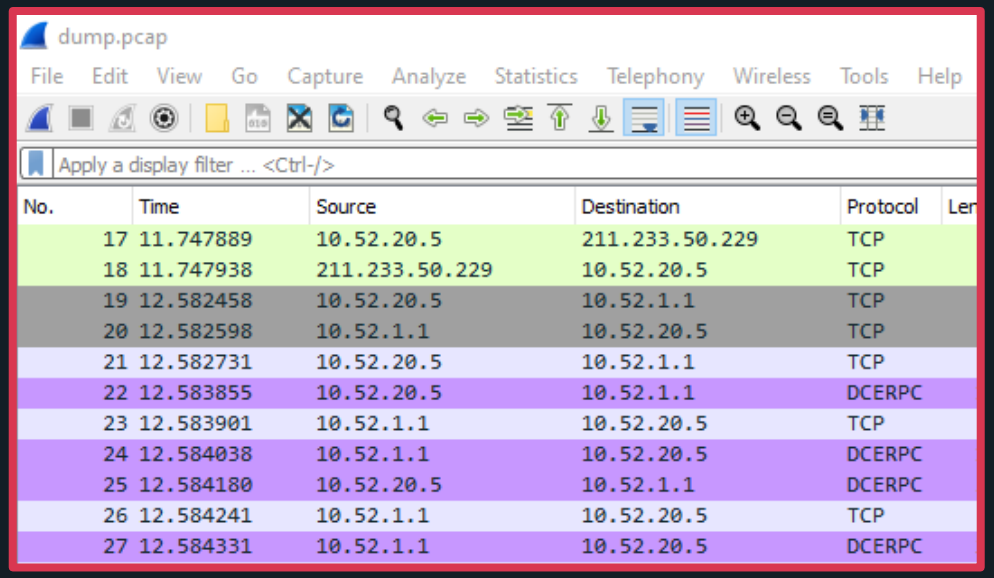
Other Third-party Tools



- 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

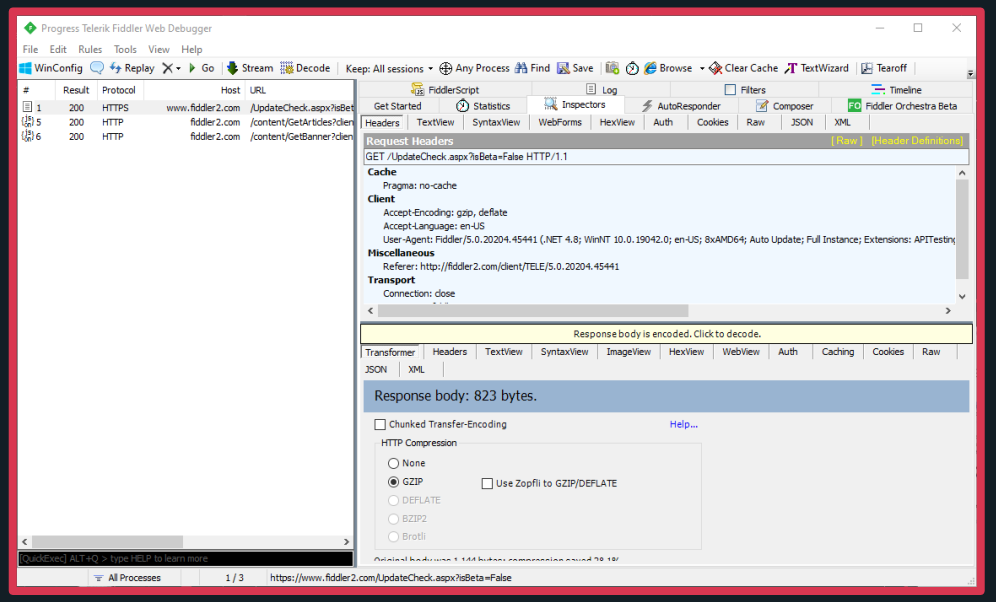
- Wireshark
  - **Wireshark** is the world's foremost and widely-used network protocol analyzer.
  - Provides an overview of the incoming and outgoing packets; useful for traffic analysis.



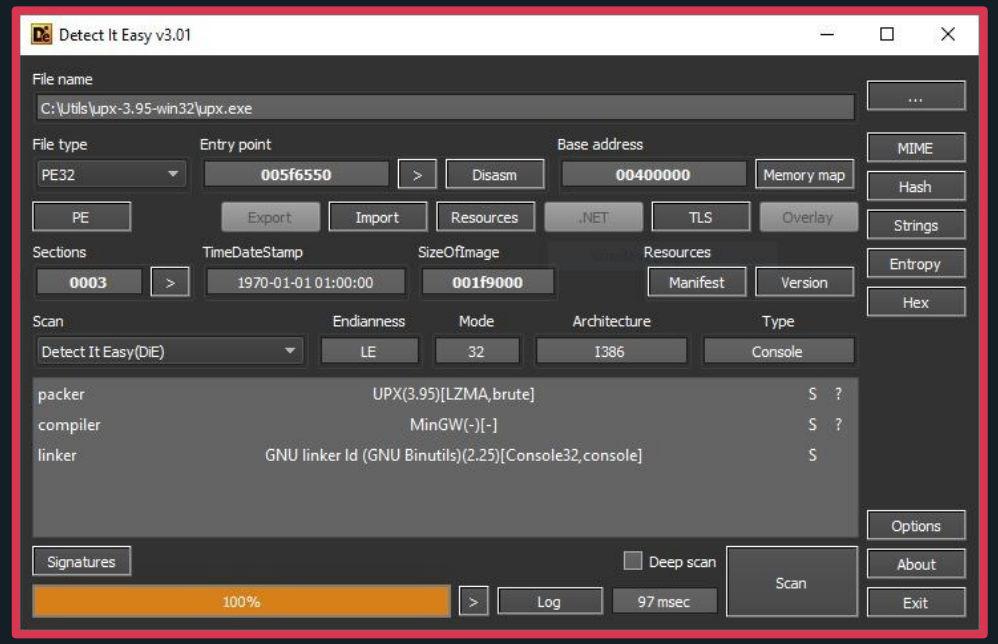
No.	Time	Source	Destination	Protocol	Len
17	11.747889	10.52.20.5	211.233.50.229	TCP	
18	11.747938	211.233.50.229	10.52.20.5	TCP	
19	12.582458	10.52.20.5	10.52.1.1	TCP	
20	12.582598	10.52.1.1	10.52.20.5	TCP	
21	12.582731	10.52.20.5	10.52.1.1	TCP	
22	12.583855	10.52.20.5	10.52.1.1	DCERPC	
23	12.583901	10.52.1.1	10.52.20.5	TCP	
24	12.584038	10.52.1.1	10.52.20.5	DCERPC	
25	12.584180	10.52.20.5	10.52.1.1	DCERPC	
26	12.584241	10.52.1.1	10.52.20.5	TCP	
27	12.584331	10.52.1.1	10.52.20.5	DCERPC	

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



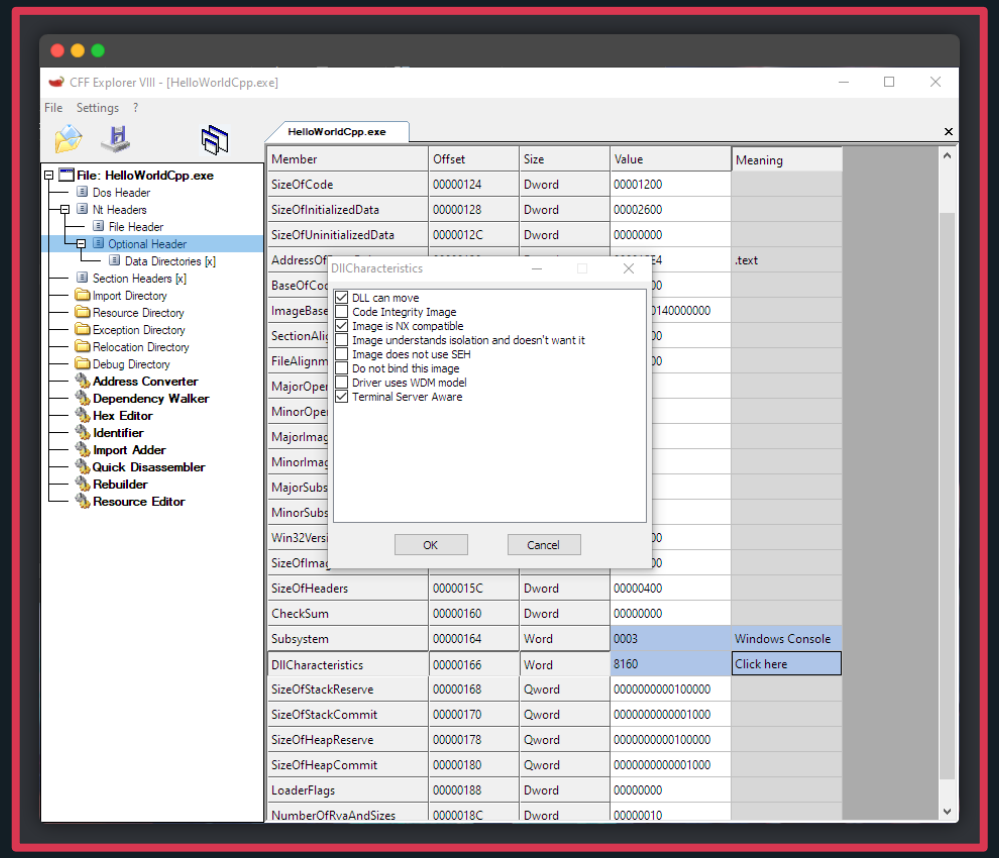
Other Third-party Tools



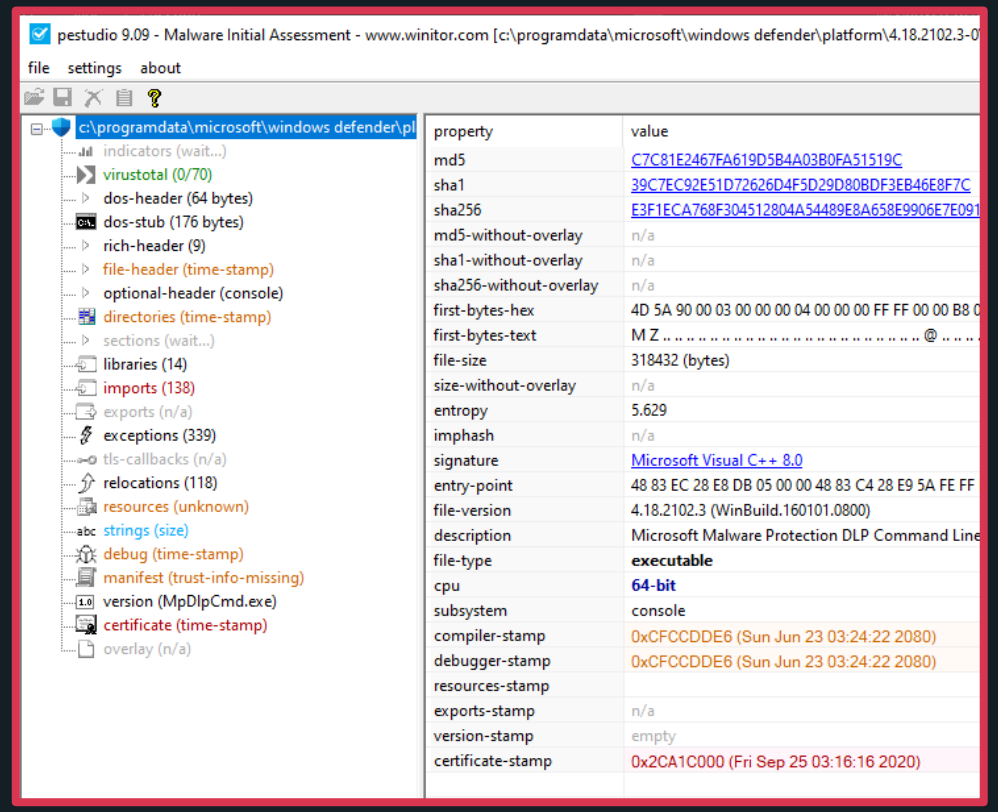
- Detect it Easy
  - Swiss-army knife of examining PEs
  - Quick overview of any specified file (incl. compiler, packer, linker, etc.) based on community-submitted signatures.
  - Examine import tables, exports, hashes of the file, strings, and more!

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



Other Third-party Tools



- 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!

Lab #2: Connecting the Dots

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

Lab #2: Connecting the Dots

Lab #2: Connecting the Dots

Start by looking up the hash
`fd866f6e1b997c31bdb6ba24361663e5`

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

TEAMT5

Putting it Together: Example

- You stumbled upon a zipped sample `86950b81df2003d08ae4a7869ecf88fe` on an online sandbox platform.

Lab #2

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!

TEAMT5

Putting it Together: Example

TEAMT5

- 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`"

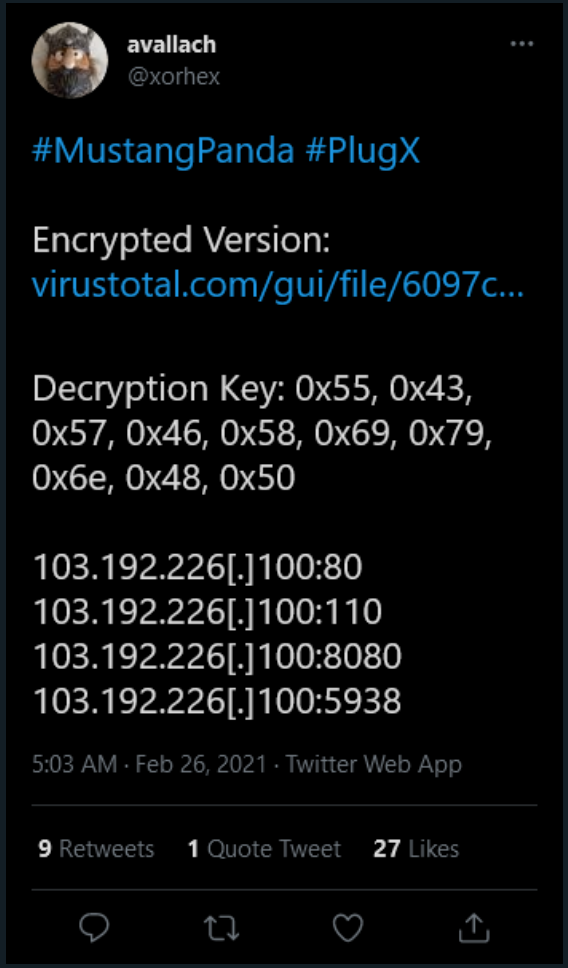
Lab #2

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

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

TEAMT5

Putting it Together: Example



(Avallach (@xorhex) / Twitter, 2021)

- 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`.

Lab #2

Who are MustangPanda and what is PlugX?

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

TEAMT5

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)

PlugX

aka: Destroy RAT, Kaba, Korplug, Sogu, TIGERPLUG

Actor(s): APT 22, APT 26, APT31, APT41, Aurora Panda, Calypso group, DragonOK, Emissary Pan
Stone Panda, UPS, Violin Panda

RSA describes PlugX as a RAT (Remote Access Trojan) malware family that is around since 200
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?

TEAMT5

- 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)

Lab #3: Your First YARA Rule

TEAMT5

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

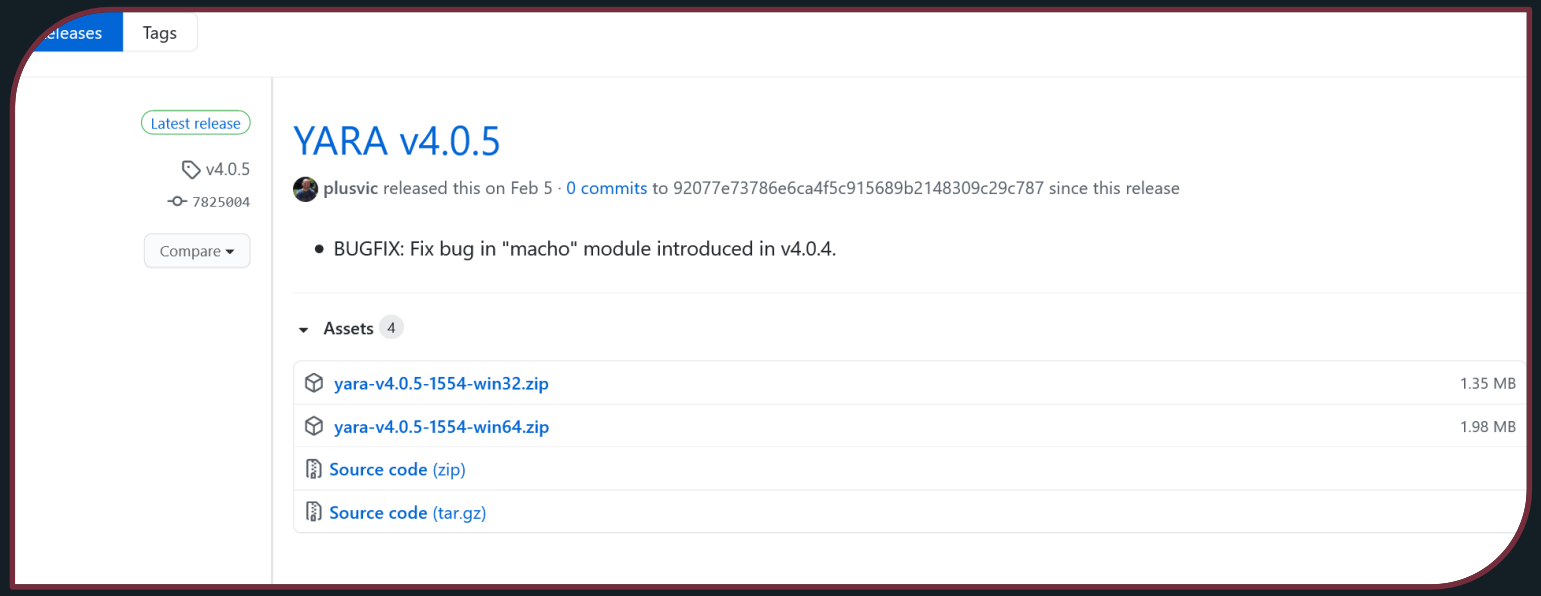
- (*VirusTotal/Yara*, 2012/2021)

```
rule REDLEAVES_DroppedFile_ObfuscatedShellcodeAndRAT_handkerchief
{
    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
}
```

Lab #3: Your First YARA Rule

TEAMT5

Install the latest YARA standalone scanner via `VirusTotal/Yara` @ GitHub



Lab #3: Your First YARA Rule

TEAMT5

- 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`

```
rule My_First_Rule
{
    strings:
    condition:
}
```

Lab #3: Your First YARA Rule

TEAMT5

- `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:
}
```

Lab #3: Your First YARA Rule

TEAMT5

- `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:
}
```

Lab #3: Your First YARA Rule

TEAMT5

- `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:
}
```

Lab #3: Your First YARA Rule

TEAMT5

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

```
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:
}
```

Lab #3: Your First YARA Rule

TEAMT5

- `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
}
```

Lab #3: Your First YARA Rule

TEAMT5

- `condition` block
  - Conditions can be chained using `or`.
  - Conditions can be limited using `and`.

```
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}
        $pastry = /slice of (cake|pie|bread)/ nocase
    condition:
        ($dead_beef and $face_booc) or
            any of them
}
```

Lab #3: Your First YARA Rule

TEAMT5

- `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_*)`

```
rule My_First_Rule
{
    strings:
        $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)
}
```

Lab #3: Your First YARA Rule

TEAMT5

- `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

Lab #2: Your First YARA Rule

TEAMT5

- `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}
```

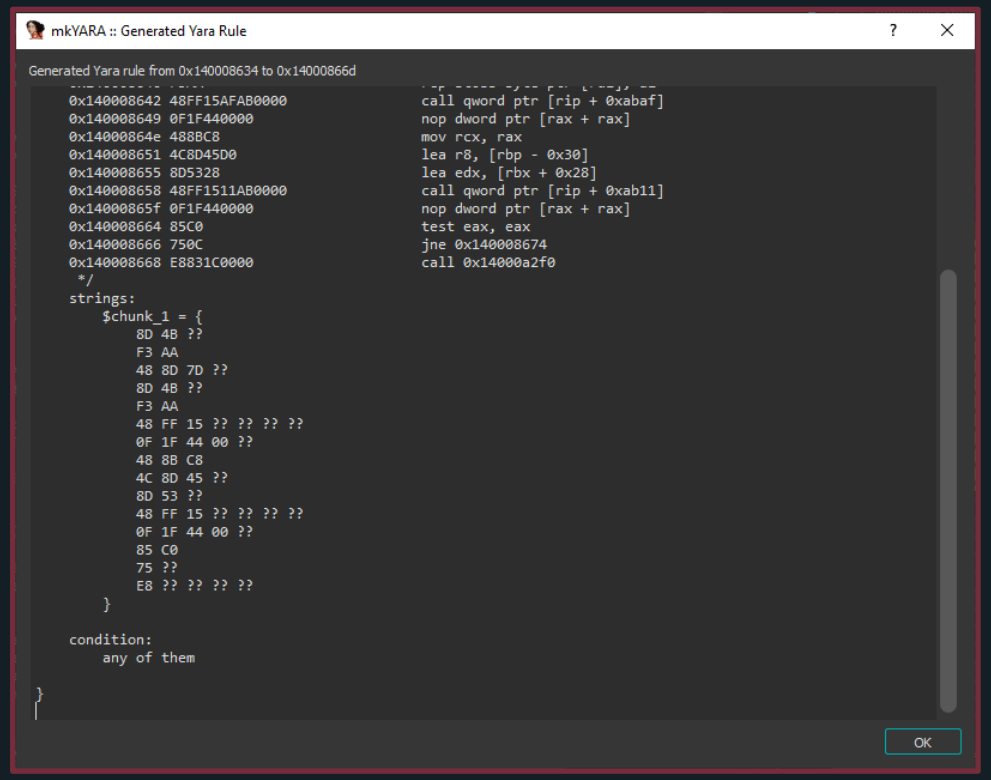
```
000093d0  55                          push ebp
000093d1  8bec                        mov ebp, esp
000093d3  6aff                        push -1
000093d5  a1f8e50210                  mov eax, dword ptr [0x1002e5f8]
000093da  50                          push eax
000093db  e8d0c4ffff                  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
000093f4  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
```

```
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:
}
```

Lab #3: Your First YARA Rule

TEAMT5

- 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



Lab #3: Your First YARA Rule

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

Lab #3: Your First YARA Rule

Now give it a try!

Try writing a YARA rule for FD866F6E1B997C31BDB6BA24361663E5.

TEAMT5

THANK YOU!

@StillAzureH

still@teamt5.org

TEAMT5

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. https://github.com/VirusTotal/yara (Original work published 2012)

TEAMT5