An Advanced Persistent Threat Exemplar

Ryan Wagner, Matthew Fredrikson, David Garlan

July 2017
CMU-ISR-17-100

Institute for Software Research
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

This material is based upon work funded and supported by the Department of Homeland Security under Contract No. FA8702-15- D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center sponsored by the United States Department of Defense, and the National Security Agency.

The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

DM17-0399

Keywords: Advanced Persistent Threat, APT, Exemplar, Cyber Security, Cybersecurity, Cyber, Attack, Adaptation, Self-Adaptive

Abstract

Security researchers do not have sufficient example systems for conducting research on advanced persistent threats, and companies and agencies that experience attacks in the wild are reluctant to release detailed information that can be examined. In this paper, we describe an Advanced Persistent Threat Exemplar that is intended to provide a real-world attack scenario with sufficient complexity for reasoning about defensive system adaptation, while not containing so much information as to be too complex. It draws from actual published attacks and experiences as a security engineer by the authors.

Contents

0.1 Introduction

This paper describes an attack scenario that can be used by researchers to investigate their hypotheses. The scenario consists of an example enterprise network architecture and a trace of an attack from initial compromise to data exfiltration. The attack trace is accompanied by a step-by-step description of considerations relevant to the attacker and defender. This example is constructed to be realistic, with sufficient depth to accommodate security researchers' needs while eliminating extraneous details.

In section 0.2, we describe the need for such an exemplar, including criteria for evaluating the usefulness of one. In section 0.3, we propose a simple, notional enterprise architecture, complete with vulnerabilities, that can be exploited by a clever adversary. Section 0.4 enumerates a specific sequence of exploits the adversary could use to gain a beachhead, establish presence, move laterally, and escalate privileges within such an environment. Section 0.6 proposes several quality attributes that can be examined in the context of the exemplar attack. In section 0.7, we evaluate our exemplar against the proposed criteria, and in section 0.8, we conclude.

Advanced persistent threats (APTs) require a more holistic, high-level understanding and response than approaches that focus on mitigating a single vulnerability or class of vulnerabilities. NIST defines an APT as:

> An adversary that possesses sophisticated levels of expertise and significant resources which allow it to create opportunities to achieve its objectives by using multiple attack vectors (e.g., cyber, physical, and deception). These objectives typically include establishing and extending footholds within the information technology infrastructure of the targeted organizations for purposes of exfiltrating information, undermining or impeding critical aspects of a mission, program, or organization; or positioning itself to carry out these objectives in the future. The advanced persistent threat: (i) pursues its objectives repeatedly over an extended period of time; (ii) adapts to defenders' efforts to resist it; and (iii) is determined to maintain the level of interaction needed to execute its objectives.[1]

APTs are adversarial, observing and reacting in real time to defensive tactics. They have a defined intent, and can leverage multiple resources to craft a complex, multi-step approach that occurs over a potentially long period of time. Because of their intelligent, adaptive, and resourceful natures, APTs present a significant threat to computer systems, including the critical systems modern society depends upon. As such, it is important for researchers to have available a realistic exemplar of an APT attack for considering proposed approaches to defending against them.

0.2 Goals

0.2.1 Purpose

The purpose of this example is to provide a reusable and realistic example of an APT type of attack against which researchers can test their approaches. Actual examples can be difficult to come by because they are often proprietary, if they occur in a business setting, or classified, if they occur in a government setting.

A successful attack scenario models an instance of an APT. It should enable consideration of the aspects of a realistic adversary that has extensive time and resources. There should be opportunities for both an adversary and defender to adapt their responses based on their observations of each others' actions. It should span multiple steps and encompass a variety of disparate attack techniques.

A number of criteria guided the creation of this example:

- Realism. To the extent possible, the example should draw on actual attacks that have occurred "in the wild." A key assumption here is that attackers and defenders have finite resources, including time. The attack complexity should be roughly commensurate with actual APT attacks, both in number of exploit types and difficulty of attack. The defender's environment should be a representative scale model of a typical network environment, with the types of components seen in real environments. The attack should have a goal (e.g., exfiltrating a specific data set) just like traditional APT attacks.

- Depth. The example should be complex enough to demonstrate a multi-stage attack scenario. At multiple points in the scenario, the attacker and defender should be confronted with trade-offs (e.g., security versus functionality) that complicate their individual goals.
- Abstraction. To guide researchers, this scenario should be simple enough to be modeled without the burden of numerous extraneous steps and details that might obscure the salient information relevant to understanding the attacker's and defender's current state and actions.
- Utility for research. The ultimate goal of this scenario is to provide researchers something that clearly presents interesting research challenges and can be used to validate research in defending against sophisticated APT-style attacks.

0.2.2 Benefit to Others

There are several benefits in having an example attack scenario such as this one. For researchers who are not security experts, a peer-reviewed example of a complex attack ensures that research into APT defenses is validated against a realistic attack scenario.

Even for researchers who are familiar with security, benefits exist. By using a consistent scenario, researchers can ensure accurate comparisons of approaches. Additionally, a corpus of accepted attack scenarios saves research time, enabling resources to focus on how to solve a security problem rather than creating an attack example.

Lastly, an example attack scenario aids dialog between researchers and often their industry or government funding sources, which are bound by confidentiality agreements. An agreed-upon neutral scenario provides researchers the ability to reason about and evaluate their approaches without being bound to non-disclosure agreements that could limit the impact and reuse of research.

0.3 Attack Victim Architecture

0.3.1 Scenario Inspiration

In 2013, the retailer Target was the subject of a sophisticated attack that resulted in the loss of information, including credit card information for up to 110 million individuals [2]. The attack was caused, in part, by the theft of a third party contractor's login credentials for a Target-operated system used by Target's suppliers [3]. The attacker used these credentials to gain access to Target's internal network. From there, the attackers were able to leverage vulnerabilities and misconfigurations within Target's corporate network to ultimately compromise Point of Sale (PoS) terminals. The compromised PoS terminals collected sensitive payment information and exfiltrated it to servers operated by the attackers [4]. While the exact details of the attack are not public, this exemplar follows the outline of that attack instantiated over a small enterprise network that is sufficiently complex to be representative of a similar style of attack. The exemplar also draws from other published examples of cyber attacks [5] [6].

0.3.2 Architecture Common Characteristics

To model an attack, we construct a notional architecture with properties analogous to prior known attacks. With the exception of scale, this architecture should have properties in common with typical enterprise architectures. For example, enterprise networks are generally delimited by a perimeter separating the enterprise network from the internet; this perimeter may have defenses to inspect and apply tactics to the traffic crossing in either direction. Network connections must exist between services of varying levels of trust, like that between an external entity and an enterprise-provided service.

A variety of hosts exist within the network; some of them contribute to only one specific function, but many hosts are necessary for the delivery of a variety of functions. An identity and access management service falls into the latter category because it is necessary across the enterprise. These hosts are likely to have a variety of vulnerabilities; some might be known, but many are not. These vulnerabilities fall into a variety of categories like software bugs (e.g., buffer overflows), misconfigurations, etc.

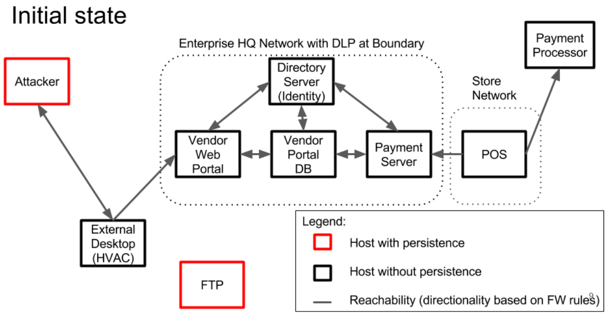

Figure 1: This is the initial state of the system.

0.3.3 Architecture Overview

The architecture at the outset of the attack is depicted in Figure 1. The attacker has an established presence on a host system outside the enterprise network. The attacker coordinates the attacks from this host and also operates a File Transfer Protocol (FTP) server elsewhere on the internet. Also, a third party contractor to the enterprise has a host that is located outside the enterprise network; the contractor logs in from this host to an enterprise web portal for the enterprise's vendors.

This web portal leverages a Microsoft SQL Server for database services. The SQL server is misconfigured to run with the privileges of the SYSTEM user. Also, the web application contains a SQL injection vulnerability. The web portal and database server are both within the enterprise network boundary.

Payments are processed by PoS terminals located within each store. To reduce latency, they connect directly from the store network to a payment processor. A redacted log of payment information is sent from the PoS terminal on the store network to a payment server on the enterprise network. The payment server in this case also stores copies of firmware for download by the PoS terminals.

To prevent exfiltration of sensitive information, a Data Loss Prevention (DLP) server sits at the network perimeter and monitors all traffic that passes through the perimeter to see if it contains prohibited sensitive information.

0.4 Exemplar Attack Sequence

0.4.1 Attack Common Characteristics

Like the architecture described in section 0.3, the attack should also have many properties in common with attacks seen in the wild. For example, the attacker should possess the attributes of an advanced persistent threat described by the National Institute of Standards and technology, including "using multiple attack vectors," a willingness to operate over a long period of time, adaptability based on defenses, and a strong incentive to carry out a particular mission [1].

Attacks consist of various stages that include reconnaissance, exploitation of a vulnerability to gain or escalate privileges, installation of software to maintain persistence, lateral movement, and continued command and control through the attack [7]. It is common for attacks to begin with phishing [8]. To make the attack coordination easier, some APTs leverage a malware toolkit like Poison Ivy to provide a turnkey malware solution [9].

0.4.2 Third Party Compromise Phase

Third Party Spear-Phishing

The attack begins when the attacker sends a spear-phishing email to the contractor. The contractor opens the email and its attachment.

Third Party Persistence

The attachment installs a multi-stage exploit toolkit on the contractor's host as depicted in Figure 2. This begins

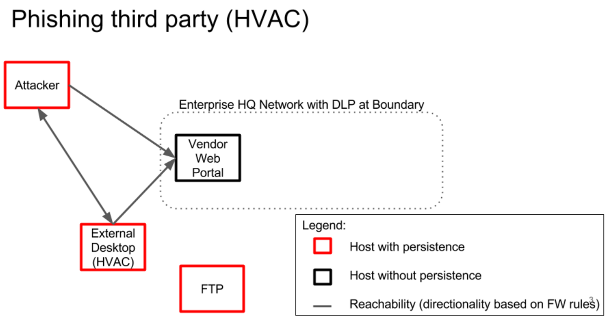
Phishing third party (HVAC)
Attacker
Enterprise HQ Network with DLP at Boundary
Vendor Web Portal
External Desktop (HVAC)
FTP
Legend:
Host with persistence
Host without persistence
Reachability (directionality based on FW rules)

Figure 2: The attacker successfully phishes the contractor.

with the installation of a dropper, a small piece of software that reaches back to the attacker for command and control (C2) and additional components of malware to install [10]. The attacker instructs the contractor's host to install a keystroke logger.

Third Party Password Capture

The logger captures the contractor's username and password for the web portal and sends them to the attacker.

0.4.3 Enterprise Compromise Phase

Web Portal Login

With the username and password, the attacker is able to login directly to the web portal (Figure 3).

Web Portal SQL Injection and Privilege Escalation

Through manual or automated testing of the form fields on the portal website, the attacker discovers and leverages a SQL injection vulnerability. The SQL server provides users a command shell-like interface with SYSTEM (superuser) privileges on the database within the enterprise, and the attacker leverages this configuration error.

Web Portal Maintain Persistence

To maintain presence, the attacker commands the database server to install a malware toolkit (Figure 4).

Enterprise Password Cracking

The attacker also dumps the Security Account Manager (SAM) password hash file from this sever and retrieves it for offline cracking using a hash table (like rainbow tables) in which a password can be quickly looked up by its hash [11].

Enterprise Reconnaissance

With a foothold on the database server, the attacker uses a network scanning tool like nmap [12] to scan the network and locate the payment server.

Enterprise Lateral Movement

Using the credentials cracked before, the attacker is able to login (via the connection to the database server) to the DLP server, which the attacker has reconfigured to not alert when payment information is exfiltrated.

Point of Sale Compromise

Using the cracked credentials, the attacker is able to login to the payment server, where she uploads a copy of malicious PoS firmware (Figure 5). The attacker configures the payment server to act as a relay for unredacted payment information, which will be received from the PoS terminals and forwarded to the FTP server (Figure 6).

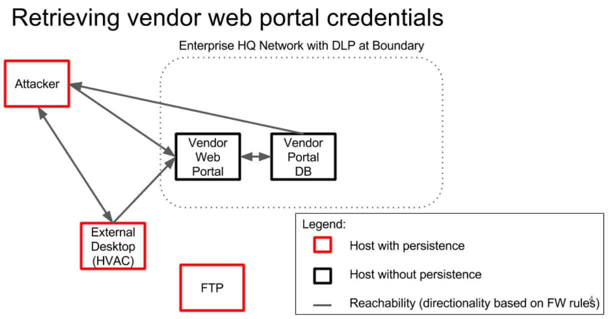
Retrieving vendor web portal credentials
Enterprise HQ Network with DLP at Boundary
Attacker
Vendor Web Portal
Vendor Portal DB
External Desktop (HVAC)
FTP
Legend:
🟥 Host with persistence
□ Host without persistence
— Reachability (directionality based on FW rules)

Figure 3: Using valid credentials, the attacker logs into the portal.

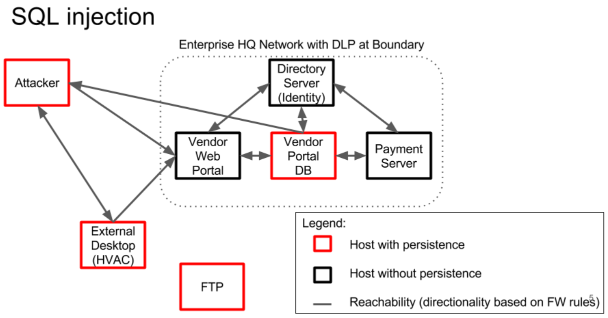
SQL injection
Enterprise HQ Network with DLP at Boundary
Attacker
Directory Server (Identity)
Vendor Web Portal
Vendor Portal DB
Payment Server
External Desktop (HVAC)
FTP
Legend:
🟥 Host with persistence
□ Host without persistence
— Reachability (directionality based on FW rules)

Figure 4: The attacker exploits a SQL injection flaw.

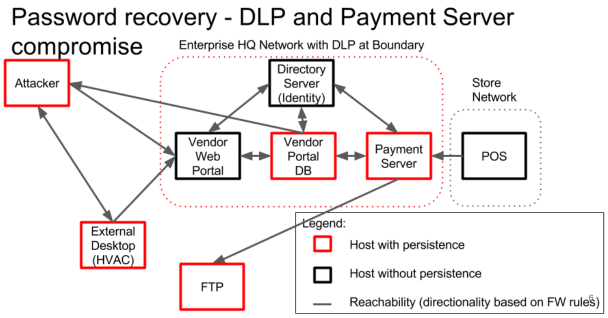
Password recovery - DLP and Payment Server compromise
Enterprise HQ Network with DLP at Boundary
Attacker
Directory Server (Identity)
Store Network
Vendor Web Portal
Vendor Portal DB
Payment Server
POS
External Desktop (HVAC)
FTP
Legend:
🟥 Host with persistence
□ Host without persistence
— Reachability (directionality based on FW rules)

Figure 5: The attacker cracks passwords offline.

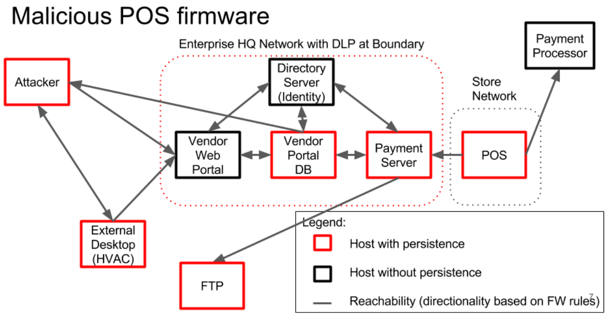

Figure 6: The attacker compromises the PoS devices and exfiltrates data.

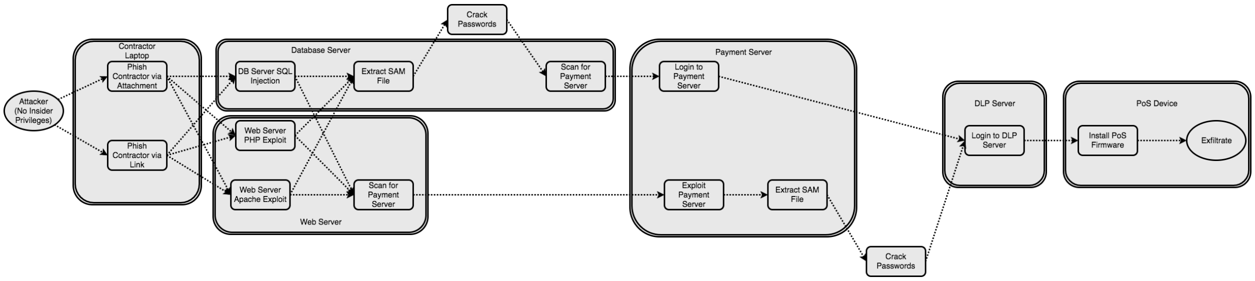

Figure 7: Simplified Attack Scenario Graph with Multiple Branches

0.4.4 Point of Sale Compromise and Exfiltration Phase

Point of Sale Persistence

At this point, the final stage of the attack begins. When the PoS terminals inquire about updates, they find a new (malicious) firmware instance available, and they download and install this software. This firmware contains the original PoS firmware functionality, but has added, malicious functionality that enables the final step of the attack.

Point of Sale Exfiltration

The PoS firmware instructs the PoS devices to send unredacted payment information to the payment server within the enterprise network.

With all the pieces in place, the attack is ready to collect stolen credit card information and send it to the attacker's FTP server without raising alerts on the DLP server.

0.4.5 Defense Common Characteristics

While the focus of this exemplar is on the defender's network architecture and the attack upon it, there are considerations for the defense of the architecture. At each stage of the attack, if the defender observes an indicator of compromise, he may choose to apply a defense tactic. Alternatively, he may proactively apply defense tactics. However, tactics have costs and may be mutually exlusive. Some tactics may take time to take effect and some might be observable to the attacker. If the attacker suspects the defender has discovered her, she may alter her tactics, techniques, and procedures to avoid further observation.

0.5 Attack Branching

More sophisticated analysis will incorporate an adaptive adversary that is making decisions in real time in response to a defender's action. At various points in time, the adversary will utilize her observations of any defense tactics to make decisions about which branch to take in the attack scenario. To enable this type of reasoning, the exemplar can be extended to include additional attack scenarios.

Because of the branching in the attack scenario, the defender will have uncertainties regarding the path to be taken by the attacker. This uncertainty is only fully resolved once the defender notices observable information emitted from the attacker taking one of the branches in the scenario.

Branch 1: Phishing. The very first branch in this scenario is based on the type of phishing the attacker uses. She could send an attachment with malware leveraging a document viewer (e.g., PDF) vulnerability that results in the installation of a keylogger on the victim's machine. The keylogger can extract passwords typed by the victim. Alternatively, the attacker can replace the malicious attachment with a malicious hyperlink. This requires the victim to visit an attacker-controlled website that leverages a browser vulnerability to install keylogger malware.

Branch 2: Intranet Service Exploit. A successful phishing attack results in the attacker gaining access to credentials to login to a web portal on the enterprise's network. This exposes the database server, which is vulnerable to SQL injection, and the web server, whic is vulnerable to a PHP exploit and an HTTP server exploit.

Branch 3: SQL Injection. The goal of the SQL injection step can be one of at least two different outcomes. The SQL injection could lead to running a command to extract a SAM (password hash) file from a host for offline password cracking. However, a SQL injection could also be used to install network scanning tools and perform reconnaissance on the database server's network. This reconnaissance leads to the discovery of a vulnerable service on the payment server. That vulnerability could be leveraged to remotely gain privileged access to the payment server. That privileged access could be used to extract a SAM file, as described above, for further lateral movement.

0.6 Quality Attributes for the Attacker and Defender

While the exemplar follows a single attack trace from beginning to end, at many steps in the attack, attackers and defenders are confronted with considerations that could cause branches in the scenario. A selected few of the considerations are explained in more detail below.

0.6.1 Observability of the Attacker

An attacker will generally wish to stay below the radar during the course of the attack. Because of this, she will choose tactics that minimize the ability of the attacker to detect, track, or reconstruct the attack. In this particular scenario, attack tactics, like the rate of data exfiltration or the aggressiveness of network reconnaissance, can affect the likelihood that the attacker will be observed.

The defender can utilize tactics to make it more difficult for an attacker to be unobserved. For example, the defender can add a honeynet (a network of fake hosts and services) so the attacker must perform additional scanning and launch additional exploits during network reconnaissance. This forces the attacker to be more observable.

0.6.2 Observability of the Defender

The defender also has a stake in maintaining low observability. In this case, the defender may wish to keep secret his understanding of the attacker. If the attacker is aware that the defender has noticed her attack, she can modify her tactics to reduce her observability. As the defender gains information to more completely understand the attack, he increases his chance that attempts to fully evict the attacker will be successful.

0.6.3 Cost to the Attacker

Each step in the attack has a cost. The cost can be in terms of the amount of time a step takes or other resources it consumes. For example, an attacker might wish to consider the judicious use of novel exploits. These are rarer than other exploits and hence more valuable. Additionally, if a novel exploit is used, it might be discovered by the defender, making the exploit difficult to reuse later in the same scenario or even with a different target (e.g., if the discovery of the exploit leads to the creation of an intrusion prevention system signature). Additionally, exploits often have a cost based on the difficulty of creating the exploit and the installation base of the targeted application (e.g., an exploit in a commonly-used program is often more valuable) [13].

0.6.4 Cost to Defender

The defender also has finite resources. For example, creating a particularly large honeynet could tax computational resources both for establishing the network and monitoring the log data. Rebooting hosts or services, or removing them from the network entirely, results in a loss of system utility.

0.7 Scenario Evaluation

0.7.1 Realism

This scenario is composed of realistic components–both from an attack and a defense perspective. The scenario is originally constructed loosely from the details of the breach of Target in 2013 [3] [2]. In this breach, the exploitation began with a phishing email sent to a third-party contractor for Target. The contractor had access to service running on a host within Target's internal network. Apparently, this host utilized a privileged service with a default username; this service ran with the same credentials across the network. Due to lack of sufficient isolation of internal subnetworks, the attacker was able to move laterally through the internal network to a point from which they could install malware on the Point of Sale systems. The malware exfiltrated credit card data to the attackers.

The exemplar in this paper begins with that scenario, but it simplifies the network, which could have a large number of hosts and network devices. Further, the exemplar incorporates assumptions about how an attacker might execute such an attack using a combination of common attack methods. These methods include buffer overflow, SQL injection, and abuse of a misconfiguration. By incorporating a variety of vulnerabilities in the scenario, researchers can examine the impacts of exploits that occur with varying levels of visibility to the defender and are mitigated using different tactics.

For example, the exemplar assumes SQL injection is possible on the vendor web portal, the web application where the contractor can submit invoices to the enterprise. Additionally, the exemplar assumes a misconfiguration of the SQL server; in this example, the SQL server is a Microsoft SQL server running with administrator privileges and has PowerShell enabled, allowing for command line interaction with the server. The SQL injection will have a different level of observability and different set of defense tactics than network reconnaissance. This enables researchers to explore strategies such as focusing defensive tactics on increasing attack observability.

0.7.2 Depth

In our analyses to date, this exemplar has proven to have sufficient complexity. It has hosts that are both within and also not within the defender's control. There are multiple potential paths to exploitation. Some hosts provide just one overall function (e.g., vendor servicing or payment processing), while other hosts contribute to multiple functions. The number of classes of exploits is sufficient to warrant a variety of defense tactics.

On the other hand, the scenario is not too complex to cause modeling problems like state explosion. The number of hosts and vulnerabilities is relatively small when compared to a full scale enterprise network environment.

0.7.3 Abstraction

For our current analyses, we found the level of abstraction in this particular scenario to be appropriate. Without understanding the categories of weakness being exploited (e.g., buffer overflow, SQL injection, misconfiguration), we cannot make assumptions about exploit observability to the defender, effective defense tactics, mitigation observability to the attacker, etc.

For modeling purposes, we have not yet needed to add more layers of detail to the attack. For example, the choice of a particular vulnerability (e.g., using a particular Common Vulnerability Enumeration entry) would add detail, but not provide useful information for our current modeling efforts. Similarly, the assignment of specific IP addresses to hosts in the network has not yet been needed, but it could prove useful to other researchers. The exemplar can easily be extended to incorporate these details if they are relevant to research.

0.7.4 Utility for Research

This attack scenario is useful to researchers with a variety of security objectives. As examples, this scenario can be used to model timing, observability, and graceful degradation.

0.7.5 Timing of Attacker Eviction

For researchers interested in the impact of timing on defenses, the scenario includes exploits with a range of timing requirements. For example, a phishing attack is, to some extent, a matter of luck, requiring a victim to open an attachment or click on a link in advance of the rest of the attack. It may take time for the first successful phishing exploit to succeed. Other aspects of the attack, like network scanning and password cracking, can take minutes, hours, or longer to complete. On the other hand, once a password has been cracked, lateral movement on the network can occur at network speed.

Similarly, defense tactics can range in timing. A password change can occur nearly instantaneously, but the instantiation of a large honeynet or rebuilding a compromised server from scratch might take longer. This range of timing for both attack and defense tactics enables researchers to explore the impact of timing on their models.

As an example, we are actively exploring the impact of timing on tactics in this exemplar. Specifically, we are exploring the tradeoff between attempting to evict an attacker based on the current level of knowledge held by the defender, or continuing to observe to gather more knowledge while increasing the likelihood of a successful attacker eviction.

0.7.6 Observability

Observability is an attribute that appears across the scenario. Some steps in the exploit might not raise the attacker's observability. For example, when the attacker uses stolen credentials to log in to the vendor web portal, this might look indistinguishable from a login by the legitimate user. On the other hand, network reconnaissance is often a noisy undertaking, with large amounts of traffic generated and much of it going to unused IP address space.

Observability of defense tactics is also captured in the scenario. Changing a password would be detected by an attacker. The decision to deploy a honeynet in response to an attack is observable; however, if the honeynet were deployed prior to an attack, similar information would be gathered without tipping off the attacker that her presence was detected. In the first case, the attacker might note the addition of numerous decoy hosts on the network; in the second case, there is no detectable change in the enterprise infrastructure in response to an attack. Of course, running a honeynet also has a cost, and its premature deployment may be unnecessary.

0.7.7 Graceful Degradation

Each of the enterprise's components has an associated function or group of functions, and these functions can be associated with a utility. For example the vendor web portal's web server only supports the function of vendor invoicing. The Point of Sale devices contribute to the function of payment processing. However, the directory server provides an identity function that is integral to both the vendor invoicing and payment processing functions. If a defender were exploring options for graceful degradation, there are a number of potential outcomes for loss of functionality. We are exploring how knowledge of an attacker's presence, combined with knowledge of the network topology, can be used to determine strategies for self-adaptive graceful degradation while under attack.

Researchers interested in graceful degradation can utilize the functionality associated with the hosts and services. The multiple overlapping functions provide a testbed for researchers who wish to explore how systems can maximize utility while managing risk in the face of an ongoing or predicted attack.

0.8 Conclusion

The attack scenario described in this paper provides a realistic example that can be used by researchers to validate models of attacker and defender behaviors. The scenario is loosely based on a real-world example, with details filled in based on other commonly-observed attack attributes. The scale of the example is sufficient to be useful in investigating a number of research questions while not being overloaded with superfluous detail or extraneous hosts and services. The example is general enough that it can be further refined by researchers to serve their specific needs, while providing a consistent construct.

.1 Attack Defenses and Considerations

The attached spreadsheet in Table 1 describes the attack in more detail. The attack steps are labeled with the goal of each step (e.g., establish beachhead, privilege escalation, lateral movement, etc.). Some of these steps require a particular vulnerability to be exploited, so the exploit technique is listed there. As the attacker moves through the network, she creates observables—things that a defender might be able to see and correlate to detect and repel the attack.

The defender has a set of tactics for slowing or reversing the course of the attack at many of the steps in the attack. These are listed in the "Defense Tactics" column and expanded on in more detail in Table 2. Of course, these defense tactics create observables of their own, and it may not be in the defender's interest to make the attacker aware that she has been noticed. More details of the defense tactics are described in the spreadsheet tab of the same name.

Last, to reason about the impact of attack and defense tactics on the network, we must have a concept of how the various services contribute to the disparate missions of the networks. This is covered in Table 3. "Critical" means that the service contributes to a function essential to the operations of the organization (e.g., email, payroll). "Core" means that the system contributes to a function that is a value-add of the organization (e.g., logistics, credit card payment processing).

.1.1 Definitions of Terms

Attack Scenario

Short Name
This is a brief title this step of the attack.

Description
A more detailed prose description of the attack step.

Phase
The attacker's intermediate goal at this step (e.g., establish a beachhead, privilege escalation, lateral movement, etc.).

Exploit Technique
(Where applicable) The type of exploit leveraged.

Observables
Log and other data that is created as a byproduct of this step of the attack and can be observed and correlated by the defender.

Sensors
The specific sensing devices that collect the observables.

Defense Tactics
Tactics that may be employed by the defender to slow or repel the attack at this step.

Targeted Host
The specific host being attacked at this point in the attack.

Min Time Needed
The minimum amount of time required for this attack step.

Probability of Success
The likelihood that this attack step will be successful.

Defense Tactics

Name
The name of the defense tactic.

Locale
Describes if the tactic is implemented across the network or somewhere at the level of the host (e.g., hardware or OS).

Startup Time
The time to enact the tactic, assuming the components are generally in place *a priori*.

Cost A rough order of magnitude of the cost of implementing the tactic.

Observability
An indication of how observable this defense tactic is to the attacker.

Observability Time
An estimate of the amount of time it takes for an attacker probing the system to be able to determine that a defense tactic has been applied (without necessarily identifying the particular tactic).

Effectiveness
The short term effectiveness of this tactic in slowing or repelling the attacker.

Impact
The expected outcome of the defense tactic, including its impact on making the attacker more observable to the defender. This could also include changes that decrease usability (e.g., disabling an account) or degrade performance (e.g., throttling a connection).

Applicability
Preconditions for use and effectiveness of the tactic. (Note that some tactics may be run well ahead of an attack, or they may be repeated.)

Trigger
What might cause a defense technique to be applied.

Timing
If the defense technique can be applied proactively before or reactively after an attack step.

Network Information

Host
The name of the host.

Business Criticality
The level of importance to the business's operations. Core means that this is fundamental to the value-add of the business (i.e., part of the business's core competency). Critical means that the host is necessary for business functions.

Time Criticality

A rough order of magnitude estimate of how long the host can be offline before business is seriously impacted.

Purpose

The mission or goal that the host helps achieve.

Bibliography

[1] R. Kissel, "Glossary of key information security terms," *NIST Interagency Reports NIST IR*, vol. 7298, no. 3, 2013.

[2] B. Krebs, "Email attack on vendor set up breach at Target," Feb 2014. [Online]. Available: https://krebsonsecurity.com/2014/02/email-attack-on-vendor-set-up-breach-at-target/comment-page-2/

[3] "A "kill chain" analysis of the 2013 Target data breach," Mar 2014. [Online]. Available: https://www.commerce.senate.gov/public/_cache/files/24d3c229-4f2f-405d-b8db-a3a67f183883/23E30AA955B5C00FE57CFD709621592C.2014-0325-target-kill-chain-analysis.pdf

[4] B. Krebs, "A first look at the Target intrusion, malware," Jan 2014. [Online]. Available: https://krebsonsecurity.com/2014/01/a-first-look-at-the-target-intrusion-malware/

[5] R. Rohozinski and R. Deibert, "Tracking ghostnet: Investigating a cyber espionage network," *Information Warfare Monitor*, vol. 29, 2009.

[6] D. McWhorter, "APT1: Exposing one of China's cyber espionage units," *Mandiant Corporation*, 2013.

[7] E. M. Hutchins, M. J. Cloppert, and R. M. Amin, "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.

[8] "Verizon Business 2016 data breach investigations report," 2016. [Online]. Available: http://www.verizonenterprise.com/resources/reports/rp_DBIR_2016_Report_en_xg.pdf

[9] J. Butler and K. Kendal, "Blackout: What really happened," *Black Hat USA*, 2008. [Online]. Available: http://www.blackhat.com/presentations/bh-usa-07/Butler_and_Kendall/Presentation/bh-usa-07-butler_and_kendall.pdf

[10] F. Li, A. Lai, and D. Ddl, "Evidence of advanced persistent threat: A case study of malware for political espionage," in *Malicious and Unwanted Software (MALWARE), 2011 6th International Conference on*. IEEE, 2011, pp. 102–109.

[11] P. Oechslin, "Making a faster cryptanalytic time-memory trade-off," in *Annual International Cryptology Conference*. Springer, 2003, pp. 617–630.

[12] "Nmap: the network mapper - free security scanner." [Online]. Available: https://nmap.org/

[13] "The digital arms trade," *The Economist*, Mar 2013. [Online]. Available: http://www.economist.com/news/business/21574478-market-software-helps-hackers-penetrate-computer-systems-digital-arms-trade

Short Name	Description	Phase	Exploit Technique	Observables	Sensors	Defense Tactics	Targeted Host(s)	Min Time Needed	Exploit Cost to Attacker
Spear phishing	Attacker sends email with malicious PDF attachment to HVAC company employee. The employee opens the attachment, and the malware dropper installs.	Establish beachhead, privilege escalation	PDF reader buffer overflow	Email from attacker, malicious attachment, buffer overflow, malware installation	None	ASLR, N-Variant, Randomized instruction set	External desktop	Seconds	High
Contractor exploit toolkit installation	Malware dropper contacts C2 server (let's say via IRC) and downloads exploit kit. This is done first with a DNS lookup to a DNS name that only briefly exists. The exploit kit is installed and calls back to the C2 server for additional instructions.	Maintain persistence		DNS lookup, IRC connection, malware download, malware installation, continued C2 calls	None	Reboot, Delete malware, Take compromised system offline, Restore from clean backup, Rebuild system from scratch	External desktop	Seconds	Medium
Contractor keylogger installation	The attacker instructs the exploit kit to install a keylogger and send captured passwords.	Lateral movement		DNS lookup, IRC connection, malware download, malware installation, continued C2 calls	None	Reboot, Delete malware, Take compromised system offline, Restore from clean backup, Rebuild system from scratch	External desktop	Seconds	Medium
Contractor keylogger exfiltration	Keylog capture is sent to dead drop location (located via dynamic DNS) via FTP. The contents are encrypted with a key provided in the keylogger.	Lateral movement		DNS lookup, FTP connection sending encrypted file to external server	None	Reboot, Delete malware, Take compromised system offline, Restore from clean backup, Rebuild system from scratch	External desktop	Seconds	Low
Attacker discovery of vendor portal credentials	The attacker looks through the contents from the keylogger and finds the username/password to contact the target enterprise's vendor portal.	Lateral movement		None	None	Password change, Multifactor authentication		Seconds	Low
Attacker vendor portal login	The attacker uses these credentials to login.	Lateral movement		Connection from previously unused IP address to vendor portal	IPS Log files: Behavioral evaluation of logins (e.g., typical source IP range, OS, etc.)	Password change, Multifactor authentication	Vendor Portal Web Server	Seconds	
DB Server SQL injection	The attacker determines a SQL injection vulnerability exists in the vendor portal's back end SQL database (and the SQL server is running as Administrator)	Lateral movement, privilege escalation	SQL injection	SQL injection code	HIPS, IPS	N-variant	Vendor Portal DB Server	Seconds	High
DB Server dropper installation	Attacker leverages SQL injection to instruct the portal server to install the dropper	Maintain persistence	SQL injection	SQL injection code, DNS lookup, malware download, malware installation, C2 calls	HIPS, IPS	N-variant, Reboot, Delete malware, Take compromised system offline	Vendor Portal DB Server	Seconds	Medium
DB Server exploit toolkit installation	Dropper downloads and installs exploit toolkit	Maintain persistence		DNS lookup, malware download, malware installation, C2 calls	HIPS, IPS	Reboot, Delete malware, Take compromised system offline, Restore from clean backup, Rebuild system from scratch	Vendor Portal DB Server	Seconds	Medium
DB Server password dump	The attacker uses the toolkit to run a tool like pwdump to pull the password hashes from the computer.	Privilege escalation		Pwdump execution, C2 calls	HIPS, IPS	Password change, Multifactor authentication	Vendor Portal DB Server	Seconds	Low
DB Server password exfiltration	The password hash file encrypted and sent to the FTP server as before.	Privilege escalation		FTP transfer of encrypted data, DNS lookup, C2	HIPS, IPS, DRM	Password change, Multifactor authentication	Vendor Portal DB Server	Seconds	Low
Password cracking	The attacker cracks the passwords offline using rainbow tables.	Privilege escalation		N/A	None	Password change, Multifactor authentication, Reset administrator accounts		Seconds	Low
Reverse shell from DB server	The attacker uses the exploit toolkit on the SQL server to cause a shell connection to be established outbound from the enterprise to the C2 server.	Maintain persistence		Shell executable, shell connection (C2), DNS lookup	HIPS, IPS	SDNA, Reboot, Delete malware, Take compromised system offline, Restore from clean backup, Rebuild system from scratch	Vendor Portal DB Server	Seconds	Low
nmap from DB server	From the SQL server, the attacker runs nmap to scan the local network and discovers the payment server and DRM appliance.	Reconnaisance		nmap scan, shell connection from SQL server to external C2 server	IPS	Camouflage, OS fingerprint evasion, SDNA, Reboot, Delete malware, Take compromised system offline, Restore from clean backup, Rebuild system from scratch	Vendor Portal DB Server, local network	Minutes	Low
DLP Appliance Reconfiguration	The attacker uses a cracked password to login to the Data Loss Prevention security appliance to reconfigure it to not alert on credit card data	Exfiltration		Login record, changes to DLP configuration file	HIPS	Password change, Multifactor authentication, Reset administrator accounts, Restore from clean backup, Rebuild system from scratch	DLP appliance		
Payment server login	The attacker logs in from the SQL server to the payment server using the cracked credentials (The rest of this attack is continued with the attacker communicating with the SQL server, which then communicates with the payment server.)	Lateral movement		Connection from SQL server to payment server, shell connection from SQL server to external C2 server	IPS	Password change, Multifactor authentication, Reset administrator accounts	Payment server	Seconds	Low
Payment server dropper installation	The attacker again installs the dropper on the payment server	Maintain persistence		Connection from SQL server to payment server, shell connection from SQL server to external C2 server, dropper download by payment server, dropper installation by payment server	HIPS, IPS	Reboot, Delete malware, Take compromised system offline, Restore from clean backup, Rebuild system from scratch	Payment server	Seconds	Medium
Payment server exploit toolkit installation	The dropper downloads and installs the exploit toolkit.	Maintain persistence		DNS query for C2 and download servers, exploit toolkit download, exploit toolkit installation	HIPS, IPS	Reboot, Delete malware, Take compromised system offline, Restore from clean backup, Rebuild system from scratch	Payment server	Seconds	Medium
Malicious firmware hosting	The attacker leverages the payment server's trust with POS devices to host a malicious firmware update for the devices to install.	Mission		Connection from SQL server to payment server, shell connection from SQL server to external C2 server, malicious firmware update download from external FTP download server to payment server	IPS	Reboot, Delete malware, Take compromised system offline, Restore from clean backup, Rebuild system from scratch	Payment server	Seconds	Medium
POS exfiltration	The POS devices send credit card information for exfiltration to the payment server.	Exfiltration		Credit card data (with additional information that should not be stored) sent from POS to payment server	IPS, DRM	Reboot, Delete malware, Take compromised system offline, Restore from clean backup, Rebuild system from scratch	POS	Seconds	Low
Payment server exfiltration	The payment server bundles the credit card information, encrypts it and sends it embedded in ICMP packets to another server (again, dynamic DNS is used to keep this server's name and IP address in motion).	Exfiltration		Credit card data (with additional information that should not be stored) stored on payment server, credit card data sent via ICMP, DNS lookups	IPS, DRM	Reboot, Delete malware, Take compromised system offline, Restore from clean backup, Rebuild system from scratch	N/A	Seconds	Low

Table 1: Attack Scenario Steps

Name	Locale	Startup Time	Cost	Observability	Observability Time	Effectiveness	Impact	Applicability	Trigger	Timing
Camouflage	Network	Seconds	Medium	Medium	Non-Instantaneous	Medium	Slows/confuses attacker. Makes attacker more visible	Attacker wishes to locate a host/service to compromise	Attack suspected or expected. Scan detection.	Before and After
SDNA	Host OS	Seconds	Medium	Medium?	Instantaneous	Medium	Slows/confuses attacker. Makes attacker more visible	Attacker wishes to locate a host/service to compromise	Attack suspected or expected. Scan detection.	Before and After
Fingerprint evasion	Host OS	Seconds	Low	Medium	Non-Instantaneous	Low	Slows/confuses attacker. Makes attacker more visible	Attacker wishes to tailor attack	Network scan detected	Before
Randomized instruction set	Host Hardware	N/A	High	Medium	N/A	High	Prevents attack	Attacker is using a host-based exploit (not configuration)	Attack expected	Before
Reboot	Host OS	Seconds	Low	High	Instantatneous	High	Remove memory-only malware	Malware resides only in memory	In-memory malware on system	After
ASLR	Host OS	N/A	Low	Medium	Instantaneous	High	Prevents attack	Attacker must desire to write to a particular location in host memory	Attack suspected	Before
N-variant application	Host	Seconds	Medium?	Medium	Non-Instantaneous	Medium	Sometimes prevents attack. Slows/confuses attacker. Makes attacker more visible	Defender has multiple different versions of a system	Attack expected	Before
Reset administrator accounts	Host	Seconds	Low	High	Instantaneous	High	Old admin account credentials are invalid	Attacker must desire to write to a particular location in host memory	Administrator account credentials are compromised	After
Rebuild user accounts	Host	Minutes	Low	High	Instantaneous	High	User accounts are free of malware	Attacker must desire to write to a particular location in host memory	User account is compromised	After
Take compromised systems offline	Host	Seconds	Low	High	Instantaneous	High	Compromised systems are disabled	A system is compromised and can be taken offline	Malware on system	After
Delete malware	Host	Minutes	Low	High	Instantaneous	High	System is free of malware	Malware is located on a system	Malware on system	After
Disable breached accounts	Host	Seconds	Low	High	Instantaneous	High	Breached accounts no longer function	An account has been determined to have been breached	Account is breached	After
Mitigate vulnerabilities	Host	Minutes / Hours	Low / Medium	High	Instantaneous	High	Vulnerabilities are no longer exploitable	A known vulnerability exists on a system	Vulnerability is detected	Before and After
Restore from clean backup	Host	Minutes / Hours	Low / Medium	High	Instantaneous	High	System is free of malware	A clean backup exists	Malware on system	After
Rebuild system from scratch	Host	Minutes / Hours	Low / Medium	High	Instantaneous	High	System is free of malware		Malware on system	After
Replace compromised files with clean versions	Host	Seconds / Minutes / Hours	Low	High	Instantaneous	High	File integrity restored to older version	Clean backups of the compromised files exist	File integrity is compromised	After
Install patches	Host	Minutes	Low	High	Instantaneous	High	Vulnerabilities are no longer exploitable	A known vulnerability exists on a system, and a patch is available	Vulnerability is detected; patch is available	Before and After
Change passwords	Host	Seconds	Low	High	Instantaneous	High	Compromised credentials are no longer useful	Passwords are known to be compromised	Password is compromised (including account is breached)	After
Change network security architecture	Network	Seconds / Minutes / Hours	Low / Medium / High	High	Instantaneous	High	Attacker path through network may be more difficult			After
Increase logging	Network/Host	Seconds	Low / Medium	Low	Non-Instantaneous	Low	Additional opportunities to observe attacker		Attack suspected or expected	Before and After
Multifactor authentication	Network	Seconds	Medium	High	Instantaneous	High	Prevents attack when attacker only has password	Attacker has password but not multifactor device	Passwords are compromised	Before and After

Table 2: Defense Tactics

Host	Business Criticality	Time Criticality	Purpose
Vendor web portal	Critical	Days	Vendor services
Vendor portal DB	Critical	Days	Vendor services
PoS Terminal	Core	Seconds	Payment
Payment Server	Core	Hours	Payment
DLP	Core	Milliseconds	Security
Directory Server	Core	Milliseconds	Vendor services, Payment

Table 3: Network Information

The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.