FireEye | CYBER DEFENSE SUMMIT

ATT&CKing FIN7

The Value of Using Frameworks for Threat Intelligence

Regina Elwell, FireEye
Katie Nickels, MITRE

OCTOBER 1 – 4, 2018 | WASHINGTON, D.C.

Agenda

- Why Should We Use Frameworks for Threat Intelligence?
  - Introduction to MITRE ATT&CK™
  - Introduction to the Attack Lifecycle
  - How ATT&CK and the Attack Lifecycle Complement Each Other
- Introduction to FIN7
- FIN7 Targeted Lifecycle Overview
- FIN7 Deep Dive

Why Use a Framework to Organize Threat Intel?

Regardless of which one you choose, it can help you…

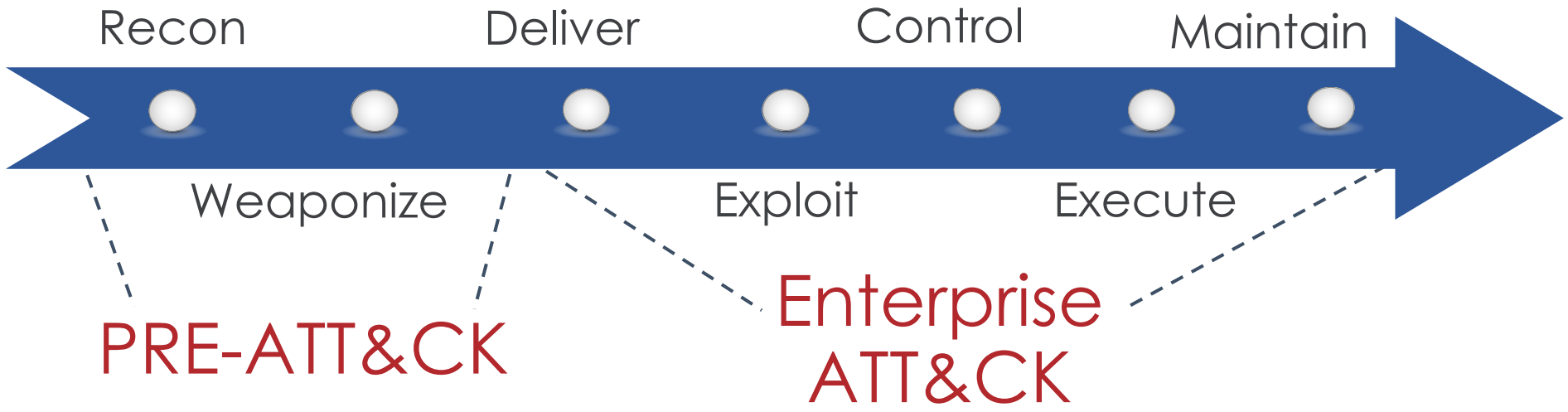
- Identify where you have gaps in knowledge
- Compare adversaries to each other
- Compare adversary behavior to defenses

Introduction to MITRE ATT&CK™

A knowledge base of adversary behavior

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

Mobile ATT&CK



Breaking Down Enterprise ATT&CK

Tactics: the adversary's technical goals

Techniques: how the goals are achieved

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Exfiltration	Command & Control
Hardware Additions	Scheduled Task			Binary Padding	Credentials in Registry	Browser Bookmark Discovery	Exploitation of Remote Services	Data from Information Repositories	Exfiltration Over Physical Medium	Remote Access Tools
Trusted Relationship	LSASS Driver		Extra Window Memory Injection		Exploitation for Credential Access	Network Share	Distributed Component	Video Capture		Port Knocking
Supply Chain Compromise	Local Job Scheduling		Access Token Manipulation						Exfiltration Over Command and Control Channel	Multi-hop Proxy
	Trap									Domain Fronting
Spearphishing Attachment	Launchctl									Data Encoding
	Signed Binary Proxy Execution	Imag						board Data	Data Encrypted	Remote File Copy
Exploit Public-Facing Application								l Collection	Automated Exfiltration	Multi-Stage Channels
	User Execution							en Capture		Web Service
Replication Through Removable Media	Exploitation for Client Execution							ta Staged	Exfiltration Over Other Network Medium	
		AppCert						ut Capture		Standard Non-Application Layer Protocol
Spearphishing via Service	CMSTP	Hookin						rom Network ared Drive	Exfiltration Over Alternative Protocol	
	Dynamic Data Exchange	Startup It							Data Transfer Size Limits	Connection Proxy
Spearphishing Link	Mshta	Launch Da						m Local System		Multilayer Encryption
Drive-by Compromise	AppleScript	Dylib Hija						the Browser	Data Compressed	Standard Application Layer Protocol
Valid Accounts	Source	Application Sh						om Removable Media	Scheduled Transfer	
	Space after Filename	AppInit D								Commonly Used Port
	Execution through Module Load	Web Sh								Standard Cryptographic Protocol
		Service Registry Permi								Custom Cryptographic Protocol
	Regsvcs/Regasm	New Serv								Data Obfuscation
	InstallUtil	File System Permiss								Custom Command and Control Protocol
	Regsvr32	Path Interc								Communication Through Removable Media
	Execution through API	Accessibility F								Multiband Communication
	PowerShell	Port Mon								Fallback Channels
	Rundll32	Kernel Modules and Extensions								Uncommonly Used Port
	Third-party Software									
	Scripting	Port Knocking								
	Graphical User Interface	SIP and Trust Provider Hijacking								
	Command-Line Interface		Exploitation for Privilege Escalation	Hidden Window						
		Screensaver								

Main page
Help
Contribute
References
Data Drilldown
Using the API

Tactics
Initial Access
Execution
Persistence
Privilege Escalation
Defense Evasion
Credential Access
Discovery
Lateral Movement
Collection
Exfiltration
Command and Control

Techniques
Technique Matrix
All Techniques
Windows
Linux
macOS
Add a Technique

Groups
All Groups
Add a Group

Software
All Software
Add Software

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.

Contents [hide]
1 Examples
2 Mitigation

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	Yes

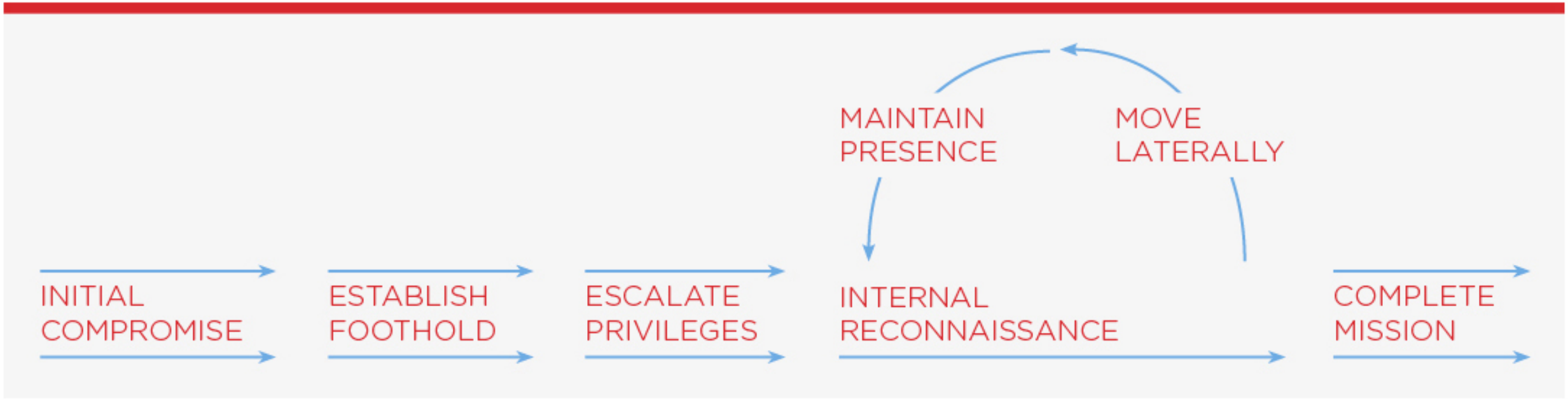
Leo Loobeek, @leoloobeek,
Alain Homewood, Insomnia Security

Procedures – Specific technique implementation

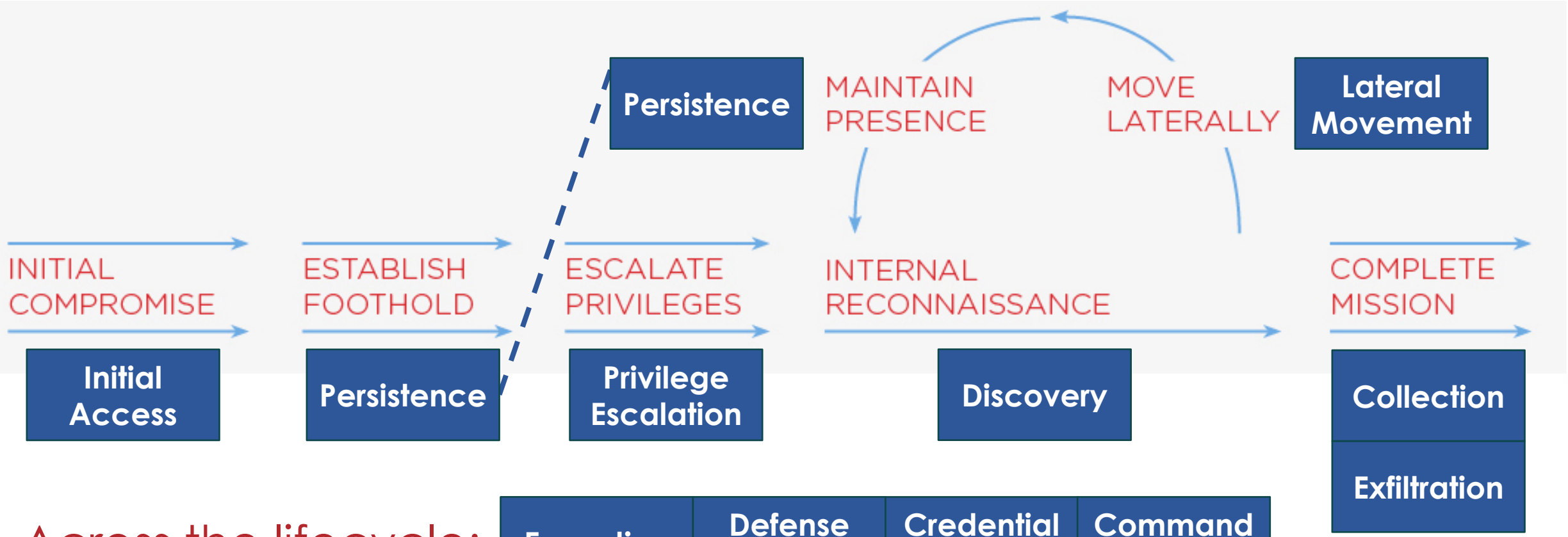
Examples

- APT18 actors used the native at Windows task scheduler tool to use scheduled tasks for execution on a victim network.[2]
- APT29 used named and hijacked scheduled tasks to establish persistence.[3]
- An APT3 downloader creates persistence by creating the following scheduled task: `schtasks /create /tn "mysc" /tr C:\Users\Public\test.exe /sc ONLOGON /ru "System"` .[4]
- APT32 has used scheduled tasks to persist on victim systems.[5]
- BRONZE BUTLER has used at and schtasks to register a scheduled task to execute malware during lateral movement.[6]
- Dragonfly 2.0 used scheduled tasks to automatically log out of created accounts every 8 hours as well as to execute tools to

The Targeted Attack Lifecycle



How ATT&CK and the Attack Lifecycle are Complementary



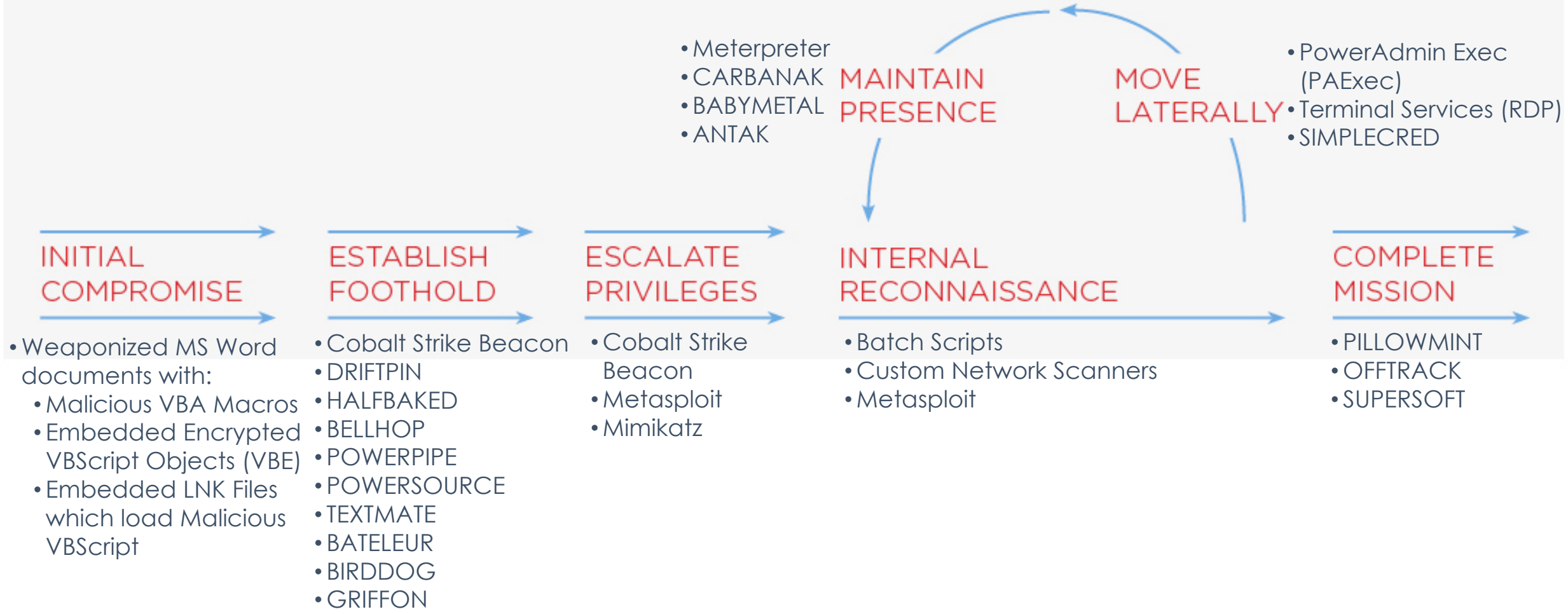
Across the lifecycle: **Execution** | **Defense Evasion** | **Credential Access** | **Command & Control**

FIN7

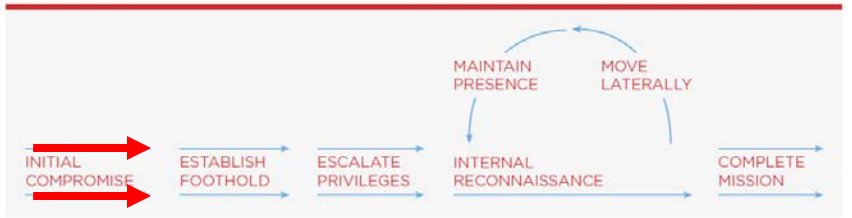
Introduction

- Active since late 2015
- Financially motivated
- Primary objective: point of sale compromise
- Mainly use spearphishing for malware distribution
- Limited use of exploits, and no known use of zero-day exploits
- Blend of publicly available and unique or altered tools

FIN7 Targeted Attack Lifecycle



Spearphishing



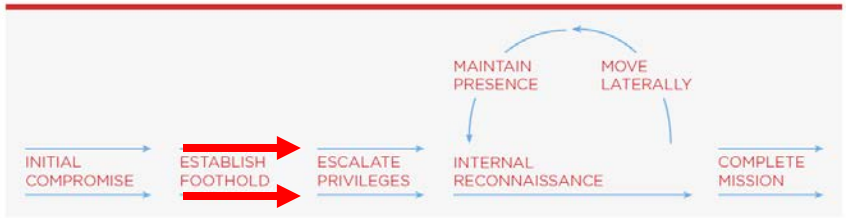
- Targeted spearphishing with customized lures **ATT&CK T1193: Spearphishing with attachment**
  - Weaponized Word documents with malicious VBA macros **T1064: Scripting**
  - LNK files used to launch VBA code embedded within document contents
  - Embedded OLE objects containing malware **T1173: Dynamic Data Exchange**
- Use social engineering to encourage response

T1204: User execution

Spearphishing: Mitigation and Detection

- User training
  - Even if they click, will they report?
  - Don't rely *just* on this
- Tools: email filtering and application whitelisting
- Use GPO to block execution of macros in documents from the Internet
- Create analytics on suspicious execution chains to detect macros
  - Example: winword.exe spawning cmd.exe, wscript.exe, or powershell.exe

HALFBAKED

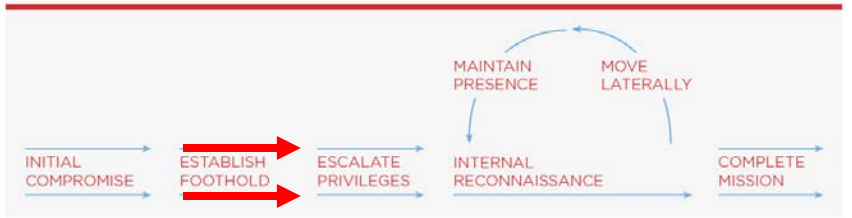


- The HALFBAKED malware has several components:
  - A dropper contained in a VBA Macro which writes out the installer and backdoor to the infected system **T1064: Scripting**
  - A VBScript installer which installs the backdoor as a persistent service **T1050: New Service**
  - A VBScript backdoor possessing typical capabilities:
    - Reverse shell **T1059: Command-Line Interface**
    - Execute shell commands
    - Upload and download files **T1105: Remote File Copy**
    - Uses Windows Management Instrumentation (WMI) to collect reconnaissance details **T1047: WMI**

HALFBAKED: Detection and Mitigation

- Implement least-privilege model for domain users
  - Ensure domain users are not in local admins group
- Monitor service creation through command-line invocation and look for low frequency services in your environment
- Monitor network traffic for WMI connections and capture command-line arguments of "**wmic**"
  - Look for anomalies in systems using WMI

BELLHOP



- BELLHOP is a javascript-based backdoor interpreted using the native Windows Scripting Host (WSH)

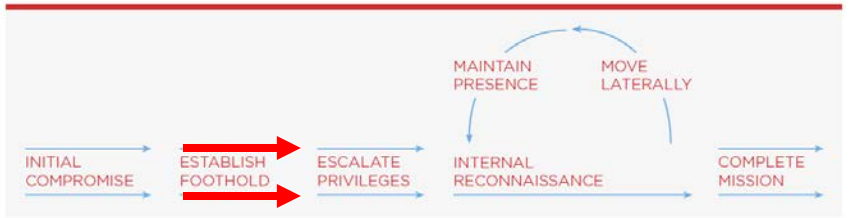
T1082: System Information Discovery

  - The BELLHOP dropper gathers basic host information and downloads a base64-encoded blob of javascript to disk and sets up persistence in three ways:
    - Creating a Run key in the Registry **T1060: Registry Run Keys**
    - Creating a RunOnce key in the Registry
    - Creating a persistent named scheduled task **T1053: Scheduled Task**
  - BELLHOP communicates using HTTP and HTTPS with primarily benign sites such as Google documents and Pastebin **T1071: Standard Application Layer Protocol** **T1102: Web Service**

BELLHOP: Mitigation and Detection

- Monitor for **ver**, **systeminfo**, and **dir** executed from the command line
  - Create a detection that chain these with other discovery commands
- Monitor for Registry run keys that do not correlate with known software
- Limit privileges of user accounts so only authorized admins can create scheduled tasks on remote systems
- Configure event logging for scheduled task creation and changes by enabling "Microsoft-Windows-TaskScheduler/Operational" in event logging
  - Example BELLHOP Scheduled Task: SysChecks

POWERSOURCE & TEXTMATE



- POWERSOURCE is a heavily obfuscated and modified version of the publicly available tool DNS_TXT_Pwnage **T1027: Obfuscated Files or Information**
  - Installed in the registry or Alternate Data Streams **T1060: Registry Run Keys** **T1096: NTFS File Attributes**
  - Uses DNS TXT requests (port 53) for command and control

    T1071: Standard App Layer Protocol **T1043: Commonly Used Port**

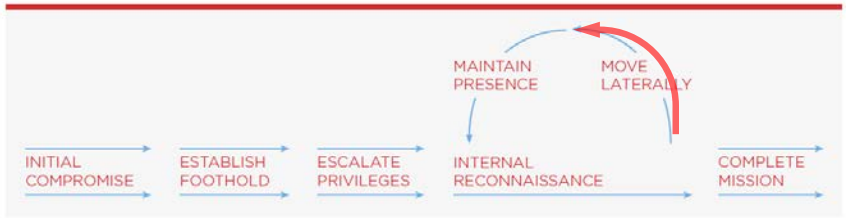
- TEXTMATE has been observed being downloaded via POWERSOURCE
  - Second-stage "file-less" payload, runs in memory via PowerShell **T1086: PowerShell**
  - Implements reverse shell via DNS TXT (port 53) commands

    T1059: Command-Line Interface **T1071: Standard Application Layer Protocol** **T1043: Commonly Used Port**

POWERSOURCE & TEXTMATE: Mitigation and Detection

- Force web traffic through a proxy
  - Including DNS traffic – do not allow Internet DNS resolution
- Flag and analyze commands containing indicators of obfuscation and known suspicious syntax such as uninterpreted escape characters like ^ and “
- Restrict PowerShell execution policy to administrators and to only execute signed scripts

PowerAdmin Exec (PAExec)



- PowerAdmin Exec (PAExec)
  - Functionally similar to SysInternals PsExec, PAExec supports execution of remote commands

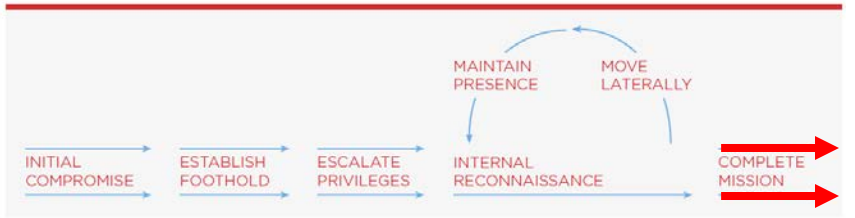
T1035: Service Execution

  - Most forensic artifacts are created on the *source* and not the target

PAExec: Mitigation and Detection

- Look for unusual file names such as “logsXXX.exe” (unique to FIN7)
- Monitor for unusual executables running from “C:\Windows\Temp\”
- If you have technology capable of it, look at binaries for:
  - CompanyName Power Admin LLC
  - FileDescription PAExec Application
  - InternalName PAExec
  - OriginalFilename PAExec.exe

PILLOWMINT

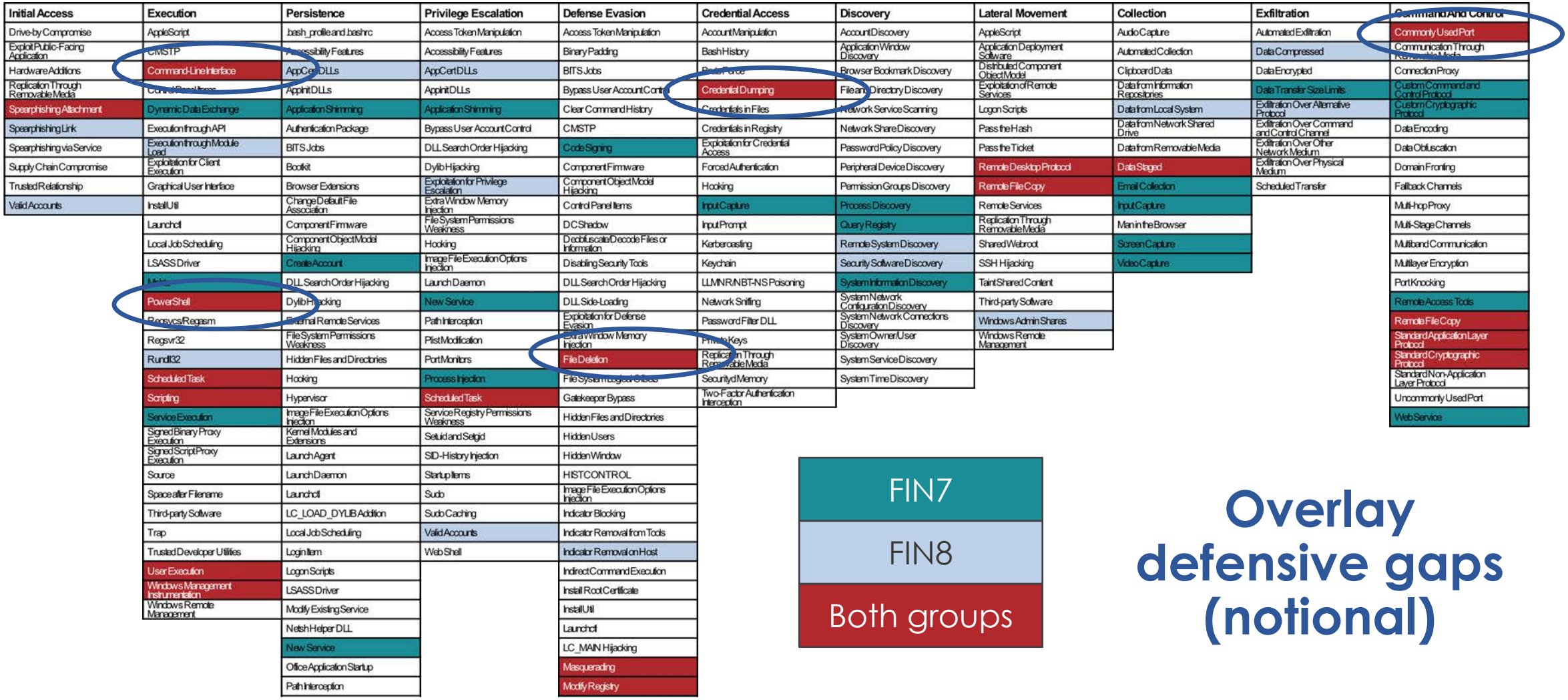


- PILLOWMINT is a Point-of-Sale malware tool used to scrape track 1 and track 2 payment card data from memory
  - Scraped payment card data is encrypted and stored in the registry and as plaintext in a file **T1074: Data Staged**
  - Contains additional backdoor capabilities including:
    - Running processes
    - Downloading and executing files **T1105: Remote File Copy**
    - Downloading and injecting DLLs **T1055: Process Injection**
  - Communicates with a command and control (C2) server over HTTP using AES encrypted messages **T1071: Standard Application Layer Protocol** **T1032: Standard Cryptographic Protocol**

PILLOWMINT: Mitigation and Detection

- Implement point-to-point encryption and tokenization
- Use data loss prevention software
- Look for registry keys:
  - HKLM\System\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\server
  - HKLM\System\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\comman
  - HKLM\System\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\PDSK21_<random>
- Look for output files in the directory: %WINDIR%\system32\sysvols\

Using Structured Threat Intelligence



FIN7

FIN8

Both groups

Overlay defensive gaps (notional)

Conclusion

- Frameworks are useful for organizing threat intel regardless of which one
- Consider which framework based on your use case, and consider combining them for analysis
- FIN7 has been successful because they use social engineering and well-disguised lures
- FIN7 continues to be successful because they are constantly adapting and evolving to prevent detection
- For the best chance of detecting FIN7, look across their attack lifecycle and ATT&CK techniques they use

Additional Resources

- Visit https://attack.mitre.org for more information on ATT&CK
  - FIN7: https://attack.mitre.org/wiki/Group/G0046
  - Contact us: attack@mitre.org
- More information on FIN7:
  - On the Hunt for FIN7: Pursuing an Enigmatic and Evasive Global Criminal Operation https://www.fireeye.com/blog/threat-research/2018/08/fin7-pursuing-an-enigmatic-and-evasive-global-criminal-operation.html
  - Tracking a Cyber Crime Group: FIN7 at a Glance https://www.fireeye.com/blog/executive-perspective/2018/08/tracking-a-cyber-crime-group-fin7-at-a-glance.html

Questions?