MTR170202
MITRE TECHNICAL REPORT

MITRE

Finding Cyber Threats with ATT&CK™-Based Analytics

Dept. No.: J83L
Project No.: 0716MM09-AA

The views, opinions and/or findings contained in this report are those of The MITRE Corporation and should not be construed as an official government position, policy, or decision, unless designated by other documentation.

Approved for Public Release; Distribution Unlimited. Case Number 16-3713.

This technical data deliverable was developed using contract funds under Basic Contract No. W15P7T-13-C-A802.

Annapolis Junction, MD

Blake E. Strom
Joseph A. Battaglia
Michael S. Kemmerer
Douglas P. Miller
Craig Wampler
Sean M. Whitley
Ross D. Wolf

June 2017

Abstract

Post-compromise intrusion detection of cyber adversaries is an important capability for network defenders as adversaries continue to evolve methods for compromising systems and evading common defenses. This paper presents a methodology for using the MITRE ATT&CK framework, a behavioral-based threat model, to identify relevant defensive sensors and build, test, and refine behavioral-based analytic detection capabilities using adversary emulation. This methodology can be applied to enhance enterprise network security through defensive gap analysis, endpoint security product evaluations, building and tuning behavioral analytics for a particular environment, and performing validation of defenses against a common threat model using a red team emulating known adversary behavior.

This page intentionally left blank.

Acknowledgments

We would like to thank the MITRE cyber security research program and leadership for supporting our research over the years, providing valuable oversight, and enabling the team to break new ground on enterprise detection. Especially Todd Wittbold for the original vision and early leadership that enabled the team to focus on the research. William Kupersanin, Adam Pennington, Xeno Kovah, Frank Duff, Eric Sheesley, Brad Crawford, and all the others who shaped FMX's research over the years by articulating the threat, and defining how to detect it more effectively. Their leadership enabled the vision to become a reality. MITRE Annapolis Junction site management, MITRE InfoSec, and MITRE Center for Information and Technology for their understanding and patience as we challenged established policies in the course of our research. The living lab environment, and continual red teaming of the environment would not have been possible without their support. And finally, we would like to acknowledge and thank the National Security Agency Adaptive Cyber Defense Systems research team. They were strong research advocates for incorporating cyber resilience concepts, participated in experiments conducted within the research environment, and engaged in multiple brainstorming sessions on detection, response, and threat modeling.

Table of Contents

Table of Figures

1 Introduction

Defending an enterprise network against an advanced persistent threat (APT) remains an increasingly difficult challenge that requires, among other things, advanced technologies and approaches for thwarting adversary goals. In current enterprise networks, it is unlikely that organizations have the ability or the resources to detect and defend against every method an adversary might use to gain access to their networks and systems. Even if an organization's enterprise patching and software compliance program is perfect, an adversary may use a zero-day exploit, or a social engineering attack to gain a foothold in a potential victim's network. Once inside, adversaries hide in the noise and complexity of their target's environment, often using legitimate mechanisms and camouflaging their activities in normal network traffic to achieve their objectives. Depending on the security sophistication of the target network, an adversary is often presented with ample time to do their work. For instance, FireEye's M-Trends states that the median time for an enterprise to discover they've been compromised was 146 days in 2015. [1]

To help address these challenges, in 2010 MITRE began researching data sources and analytic processes for detecting APTs more quickly under an "assume breach" mentality through the use of endpoint telemetry data. Specifically, MITRE's work centered on post-compromise detection, focusing on adversary behavior after they have gained access to a system within a network. One driver for MITRE's approach was that public information on cyber intrusions suggests that adversaries tend to exhibit consistent patterns of behavior while interacting with endpoint or victim systems. [2] The goal of MITRE's research was to show that automated measuring of endpoint data or telemetry could be used to detect post-compromise operations in a useful way that distinguished such behavior from the typical noise generated through normal system use. The results of this research indicated that using analytics based on a combination of host and network behaviors provides a useful way to detect post-compromise adversary behavior.
As part of its research effort, starting in 2013 MITRE also developed a process for modeling an adversary's post-compromise behavior at a granular level. This model is named ATT&CK™ (Adversarial Tactics, Techniques, and Common Knowledge) [3], and it serves as both the adversary emulation playbook and as a method for discovering analytic coverage and defense gaps inside a target network. ATT&CK was released in 2015 and is available at https://attack.mitre.org.

Additionally, MITRE researchers created a method for describing behavioral intrusion detection analytics and a suite of analytics aligned to the ATT&CK model, both of which have been made publicly available to the information security community through the MITRE Cyber Analytics Repository. [4]

Both the creation of behavioral detection analytics and the efficacy of this approach in detecting threat behavior were validated through a series of "cyber games" that pitted a Red Team performing adversary emulation using APT behavior (as described in the ATT&CK model) against a Blue Team using analytics to detect the Red Team's intrusion and the scope of its actions throughout the targeted network. The games were performed on an approximately two-hundred-fifty-node production enclave on MITRE's live corporate network. A live network was used to ensure realistic network and system noise typically generated through normal use of a corporate network environment so that MITRE's analytics could be tuned appropriately to detect malicious behavior.

The purpose of this paper is to: 1) educate the reader about how to apply MITRE's ATT&CK model for better detection of an APT operating within a given system or network; 2) describe the processes and methods for developing this model and its associated methods and analytics; and 3) to share some practical experiences and anecdotes to give the reader ideas on how MITRE's research might help a specific organization protect its networks and systems. To these ends, this paper begins by orienting the reader with a hypothetical APT campaign that is described using tactics, techniques, and procedures (TTPs) from the ATT&CK model. MITRE's high level research process and the behavioral detection paradigm it developed are then described in Section 2. The ATT&CK model itself is described in Section 3. Section 4 explains the process of applying ATT&CK for developing behavioral intrusion detection analytics. Section 5 details some of MITRE's experiences in applying our methodology. Section 6 provides a summary of MITRE's work.

1.1 Frame of Reference

To assist with understanding the applicability of the research described in this paper, it is helpful to first consider a hypothetical APT campaign based on an adversary's post-compromise tactics, techniques, and procedures (TTPs). The hypothetical adversary in this example uses publically reported TTPs, each of which are described in the ATT&CK model itself and referenced in footnotes that should not be confused with the endnotes used throughout this paper and signified with a bracketed number nomenclature such as [#]. The ATT&CK model is explained further in Section 3, but referenced here to show how it describes post-compromise adversary behavior. The hypothetical campaign begins with an adversary sending a spear phishing e-mail designed around a current event of interest. [5] The payload is a *.zip* file that contains a decoy portable document format (PDF) file and a malicious executable that uses the PDF to disguise itself on systems with Acrobat Reader installed. [6]

When run, the executable downloads a second stage[1] Remote Access Tool (RAT) payload, giving a remote operator access to the victim computer as well as an initial access point into the network. The adversary then generates new domain names for Command and Control (C2) purposes and sends these domains to the RAT on the compromised network through periodically changing twitter handles[2]. [7] The domains and Internet Protocol (IP) addresses used for C2 are short-lived and are routinely changed by the adversary after only a few days. The adversary persists or maintains a seemingly legitimate presence on the victim computer by installing a Windows® service[3] with a name that is easily assumed by the computer's owner to be that of a

[1] ATT&CK: T1104 – Command and Control/Multi-Stage Channels

[2] ATT&CK: T1102 – Command and Control/Web Service

[3] ATT&CK: T1050 – Persistence/New Service

legitimate system service[4]. Before deploying this malware, the adversary has previously tested it on a wide array of anti-virus (AV) products to ensure it does not match any existing or known malware signatures[5].

To interact with its host victim, the remote operator uses the RAT to start a Windows command prompt[6]. The adversary then uses tools already available on the compromised computer to learn more about the victim's system and the surrounding network with the goal of increasing his level of access on additional systems and to move the adversary closer to achieving his objectives. More specifically, the adversary uses built-in Windows tools or legitimate third-party administration tools to discover internal host and network resources, and to discover information such as accounts[7], permissions groups[8], processes[9], services[10], network configuration[11], and nearby network resources[12][13]. The remote operator may then bulk-capture cached authentication credentials[14] using Invoke-Mimikatz[15], a Windows PowerShell®[16] wrapper for the popular Mimikatz [8] tool that allows the tool to run without having to write an executable to the computer's hard drive. [9] Lateral movement from one computer to another may then occur after enough information is gathered, which may typically happen using mapped Windows admin shares[17] and remote Windows (Server Message Block [SMB]) file copies[18] coupled with remotely scheduled tasks[19]. With each increased access, the adversary finds documents of interest within

---

4 ATT&CK: T1036 – Defense Evasion/Masquerading

5 ATT&CK: T1066 – Indicator Removal from Tools

6 ATT&CK: T1059 – Execution/Command-Line Interface

7 ATT&CK: T1087 – Discovery/Account Discovery

8 ATT&CK: T1069 – Discovery/Permission Groups Discovery

9 ATT&CK: T1057 – Discovery/Process Discovery

10 ATT&CK: T1007 – Discovery/System Service Discovery

11 ATT&CK: T1016 – Discovery/Local Network Configuration Discovery

12 ATT&CK: T1049 – Discovery/Local Network Connections Discovery

13 ATT&CK: T1018 – Discovery/Remote System Discovery

14 ATT&CK: T1003 – Credential Access/Credential Dumping

15 ATT&CK: T1064 – Defense Evasion/Scripting

16 ATT&CK: T1086 – Execution/PowerShell

17 ATT&CK: T1077 – Lateral Movement/Windows Admin Shares

18 ATT&CK: T1105 – Lateral Movement/Remote File Copy

19 ATT&CK: T1053 – Execution/Scheduled Task

the network[20][21][22][23]. The adversary then stages these documents to a central location[24], compresses[25] and encrypts[26] the compressed file through a remote command-line shell using utilities such as RAR, and finally, exfiltrates or exports them from the victim's host computer, via a well-formed Hypertext Transfer Protocol (HTTP)[27] session inside Secure Sockets Layer/Transport Layer Security (SSL/TLS)[28]. Thus, the adversary has committed cyber theft of critical documents of interest and value from the victim computer and network to his own remote computer where he can analyze and use the exfiltrated information at his convenience.

1.2 Shortcomings of Contemporary Approaches for Detection

Contemporary network security approaches have trouble detecting an APT similar to the one in the aforementioned hypothetical example. Most AV applications may not reliably detect custom tools because these tools have been tested by adversaries against various products before use and may even contain obfuscation techniques used to evade other types of malware detection. [10] Malicious remote operators also have the ability to use legitimate functionality on systems they compromise, essentially "living off the land" [11] to avoid detection since many defensive tools do not collect sufficient data to detect this kind of malicious use of otherwise appropriate system behavior.

Other current approaches to network security, such as threat intelligence feed subscription or sharing may not help with detecting the use of adversary infrastructure, because adversary indicators can change too rapidly. Typical network traffic inspection will not be useful either, since the APT's traffic, such as that described in the previous hypothetical example, is encrypted by valid SSL. SSL interception may be useful, but without a priori knowledge of components with known signatures (i.e., a unique pattern that exists in network traffic that can be used to detect a specific kind of known malicious activity) in the correctly formed HTTP traffic, finding this level of malicious activity and distinguishing it from benign network behavior is simply too difficult. The shortcomings of these contemporary approaches led MITRE to develop a different method for detecting APTs over the course of the multi-year research effort by focusing on an adversary's post-compromise behavior using the ATT&CK threat model.

---

[20] ATT&CK: T1083 – Discovery/File and Directory Discovery

[21] ATT&CK: T1005 – Collection/Data from Local System

[22] ATT&CK: T1025 – Collection/Data from Removable Media

[23] ATT&CK: T1039 – Collection/Data from Network Shared Drive

[24] ATT&CK: T1074 – Collection/Data Staged

[25] ATT&CK: T1002 – Exfiltration/Data Compressed

[26] ATT&CK: T1022 – Exfiltration/Data Encrypted

[27] ATT&CK: T1071 – Command and Control/Standard Application Layer Protocol

[28] ATT&CK: T1032 – Command and Control/Standard Cryptographic Protocol

2 Threat-Based Security Approach

MITRE's threat-based approach to network compromise detection uses a behavioral methodology and is guided by five principles that were developed over the course of its research. These principles describe critical tenets of an effective threat-based approach to network security. They are summarized here (Figure 1) and explained in further detail throughout the rest of this section.

Principle 1: Include Post-Compromise Detection – Over time, previously effective perimeter and preventative defenses may fail to keep persistent threats out of a network. Post-compromise detection capabilities are necessary for when a threat bypasses established defenses or uses new means to enter a network.

Principle 2: Focus on Behavior – Signatures and indicators are useful with a priori knowledge of adversary infrastructure and tool artifacts, but defensive tools that rely on known signatures may become unreliable when signatures become stale in relation to a changing threat. Sophisticated defenses should also incorporate detecting and learning from post-compromise adversary behavior.

Figure 1. Five Principles of Threat-Based Security

Principle 3: Use a Threat-based Model – An accurate and well-scoped threat model is necessary to ensure that detection activities are effective against realistic and relevant adversary behaviors.

Principle 4: Iterate by Design – The adversarial tool and technique landscape is constantly evolving. A successful approach to security requires constant, iterative evolution and refinement of security models, techniques, and tools to account for changing adversary behavior and to understand how networks are compromised by an APT.

Principle 5: Develop and Test in a Realistic Environment – Analytic development and refinement should be performed in a production network environment, or one that matches realistic network conditions as closely as possible. Behavior generated by real network users should be present to account for the expected level of sensor noise generated by standard network use. Whenever possible, detection capabilities should be tested by emulation of adversary behavior within that environment.

2.1 Principle 1: Include Post-Compromise Detection

There are likely ways to penetrate even the most well-defended network perimeter. For instance, currently there is no entirely effective way to prevent every zero-day vulnerabilities from being exploited [12, 13], no method to instantaneously patch software, no way to prevent humans from exposing passwords, and no cost-effective means of securing supply chains. The sheer size and complexity of the attack surface means that APTs will likely continue to find ways to circumvent common security practices and be successful at penetrating enterprise networks to accomplish their objectives. As such, any effective network security should take into account post-compromise adversary behavior in order to minimize damage caused by an adversary who successfully penetrates your initial defenses.

2.2 Principle 2: Focus on Behavior

Many contemporary defenses focus on signatures or indicators of compromise (IOCs). IOCs are artifacts or combinations of artifacts that may be observed in the presence of known malicious software or activity. [14] In many cases these are brittle and easy for adversaries to bypass by modifying malware or infrastructure. Indicators like file hashes, IP addresses, and domain names have become the focal point for many network defenders, yet each of these are trivial for an adversary to change in order to avoid detection. In addition, the defending organization needs to have access to relevant and up-to-date indicators through a threat indicator sharing program or commercial data feed, all of which may still not ensure that defenders are able to keep pace with adversary changes.

An intrusion detection program incorporating behavioral detection analytics is more resilient to attempts by adversaries to avoid signature-based detection through indicator modification. Behavioral detection approaches help identify the common behaviors that are highly likely to be performed across many adversary groups during an intrusion, and are independent of specific changes to indicators that adversaries make. This is the premise that drove the development of ATT&CK-based analytics.

2.3 Principle 3: Use a Threat-based Model

The use of a threat model has long been the foundation of a robust security process. The compilation of adversary actions and behaviors in a threat model allows defenders to adequately plan and evaluate their defenses. In keeping with Principles 1 and 2, the threat model encapsulated in ATT&CK describes post-compromise adversary behaviors within enterprise networks. Figure 2 shows how the ATT&CK model takes the three post-compromise or post-exploit stages of the cyber-attack lifecycle [15][29] and expands them into 10 distinct tactics that are employed by APTs.

---

[29] First articulated by Lockheed Martin as the Cyber Kill Chain® [29]

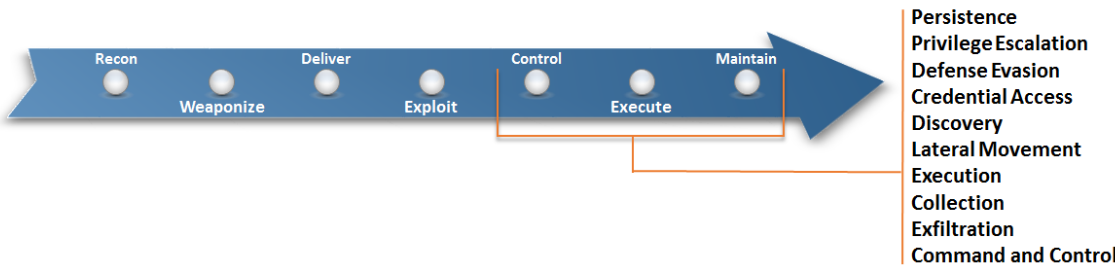

Figure 2. The ATT&CK Tactic Categories

2.4 Principle 4: Iterate by Design

An iterative process was critical to creating an effective behavioral analytics and detection system aligned with the ATT&CK model. An important part of this process involved testing behavioral analytics every few months against a Red Team that emulated known APT behaviors documented in ATT&CK-based cyber games. This approach is covered in more detail in Section 4. One of the benefits of employing an iterative process is the quick feedback loop provided to analytic developers. By only targeting a few analytics per iteration and then validating their performance against a simulated threat within a real network, analysts can quickly see which ideas are useful, which ones need to be developed further, and which ones should be discarded. This approach provides numerous opportunities for a developer to test hypotheses about how best to detect given behaviors on a real network.

Another benefit of iteration is the ability for network defenses to adapt to a changing threat landscape. As previously stated, a behavioral detection approach provides an effective method for dealing with APTs that rapidly change their behavior to overcome defensive barriers. As in the hypothetical APT example described in Section 1.1of this paper, adversaries use legitimate web services and strong encryption to bypass network protocol signature detection and blacklists, but they will still need to interact with an endpoint system in a way that is consistent with how that system operates. In this paper's hypothetical APT example, the remote operator uses a new Windows service to install persistent malware. However, regardless of what malware the intruder tries to run, the presence of something like a new Windows service is one of the adversary behaviors that the ATT&CK model describes and may combine with other events to recognize distinct patterns of adversary behaviors.

Behavioral detection makes it harder for adversaries to avoid detection. However, adversaries will still continue to change their behavior over time and the quick feedback that an iterative approach provides regarding defensive capabilities will remain essential for effectively detecting the latest threat behaviors.

2.5 Principle 5: Develop and Test in a Realistic Environment

It is important for the iterative development of analytics and detection capabilities (2.4) to be performed in a live production environment, or one that is as realistic as possible. To do this, MITRE's Red Team emulated APT behavior based on the latest understanding of current threats within ATT&CK. The actual development and testing of analytics was performed on a 250-node enclave on the MITRE corporate network (See Section 5 for more details.), with real users performing their normal daily work to ensure realistic background system noise. By using a Red Team that closely mimicked real APT behaviors on a live network, the analytics were tested through the cyber games process in an environment that helped ensure their ability to detect desired adversary behaviors effectively.

If the analytics had been developed and tested in a laboratory environment without real users performing real work using real mission applications, the adversary behaviors that would have been detected may have ended up being behaviors routinely performed by users or system administrators, thus making them useless for detecting APT behavior on a production network.

3 ATT&CK

ATT&CK is a model and framework for describing the actions an adversary takes while operating within an enterprise network. The model can be used to better characterize post-compromise adversary behavior with the goal of distilling common behaviors across known intrusion activity into individual or combinations of actions that an adversary may take to achieve their goals. The TTPs described in ATT&CK were selected based on observed APT intrusions from public reporting, and are included in the model at a level of abstraction necessary for effectively prioritizing defensive investments and comparing endpoint intrusion detection capabilities.

3.1 Post-Compromise Threat-Based Modeling

ATT&CK addresses a gap in the community's understanding of specific post-compromise intrusion details. Public reports on adversary groups and intrusions mention high-level details about adversary behavior, but often lack critical information that could be used to defend against intruder techniques. For example, reports that mention lateral movement in general terms without providing details about how an adversary performs these movements do not help an organization defend against that specific tactic. For instance, adversarial use of the Windows remote desktop feature is a very different way of moving laterally than copying a remote access tool to a shared directory and issuing a remote service start command to execute it.

MITRE created the ATT&CK model to address the need for additional details while remaining grounded in observed and plausible adversary behavior. Information sources that informed the model included public threat intelligence reporting, penetration testing, red teaming knowledge, and security research. The techniques in ATT&CK largely focus on Windows enterprise systems due to the amount of publicly available intrusion reports containing details and adversary tools that work against Windows—suggesting that enterprise network intrusions tend to focus on these types of systems. Even so, because adversaries' higher level objectives tend to remain the same regardless of the platform, the ATT&CK model can be expanded into non-Windows-based enterprise systems, although doing so would require additional information about adversary techniques on these platforms to maintain the desired level of empirical realism.

ATT&CK is broken down into high-level adversary tactic categories and individual techniques that adversaries may use within each of the tactic categories. Tactics describe *why* an adversary performs an action, and techniques describe *how* they do it. Techniques are described in the ATT&CK model from both the offensive and defensive points of view so they are a useful reference and pivot between both disciplines. ATT&CK techniques also contain references to known examples of the technique having been used and links to public threat reporting on adversary groups known to use that technique or related tools to provide examples of adversary behavior *in the wild*.

3.2 Tactics

Tactics represent the highest level of abstraction within the ATT&CK model. They are the tactical goals an adversary has during an operation. The ATT&CK tactic categories are listed here:

- **Persistence** – Any access, action, or configuration change to a system that gives an adversary a persistent presence on that system. Adversaries will often need to maintain access to systems through interruptions such as system restarts, loss of credentials, or other failures.
- **Privilege Escalation** – The result of techniques that cause an adversary to obtain a higher level of permissions on a system or network. Certain tools or actions require a higher level of privilege to work and are likely necessary at many points throughout a remote operation.
- **Defense Evasion** – Techniques an adversary may use for the purpose of evading detection or avoiding other defenses.
- **Credential Access** – Techniques resulting in the access of, or control over, system, domain, or service credentials that are used within an enterprise environment.
- **Discovery** – Techniques that allow an adversary to gain knowledge about a system and its internal network.
- **Lateral Movement** – Techniques that enable an adversary to access and control remote systems on a network. Often the next step for lateral movement is remote execution of tools introduced by an adversary.
- **Execution** – Techniques that result in execution of adversary-controlled code on a local or remote system.
- **Collection** – Techniques used to identify and gather information, such as sensitive files, from a target network prior to exfiltration.
- **Exfiltration** – Techniques and attributes that result or aid in an adversary removing files and information from a target network. This category also covers locations on a system or network where an adversary may look for information to exfiltrate.
- **Command and Control** – Techniques and attributes of how adversaries communicate with systems under their control within a target network. Examples include using legitimate protocols such as HTTP to carry C2 information.

At this level, ATT&CK closely resembles other threat models that describe the adversary life cycle. The difference between ATT&CK and these other models resides in its scope. ATT&CK tactic categories are applicable from one individual endpoint system to the next as an adversary moves across a network, whereas a model such as the cyber attack lifecycle (see Figure 2) takes into account a much broader scope of the full operation according to an adversary's life cycle. The tactics in the ATT&CK model also represent an adversarial process of navigating an enterprise network.

Figure 3 provides an overview of the tactic categories and associated techniques described in the ATT&CK model. The figure visually aligns individual techniques under each tactic category in which they can be applied.

Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Execution	Collection	Exfiltration	Command and Control
DLL Search Order Hijacking	DLL Search Order Hijacking	DLL Search Order Hijacking	Brute Force	Account Discovery	Windows Remote Management	Windows Remote Management	Automated Collection	Automated Exfiltration	Commonly Used Port
Legitimate Credentials	Legitimate Credentials	Legitimate Credentials	Credential Dumping	Application Window Discovery	Third-party Software	Third-party Software	Clipboard Data	Data Compressed	Communication Through Removable Media
Accessibility Features	Accessibility Features	Binary Padding	Credential Manipulation	File and Directory Discovery	Application Deployment Software	Command-Line	Data Staged	Data Encrypted	Custom Command and Control Protocol
AppInit DLLs	AppInit DLLs	Code Signing	Credentials in Files	Local Network Configuration Discovery	Exploitation of Vulnerability	Execution through API	Data from Local System	Data Transfer Size Limits	Custom Cryptographic Protocol
Local Port Monitor	Local Port Monitor	Component Firmware	Input Capture	Local Network Connections Discovery	Logon Scripts	Graphical User Interface	Data from Network Shared Drive	Exfiltration Over Alternative Protocol	Data Obfuscation
New Service	New Service	DLL Side-Loading	Network Sniffing	Network Service Scanning	Pass the Hash	InstallUtil	Data from Removable Media	Exfiltration Over Command and Control Channel	Fallback Channels
Path Interception	Path Interception	Disabling Security Tools	Two-Factor Authentication Interception	Peripheral Device Discovery	Pass the Ticket	PowerShell	Email Collection	Exfiltration Over Other Network Medium	Multi-Stage Channels
Scheduled Task	Scheduled Task	File Deletion	Exploitation of Vulnerability	Permission Groups Discovery	Remote Desktop Protocol	Process Hollowing	Input Capture	Exfiltration Over Physical Medium	Multiband Communication
Service File Permissions Weakness	Service File Permissions Weakness	File System Logical Offsets		Process Discovery	Remote File Copy	Regsvcs / Regasm	Screen Capture	Scheduled Transfer	Multilayer Encryption
Service Registry Permissions Weakness	Service Registry Permissions Weakness	Indicator Blocking		Query Registry	Remote Services	Regsvr32			Peer Connections
Web Shell	Web Shell	Exploitation of Vulnerability		Remote System Discovery	Replication Through Removable Media	Rundll32			Remote File Copy
Basic Input/Output System	Exploitation of Vulnerability	Bypass User Account Control		Security Software Discovery	Shared Webroot	Scheduled Task			Standard Application Layer Protocol
Bootkit	Bypass User Account Control	DLL Injection		System Information Discovery	Taint Shared Content	Scripting			Standard Cryptographic Protocol
Change Default File Association	DLL Injection	Indicator Removal from Tools		System Owner/User Discovery	Windows Admin Shares	Service Execution			Standard Non-Application Layer Protocol
Component Firmware		Indicator Removal on Host		System Service Discovery		Windows Management Instrumentation			Uncommonly Used Port
Hypervisor		InstallUtil							Web Service
Logon Scripts		Masquerading							
Modify Existing Service		Modify Registry							
Redundant Access		NTFS Extended Attributes							
Registry Run Keys / Start Folder		Obfuscated Files or Information							
Security Support Provider		Process Hollowing							
Shortcut Modification		Redundant Access							
Windows Management Instrumentation Event Subscription		Regsvcs / Regasm							
Winlogon Helper DLL		Regsvr32							
		Rootkit							
		Rundll32							
		Scripting							
		Software Packing							
		Timestomp							

Figure 3. The MITRE ATT&CK Matrix™ [16]

3.3 Techniques

The techniques in the ATT&CK model describe the actions adversaries take to achieve their tactical objectives. Within each tactic category there are a finite number of actions that will accomplish that tactic's goal. Throughout the course of their post-compromise operations, an adversary constantly makes decisions about which technique to use based on knowledge, information obtained about the target environment, information needed for future actions, and capabilities currently available. Techniques describe actions in a way that is independent of specific adversary malware and tools. The benefit from this approach is that it covers behavior exhibited by an adversary through remote access tools, scripts, or interaction at a command-line interface without tying defenses to specific adversary malware and tools that are likely to change over time.

An important distinction between a technique in ATT&CK and an IOC is that many of the ATT&CK techniques are legitimate system functions that can be used for malicious purposes, whereas an IOC deployed as an intrusion detection mechanism is typically an indication of an action known to be caused by or under the influence of an adversary. For example, a scheduled task using the Microsoft Windows *schtasks.exe* utility[30] is a technique that can be used for persistence or for executing a binary file remotely as part of lateral movement. The appearance or invocation of *schtasks.exe* on a system is not inherently a malicious act because it is a legitimate administrative feature of the operating system. Adversaries are aware of this and other legitimate system functions and how to use those features to their advantage.

Individual occurrences of techniques are not performed in a vacuum, but normally follow sets of behavior or sequences of events. In addition to the steps called out in the aforementioned example, the adversary may have gathered credentials used to execute *schtasks.exe* from a credential dumper on the local system, or he may have used a keylogger. Likewise, the remote file executed by *schtasks.exe* may also have its own behavioral indicators that result after execution, such as an unknown process using the network to call back to a C2 server or another credential dumper. Being able to link these kinds of events together is an important part of defense, and is a feature of the ATT&CK model.

The techniques described in the ATT&CK model should, therefore, not be viewed in isolation or as singular APT actions but as parts of an adversary's playbook that maps malicious behavior that defensive systems can be built to detect. Data from each step in an adversary's sequence of events can be used to build a more accurate case for whether or not a set of activities constitutes malicious or benign behavior.

[30] ATT&CK: S0111 – Scheduled Tasks

ATT&CK does not claim to enumerate all possible techniques in a given tactic category, but it is based on a community of knowledge about actions that adversaries have used for a particular purpose and about how those actions relate to one another to form identifiable sequences of behavior. To further expand on the *schtasks.exe* example, when using an occurrence of *schtasks.exe* as an indicator of suspicious activity, the ATT&CK modeling approach emphasizes the importance of understanding the context under which the *schtasks.exe* was used, along with the resulting effects as described in the following example. This contextual way of viewing indicators provides the best way to gather effective information about the use of adversary techniques and how to relate them to other data points to form useful analytics.

Example ATT&CK Technique

Behavior: Remote execution via scheduled tasks using *schtasks.exe.*
Requirements:

1. Credentials or existing domain permissions providing SMB (Server Message Block – the Windows file sharing mechanism) access to the remote system.
2. Ability to move a file to the remote system for execution by the scheduled task.
3. Permission to run *schtasks.exe* on the local system. Any user can run *schtasks.exe* by default.
4. Administrative access to the remote system to schedule the task over Remote Procedure Calls (RPCs).

Cause:

1. Invocation of *schtasks.exe* at a command-line interface with arguments to execute a file on a remote system.

Effects:

1. The *schtasks.exe* process starts on the local system.
2. An RPC connection is established from the local system to the destination system.
3. An entry for the task is made under the remote system's "%SystemRoot%\Tasks\" directory.
4. The file on the remote system is executed at a specified time as a child process of *taskeng.exe*.
5. Subsequent system changes are caused by execution of the binary or script. For example, if the program is a remote access tool, the resulting process may attempt to open a network connection.

3.4 Operational Use Cases

The ATT&CK model has shown the potential for multiple applications across the offensive and defensive spectrums. For instance, in MITRE's cyber game exercises (described in Section 5.1), ATT&CK served as a model for adversary emulation. We introduced a White Team which was used for developing threat scenarios for testing defenses. The White Team constructed scenarios that emulated APT use cases (described in Section 4.4) that were based on behaviors and actions described in the ATT&CK model to test specific sensors and analytics. The model provided a common language among Red and Blue Team members to discuss actions within the network environment and how these actions were detected or not detected during a simulated adversary operation. ATT&CK's robust threat model of post-compromise behavior also made it possible to perform a defensive gap analysis (Section 4.2.1) and measure the effectiveness of existing and new defensive tools and services. We assessed these tools using ATT&CK-based operational use cases to determine how well tools detected or prevented a known set of adversary techniques that were chosen and prioritized based on their frequency of known adversary usage.

4 ATT&CK-Based Analytics Development Method

MITRE used the ATT&CK-based analytics development method to create, evaluate, and revise analytics with the intent of more accurately detecting cyber adversary behavior. Since the cyber games began in 2012, this process has been refined using experience investigating adversary behaviors, building sensors to acquire data, and analyzing data to detect adversary behavior. In describing this development method, the terms White Team, Red Team, and Blue Team performed the following roles:

- **White Team** – Developed threat scenarios for testing defenses. Worked across Red and Blue Teams to address issues that arose during the cyber game and ensured the testing objectives were met. This team interfaced with network administrators to ensure network equities were maintained.
- **Red Team** – Played the adversary for this cyber game. Executed the planned scenario with a focus on adversary behavior emulation and interfaced with White Team as needed. Any system or network vulnerabilities that were discovered were reported to the White Team.
- **Blue Team** – Acted as network defenders that used analytics to detect Red Team activity. They were also thought of as a *hunting team*.

The ATT&CK-based analytics development method contains seven steps that are shown in Figure 3 and described here as they were applied to the MITRE cyber games:

1. ***Identify Behaviors*** – Identify and prioritize adversary behaviors from the threat model to detect.
2. ***Acquire Data*** – Identify the data that is necessary to detect a desired adversary behavior. If the capability to acquire the data does not exist, a sensor must be created to collect this data.
3. ***Develop Analytics*** – Create analytics from collected data to detect identified behaviors. It is also important to ensure that analytics do not have an unacceptable false positive rate on benign environmental events.
4. ***Develop an Adversary Emulation Scenario*** – The White Team develops an adversary emulation scenario, based on ATT&CK, that includes behaviors identified in Step 1

(Identify Behaviors). The scenario includes specific techniques that should be used by the Red Team.

5. ***Emulate Threat*** – The Red Team attempts to achieve the objectives outlined by the White Team by exercising behaviors and techniques described in the ATT&CK model.
6. ***Investigate Attack*** – The Blue Team attempts to recreate the timeline of Red Team activity using analytics and data developed in Step 3 (Develop Analytics).
7. ***Evaluate Performance*** – White, Red, and Blue Teams review the engagement to evaluate to what extent the Blue Team was able to use the analytics and sensor data to successfully detect the simulated APT behaviors. After this evaluation, the cycle repeats and returns to Step 1.

Asynchronous Blue/Red Assessments

Figure 4. ATT&CK-Based Analytics Development Method

It is important to note that the Blue and Red Teams of the cyber game exercises were held asynchronously from each other for the duration of MITRE's research. This typically meant performing Blue Team evaluations several weeks after Red Team events. There were several reasons for taking this approach over the more traditional synchronous method. The first was that the research was focused on detecting the adversary, not remediating or attempting to hinder it. In fact, detecting Red Team activity and stopping their behavior in real time would have prevented using the engagement to ascertain multiple ways of detecting the emulated threat. Secondly, asynchronous exercises more accurately emulate the all-too-common real-world situation of defenders not being notified of adversary activity until after the fact. This is often the more challenging case, as defenders usually have little indication of the time ranges on which to focus their search.

Although MITRE chose not to use a synchronous methodology, exercises held synchronously can aid in developing and refining other necessary skills of defenders, like being able to react to an adversary in real time. Correctly applied, these skills can potentially prevent extensive damage to the enterprise. Synchronous engagements can also drive attackers and defenders to push the limits of their respective talents as they try to outdo each other. Ultimately, it is up to the organizers of the exercise to determine which method best suits their needs.

4.1 Step 1: Identify Behaviors

The analytic development process starts by identifying the adversary behaviors to detect. Several factors need to be considered when deciding how to prioritize adversary behaviors:

- *Which behaviors are most common?*

  Prioritizing TTPs that are most commonly used by adversaries of interest will have the broadest impact on an organization's security posture by addressing threat techniques that are most prevalent and therefore most likely to be encountered. A robust threat intelligence capability can inform an organization on the ATT&CK tactics and techniques on which to focus.

- *Which behaviors have the most adverse impact?*

  Organizations must consider which adversary TTPs have the greatest potential adverse impact to the organization. These impacts may take the form of physical destruction, loss of information, system compromise, or other negative consequences.

- *For which behaviors is data readily available?*

  Behaviors for which requisite data is already available will make it easier to create related analytics than those that require new sensors or data sources to be developed and deployed.

- *Which behaviors are most likely to indicate malicious behavior?*

  Behaviors that usually result only from adversaries and not from legitimate users are the most useful to defenders because they result in few false positives.

4.2 Step 2: Acquire Data

In preparation for creating analytics, organizations must identify, collect, and store the data needed for developing the analytics. To identify which data an analyst needs to collect to create analytics, it is important to understand what data is already being collected by existing sensors and logging mechanisms. In some cases, this data may fulfill the data requirements for a given set of analytics. In many instances, however, settings or rules for existing sensors and tools may need to be modified to begin collecting required data. In other cases, new tools or capabilities may need to be installed to collect the required data. For a list of sensors used in the MITRE implementation, refer to Appendix 6A.2.

After the data required to create a given set of analytics has been identified, it must be collected and stored on the platform where the analytics will be written. The MITRE implementation currently relies primarily on a Splunk [17]-based architecture that uses Splunk Universal Forwarders installed on all of its endpoints to send the collected data to a Splunk Heavy Forwarder. To facilitate analytic development on other platforms, MITRE was able to store selected data sources in additional storage technologies by configuring its Splunk Heavy Forwarder to send data to a Hadoop [18] cluster and to Elasticsearch [19] in addition to the Splunk Indexers.

4.2.1 Endpoint Sensing

Many enterprises rely on network sensing at the perimeter due to the ease of sensor deployment at network ingress and egress points. However, this limits visibility to only network traffic that enters or exits the network and does not help defenders gain a view of what occurs within their network and between their systems. Defenders under this perimeter-based sensing paradigm typically rely on netflow, packet capture, firewalls, proxies, network-based intrusion detection system, and other network packet analysis or blocking systems. If an adversary is able to successfully gain access to a system within a monitored perimeter and establish a command and control presence that circumvents network protections, a defender may be blind to the adversary's activity within its own network. As with the frame of reference example in Section 1.1, an adversary's use of legitimate web services and encrypted communications that are commonly allowed to traverse network perimeters makes it difficult for defenders to identify malicious activity within their network.

Our research was driven by the hypothesis that endpoint sensing is necessary to reliably identify post-compromise operations by detecting ATT&CK behaviors that cannot typically be detected using perimeter-based methods. This is due to the higher fidelity of information that can be gathered about adversary actions and effects on endpoint systems, inside the network perimeter. Figure 5 shows an ATT&CK Matrix representing ATT&CK coverage with only perimeter-based network sensors on a notional enterprise network. Red highlighted cells represent no capability to detect a behavior, and yellow cells represent partial ability. Without sensors on endpoints detecting network events, such as process starts and new network connections, it is difficult to detect many of the behaviors described by the ATT&CK model at a high enough confidence level to identify an intrusion without certain a priori knowledge of an adversary (infrastructure and/or C2 protocols) and without certain established defender capabilities (ability to rapidly identify malicious network traffic, decode it, and process it) to determine adversary actions.

Other approaches to intrusion detection that rely on endpoint data gathered through scanning endpoints for IOCs or to acquire snapshots of data may also fail to detect an adversary who has breached the network perimeter and is operating inside the network. Data being collected intermittently could lead to missed opportunities to detect behaviors that occur between snapshots. For example, an adversary may use techniques to reflectively[31] load an unknown RAT into a legitimate process[32] (such as *explorer.exe*) that then proceeds to interact with the system through a remote shell using the *cmd.exe* command-line interface[33]. The adversary's sequence of actions will likely occur over a short period of time and leave very little in the way of artifacts for a network defender to discover after the fact. Snapshots collecting information such as running processes, process trees, loaded DLLs, Autoruns locations, open network connections, and known malware signatures within files may only see a reflectively loaded DLL running

---

[31] Reflective DLL Injection is a method of injecting a DLL into a process without writing anything to disk. This makes detecting this forensically more challenging.

[32] ATT&CK: T1055 – Defense Evasion/DLL Injection

[33] ATT&CK: T1059 – Execution/Command-Line Interface

within *explorer.exe*, if the scan is executed while the RAT is loaded. However, the snapshot will miss the actual injecting of the RAT into *explorer.exe*, the *cmd.exe* launching, the resulting process tree, and additional behaviors performed by the adversary through the command shell because the data was not collected in a persistent manner.

Figure 5. Color Coded ATT&CK Matrix Covering Notional Perimeter-based Defenses

4.3 Step 3: Develop Analytics

Once organizations possess the requisite sensors and data, they can then develop analytics. Developing analytics requires a hardware and software platform on which to design and run analytics, as well as data scientists to design analytics. While this is often done through an organization's SIEM (Security Information and Event Management) platform, this may not be the only way. The MITRE implementation instantiated analytics using Splunk's query language. MITRE researchers categorized the ATT&CK-related analytics into four major types:

- **Behavioral** – An analytic designed to detect a specific adversary behavior, such as creating a new Windows service. The behavior by itself may or may not be malicious. These behaviors should map back to techniques identified within the ATT&CK model.
- **Situational Awareness** – Analytics geared towards a general understanding of what is occurring within a network environment at a given time. Not all analytics need to be geared towards generating alerts on malicious behavior. Rather, an analytic can prove valuable to an organization by providing general information on the state of the environment. Information such as login times do not indicate malicious activity, but when coupled with other indicators, even this type of data can provide much needed information about adversary behaviors. Situational awareness analytics can also be helpful for monitoring the *health* of a network environment (e.g., determining on which hosts sensors are not operating correctly).
- **Anomaly/Outlier** – Analytics that may detect behavior that is not malicious, but which is unusual and may be suspect. Some examples would be detecting executables that have never been run before or identifying processes running on the network that do not typically do so. Like Situational Awareness analytics, these types of analytics do not necessarily indicate an attack.
- **Forensic** – Analytics that are most useful when conducting an investigation regarding an event. Oftentimes, forensic analytics will need some kind of input to be most useful. For example, if an analyst finds that a credential dumper was used on a host, an analytic can be run that will reveal all the users whose credentials were compromised.

A team of defenders may use a combination of these four types of analytics during a cyber game exercise or when developing analytics for real-life application. Examples of how to combine these types of analytics can be demonstrated in the context of the previously described example in Section 1.1 as follows:

1. Security Operation Center (SOC) analysts are first alerted that there may be an attack in progress through an alert generated by an analytic looking for remote creation of a scheduled task (Behavioral).
2. After seeing this alert from the compromised machine, the analyst runs an analytic looking for any anomalous services that have been scheduled on that host. This analytic reveals that not long before the remote task was scheduled, a new service was in fact created on the originating host (Anomaly/Outlier).
3. After identifying the new suspicious service, the analyst then investigates further. He runs an analytic that identifies all child processes of the suspicious service. Investigating in this way may reveal indicators of what activity was being performed on the host. This

investigation exposes the behavior of the RAT. Running the same analytic again, looking for children of the RAT child process reveals the execution of PowerShell by the RAT (Forensic).

4. Having suspected the remote access of other hosts from the compromised machine, the analyst decides to investigate any other remote connections that may have been attempted from that machine. To this end, the analyst runs an analytic detailing all of the remote logins that have occurred in the environment from the machine in question and discovers other hosts to which connections were made (Situational Awareness).

Section 5.2 provides a detailed example of how MITRE iteratively developed analytics during its cyber games.

4.4 Step 4: Develop an Adversary Emulation Scenario

There are many definitions of what activities and types of tests are covered under the umbrella of offensive cyber security testing. Traditional penetration tests focus on highlighting vulnerabilities across different types of systems that an adversary may leverage at some point so that they are remediated. Red Team engagements may focus on a long-term, impactful goal within a target network, such as taking control over a mission critical system. Over the course of a test operation, the Red Team will likely discover vulnerabilities that should be fixed, but the team's scope is generally limited to the path they took to reach an objective and may not cover the breadth of discoveries found while conducting their penetration testing. [20]

MITRE's approach to adversary emulation differs from these traditional approaches. Its goal is for Red Team members to execute behaviors and techniques based from a particular or many known adversaries to test specific aspects of a system or network's defense. The adversary emulation exercises consist of small-scale, repeated engagements that are designed to improve and test defenses on a live network through systematically introducing a variety of new malicious behaviors into the environment. The threat-emulating Red Team works closely with the Blue Team (often referred to as purple teaming [21]) to ensure open lines of communication that are important to quickly honing an organization's defenses so they are tested before an actual adversary begins operating in a targeted network environment. As such, adversary emulation tests are often conducted in a more rapid and focused manner than fully-scoped penetration tests or mission objective focused Red Teams.

As novel detection methods are developed, matured, and deployed across the cyber community, security research will focus on circumventing them, and adversaries will continue to adapt. [22] [23] Adversary emulation scenarios should be developed around this idea and should always keep in mind that most real adversaries have specific objectives, such as gaining access to sensitive information. During testing operations, the Red Team may be given specific objectives as well, but their simulated adversary operations should focus on how they go about trying to achieve their objectives and not whether or not they achieve them, so that the Blue Team will be able to conduct a successful and thorough test of network defenses and capabilities against the most likely adversary techniques.

4.4.1 Scenario Development

When developing an adversary emulation scenario for the purposes of testing network defenses, cyber games may require a high-level plan to facilitate the communication of operational goals without giving away the details of the operational test scenario to either the Red or Blue Team. The White Team should craft this plan, using their knowledge of Blue Team sensor and analytic detection gaps against threat behavior, and changes that the Blue Team has made or new analytics that need to be evaluated. The White Team should also determine whether the Red Team has the capabilities to adequately test adversary behavior. If not, the White Team should work with the Red Team to address any gaps, including any tool development, acquisition, and testing that may be required. The high-level plan can be used as the basis for the full adversary emulation scenario, to communicate requirements, and to coordinate with asset owners and other stakeholders.

Details of the high-level scenario plan may include:

1. Sensor/analytic and defensive capabilities to be tested
2. Common adversary behavior to be used
3. Rough plan with sequences of actions suggested to verify defensive capabilities
4. System, network, or other resources needed for the cyber game/test.

The adversary emulation scenario can be, but is not necessarily, a detailed command-by-command script. It should be detailed enough to give direction to the Red Team to verify defense capabilities but also should be flexible enough as to free the Red Team to adapt their operations as necessary during the exercise to test behavior variations that may not have been considered by the Blue Team. Since Blue Team defenses may also reach a maturity that covers known threat behaviors, the Red Team must also be free to expand beyond pure emulation. By using the White Team to decide which new behaviors should be tested, the Blue Team can remain unaware of what specific activity to expect, and the Red Team can remain untethered by assumptions about Blue Team capabilities that may impact Red Team decisions. The White Team also continues to be free to inform the Red Team of details about the environment to fully test detection methods against sequences of adversary behavior.

4.4.1.1 Example Scenario 1

In Scenario 1, an adversary uses functionality of the Windows operating system and available system utilities to perform certain actions. [11] An adversary's custom tools provide an access point and a C2 channel, but the adversary chooses to interact with systems through an interactive command shell. The Blue Team has deployed Microsoft Sysinternal's Sysmon for persistent process monitoring and command-line argument collection. The goal of this scenario is to test and develop post-compromise detection analytics based on the telemetry data Sysmon collects from endpoints within the live network.

High-level scenario:

1. The Red Team may be given a specific end goal. For example, to gain access to a particular system, domain account, or to gather specific information to exfiltrate.

2. Assume breach and give the Red Team access to an internal system to test post-compromise behavior. The Red Team is given execution of a loader or RAT on one system in the environment to simulate the success of pre-compromise actions and gaining an initial foothold regardless of prior knowledge, access, exploit, or social engineering variables.
3. The Red Team must use discovery techniques from the ATT&CK model to learn about the environment and to gather data to continue. This is done using available Windows utilities.
4. The Red Team dumps credentials on the initial system and attempts to locate nearby systems against which the credentials can be leveraged.
5. The Red Team moves laterally until the target system/account/information is obtained

The high-level scenario is used to construct a more defined plan for the Red Team using ATT&CK as an adversary emulation guidebook. The selection of techniques is focused on those that are required to meet the test objective based on how they are commonly applied in known intrusion activity, but allow for some variation in technique use by the red team to introduce additional behavior.

Figure 6 shows the ATT&CK matrix representation of Scenario 1. Green highlighted cells depict techniques required to achieve the test objective. The operation may be conducted in a shallow manner with just the main techniques or as a more comprehensive test with the Red Team performing additional actions that are left open-ended. The yellow highlighted cells represent general suggested techniques for a more comprehensive adversary emulation scenario. See Appendix B for the full scenario sequence breakdown and ATT&CK tactic and technique mapping with specific tools and commands.

Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Execution	Collection	Exfiltration	Command and Control
DLL Search Order Hijacking			Brute Force	Account Discovery	Windows Remote Management		Automated Collection	Automated Exfiltration	Commonly Used Port
Legitimate Credentials			Credential Dumping	Application Window Discovery	Third-party Software		Clipboard Data	Data Compressed	Communication Through Removable Media
Accessibility Features		Binary Padding			Application Deployment Software	Command-Line	Data Staged	Data Encrypted	
AppInit DLLs		Code Signing	Credential Manipulation	File and Directory Discovery		Execution through API	Data from Local System	Data Transfer Size Limits	Custom Command and Control Protocol
Local Port Monitor		Component Firmware			Exploitation of Vulnerability	Graphical User Interface	Data from Network Shared Drive	Exfiltration Over Alternative Protocol	
New Service		DLL Side-Loading	Credentials in Files	Local Network Configuration Discovery		InstallUtil			Custom Cryptographic Protocol
Path Interception		Disabling Security Tools	Input Capture		Logon Scripts	PowerShell	Data from Removable Media	Exfiltration Over Command and Control Channel	
Scheduled Task		File Deletion	Network Sniffing	Local Network Connections Discovery	Pass the Hash	Process Hollowing			Data Obfuscation
Service File Permissions Weakness		File System Logical Offsets	Two-Factor Authentication Interception		Pass the Ticket	Regsvcs / Regasm	Email Collection		Fallback Channels
Service Registry Permissions Weakness				Network Service Scanning	Remote Desktop Protocol	Regsvr32	Input Capture	Exfiltration Over Other Network Medium	Multi-Stage Channels
Web Shell		Indicator Blocking		Peripheral Device Discovery	Remote File Copy	Rundll32	Screen Capture		Multiband Communication
Basic Input/Output System	Exploitation of Vulnerability				Remote Services	Scheduled Task		Exfiltration Over Physical Medium	
	Bypass User Account Control			Permission Groups Discovery	Replication Through Removable Media	Scripting			Multilayer Encryption
Bootkit	DLL Injection					Service Execution		Scheduled Transfer	Peer Connections
Change Default File Association		Indicator Removal from Tools		Process Discovery	Shared Webroot	Windows Management Instrumentation			Remote File Copy
				Query Registry	Taint Shared Content				
Component Firmware		Indicator Removal on Host		Remote System Discovery	Windows Admin Shares				Standard Application Layer Protocol
Hypervisor				Security Software Discovery					Standard Cryptographic Protocol
Logon Scripts		InstallUtil							
Modify Existing Service		Masquerading		System Information Discovery					Standard Non-Application Layer Protocol
Redundant Access		Modify Registry							
Registry Run Keys / Start Folder		NTFS Extended Attributes		System Owner/User Discovery					Uncommonly Used Port
									Web Service
Security Support Provider		Obfuscated Files or Information		System Service Discovery					
Shortcut Modification		Process Hollowing							
Windows Management Instrumentation Event Subscription		Redundant Access							
		Regsvcs / Regasm							
		Regsvr32							
Winlogon Helper DLL		Rootkit							
		Rundll32							
		Scripting							
		Software Packing							
		Timestomp							

Figure 6. Scenario 1 Plan ATT&CK Matrix

4.4.1.2 Example Scenario 2

A second cyber game (Scenario 2) could be developed based on the successes and failures of the first. The goal of the second would be to verify any new data sources and analytics that were found to be working against the previously emulated behavior in the first cyber game. The second cyber game would also be used to introduce new adversarial behavior by the Red Team.

Since the end of Scenario 1, the Blue Team has deployed persistent process monitoring tools and capabilities, such as Sysmon, to capture process invocation and command-line arguments. Doing so makes the ATT&CK behaviors from the first cyber game available for defensive analysis.

In Scenario 2, it is assumed the Blue Team now has a high detection rate of the Red Team's adversary emulation activities from Scenario 1. As such, part of the goal for Scenario 2 is to exercise sensor and analytic capabilities against the prior ATT&CK behaviors and technique variations that may not be effectively detected by process monitoring. In Scenario 2, command-line arguments are used by the Blue Team to capture the resilience of its analytics.

Recent threat reporting has indicated some adversaries have adopted the use of PowerShell as is referenced in the ATT&CK PowerShell entry. [24] Other reports have suggested PowerShell is being used by both advanced targeted attacks and crimeware to evade detection and that the use of PowerShell in this manner is a growing trend. [25] Depending on how PowerShell is used and the existence and type of PowerShell logging on the target endpoint systems, it could effectively mask adversary techniques that were effectively detected by process monitoring and command-line arguments in Scenario 1.

The high-level sequence of events matches that of Scenario 1. Scenario 2 mirrors the same goal and use of tactics and techniques, but the techniques are executed through PowerShell equivalents instead of common Windows-based utilities.

Freely available PowerShell tools, such as Empire [26], can be used in place of some common Windows utilities to test defenses against PowerShell-based behaviors. For Scenario 2, the Red Team could use PowerShell alternatives of a subset of ATT&CK techniques successfully used during Scenario 1 in an attempt to avoid detection by Blue Team analytics. Figure 7 is a color-coded ATT&CK matrix displaying the techniques used in Scenario 2. The techniques that are could be modified using PowerShell alternatives appear with an orange border. See Appendix B for a detailed list of techniques for Scenario 2.

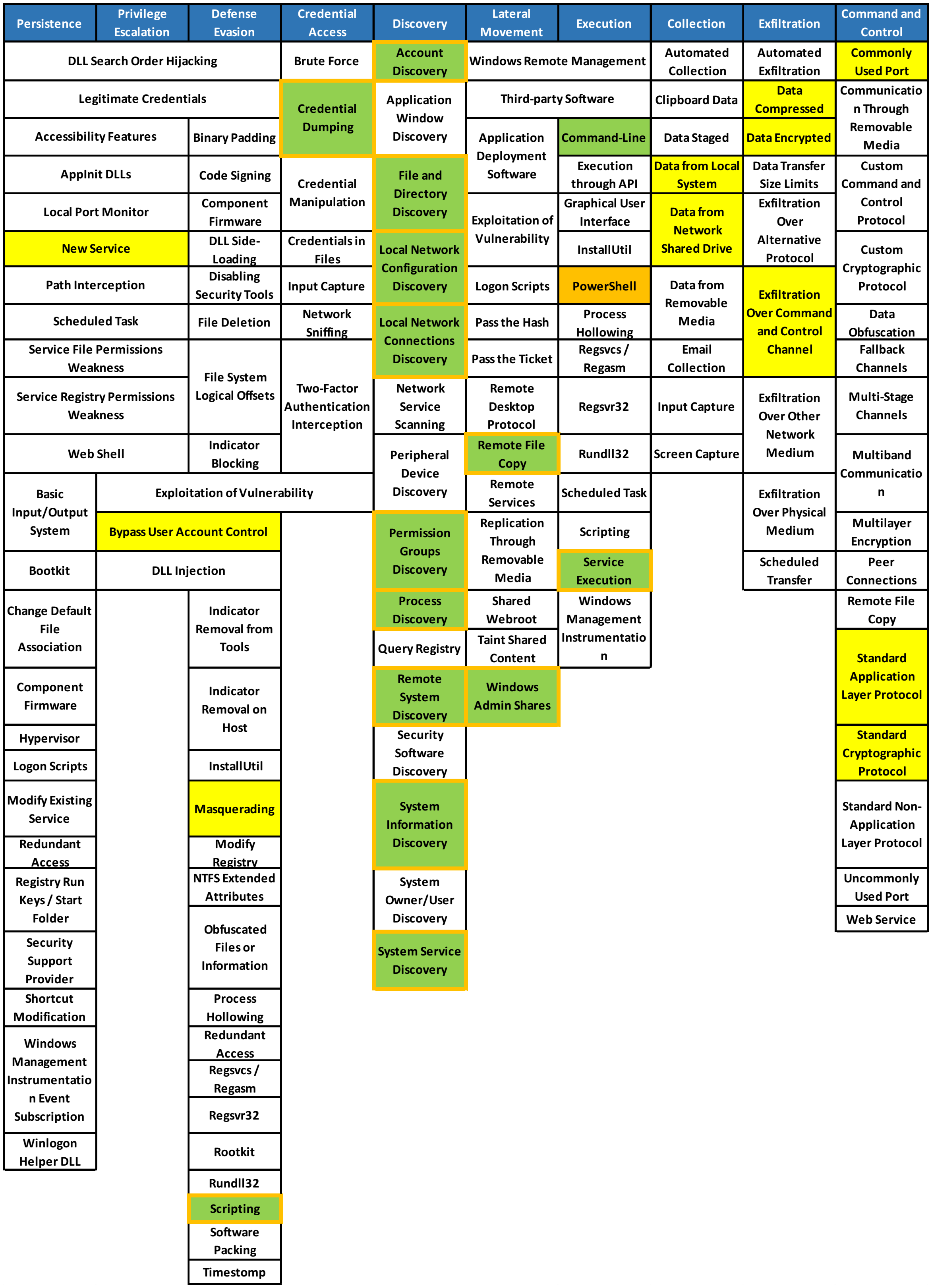

Figure 7. Scenario 2 ATT&CK Matrix

4.5 Step 5: Emulate Threat

After designing the scenarios and the analytics, it is time to use the scenarios to emulate the adversary to test the functionality of the analytics in Step 6. This is done by having a Red Team emulate threat behavior and perform techniques as scoped by a White Team. Adversary emulation operations allow analytic developers to verify the efficacy of their cyber defenses. To place the focus on post-compromise adversary behavior, the Red Team begins with access to the enterprise network through a remote access tool on a particular system in a given network environment. This access expedites the assessment and ensures that post-compromise defenses are adequately tested. The Red Team then follows the plan and guidelines outlined by the White Team.

The White Team should coordinate any adversary emulation activity with an organization's network asset owners and security organizations to ensure awareness in case of network issues, user concerns, security incidents, or other issues that may arise.

4.6 Step 6: Investigate Attack

Once the Red Team component has been conducted in a given cyber game, the Blue team gathers to attempt to discover what the Red Team did. In many of MITRE's cyber games, the Blue Team included the developers responsible for creating the analytics being used. The benefit of this is that the developers experience first-hand how well their analytics perform in a near real-life situation, and their lessons learned can drive future development and refinement. On the other hand, it is beneficial to occasionally conduct a Blue Team with analysts who have no experience with the analytics being used. Testing in this way helps to ensure that the Blue Team's success is not dependent on institutional knowledge and that the analytics are intuitive for all users, not just for those who developed them.

The Blue Team starts their portion of the cyber game with a set of high-confidence analytics that, if successful, provide an initial indicator of where and when the Red Team may have been active. This is important, as the Blue Team is not given any information regarding Red Team activity, other than a vague window of time, normally on the order of a month. Oftentimes the Blue Team's high confidence analytics fall into the "behavioral" category of analytics, although some may belong in the "anomaly/outlier" category. The results of applying these high-confidence analytics drive the Blue Team to further investigate individual hosts using the other types of analytics previously described (situational awareness, outlier/anomaly, and forensic). The information derived from the higher confidence analytics helps to focus and refine the higher noise analytics, increasing the confidence that their output is indicative of Red Team activity. This process of using the output of one analytic to help refine another's is iterative and is repeated throughout the exercise as new information is gathered.

Eventually, as events are identified as belonging to the Red Team, a timeline of activity begins to form. Understanding the timeline of events is important and can help analysts infer information that cannot be gained purely through the analytics. Gaps in activity in the timeline can identify windows of time where further investigation is needed. Also, by looking at the data this way, Blue Team members may be able to infer where activity may be found, even without having any other evidence of such activity. For example, seeing a new executable run but having no evidence of how it was placed on the machine may alert analysts of potential Red Team behavior and can provide details on how the Red Team accomplished its lateral movement. These clues can also lead to ideas for new analytics that need to be written for the next iteration of the ATT&CK based analytic development method.

While investigating the Red Team's attack, the Blue Team develops several overarching categories of information as their portion of the exercise progresses. These can take the form of pieces of information they wish to discover, such as:

- **Hosts Involved/Compromised** – This is often represented during the exercise as a list of hosts and the reason(s) why each host has been identified as suspicious. This is critical information when trying to remediate an incident.
- **Accounts Compromised** – It is crucial that the Blue Team be able to identify the accounts that have been compromised on a network. Failure to do so allows the Red Team, or an adversary in real life, to regain access to the network from other vectors, thus negating all previously made remediation efforts.
- **Objective** – The Blue Team also needs to endeavor to discover the Red Team's objectives and whether they achieved them or not. This is often one of the hardest aspects to uncover as it requires a large corpus of data to determine with confidence.
- **TTPs Used** – Red Team TTPs are important to note at the end of the exercise as a way to identify future work that needs to be done. The Red Team may have exploited misconfigurations in a network that need to be addressed, or the Blue Team may discover a technique that the Blue Team cannot identify currently without further sensing. The TTPs that the Blue Team identifies should be compared to the list of TTPs the Red Team claims to have used to identify any defense gaps.

4.7 Step 7: Evaluate Performance

After both Blue and Red Team activities are complete, the White Team facilitates analysis by team members that compares the Red Team activity to what was reported by the Blue Team. This allows a comprehensive comparison from which the Blue Team gains invaluable information about how successful they were at discovering Red Team actions. Using this information, the Blue Team refines existing analytics and identifies adversary behaviors for which they need to develop or install new sensors, collect new sets of data, or create new analytics.

5 Real-World Experiences

To validate its approach, MITRE created a *living lab*—an enclave of about 250 computers on MITRE's corporate networks. These were the real PCs that MITRE employees use every day for their actual work. Host and network sensors were then added to these PCs before developing behavioral analytics using the ATT&CK-based analytics development method outlined in Section 4. MITRE developed its analytics in a live environment (as opposed to a lab environment), because analytics that work well in the lab may fail in the real world due to an unacceptable level of false positives.

5.1 Cyber Game Experiences

MITRE performed six cyber games of experimentation between 2012-2013. Over the course of these games, both the Red and Blue Teams refined their skills and repeatedly adapted to each other's capabilities. Each game provided a learning experience for how future exercises should be structured and scoped, and subsequent exercises attempted to integrate these lessons learned. This section discusses the first three cyber games, examines lessons learned, and describes how these lessons drove the refinement of future cyber games. This section also annotates the cyber game narratives with ATT&CK tactics and techniques in the footnotes to show how the ATT&CK model was incorporated into MITRE's plans.

As discussed in Section 4, these games were performed several weeks after the simulated compromise. Part of the reason for this is that the cyber games were performed by researchers and not full-time network defenders. If synchronous games had been conducted, they would have required researchers to constantly monitor the network, or it would have required them to monitor the network during a pre-arranged date range. The constant monitoring of networks would have precluded MITRE researchers from conducting their actual research. Alternatively, giving researchers a set of days to constantly monitor the network would have simulated an APT telling network defenders when they were going to attack—an unrealistic scenario at best.

MITRE's analytic writers, on the other hand, monitored their analytics every few days, just in case something truly malicious did occur on the portion of the MITRE network being used for the cyber games. Whenever a researcher discovered something suspicious, he or she asked the White Team lead to identify whether the identified activity was part of an exercise or not. Once the analytics reached maturity, the Blue Team usually detected the Red Team during the adversary emulation operation. Any researchers who detected Red Team activities were asked not to participate in the Blue Team exercise until the other Blue Team members had narrowed down Red Team activity to a specific time frame, so as not to bias the rest of the team members with a priori information about the exercise.

Cyber Game #1 - For the first cyber game, the Blue Team was composed of people from various backgrounds, none of which were directly related to Blue Teaming. The Blue Team grew in experience over subsequent cyber games. Conversely, all Red Team members had several years of Red Team experience, thus making the apparent outcome favor the Red Team. This first exercise was used as a test to evaluate the usefulness of several potential sensors, and as such, these sensors were not widely deployed at the time. This led to limited visibility and made detecting the Red Team activity across hosts challenging. The necessity of widespread deployment of these sensors was one of the fundamental lessons learned from this exercise. Despite the disparity in past experience between the two teams, the Blue Team was able to identify a significant amount of the Red Team activity. Red Team activity consisted mostly of being intentionally noisy and of writing *flag* files to different hosts using Windows file shares[34] (SMB). This provided evidence that the Red Team had established some control on those hosts and created a metric by which Blue Team success could be measured. The Red Team also ran credential harvesting tools[35]. Overall the Blue Team was highly successful in their investigation despite their limited experience. This was in large part due to the use of endpoint sensors with visibility into the command lines of process start events on some hosts and visibility into SMB file copies performed by the Red Team on others.

Cyber Game #2 - For the second cyber game, the Red Team decided to use more advanced tactics in light of the Blue Team's prior success. Many of the analytics that were developed in response to the first cyber game failed to yield any results in the second cyber game because the Red Team renamed built-in windows commands[36] and did not perform obvious actions such as writing files with names such as *flag.txt*. In addition to this, the sensor environment was still in a state of flux. This led to gaps where the Blue Team had little visibility.

Despite having the lowest success rate of any of the cyber games, in the second cyber game, the Blue Team learned valuable lessons for future exercises. One of the major efforts undertaken as a result of this exercise was the development of analytics for analyzing user login activity across the network—with a focus on admin activity. On the planning side, MITRE discovered how much more difficult it is for a Blue Team to discover malicious activity when the environment is not stable and sensor deployment is occurring concurrently. In future games, a greater effort was made by the White Team to ensure a stable network environment prior to the beginning of the exercise.

Cyber Game #3 - While the first cyber game was too easy and the second was too hard for the Blue Team, the third game was somewhere in between. In this cyber game, the White Team wanted to ensure that the Blue Team would be challenged and be exposed to new TTPs, while at the same time being allowed to validate previously used analytics. It was during this exercise that the White Team planners decided to intentionally drive the Red Team towards certain TTPs that would generate alerts from Blue Team analytics and to use TTPs unfamiliar to the Blue Team. The Blue Team achieved a similar success rate to that of the first exercise. They were able to identify the majority of Red Team activity, but there were still a couple of gaps that they

---

[34] ATT&CK: T1077 – Lateral Movement/Windows Admin Shares

[35] ATT&CK: T1003 – Credential Access/Credential Dumping

[36] ATT&CK: T1036 – Defense Evasion/Masquerading

struggled to fill, such as identifying the information that was actually exfiltrated[37] and determining how the Red Team maintained its persistence[38].

Subsequent exercises followed the same methodology utilized in the third game. As new analytics were developed, the Red Team intentionally performed activity designed to trigger them. The Red Team also adopted new TTPs to push the Blue Team to develop new and better ways to detect APT activity. How the Blue Team approached the exercises changed over time as well; at first there was very little coordination or communication between team members. As exercises were held, the Blue Team tried different methods to address this problem so as to minimize duplication of effort. They began to use collaboration software (Microsoft SharePoint and OneNote) to share discoveries and information with each other. The following list details some of the important points of information the Blue Team found helpful regarding collaboration during a cyber game:

- A single timeline was maintained that all the team members could reference and add to, as new information was gained. This helped to minimize duplicative effort and helped to identify gaps in knowledge regarding Red Team activity.
- A suspicious host list was built over the course of the engagement that outlined all of the hosts the Blue Team felt was worthy of investigation. This list also included some basic system information from each host, such as IP addresses, usernames, and why that host was identified as being suspicious. Having this extended information aided other team members when they saw network sessions from a host or IP address and wanted to know if that host was already suspected of being the victim of malicious activity.
- Drawing a network graph of how all the suspicious hosts were related was also very helpful. By including details like how the connections between hosts were made and the temporal ordering of events analysts could more easily see the progression of the compromise. It also helped to see how all the hosts were related. *Note: Nodes of activity might also be easier to identify using this method.*
- The Blue Team initially failed to realize the importance of capturing detailed notes about their actions during the exercise. This impacted their ability to relay details about discovered events to other team members and made it very difficult to write up the Blue Team report after the end of the exercise. As time went on, the Blue Team agreed on a standard template that could be used by all member of the team for recording the data they looked at and the results of their investigations. This template included data such as time of execution of the analytic or query, the actual analytic or query initiated, what they were searching for and why, and the results.

5.2 Analytic Iteration

As discussed previously, one of the core aspects of ATT&CK-based analytics is iteration. A good example of this process of analytic development across cyber games was the evolution of

[37] ATT&CK: Tactic – Exfiltration

[38] ATT&CK: Tactic – Persistence

the Suspicious Commands Analytic. (See Section 0 for analytic details.) This analytic was originally created to identify instances where certain commands were used in the network environment. The list of commands it looked for was originally based on open source threat information and ATT&CK techniques. The commands came from over 20 different ATT&CK techniques, such as Discovery/Process Discovery, Persistence/New Service, and Execution/Scheduled Task.

After the first cyber game, MITRE researchers learned several lessons about this analytic. One of these lessons was that the initial noise level was too high, requiring researchers to modify the analytic to look for groups of commands that were run together. This both reduced the noise level as well as added more context to each command, better enabling analysts to determine if a command was used maliciously or not. The Red Team also used commands that were not previously on the list of commands being monitored. After identifying this gap, the Blue Team added those commands to the list of commands being monitored and began monitoring the updated analytic.

One of the major changes the Red Team made during the second cyber game was to rename many of the tools they used to better hide them from the Blue Team. This tactic thwarted the Suspicious Command Analytic and forced the Blue Team to identify a new way of finding these commands. To accomplish this, the Blue Team started looking for arguments that were commonly used by these commands. This method was noisier but it did enable the Blue Team to successfully identify Red Team activity. After the exercise ended, the Blue Team made a note of this analytic idea in their report and later refined it into a complete analytic. Like the analytic that preceded it, this new *suspicious arguments* analytic was further refined as a result of day-to-day monitoring and more cyber games.

The analytic products of this research effort were documented and published in the MITRE Cyber Analytic Repository (CAR). CAR contains analytics mapped to specific ATT&CK techniques and describes the high-level analytic hypothesis, pseudocode analytic implementation, unit tests, and the data model used to develop them so the analytics can be transcribed to various analytic platforms. CAR is intended to be used by cyber defenders throughout the community and serves as a mechanism for sharing behavioral-based analytics that can be used for adversary detection. See Appendix A.1 for example analytics. CAR was released in 2016 and is available at https://car.mitre.org.

6 Summary

APTs have been and will likely continue to be successful at gaining initial access into targeted enterprise networks. The ATT&CK-based analytics development method is a powerful tool for network defenders to use for creating and maintaining a capability to detect these threats. Detection using these methods does not rely on typical known-bad IOCs or external notification of a network breach and can lead to the rapid discovery of a network compromise by detecting an adversary's use of techniques described in the ATT&CK model.

The analytics development method is based on five principles:

- Include post-compromise detection
- Focus on behavior
- Base on a threat model
- Iterate by design
- Develop in a realistic environment.

MITRE researchers used the ATT&CK model to inform both the development of analytics and the structuring of Red Teams for cyber games. MITRE utilized the seven steps of the ATT&CK-based analytics development method to iteratively improve its defensive posture in its cyber games with new and refined analytics. MITRE has published CAR to serve as the analytic sharing platform to document the analytics developed through this effort for the community to use and expand upon.

This page intentionally left blank.

References

[1] FireEye, "M-Trends 2016," [Online]. Available: https://www2.fireeye.com/rs/848-DID-242/images/Mtrends2016.pdf.

[2] S. Tomonaga, "Windows Commands Abused by Attackers," 26 January 2016. [Online]. Available: http://blog.jpcert.or.jp/2016/01/windows-commands-abused-by-attackers.html. [Accessed 17 September 2016].

[3] The MITRE Corporation, "Adversarial Tactics, Techniques and Common Knowledge," July 2016. [Online]. Available: https://attack.mitre.org/. [Accessed 6 September 2016].

[4] The MITRE Corporation, "Cyber Analytics Repository," September 2016. [Online]. Available: https://car.mitre.org/wiki/Main_Page. [Accessed 13 September 2016].

[5] Financial Times, "A sobering day," [Online]. Available: http://labs.ft.com/2013/05/a-sobering-day/.

[6] SecureList, "THE MsnMM Campaigns," SecureList, 2015.

[7] FireEye, "HAMMERTOSS: Stealthy Tactics Define a Russian Cyber Threat Group," 2015.

[8] B. Delpy, "mimkatz," [Online]. Available: http://blog.gentilkiwi.com/mimikatz. [Accessed 19 05 2016].

[9] "PowerSploit," [Online]. Available: https://github.com/PowerShellMafia/PowerSploit. [Accessed 19 05 2016].

[10] HarmJ0y, "Veil-Evasion," [Online]. Available: https://www.veil-framework.com/framework/veil-evasion/. [Accessed 25 05 2016].

[11] SecureWorks, "Living Off the Land," 28 May 2015. [Online]. Available: https://www.secureworks.com/blog/living-off-the-land. [Accessed 26 January 2016].

[12] L. Szekeres, M. Payer, T. Wei and D. Song, "Sok: Eternal war in memory," in *IEEE Symposium on Security and Privacy*, 2013.

[13] Wikipedia, "Stuxnet," [Online]. Available: https://en.wikipedia.org/wiki/Stuxnet. [Accessed 06 09 2016].

[14] W. Gragido, "Understanding Indicators of Compromise (IOC) Part 1," [Online]. Available: http://blogs.rsa.com/understanding-indicators-of-compromise-ioc-part-i/. [Accessed 07 07 2016].

[15] The MITRE Corporation, "Threat-based Defense," [Online]. Available: https://www.mitre.org/capabilities/cybersecurity/threat-based-defense. [Accessed September 2016].

[16] The MITRE Corporation, "ATT&CK Matrix," 9 2016. [Online]. Available: https://attack.mitre.org/wiki/ATT%26CK_Matrix. [Accessed 11 9 2016].

[17] Splunk, "Splunk," [Online]. Available: http://www.splunk.com/. [Accessed 25 05 2016].

[18] Apache, "Welcome to Apache(TM) Hadoop," [Online]. Available: http://hadoop.apache.org/. [Accessed 06 09 2016].

[19] Elastic, "Elasticsearch | Elastic," [Online]. Available: https://www.elastic.co/. [Accessed 06 09 2016].

[20] R. Mudge, "Adversary Simulation Becomes a Thing...," 12 11 2014. [Online]. Available: http://blog.cobaltstrike.com/2014/11/12/adversary-simulation-becomes-a-thing/. [Accessed 11 9 2016].

[21] C. Gates, "More on Purple Teaming," 21 March 2016. [Online]. Available: http://carnal0wnage.attackresearch.com/2016/03/more-on-purple-teaming.html. [Accessed 4 April 2016].

[22] M. Mimoso, "Core Windows Utility Can Be Used to Bypass AppLocker," 21 4 2016. [Online]. Available: https://threatpost.com/core-windows-utility-can-be-used-to-bypass-applocker/117604/. [Accessed 11 9 2016].

[23] FireEye, "Angler Exploit Kit Evading EMET," [Online]. Available: https://www.fireeye.com/blog/threat-research/2016/06/angler_exploit_kite.html. [Accessed 07 09 2016].

[24] The MITRE Corporation, "PowerShell - ATT&CK," July 2016. [Online]. Available: https://attack.mitre.org/wiki/Technique/T1086. [Accessed 13 September 2016].

[25] Carbon Black, "'PowerShell' Deep Dive: A United Threat Research Report," 21 April 2016. [Online]. Available: https://www.carbonblack.com/wp-content/uploads/2016/04/Cb-Powershell-Deep-Dive-A-United-Threat-Research-Report-1.pdf. [Accessed 18 September 2016].

[26] W. Schroeder, J. Warner and M. Nelson, "Empire," September 2016. [Online]. Available: https://github.com/PowerShellEmpire/Empire. [Accessed 18 September 2016].

[27] Microsoft, "Sysmon," [Online]. Available: https://technet.microsoft.com/en-us/sysinternals/sysmon. [Accessed 24 05 2016].

[28] Microsoft, "Autoruns for Windows v13.51," [Online]. Available: https://technet.microsoft.com/en-us/sysinternals/bb963902. [Accessed 24 05 2016].

[29] Microsoft, "About Event Tracing," [Online]. Available: https://msdn.microsoft.com/en-us/library/windows/desktop/aa363668(v=vs.85).aspx. [Accessed 24 05 2016].

[30] E. M. a. C. M. J. a. A. R. M. Hutchins, "Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains," *Leading Issues in Information Warfare & Security Research,* vol. 1, p. 80, 2011.

This page intentionally left blank.

Appendix A Details on MITRE's Implementation

This appendix contains information about the sensors, data, and analytics used within the MITRE environment during its cyber games. The data was gathered from sensors on MITRE's living lab network. The data that was used was actual user data from real employee workstations. This provided a real-life environment for testing analytics and sensors and for developing a true feeling for noise and efficacy that would otherwise be impossible to ascertain in an artificial test network.

A.1 Example Analytics

This section details some of the analytics that our Blue Team used to uncover Red Team activity. Several exemplars are described below along with the analytic type for each (See Section 4.3). A link is also provided to the public MITRE CAR page that references each analytic.

Analytic 1: Suspicious Commands
CAR URL: https://car.mitre.org/wiki/CAR-2013-04-002
Type: Behavioral
Data Required: Process Creation Events with Command Line Information

This analytic identifies sequences of executed processes that have been identified as suspicious. In this context, *suspicious* refers to those processes that are built-in, freely available, or used by legitimate administrators, but which are also known to be used by adversaries. This analytic attempts to identify when groups of these processes are executed together based on several criteria: the amount of time between each occurrence; the parent process; and the host. By looking for groups of processes being executed simultaneously, the noise an analyst must sift through is reduced, and valuable context is added to the event. Examples of these processes include *net.exe*, *reg.exe*, *dsquery.exe*, and *schtasks.exe*. This analytic finds behavior that touches on over 20 different ATT&CK TTPs. The following table lists only a few of the tactics/techniques found by this analytic. The entire list can be viewed on the CAR site.

Analytic 1 ATT&CK Coverage:

Tactic	Technique
Discovery	Process Discovery
Defense Evasion	Disabling Security Tools
Execution	Command Line Interface
More on CAR site	…

Analytic 2: Remotely Launched Executables via Services
CAR URL: https://car.mitre.org/wiki/CAR-2014-03-005
Type: Behavioral
Data: Process Creation Events, Process Network Connection Events with Remote Procedure Call (RPC) Metadata

This analytic correlates data from multiple sensors to identify instances where an adversary uses the RPC functionality of Windows to remotely start the execution of an executable on a host. Such activity is identified by correlating an RPC connection to *services.exe* with a near immediate process start event of *services.exe*.

Analytic 2 ATT&CK Coverage:

Tactic	Technique
Execution	New Service
Execution	Modifying Existing Service
Execution	Service Execution

Analytic 3: User Login Activity Monitoring
CAR URL: https://car.mitre.org/wiki/CAR-2013-10-001
Type: Informational
Data: Login, Logout and Remote Login Events (from Windows Event Logs)

Situational awareness analytics provide valuable information to analysts about when an incident occurs. This analytic uses Windows security logs to track user logon sessions, including both local and remote logon sessions. This is useful to help establish which user account(s) was responsible for, or impacted by, the compromise of a host. It also provides general user account logon patterns which can be useful for advanced anomaly and behavioral analysis.

Analytic 3 ATT&CK Coverage:

Tactic	Technique
Defense Evasion	Legitimate Credentials
Lateral Movement	Remote Desktop Protocol

Analytic 4: Server Message Block (SMB) Copy and Execution
CAR URL: https://car.mitre.org/wiki/CAR-2013-05-005
Type: Behavioral
Data: Process Creation Events, Network Flow Events with SMB Metadata

Adversaries commonly use the SMB Protocol to write malicious executables or scripts to a remote host that are then executed at a later time. This analytic looks for an SMB file write event (via file or network monitoring) and the process execution event corresponding to that executable or script.

Analytic 4 ATT&CK Coverage:

Tactic	Technique
Lateral Movement	Windows Admin Shares
Defense Evasion, Lateral Movement	Legitimate Credentials
Lateral Movement	Remote File Copy

Analytic 5: Outlier Parents of *cmd.exe*
CAR URL: https://car.mitre.org/wiki/CAR-2014-11-002
Type: Anomaly/Outlier
Data: Process Creation Events

Some programs execute the Windows command shell frequently as part of their normal operation. Malicious actors commonly spawn command prompts from processes that do not normally spawn command prompts, either by using custom malware or by hijacking other processes. By identifying instances where a command prompt is started by a process that does not normally start command prompts, this kind of behavior can be observed.

Analytic 5 ATT&CK Coverage:

Tactic	Technique
Execution	Command Line Interface

A.2 Sensors

During the course of our research, a wide variety of different sensors were used to collect data in the living lab. Because most of the data that was necessary to build the ATT&CK-based analytics had to be collected from endpoint hosts, MITRE's sensing efforts gravitated towards sensors that could be installed on endpoint systems. At the time of this writing, the endpoint tools or data sources that are currently in use are:

- **Windows Event Logs**

Standard Windows logs from hosts. These are used for tracking user session information, including login attempts.

- **Sysmon [27]**

A Microsoft Sysinternals tool that captures a variety of endpoint data. It is used for capturing process creation events and their associated command lines, but it can also capture network connection creation, driver loads, and dynamic linking library (DLL) module loads. It can also detect when a file has its creation time altered (also known as timestomping).

- **Autoruns [28]**

A Microsoft Sysinternals tool that provides data relating to all programs that are scheduled to start automatically on a host. Configuring a program to automatically start is a common technique used by adversaries to achieve persistence. *Note: Although this tool is scanning-based and not event-driven, as the other sensors are, MITRE is not aware of any fully-featured, freely-available, and event-driven alternative at the time of writing*.

- **Custom Endpoint Netflow Sensor**

A MITRE built endpoint network flow program. Similar to a NetFlow sensor, it captures network connection flows, but augments them with data from the end host, such as process ID. (This is included in Sysmon as well.) This sensor also decodes portions of a session and enriches flow records with that metadata. The two primary advantages of this are the ability to associate a process with network connections and the ability to perform protocol decoding of the Windows SMB Protocol that underlies much of the native functionality of Windows Enterprise Domains.

- **Custom Event Tracing for Windows [29] Sensor**

A MITRE-built sensor that fills several gaps in sensing. This sensor is primarily used to detect process injection (which can also be detected by the later versions of Sysmon), which is a common tactic used by credential dumpers to obtain user logon information. This sensor also captures process creation and termination, file creation, and file deletion events for certain files and paths.

- **Computer Properties and Sensor Checks**

Several times each day, every computer collects an informational snapshot of its current status. This status includes information related to network devices and configuration, hardware profile, disk utilization, memory utilization, last boot time, operating system information, and a list of all users who have logged on since the last boot. This data is primarily used in situational awareness analytics, but it can also enrich queries related to credential access to help determine which credentials may have been compromised. Additionally, there are periodic checks performed to return the status of all of our sensor services and executables in order to determine if they are running, stopped, or not present on each system.

Appendix B Scenario Details

The sections below expand and give more details to the scenarios outlined in Section 4.4.1, as well as detailing which ATT&CK techniques a Red Team should use for adversarial emulation.

B.1 Scenario 1

Example of a high-level tactic sequence and selection of ATT&CK techniques:

1. The emulated adversary gains execution through an initial foothold that was provided by some means provided by the White Team. The following could represent an embedded protocol within HTTP over TCP port 80 which could also be used to move additional tools into the network:

ATT&CK Tactic	Technique	ID
Command and Control	Standard Application Layer Protocol	T1071
Command and Control	Commonly Used Port	T1043
Command and Control	Remote File Copy	T1105

2. Once established, initiate a reverse shell command interface through the remote access tool:

ATT&CK Tactic	Technique	ID	Tool/Command
Execution	Command-Line Interface	T1059	cmd.exe

3. Perform Discovery techniques through the command interface:

ATT&CK Tactic	Technique	ID	Tool/Command
Discovery	Account Discovery	T1087	net localgroup administrators net group <groupname> /domain net user /domain
Discovery	File and Directory Discovery	T1083	dir cd
Discovery	Local Network Configuration Discovery	T1016	ipconfig /all

Discovery	Local Network Connections Discovery	T1049	netstat -ano
Discovery	Permission Groups Discovery	T1069	net localgroup net group /domain
Discovery	Process Discovery	T1057	tasklist /v
Discovery	Remote System Discovery	T1018	net view
Discovery	System Information Discovery	T1082	systeminfo
Discovery	System Service Discovery	T1007	net start

4. After sufficient information is learned perform open-ended tactics and techniques as needed. The following techniques are based on suggested actions from ATT&CK to establish persistence or to escalate privileges and establish persistence. After sufficient privileges are obtained, then dump credentials using Mimikatz or attempt to acquire credentials with a keylogger to capture user input.

ATT&CK Tactic	Technique	ID
Persistence	New Service	T1050
Persistence	Registry Run Keys / Start Folder	T1060
Privilege Escalation, Defense Evasion	Bypass User Account Control	T1088
Credential Access	Dump Credentials	T1003
Credential Access	Input Capture	T1056

5. If credentials are obtained and knowledge of systems is sufficient from the Discovery techniques, then attempt to move laterally to exercise the main objective of this scenario.

ATT&CK Tactic	Technique	ID	Tool/Command
Lateral Movement	Windows Admin Shares	T1077	net use * \\<remote system>\ADMIN$ <password> /user:<domain>\<account>

Lateral Movement	Remote File Copy	T1105	copy <source path to file> <remote share destination>
Execution	Service Execution	T1035	psexec

6. Continue spreading laterally using the previous techniques as needed to obtain the target sensitive information and exfiltrate it. Files may be collected and exfiltrated using the following suggested ATT&CK techniques:

ATT&CK Tactic	Technique	ID
Collection	Data from Local System	T1005
Collection	Data from Network Shared Drive	T1039
Exfiltration	Data Compressed	T1002
Exfiltration	Data Encrypted	T1022
Exfiltration	Exfiltration Over Command and Control Channel	T1041

B.2 Scenario 2

Example high level tactic sequence and selection of ATT&CK techniques:

1. The emulated adversary gains execution through initial foothold that was provided. The command and control channel for this scenario was left open-ended. The following was suggested and could represent an embedded protocol within HTTPS using SSL/TLS over TCP port 443 which could also be used to move additional tools into the network:

ATT&CK Tactic	Technique	ID
Command and Control	Standard Application Layer Protocol	T1071
Command and Control	Standard Cryptographic Protocol	T1032
Command and Control	Commonly Used Port	T1043
Command and Control	Remote File Copy	T1105

2. Depending on the RAT used, a reverse shell command interface may be invoked to issue PowerShell commands:

ATT&CK Tactic	Technique	ID	Tool/Command
Execution	Command-Line Interface	T1059	cmd.exe

3. Perform Discovery techniques through the command interface using equivalent PowerShell scripts or offensive framework. This example uses the Empire framework. [26] The first two rows act as modifiers to the remaining techniques in the table:

ATT&CK Tactic	Technique	ID	Tool/Module/Command
Execution	PowerShell	T1086	powershell.exe
Defense Evasion	Scripting	T1064	powershell.exe
Discovery	Account Discovery	T1087	situational_awareness/network/netview
Discovery	File and Directory Discovery	T1083	dir/ls cd
Discovery	Local Network Configuration Discovery	T1016	ipconfig
Discovery	Local Network Connections Discovery	T1049	situational_awareness/network/netview
Discovery	Permission Groups Discovery	T1069	situational_awareness/network/netview
Discovery	Process Discovery	T1057	tasklist/ps
Discovery	Remote System Discovery	T1018	situational_awareness/network/netview
Discovery	System Information Discovery	T1082	sysinfo
Discovery	System Service Discovery	T1007	situational_awareness/network/netview

4. After sufficient information is learned perform open-ended tactics and techniques as needed. The following techniques are based on suggested actions from ATT&CK.

Establish persistence or escalate privileges and establish persistence. After sufficient privileges are obtained, then dump credentials using the Invoke-Mimikatz module.

ATT&CK Tactic	Technique	ID
Persistence	New Service	T1050
Privilege Escalation, Defense Evasion	Bypass User Account Control	T1088
Credential Access	Dump Credentials	T1003

5. If credentials are obtained and knowledge of systems is sufficient from the Discovery techniques, then attempt to move laterally to exercise the main objective of this scenario.

ATT&CK Tactic	Technique	ID	Tool/Module
Lateral Movement	Windows Admin Shares	T1077	lateral_movement/invoke_psexec
Lateral Movement	Remote File Copy	T1105	lateral_movement/invoke_psexec
Execution	Service Execution	T1035	lateral_movement/invoke_psexec

6. As with Scenario 1, continue spreading laterally using the previous techniques as needed to reach the objective. For data exfiltration, the following ATT&CK techniques are suggested and should be perform through PowerShell when possible:

ATT&CK Tactic	Technique	ID
Collection	Data from Local System	T1005
Collection	Data from Network Shared Drive	T1039
Exfiltration	Data Compressed	T1002
Exfiltration	Data Encrypted	T1022
Exfiltration	Exfiltration Over Command and Control Channel	T1041