Intro to Threat Emulation

Introduction

Red and Blue Teams are essential in identifying, detecting and addressing vulnerabilities. In a continually evolving threat landscape, security operations centres (SOC) need to reduce the impact of the cyber security skills gap, gain confidence in their ability to prevent a data breach, and get real-world training through experience. This can be facilitated through effective collaboration between the teams while addressing security breaches and during training and emulation practices. Understanding the difference between cyber security simulation and emulation can help you build a more robust threat detection and response program that strengthens security.

Learning Objectives

- Understand what Threat Emulation is.
- Identify various frameworks used in Threat Emulation.
- Understand how to plan, execute and report emulation exercises.

Prerequisites

Before embarking on this room, it is highly recommended to review the rooms in the following modules:

- Cyber Defence Frameworks
- Cyber Threat Intelligence

What is Threat Emulation?

Purpose of Threat Emulation

Threat emulation is meant to assist security teams and organisations, in general, in better understanding their security posture and their defence mechanisms and performing due diligence in their compliance. This ensures they are provided with an adversary's perspective of an attack without the hassle of dealing with an actual threat with malicious intent. Additionally, the organisation will be well prepared if a real-time and sophisticated attack is initiated against them. With this know-how, the following common assumptions about an attack would be avoided:

- "We applied all patches."
- "Our applications have multi-factor authentication applied."
- "We have the network segmented, implemented a DMZ, and traffic flows through a proxy."
- "Nothing will go through our firewalls, antivirus and IDS solutions."

Cyber Security Assessment Issues

Network owners and IT executives often wish to understand their cyber security effectiveness. They will usually line up the following common questions to be answered:

- Are our people trained and alert?
- Are our internal processes effective?
- Has the technology in use properly configured and delivered value to the business?

These questions are addressed through cyber security assessments, mainly red team engagements, vulnerability assessments and penetration tests. With that, let's understand how these practices contribute to security assessments.

Vulnerability assessments are conducted to identify vulnerabilities in assets under a defined scope. The focus here is comprehensive and based on the rules of engagement defined, as assessments do not include exploitation.

Penetration testing involves exploiting vulnerabilities within an organisation under strict control of the scope and rules of engagement. Pentests provide organisations with information about their security posture, patching vulnerabilities and where to invest in security training or practices.

Red teaming provides a means of looking at cyber security issues from an adversary's perspective. This generates attacks on the organisation as though the actual adversary is attacking. This is done in the hope of making the defensive measures better and improving their detections.

The challenge of these assessments is that they do not represent real-world threats. They commonly identify initial access vectors that attackers may use and do not facilitate the entire attack cycle.

Another challenge from these assessments is the lack of incentivisation of Red and Blue teams during engagements. No party wishes to share their intel and TTPs (Tactics, Techniques, and Procedures) even when the benefits overlap for both teams.

Emulation is here to address these challenges and provide a holistic security evaluation.

Emulation vs Simulation

There is no standard definition of this discipline within the industry, as people use different terminologies to mean roughly the same thing. The familiar words used are threat emulation, adversary emulation, attack simulation and purple teaming. However, we shall use threat and adversary emulation interchangeably for this room and differentiate them from simulation. Therefore, what is Threat Emulation?

Threat emulation is an intelligence-driven impersonation of real-world attack scenarios and TTPs in a controlled environment to test, assess and improve an organisation's security defences and response capabilities. This means that you seek to behave as the adversary would. Threat emulation aims to identify and mitigate security gaps before attackers exploit them.

Emulation can be conducted as a blind operation - mainly as a Red Team engagement and unannounced to the rest of the defensive security team - or as a non-blind operation involving all security teams and ensuring knowledge sharing.

In contrast, threat simulation commonly represents adversary functions or behaviour through predefined and automated attack patterns that pretend to represent an adversary. This implies that the actions taken during the exercise will combine TTPs from one or more groups but not an exact imitation of a particular adversary.

Key Concepts

Threat emulation would be seen to have several key characteristics which line up well with the Pyramid of Pain. For more information on the Pyramid, check out the linked room. The concepts include the following:

- Real-world threats: The MITRE ATT&CK framework and cyber threat intelligence are common information sources to ensure threat TTPs are based on actual breaches, APTs and campaigns.
- Behaviour-focused: The execution of TTPs during an emulation exercise aims to tune defences based on behaviours and not signatures, thus adapting to the elements of the Pyramid.
- Transparency: Disclosure of activities between the Red and Blue teams during execution ensures that the security posture is improved holistically.
- Collaborative: Due to the common goal of improving organisational security, threat emulation allows teams to collaborate in their efforts.
- Repeatable: Some emulation tasks would be done multiple times in the course on an exercise or numerous exercises. These tasks can be automated, creating a baseline of continuous practical security assessments and deployment.

Emulation can be applied to numerous instances, each with its own goals:

- Assessments & Improvement: The goal is to test personnel, assess security processes and evaluate the technology adopted.

- Capability Development: Emulation enables the creation, modification and application of tools and analytics derived from TTPs.
- Professional Development: What better way to teach and promote knowledge sharing about adversary behaviours and frameworks? This breaks down the barriers between red and blue teams and fosters collaboration missions.

Threat emulation exercises provide vital insights to organisations to assess, manage and improve their abilities to protect their systems effectively against adversaries. At this point, one may ask how you conduct threat emulation and what methodologies can be followed for a successful exercise. We shall dive into that in the next task.

****

Answer the questions below:

What can be defined as an intelligence-driven impersonation of real-world attacks?
Answer: Threat Emulation

What is the exercise of representing adversary functions through predefined and automated attack patterns?
Answer: Threat Simulation

Emulation Methodologies

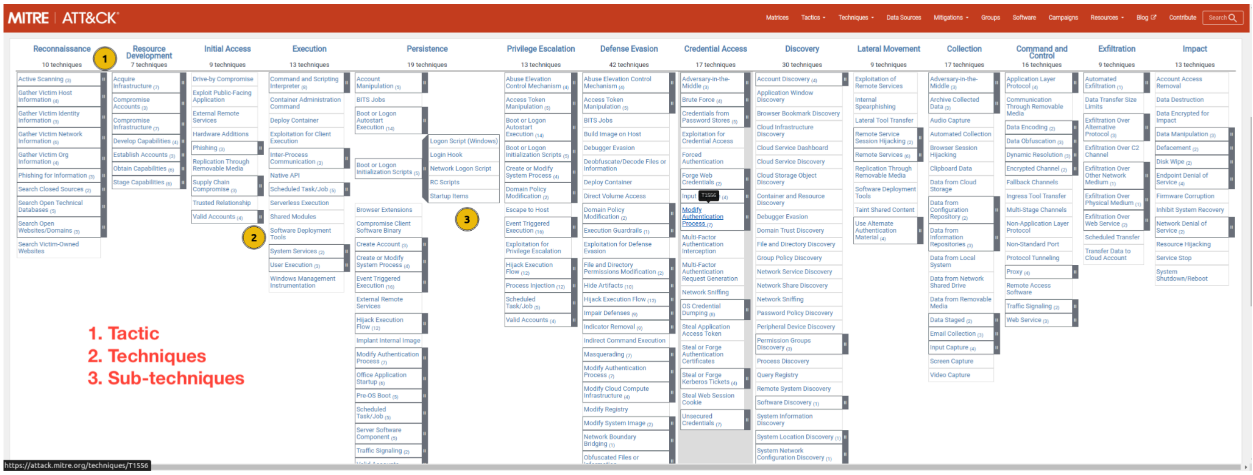
Threat emulation methodologies are strategies, plans and procedures used to simulate and test network defences and systems against adversaries. There are various methodologies, each with its unique approach and level of technicality, but all share the goal of discovering weaknesses in security. It is essential to know that no adversary is alike. However, they would follow a methodology and have their workflows. The methodologies described in this task seek to provide a knowledge base for organisations when dealing with threats and to plan their emulation exercises.

Additionally, these methodologies can be combined when formulating your emulation plan; as we shall see, some are already integrated. Let us take a look at some of the methodologies.

MITRE ATT&CK

The MITRE ATT&CK Framework is an industry-known knowledge base that provides information about known adversarial TTPs observed in actual attacks and breaches. Threat emulation teams can extract many benefits from integrating ATT&CK with their engagements as it would make it efficient when writing reports and mitigations related to the behaviours experimented with.

The MITRE ATT&CK matrix visually represents attackers' techniques to accomplish a specific objective. It showcases 14 tactics, from reconnaissance to impact and within each tactic, several adversary techniques are listed and describe the activity carried out. An extension of the ATT&CK matrix is the Navigator, a web-based tool for exploring ATT&CK matrices by creating colour-coded heatmap layers of techniques and sub-techniques used by a particular adversary.



Reconnaissance
10 techniques
- Active Scanning (3)
- Gather Victim Host Information (4)
- Gather Victim Identity Information (3)
- Gather Victim Network Information (6)
- Gather Victim Org Information (4)
- Phishing for Information (3)
- Search Closed Sources (2)
- Search Open Technical Databases (5)
- Search Open Websites/Domains (3)
- Search Victim-Owned Websites

Resource Development
7 techniques
- Acquire Infrastructure (7)
- Compromise Accounts (3)
- Compromise Infrastructure (7)
- Develop Capabilities (4)
- Establish Accounts (3)
- Obtain Capabilities (6)
- Stage Capabilities (6)

Initial Access
9 techniques
- Drive-by Compromise
- Exploit Public-Facing Application
- External Remote Services
- Hardware Additions
- Phishing (3)
- Replication Through Removable Media
- Supply Chain Compromise (3)
- Trusted Relationship
- Valid Accounts (4)

Execution
13 techniques
- Command and Scripting Interpreter (8)
- Container Administration Command
- Deploy Container
- Exploitation for Client Execution
- Inter-Process Communication (3)
- Native API
- Scheduled Task/Job (5)
- Serverless Execution
- Shared Modules
- Software Deployment Tools
- System Services (2)
- User Execution (3)
- Windows Management Instrumentation

Persistence
19 techniques
- Account Manipulation (5)
- BITS Jobs
- Boot or Logon Autostart Execution (14)
- Boot or Logon Initialization Scripts (5)
  - Logon Script (Windows)
  - Login Hook
  - Network Logon Script
  - RC Scripts
  - Startup Items
- Browser Extensions
- Compromise Client Software Binary
- Create Account (3)
- Create or Modify System Process (4)
- Event Triggered Execution (16)
- External Remote Services
- Hijack Execution Flow (12)
- Implant Internal Image
- Modify Authentication Process (7)
- Office Application Startup (6)
- Pre-OS Boot (5)
- Scheduled Task/Job (5)
- Server Software Component (5)
- Traffic Signaling (2)
- Valid Accounts

Privilege Escalation
13 techniques
- Abuse Elevation Control Mechanism (4)
- Access Token Manipulation (5)
- Boot or Logon Autostart Execution (14)
- Boot or Logon Initialization Scripts (5)
- Create or Modify System Process (4)
- Domain Policy Modification (2)
- Escape to Host
- Event Triggered Execution (16)
- Exploitation for Privilege Escalation
- Hijack Execution Flow (12)
- Process Injection (12)
- Scheduled Task/Job (5)
- Valid Accounts (4)

Defense Evasion
42 techniques
- Abuse Elevation Control Mechanism (4)
- Access Token Manipulation (5)
- BITS Jobs
- Build Image on Host
- Debugger Evasion
- Deobfuscate/Decode Files or Information
- Deploy Container
- Direct Volume Access
- Domain Policy Modification (2)
- Execution Guardrails (1)
- Exploitation for Defense Evasion
- File and Directory Permissions Modification (2)
- Hide Artifacts (10)
- Hijack Execution Flow (12)
- Impair Defenses (9)
- Indicator Removal (9)
- Indirect Command Execution
- Masquerading (7)
- Modify Authentication Process (7)
- Modify Cloud Compute Infrastructure (4)
- Modify Registry
- Modify System Image (2)
- Network Boundary Bridging (1)
- Obfuscated Files or Information

Credential Access
17 techniques
- Adversary-in-the-Middle (3)
- Brute Force (4)
- Credentials from Password Stores (5)
- Exploitation for Credential Access
- Forced Authentication
- Forge Web Credentials (2)
- Input T1556 (4)
- Modify Authentication Process (7)
- Multi-Factor Authentication Interception
- Multi-Factor Authentication Request Generation
- Network Sniffing
- OS Credential Dumping (8)
- Steal Application Access Token
- Steal or Forge Authentication Certificates
- Steal or Forge Kerberos Tickets (4)
- Steal Web Session Cookie
- Unsecured Credentials (7)

Discovery
30 techniques
- Account Discovery (4)
- Application Window Discovery
- Browser Bookmark Discovery
- Cloud Infrastructure Discovery
- Cloud Service Dashboard
- Cloud Service Discovery
- Cloud Storage Object Discovery
- Container and Resource Discovery
- Debugger Evasion
- Domain Trust Discovery
- File and Directory Discovery
- Group Policy Discovery
- Network Service Discovery
- Network Share Discovery
- Network Sniffing
- Password Policy Discovery
- Peripheral Device Discovery
- Permission Groups Discovery (3)
- Process Discovery
- Query Registry
- Remote System Discovery
- Software Discovery (1)
- System Information Discovery
- System Location Discovery (1)
- System Network Configuration Discovery (1)

Lateral Movement
9 techniques
- Exploitation of Remote Services
- Internal Spearphishing
- Lateral Tool Transfer
- Remote Service Session Hijacking (2)
- Remote Services (6)
- Replication Through Removable Media
- Software Deployment Tools
- Taint Shared Content
- Use Alternate Authentication Material (4)

Collection
17 techniques
- Adversary-in-the-Middle (3)
- Archive Collected Data (3)
- Audio Capture
- Automated Collection
- Browser Session Hijacking
- Clipboard Data
- Data from Cloud Storage
- Data from Configuration Repository (2)
- Data from Information Repositories (3)
- Data from Local System
- Data from Network Shared Drive
- Data from Removable Media
- Data Staged (2)
- Email Collection (3)
- Input Capture (4)
- Screen Capture
- Video Capture

Command and Control
16 techniques
- Application Layer Protocol (4)
- Communication Through Removable Media
- Data Encoding (2)
- Data Obfuscation (3)
- Dynamic Resolution (3)
- Encrypted Channel (2)
- Fallback Channels
- Ingress Tool Transfer
- Multi-Stage Channels
- Non-Application Layer Protocol
- Non-Standard Port
- Protocol Tunneling
- Proxy (4)
- Remote Access Software
- Traffic Signaling (2)
- Web Service (3)

Exfiltration
9 techniques
- Automated Exfiltration (1)
- Data Transfer Size Limits
- Exfiltration Over Alternative Protocol (3)
- Exfiltration Over C2 Channel
- Exfiltration Over Other Network Medium (1)
- Exfiltration Over Physical Medium (1)
- Exfiltration Over Web Service (2)
- Scheduled Transfer
- Transfer Data to Cloud Account

Impact
13 techniques
- Account Access Removal
- Data Destruction
- Data Encrypted for Impact
- Data Manipulation (3)
- Defacement (2)
- Disk Wipe (2)
- Endpoint Denial of Service (4)
- Firmware Corruption
- Inhibit System Recovery
- Network Denial of Service (2)
- Resource Hijacking
- Service Stop
- System Shutdown/Reboot

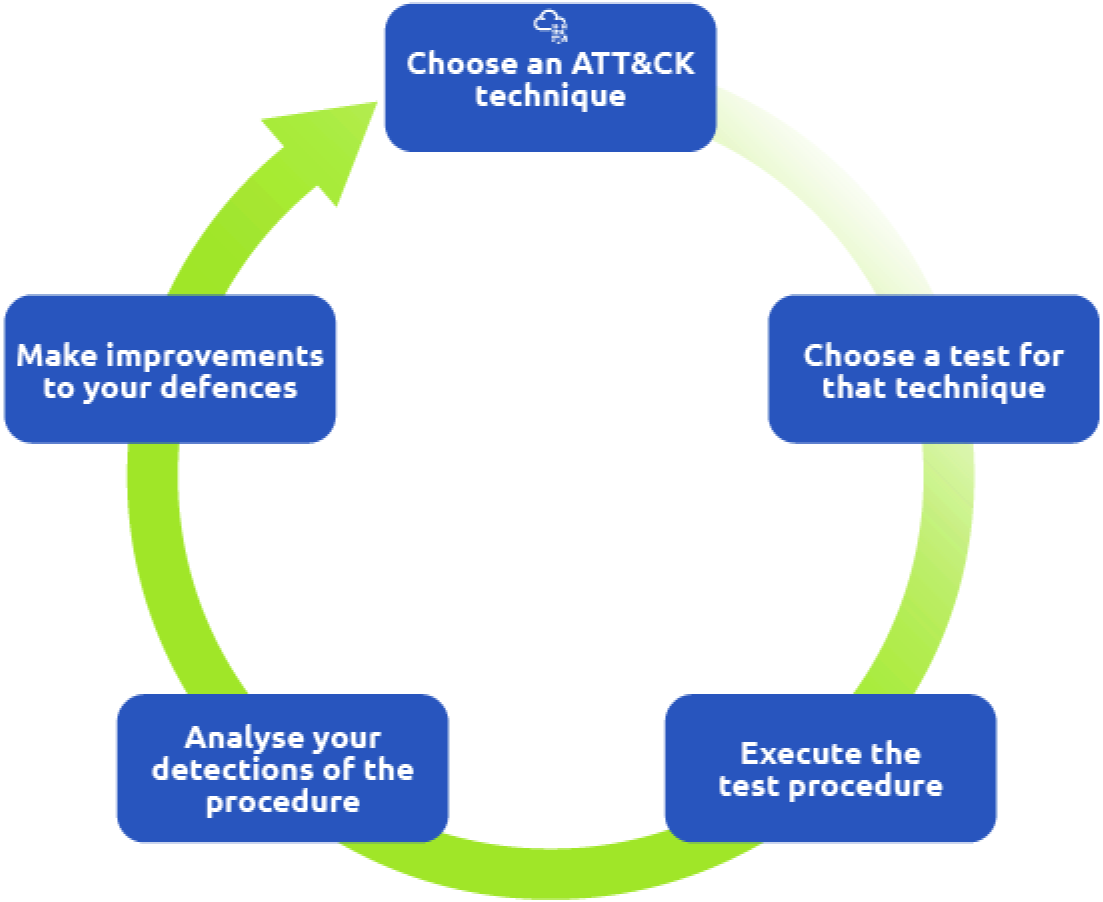
1. Tactic
2. Techniques
3. Sub-techniques

https://attack.mitre.org/techniques/T1556

Atomic Testing

The Atomic Red Team is a library of emulation tests developed and curated by Red Canary that can be executed to test security defences within an organisation. The testing framework provides a mechanism for learning what malicious activities look like and provides telemetry from every test to facilitate defence improvements.

The atomics (individual tests) are mapped to the MITRE ATT&CK framework, providing a pivot between threat profiles and emulation. Atomic Red Team supports emulation on a wide range of platforms, not only on known Operating Systems but also in Cloud Environments. More on the Atomic Red Team has been covered in these rooms: Atomic Red Team & Caldera.


Choose an ATT&CK technique
Choose a test for that technique
Execute the test procedure
Analyse your detections of the procedure
Make improvements to your defences

TIBER-EU Framework

The Threat Intelligence-based Ethical Red Teaming (TIBER-EU) is the European framework developed to deliver controlled, bespoke, intelligence-led emulation testing on entities and organisations' critical live production systems. It is meant to provide a guideline for stakeholders to test and improve cyber resilience through controlled adversary actions.

The TIBER_EU framework follows a three-phase process for end-to-end adversary testing:

1. Preparation Phase

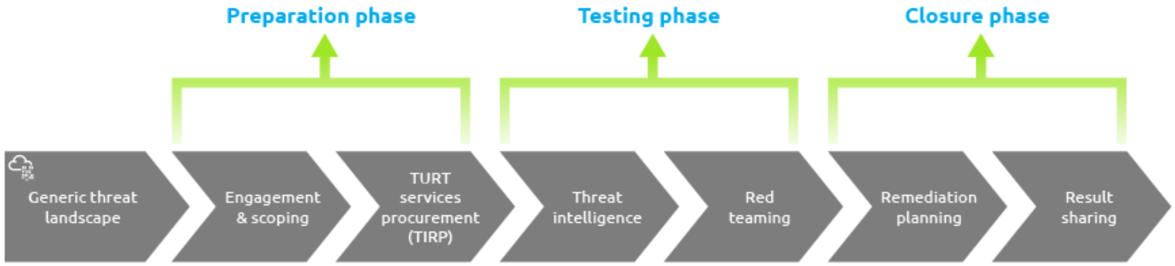
During this phase, the security teams involved in the test are established, and the entity's management determines and approves the scope. This represents the formal launch of adversarial testing by ensuring all the planning and procurement processes are fulfilled.

2. Testing Phase

The Threat Intelligence team will prepare a detailed report to showcase the threat areas for the organisation and set up the necessary attack scenarios based on interested adversarial behaviour. Meanwhile, the Red Team will use this report to craft the emulation tests against the systems, people and processes that underpin critical functions. The Blue Team will look for the attacks and assess how their defence systems perform against them. This ultimately forms the aspect of collaboration between the various security teams.

3. Closure Phase

Once tests are run and defences measured, the emulation team must consider reporting and remediation measures. Each group will draft their analysis reports, including details of the tests conducted, findings and recommendations for technical controls, policies, procedures and awareness training.



CTID Adversary Emulation Library

The Center for Threat-Informed Defense is a non-profit research and development organisation operated by MITRE Engenuity. Its mission is to promote the practice of threat-informed defence. With this mission, they have curated an open-source adversary emulation plan library, allowing organisations to use the plans to evaluate their capabilities against real-world threats.

The library provides users with two approaches to their emulation:

Full Emulation: This is a comprehensive approach to emulating a particular adversary, for example, APT29, typically from initial access to exfiltration. An example of this approach can be found in this APT29 Adversary Emulation repository.

Micro Emulation: This approach is more focused, emulating behaviours across multiple adversaries, such as file access or process injection techniques. You can view existing emulation plans from CTID micro emulation plans on the linked repository.

The Adversary Emulation Process tasks will look at developing an emulation plan for an adversary in ways that will utilise some of the methodologies discussed here.

****
Answer the questions below:

Under TIBER-EU, under which phase would Engagement and Scoping fall?
Answer: Preparation

What is the library that provides technical emulation tests based on TTPs?
Answer: Atomic Red Team

Threat Emulation Process I

Scenario

VASEPY Corp is a multi-billion dollar U.S. retail establishment who are aware of the numerous cyber threats plaguing the retail industry. As a result of the news of a recent breach of one of their competitors, executives have become more concerned about the company's security posture. They have hired you as a Threat Emulation Engineer to plan and execute an adversary emulation exercise based on known threat groups that would target their business.

Developing an internal emulation engagement process for an organisation must be iterative, aligned with recorded cyber security goals, intelligence-driven, and methodical. Security teams can revisit the iterative process, make adjustments where fit and improve the exercise while emulating a given adversary. As an overview, the process's steps are as follows:

- Define Objectives
- Research Adversary TTPs
- Planning the Threat Emulation Engagement
- Conducting the Emulation
- Concluding and Reporting

1. Defining Emulation Objectives

This step is crucial to ensure the exercise remains focused and on track. The objectives should be clearly defined, specific, and measurable. For example, the aim might be to identify how an attacker could gain access to sensitive data on a particular server or, in the case of VASEPY, have to set our objective to identify areas of protection against credit card fraud and ransomware attacks. The scope of the exercise should also be clearly defined, including what the emulation will target as specific systems and data.

2. Research Adversary TTPs

This step aims to accurately model the behaviour of the target adversary so that the emulation team can conduct the exercise realistically and practically. It involves gathering as much information about the threat and identifying behaviours that can be tested based on the set environment. We can break down this process into the following steps:

2.1. Information Gathering

This step starts by gathering information about threats you would be concerned about. This is to avoid instances of selecting arbitrary and non-concerning adversaries such as APT41 that deals with cyber espionage, as opposed to financial fraud. This information can be gathered from various sources, including threat intelligence reports, previous attacks, and publicly available information. More important to note is that as a threat emulation engineer, you would need to start with internal sources, such as network owners, CTI analysts, the cyber defence team and system administrators. These groups will provide information about threats they have seen or heard about and characterise threats based on threat intelligence reports and insights from prior incidents.

Using this for our case, we identify that in Vasepy Corp, security teams are more concerned about financially motivated threat actors that target retail businesses and utilise ransomware. We can establish a shortlist of candidate adversaries that can be emulated for this case using the ATT&CK framework. A quick search provides information about APT groups such as FIN6, FIN7 and FIN8, which all target retail organisations and compromise point-of-sale systems.

2.2. Select The Emulated Adversary

With our shortlist of adversaries, we must narrow it down to one we can emulate. To do so, we can follow a set of critical factors that will influence our selection.

- Relevance: Here, we need to align the adversary to be selected to the engagement objectives and the company's goals. This may even include looking at the geographical relevance of particular APTs.
- Available CTI: Threat intelligence is vital for providing trustworthy information about a threat. For a robust emulation plan, you would need enough reliable resources around the TTPs.
- TTP Complexity: Executing a fruitful emulation plan for complex adversaries who use sophisticated tools and procedures may take a long

time. Here, we must establish whether existing tools can handle the emulation or whether custom ones are required.

- <u>Available Resources</u>: These are primarily in-house resources that must be provisioned for a smooth operation. Budget, time and personnel must be allocated appropriately during the emulation process.

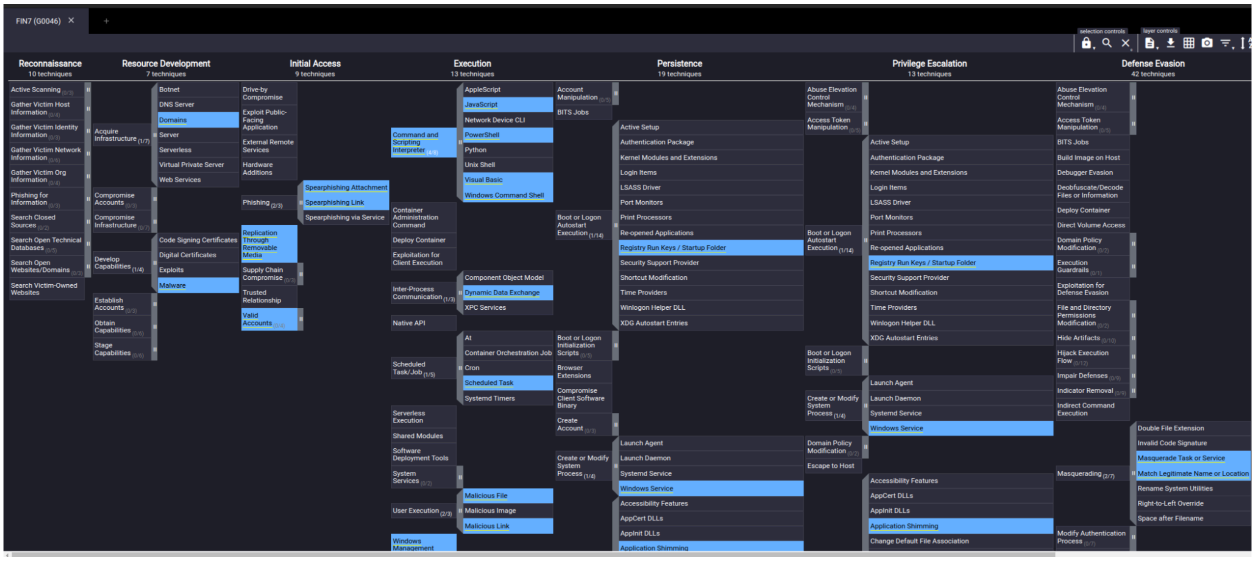
As the Emulation Engineer, you can review these factors, assessing the shortlist of TTPs against them and narrowing down to the appropriate selection. Based on our information, we can select FIN7 as our suitable adversary to emulate, as they target U.S based retail entities such as Vasepy Corp.

2.3. Select The Emulated TTPs

This step aims to accurately model the behaviour of the target adversary so that the exercise can be conducted realistically and practically. The TTPs selected to emulate will drive the rules of engagement, implementations and operations flow of the emulation.

In the case of the FIN7 APT, their TTPs include spear phishing, social engineering, and watering hole attacks. To select which one to emulate, we must understand the TTPs to prioritise our selection. We can visualise these TTPs using the ATT&CK Navigator and look at the specific behaviours the threat group is known to use.

After this, pivoting to CTI resources, such as those listed within the ATT&CK description, would provide information about how the original TTP was executed. This will be followed by creating a scenario outline for the selected TTP, with appropriate context and sources to fulfil its emulation.



Reconnaissance
10 techniques
Active Scanning (0/3)
Gather Victim Host Information (0/4)
Gather Victim Identity Information (0/3)
Gather Victim Network Information (0/6)
Gather Victim Org Information (0/4)
Phishing for Information (0/3)
Search Closed Sources (0/2)
Search Open Technical Databases (0/5)
Search Open Websites/Domains (0/3)
Search Victim-Owned Websites

Resource Development
7 techniques
Acquire Infrastructure (1/7)
Botnet
DNS Server
Domains
Server
Serverless
Virtual Private Server
Web Services
Compromise Accounts (0/3)
Compromise Infrastructure (0/7)
Develop Capabilities (1/4)
Code Signing Certificates
Digital Certificates
Exploits
Malware
Establish Accounts (0/3)
Obtain Capabilities (0/6)
Stage Capabilities (0/6)

Initial Access
9 techniques
Drive-by Compromise
Exploit Public-Facing Application
External Remote Services
Hardware Additions
Phishing (2/3)
Spearphishing Attachment
Spearphishing Link
Spearphishing via Service
Replication Through Removable Media
Supply Chain Compromise (0/3)
Trusted Relationship
Valid Accounts (0/4)

Execution
13 techniques
Command and Scripting Interpreter (4/8)
AppleScript
JavaScript
Network Device CLI
PowerShell
Python
Unix Shell
Visual Basic
Windows Command Shell
Container Administration Command
Deploy Container
Exploitation for Client Execution
Inter-Process Communication (1/3)
Component Object Model
Dynamic Data Exchange
XPC Services
Native API
Scheduled Task/Job (1/5)
At
Container Orchestration Job
Cron
Scheduled Task
Systemd Timers
Serverless Execution
Shared Modules
Software Deployment Tools
System Services (0/2)
User Execution (2/3)
Malicious File
Malicious Image
Malicious Link
Windows Management

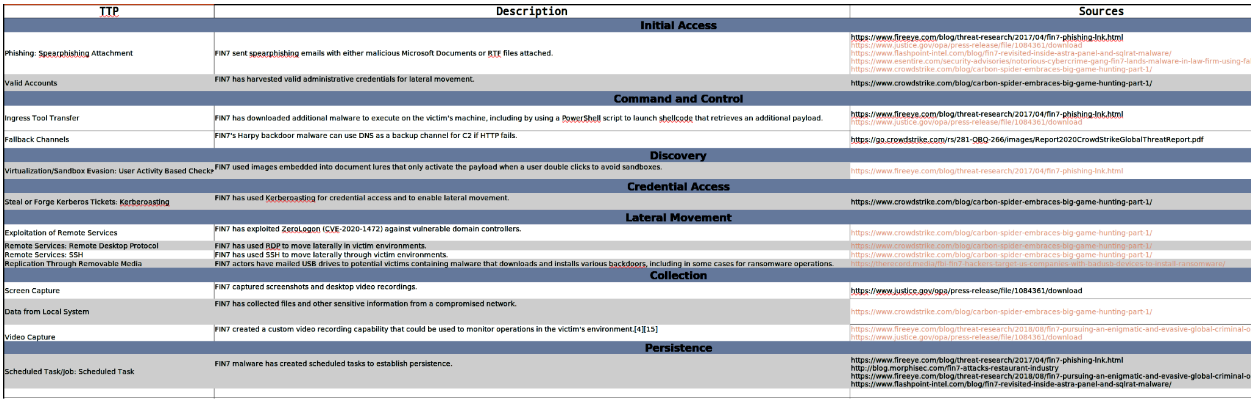
Persistence
19 techniques
Account Manipulation (0/5)
BITS Jobs
Boot or Logon Autostart Execution (1/14)
Active Setup
Authentication Package
Kernel Modules and Extensions
Login Items
LSASS Driver
Port Monitors
Print Processors
Re-opened Applications
Registry Run Keys / Startup Folder
Security Support Provider
Shortcut Modification
Time Providers
Winlogon Helper DLL
XDG Autostart Entries
Boot or Logon Initialization Scripts (0/5)
Browser Extensions
Compromise Client Software Binary
Create Account (0/3)
Create or Modify System Process (1/4)
Launch Agent
Launch Daemon
Systemd Service
Windows Service
Accessibility Features
AppCert DLLs
AppInit DLLs
Application Shimming

Privilege Escalation
13 techniques
Abuse Elevation Control Mechanism (0/4)
Access Token Manipulation (0/5)
Boot or Logon Autostart Execution (1/14)
Active Setup
Authentication Package
Kernel Modules and Extensions
Login Items
LSASS Driver
Port Monitors
Print Processors
Re-opened Applications
Registry Run Keys / Startup Folder
Security Support Provider
Shortcut Modification
Time Providers
Winlogon Helper DLL
XDG Autostart Entries
Boot or Logon Initialization Scripts (0/5)
Create or Modify System Process (1/4)
Launch Agent
Launch Daemon
Systemd Service
Windows Service
Domain Policy Modification (0/2)
Escape to Host
Accessibility Features
AppCert DLLs
AppInit DLLs
Application Shimming
Change Default File Association

Defense Evasion
42 techniques
Abuse Elevation Control Mechanism (0/4)
Access Token Manipulation (0/5)
BITS Jobs
Build Image on Host
Debugger Evasion
Deobfuscate/Decode Files or Information
Deploy Container
Direct Volume Access
Domain Policy Modification (0/2)
Execution Guardrails (0/1)
Exploitation for Defense Evasion
File and Directory Permissions Modification (0/2)
Hide Artifacts (0/10)
Hijack Execution Flow (0/12)
Impair Defenses (0/9)
Indicator Removal (0/9)
Indirect Command Execution
Masquerading (2/7)
Double File Extension
Invalid Code Signature
Masquerade Task or Service
Match Legitimate Name or Location
Rename System Utilities
Right-to-Left Override
Space after Filename
Modify Authentication Process (0/7)

2.4. Construct TTP Outline

The outline aims to drive follow-up threat emulation activities, such as explaining the planned emulation activities, stating the scope and rules of engagement and how the TTPs will be implemented during the exercise. For emulating FIN7, the TTP outline would look similar to the image below:



TTP	Description	Sources
Initial Access		
Phishing: Spearphishing Attachment	FIN7 sent spearphishing emails with either malicious Microsoft Documents or RTF files attached.	https://www.fireeye.com/blog/threat-research/2017/04/fin7-phishing-lnk.html https://www.justice.gov/opa/press-release/file/1084361/download https://www.flashpoint-intel.com/blog/fin7-revisited-inside-astra-panel-and-sqlrat-malware/ https://www.esentire.com/security-advisories/notorious-cybercrime-gang-fin7-lands-malware-in-law-firm-using-fal https://www.crowdstrike.com/blog/carbon-spider-embraces-big-game-hunting-part-1/
Valid Accounts	FIN7 has harvested valid administrative credentials for lateral movement.	https://www.crowdstrike.com/blog/carbon-spider-embraces-big-game-hunting-part-1/
Command and Control		
Ingress Tool Transfer	FIN7 has downloaded additional malware to execute on the victim's machine, including by using a PowerShell script to launch shellcode that retrieves an additional payload.	https://www.fireeye.com/blog/threat-research/2017/04/fin7-phishing-lnk.html https://www.justice.gov/opa/press-release/file/1084361/download
Fallback Channels	FIN7's Harpy backdoor malware can use DNS as a backup channel for C2 if HTTP fails.	https://go.crowdstrike.com/rs/281-OBQ-266/images/Report2020CrowdStrikeGlobalThreatReport.pdf
Discovery		
Virtualization/Sandbox Evasion: User Activity Based Checks	FIN7 used images embedded into document lures that only activate the payload when a user double clicks to avoid sandboxes.	https://www.fireeye.com/blog/threat-research/2017/04/fin7-phishing-lnk.html
Credential Access		
Steal or Forge Kerberos Tickets: Kerberoasting	FIN7 has used Kerberoasting for credential access and to enable lateral movement.	https://www.crowdstrike.com/blog/carbon-spider-embraces-big-game-hunting-part-1/
Lateral Movement		
Exploitation of Remote Services	FIN7 has exploited ZeroLogon (CVE-2020-1472) against vulnerable domain controllers.	https://www.crowdstrike.com/blog/carbon-spider-embraces-big-game-hunting-part-1/
Remote Services: Remote Desktop Protocol	FIN7 has used RDP to move laterally in victim environments.	https://www.crowdstrike.com/blog/carbon-spider-embraces-big-game-hunting-part-1/
Remote Services: SSH	FIN7 has used SSH to move laterally through victim environments.	https://www.crowdstrike.com/blog/carbon-spider-embraces-big-game-hunting-part-1/
Replication Through Removable Media	FIN7 actors have mailed USB drives to potential victims containing malware that downloads and installs various backdoors, including in some cases for ransomware operations.	https://therecord.media/fbi-fin7-hackers-target-us-companies-with-badusb-devices-to-install-ransomware/
Collection		
Screen Capture	FIN7 captured screenshots and desktop video recordings.	https://www.justice.gov/opa/press-release/file/1084361/download
Data from Local System	FIN7 has collected files and other sensitive information from a compromised network.	https://www.crowdstrike.com/blog/carbon-spider-embraces-big-game-hunting-part-1/
Video Capture	FIN7 created a custom video recording capability that could be used to monitor operations in the victim's environment.[4][15]	https://www.fireeye.com/blog/threat-research/2018/08/fin7-pursuing-an-enigmatic-and-evasive-global-criminal-o https://www.justice.gov/opa/press-release/file/1084361/download
Persistence		
Scheduled Task/Job: Scheduled Task	FIN7 malware has created scheduled tasks to establish persistence.	https://www.fireeye.com/blog/threat-research/2017/04/fin7-phishing-lnk.html http://blog.morphisec.com/fin7-attacks-restaurant-industry https://www.fireeye.com/blog/threat-research/2018/08/fin7-pursuing-an-enigmatic-and-evasive-global-criminal-o https://www.flashpoint-intel.com/blog/fin7-revisited-inside-astra-panel-and-sqlrat-malware/

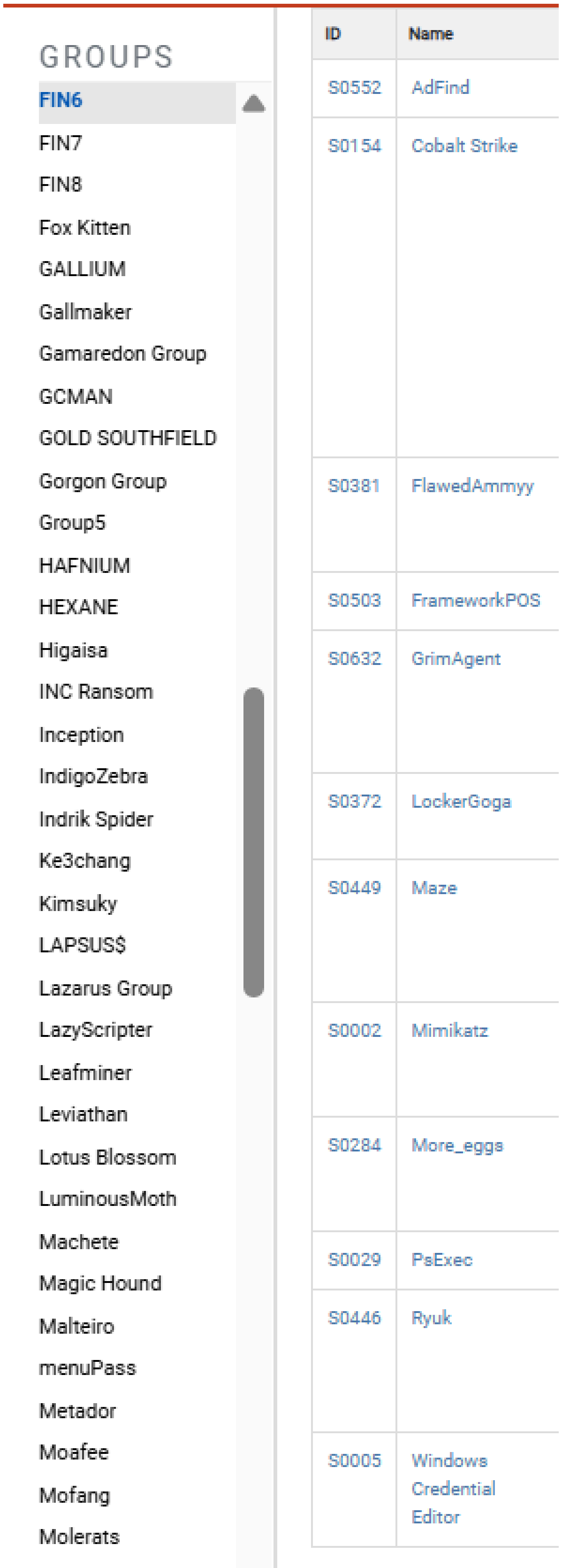
For continuous emulation exercises, it is worth noting that adversary TTPs can change over time, so staying up-to-date with the latest threat intelligence is essential. For example, FIN7 has been known to alter their TTPs over time, so it is crucial to continually gather new information and update the emulation.

****

Answer the questions below:

There's a set of 3 software used by FIN6 & FIN7. Can you identify them? Answers are in alphabetical order, separated by a comma.

Comparing FIN6 and FIN7 on the ATT&CK Framework under the software section we see for FIN6 the following software:

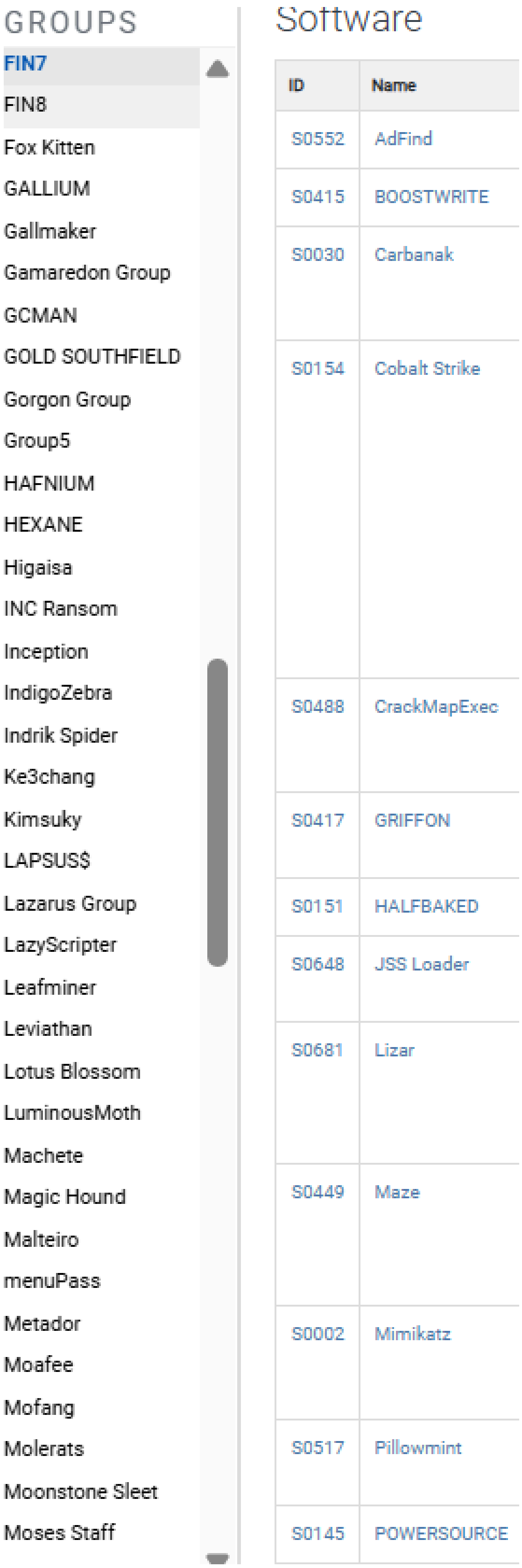


GROUPS

- **FIN6**
- FIN7
- FIN8
- Fox Kitten
- GALLIUM
- Gallmaker
- Gamaredon Group
- GCMAN
- GOLD SOUTHFIELD
- Gorgon Group
- Group5
- HAFNIUM
- HEXANE
- Higaisa
- INC Ransom
- Inception
- IndigoZebra
- Indrik Spider
- Ke3chang
- Kimsuky
- LAPSUS$
- Lazarus Group
- LazyScripter
- Leafminer
- Leviathan
- Lotus Blossom
- LuminousMoth
- Machete
- Magic Hound
- Malteiro
- menuPass
- Metador
- Moafee
- Mofang
- Molerats

ID	Name
S0552	AdFind
S0154	Cobalt Strike
S0381	FlawedAmmyy
S0503	FrameworkPOS
S0632	GrimAgent
S0372	LockerGoga
S0449	Maze
S0002	Mimikatz
S0284	More_eggs
S0029	PsExec
S0446	Ryuk
S0005	Windows Credential Editor

And for FIN7:

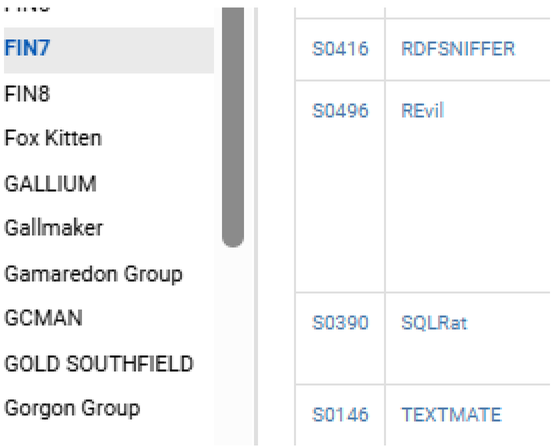


GROUPS

- FIN7
- FIN8
- Fox Kitten
- GALLIUM
- Gallmaker
- Gamaredon Group
- GCMAN
- GOLD SOUTHFIELD
- Gorgon Group
- Group5
- HAFNIUM
- HEXANE
- Higaisa
- INC Ransom
- Inception
- IndigoZebra
- Indrik Spider
- Ke3chang
- Kimsuky
- LAPSUS$
- Lazarus Group
- LazyScripter
- Leafminer
- Leviathan
- Lotus Blossom
- LuminousMoth
- Machete
- Magic Hound
- Malteiro
- menuPass
- Metador
- Moafee
- Mofang
- Molerats
- Moonstone Sleet
- Moses Staff

Software

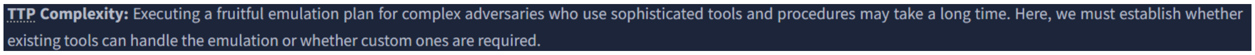
ID	Name
S0552	AdFind
S0415	BOOSTWRITE
S0030	Carbanak
S0154	Cobalt Strike
S0488	CrackMapExec
S0417	GRIFFON
S0151	HALFBAKED
S0648	JSS Loader
S0681	Lizar
S0449	Maze
S0002	Mimikatz
S0517	Pillowmint
S0145	POWERSOURCE



S0416	RDFSNIFFER
S0496	REvil
S0390	SQLRat
S0146	TEXTMATE

And comparing the two lists we see that the shared software is Adfind, Cobalt Strike, Maze, and Mimikatz, but for the answer they only want Adfind, Cobalt Strike, and Mimikatz.
Answer: AdFind, Cobalt Strike, Mimikatz

Which factor will be considered when analysing whether to use existing or custom tools during the emulation?



TTP Complexity: Executing a fruitful emulation plan for complex adversaries who use sophisticated tools and procedures may take a long time. Here, we must establish whether existing tools can handle the emulation or whether custom ones are required.

Answer: TTP Complexity

Threat Emulation Process II

3. Planning the Threat Emulation Engagement

Since threat emulation involves conducting and mimicking actual cyber attacks, significant problems may ensue if not properly planned and coordinated. The issues may include disclosure of private data, data loss and unplanned system downtime.

Planning the emulation activities through defining the rules of engagement for the exercise, including communication and approvals, is vital to avert these risks. Planning also involves determining the resources needed for the activity, such as personnel, time, and equipment.

3.1 Threat Emulation Plans

Threat Emulation Plans are a collection of resources used to organise and set a step-by-step execution of instructions for adversary behaviours based on a particular set of TTPs. As discussed in the previous tasks, we can find available emulation plans from CTID that capture adversary behaviours based on specific scenarios and step-by-step procedures to execute the emulation using tools.

A well-defined plan will contain the elements of the threat emulation process as well as the following components:

- Engagement Objectives: We have seen that the objectives are defined at the beginning of the process to understand the need for threat emulation.
- Scope: The departments, users and devices upon which emulation activities are permitted should be defined explicitly.
- Schedule: The dates and times when the activities should occur and when deliverables are due should be defined. This helps avoid conflicts between emulation activities and legitimate business operations.
- Rules of Engagement: The acceptable adversary behaviour to be emulated must be planned and discussed. This also includes mitigation actions for any high-risk TTPs used during the exercise.
- Permission to Execute: Explicit written consent to conduct the emulation activities must be provided by sufficient authority from the organisation. This helps avoid acting out independently and risking legal or criminal problems.
- Communication Plan: The emulation team and organisation stakeholders must have a plan to communicate information concerning the emulation activities. You need to define the timings, communication channels, and any collaboration efforts to be included.

4. Conducting the Emulation

This step involves carrying out the attack using the TTPs identified in the research phase. This step requires skilled professionals who can accurately replicate the tactics and techniques of the target adversary. The exercise should be conducted controlled and safely, and any issues should be addressed immediately.

We will not go through a technical implementation of TTPs in this room. However, we shall provide a decent breakdown of the process.

During the execution of the emulation, some resources would be needed to implement the TTPs. This will be your emulation lab, comprising an attack platform, an analysis platform used to gather forensic details and analyse artefacts and your test systems where the TTPs would be deployed.

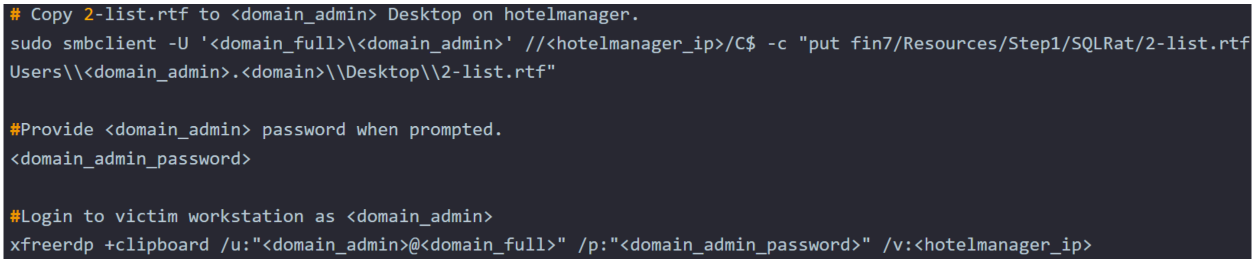
4.1. Planning the Deployment

As the emulation engineer for VASEPY, we can revisit the Research phase tackled in the previous task, where we identify the TTPs to emulate for FIN7. Let's say we wish to emulate the Initial Access TTPs. We can use ATT&CK to understand the TTPs and map them out using the Navigator. This would be combined with CTI resources, such as Mandiant's FIN7 Evolution Report and ESentire FIN7 Report, outlining how FIN7 used Windows document lures to execute their campaign.

The lab environment needed for the exercise should be effectively set up, and security teams should know their responsibilities.

4.2. Implementation of TTP

This is where the deployment of actual TTPs happens. In our case scenario, an Initial Access payload for FIN7 would be created and obfuscated using an RTF document, delivered through a spear phishing email. The lures used by attackers such as FIN7 tend to be convincing using DOCX and RTF files with malicious Windows Shortcut File (.LNK) embedded. A code snippet demonstrating this execution based on the CTID FIN7 Emulation plan would look as follows:



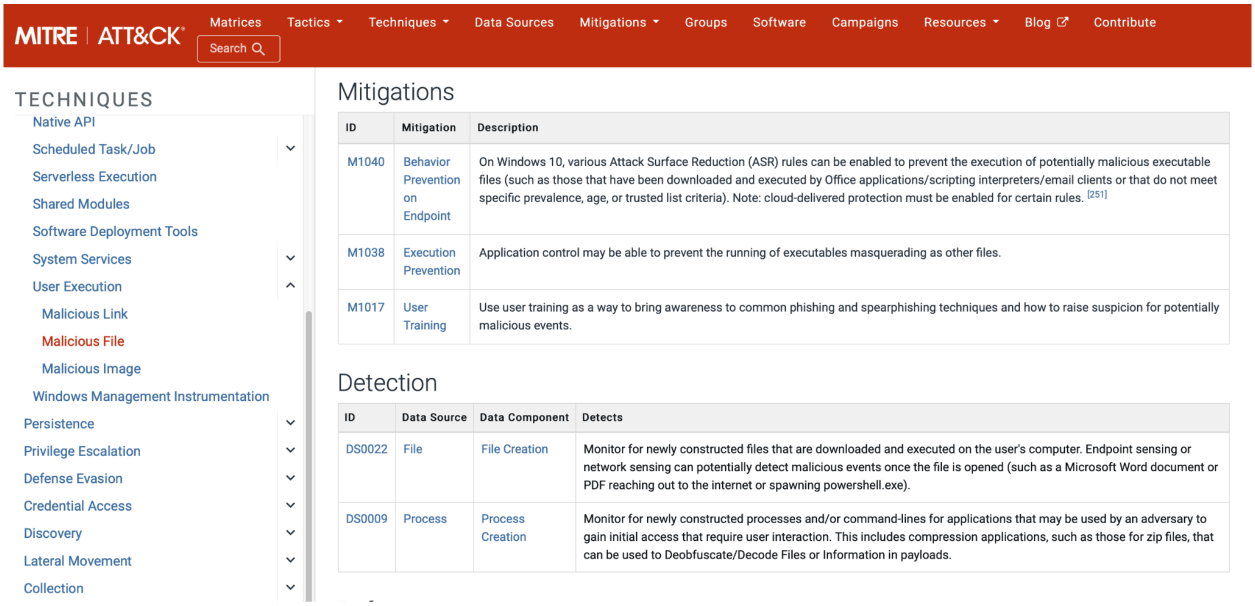
```
# Copy 2-list.rtf to <domain_admin> Desktop on hotelmanager.
sudo smbclient -U '<domain_full>\<domain_admin>' //<hotelmanager_ip>/C$ -c "put fin7/Resources/Step1/SQLRat/2-list.rtf
Users\\<domain_admin>.<domain>\\Desktop\\2-list.rtf"

#Provide <domain_admin> password when prompted.
<domain_admin_password>

#Login to victim workstation as <domain_admin>
xfreerdp +clipboard /u:"<domain_admin>@<domain_full>" /p:"<domain_admin_password>" /v:<hotelmanager_ip>
```

4.3. Detections & Mitigations

Since emulation is a cross-team and collaborative endeavour, the defence team must find ways to detect and mitigate against emulated TTPs. Depending on the organisational setup, the SOC would use standard cyber security tools to collect, correlate and analyse TTP behaviour and logs for detection. MITRE provides a list of mitigation efforts for the adversarial TTPs, and this can be provided as recommendations and implemented as part of the emulation. In our case, we can look at the Malicious File T1204.002 mitigations and detection strategies.



Mitigations

ID	Mitigation	Description
M1040	Behavior Prevention on Endpoint	On Windows 10, various Attack Surface Reduction (ASR) rules can be enabled to prevent the execution of potentially malicious executable files (such as those that have been downloaded and executed by Office applications/scripting interpreters/email clients or that do not meet specific prevalence, age, or trusted list criteria). Note: cloud-delivered protection must be enabled for certain rules. [251]
M1038	Execution Prevention	Application control may be able to prevent the running of executables masquerading as other files.
M1017	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.

Detection

ID	Data Source	Data Component	Detects
DS0022	File	File Creation	Monitor for newly constructed 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).
DS0009	Process	Process Creation	Monitor for newly constructed processes and/or command-lines 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.

To end the task, answer the questions provided based on what you have learnt. Click the View Site button at the top of the task to launch the static site and complete the activity.

****

Answer the questions below:

The emulation plan component determining which activities are to be conducted is known as the?
Answer: Scope

What is flag one obtained after completing the exercise?
Answer: THM{C4RB0N_$P1D3R_1$_F1N7}

What is flag two obtained after completing the exercise?
Answer: THM{3$P1ON4G3_F0R_R34P3R}

Threat Emulation Process III

5. Observe Results

While going through the emulation engagement, the observing team (typically Blue Team) must identify artefacts that point to the emulation activity. This will be through the analysis of logs, evaluation of event logs and tracking of networking traffic.

Additionally, like in the case of FIN7, detection rules would be vital to detect the threat. One collection of rules useful for this would be the YARA rules. Have a look at the rules to detect the pillowMint.exe malware.

The output of these results would help to understand if the TTP was successful at its mission, blocked or detected by the security measures available.

6. Document & Report Findings

Once results have been obtained, the teams must document and report the findings. Documentation provides empirical evidence to demonstrate the cyber security effectiveness of the process.

Reporting should cover the exercise procedures, as outlined in the emulation plan and what was executed, the impact faced and recommendations that would be offered to avert the threat.

****

Answer the questions below:

Click the View Site button at the top of the task to launch the static site. What is flag three obtained after completing the exercise?

Answer: THM{D3F3NC3_1N_3MUL4T10N}

What is flag four obtained after completing the exercise?

Answer: THM{S3CUR3_4LL_W3B_4553T5}

Conclusion

Fantastic work going through the Introduction to Threat Emulation room.

Throughout the room, we have seen how adversary behaviours can be emulated by following a detailed process and formulating a plan. As the room covered these concepts in theory, you should expect to meet them more practically and technically in future rooms within the module.

Threat Emulation is commonly viewed as a Red Team concept, yet it is all-rounded and involves every team in the security domain. Knowing how attackers do it simplifies understanding how to defend it, more like learning the moves in a video game.