Retrospectively Using Multilayer Deception in Depth Against Advanced Persistent Threats

Jason A. Landsborough
Naval Information Warfare Center Pacific
jason.a.landsborough.civ@us.navy.mil

Thuy D. Nguyen
Naval Postgraduate School
tdnguyen@nps.edu

Neil C. Rowe
Naval Postgraduate School
ncrowe@nps.edu

Abstract

Defensive cyber deception is useful in both the information and cognitive domains of warfare. Such deception works better when it is multilayer as a defense-in-depth strategy. We developed a tool to analyze the offensive tactics in the MITRE ATT&CK Enterprise framework that were popular with sixteen Advanced Persistent Threat (APT) groups, and identified deceptive defense methods that can counter each technique. With this knowledge defenders can make more informed decisions while planning the deception to use in different layers. We use as examples three recent high-profile APT events, and review how well the deception methods could interfere with them.

Keywords: multilayer defense, deception in depth, advanced persistent threat, MITRE ATT&CK, APT analysis

This paper appeared in the Proc. Hawaii Intl. Conf. on System Sciences, Hawaii, HI, January 2024.

1. Introduction

Defenders of computer systems and networks must defend all systems within an organization from any attack, although those systems are porous with both known and unknown vulnerabilities. An attacker needs just one way inside a system or network, the easier the better, so often it is through the user (Schneier, 2012). Although the median dwell time for attackers has decreased over time as defenders get better tools, most compromise notifications come from outside their organizations (Mandiant, 2022b). Many security tools concentrate on defending the perimeter with little focus on the security of the internal systems, allowing attackers to roam around an organization‘s network for months before detection (Newman, 2020; Sophos, 2023), as with sophisticated attackers or Advanced Persistent Threat (APT) groups. These groups are well resourced, well organized, and are typically associated with nation-states or large criminal organizations (Cole, 2012). Activities of APT groups are often undetected for a long time because they evade detection by blending in with existing behavior such as using tools that already exist on a system (“living off the land”).

APTs do everything possible to achieve their goals. That means they try multiple attack options if they encounter obstacles in defenses. Cyber defense is difficult and it will be difficult for defenders to thwart all attack options. For instance, many ways exist to gain a foothold on a system by exploiting vulnerabilities, not all of them public knowledge. For this reason, it is desirable for defense against APTs to be multilayer, following the military principle of defense-in-depth, which was adapted for computer systems by the National Security Agency and is recommended by the National Institute of Standards and Technology (NIST, 2020; NSA, 2009). Multilayer means that once the attacker has breached one line of defense, they encounter a new line of defense. Multilayer defenses require careful planning because they must use different methods at later layers so attackers cannot use what they have already learned about how to quickly circumvent similar obstacles.

Among possible active defenses, deception is particularly useful since it is often unexpected. Deception can influence someone to have a false belief (Rowe and Rrushi, 2016). This can aid defenders in reducing the asymmetric advantage attackers have over defenders (Ferguson-Walter et al., 2019; Shade et al., 2020) by acting as an early detection system (Rowe and Rrushi, 2016), that wastes the attacker’s time (Ferguson-Walter et al., 2018; Landsborough et al., 2021; Michael and Riehle, 2001) or provides freedom of maneuver (Ferguson-Walter et al., 2019) for defenders. Deception can also reduce the number of false positives in identifying attacks (Chacon et al., 2020), which helps defenders.

The possibilities for defensive cyber deception are many, so consideration of priorities is important. To help with prioritization, we created an APT analysis tool that inspects MITRE’s APT group data (MITRE,

2023) and visualizes attack techniques used by different APT groups. It displays all techniques associated with each tactic to reveal the most common techniques for a given tactic. With this information we identify deception methods that can be used against the most common techniques and develop a multilayered deception plan.

2. Threat scenarios

During an attack, APT groups use different techniques at different stages of their campaign. The MITRE ATT&CK framework (Strom et al., 2018) (https://attack.MITRE.org/), calls these stages *tactics*. Table 1 shows the offensive tactics and adversarial objectives in the recent version 13 of the Enterprise ATT&CK framework; ATT&CK identifiers are in parentheses. Each tactic has several associated techniques, some of which are shared by other tactics. As examples, we now discuss the tactics, techniques, and procedures (TTPs) used in three real-world attacks.

2.1. Motivating example 1: 2020 U.S. government data breach

A frequent goal for an attack is data exfiltration. A high-profile example was the 2020 US Federal Government data breach attributed to the Russian group APT29 (Mandiant, 2022a). This group exploited vulnerabilities in products by three vendors with many customers: Microsoft, SolarWinds, and VMware. The attack was undetected for months and was first disclosed in December 2020 when FireEye discovered that an APT had targeted them to steal their red-team tools (Newman, 2020). Shortly thereafter, it was reported that attackers had been monitoring internal email traffic at the U.S. Treasury and Commerce departments (Bing, 2020).

The most significant aspect of this breach was the supply-chain compromise of the SolarWinds Orion security platform. The foothold in SolarWinds' network occurred on or before September 2019 (Cimpanu, 2021), and an initial proof of concept modification to the Orion software was made on October 2019. The APT actors then set up their command and control (C2) infrastructure from December 2019 to February 2020 (Kovacs, 2020). In March 2020, the first Trojan updates to SolarWinds Orion occurred, which were included in 18,000 downloads of the compromised version according to SolarWinds (Stubbs et al., 2020).

Cloudflare identified the first contact of infections with remote C2 to avsvmcloud.com (and its subdomains) in April 2020 (Tadeusz and Kipp, 2020). After gaining access on the victim system, the attacker would install frequently used exploit tools such as the Cobalt Strike Beacon (Mandiant, 2020) to execute payloads. Attackers successfully stole credentials and established persistence even after the compromised Orion software was disabled (Stubbs et al., 2020).

Many organizations using the Orion software were attacked, but not all successfully. For example, Microsoft found traces of the malicious code in their systems and alerted Crowdstrike that an attacker might be trying to access the latter's email system. However since CrowdStrike did not use Office 365 email software and their email systems were not affected (Vavra, 2020).

Table 1. Adversarial tactics and objectives

Tactic	Adversary's Objective
Reconnaissance (TA0043)	Gather information to plan future operations.
Resource Development (TA0042)	Establish resources to support operations.
Initial Access (TA0001)	Get onto victim network and drop malware.
Code Execution (TA0002)	Run malicious code.
Persistence (TA0003)	Maintain attack foothold.
Privilege Escalation (TA0004)	Gain more permissions and capabilities.
Defense Evasion (TA0005)	Take additional measures to avoid being detected.
Credential Access (TA0006)	Steal account names and passwords.
Discovery (TA0007)	Map out victim environment.
Lateral Movement (TA0008)	Attack other vulnerable systems in victim environment.
Collection (TA0009)	Gather data of interest to the mission.
Command and Control (TA0011)	Control compromised systems using covert communication channels.
Exfiltration (TA0010)	Steal data.
Impact (TA0040)	Manipulate, interrupt, or destroy target systems and data.

The information stolen in this breach could be exploited to access high-value assets for years to come. In June 2021, Google's Threat Analysis Group disclosed that APT29 had used an iOS zero-day vulnerability to target and steal credentials for government employees on LinkedIn (Goodin, 2021).

Seventeen TTPs described in the MITRE ATT&CK framework (MITRE, 2022c; Strom et al., 2018) were used in the SolarWinds breach (Ozarslan, 2020).

2.2. Motivating example 2: Snake espionage implant

A Joint Cybersecurity Advisory published by The U.S. Cybersecurity and Infrastructure Security Agency (CISA) describes an espionage implant tool used by the Russian Federal Security Service (FSB) (CISA, 2023b). The malware, named "Snake" has been used and maintained by Russia for decades (the original version developed in 2003 was known as "Uroburos") and variants have been found in over 50 countries, including within Russia. According to the advisory, U.S. targets included "education, small businesses, and media organizations, as well as critical infrastructure sectors including government facilities, financial services, critical manufacturing, and communications" (CISA, 2023b).

Snake is typically deployed to externally facing machines for communication among infected machines. Once inside the network, malware operators use other methods such as network enumeration and lateral movement. Unless necessary, Snake's operators did not deploy additional tools, but relied on tools that exist in the internal network as a form of "living off the land" attack. According to the CISA report, 40 techniques were used by Snake to carry out its mission (CISA, 2023b).

Despite the sophistication of Snake, its developers and operators made several mistakes which helped defenders analyze it. For example, they used a small bit length for Diffie-Hellman key exchange which helped in cracking it. Some instances of Snake were also deployed without stripping the binary of identifiers useful in reverse engineering.

2.3. Motivating example 3: Ransomware attacks

Ransomware is a common type of cyber attack on a broad range of targets. It is a type of malware that encrypts files on a target's machine and demands the victims (organizations or individuals) to pay a ransom to obtain the decryption key or keys. Some attacks also exfiltrate data which the attacker threatens to disclose or sell if the ransom is not paid. A study found that in 2021, 74% of ransomware groups were affiliated with Russian organizations (Chainalysis, 2022). Wipers and ransomware have been used against Ukraine (ESET, 2023) recently, but it remains to be seen whether these are government-sponsored or hacktivists. Most ransomware is not affiliated with governments but is used by criminal organizations for extortion (Crowdstrike, 2022).

North Korea actively uses ransomware. In 2022, CISA released a cybersecurity advisory about the Maui ransomware used by North Korea that targeted healthcare and public health organizations (CISA, 2022) with activity dating back to 2017 (CISA, 2023c). Several TTPs relating to manual execution and data exfiltration were observed with these ransomware campaigns. In 2023, CISA released another advisory about new North Korean ransomware attacks (CISA, 2023a). Iranian state attackers known as "HomeLand Justice" also have used ransomware and wipers while exfiltrating data (FBI and CISA, 2022).

3. Common techniques of APTs

We created a tool in Python to analyze the JavaScript Object Notation (JSON) data in MITRE's APT dataset (https://attack.mitre.org/groups/) and identify the most frequent techniques. Our tool uses programming interfaces from the mitreattack-python project (https://github.com/mitre-attack/mitreattack-python) to extract Structured Threat Information Expression (STIX) data from the MITRE ATT&CK Enterprise matrix, which lets us map the TTP ID to its name, something not included in the JSON data. We also extract data of the APT groups and the techniques used from the JSON data. We organize the techniques by tactic and frequency of use to highlight the most common techniques used by the groups. We also graph using Pyvis the connections between the APT groups to show common techniques (Figure 1). This graph approach works best with a small number of APT groups; the visualization of a graph of a large number of APTs will be unintelligible.

To focus our research, we studied the 16 APT groups identified by numeric values such as APT29 and APT41 since these groups are well documented by MITRE. To limit our scope, we only considered the two most common techniques for a given tactic as summarized in Table 2, though some techniques may be belong multiple tactics.

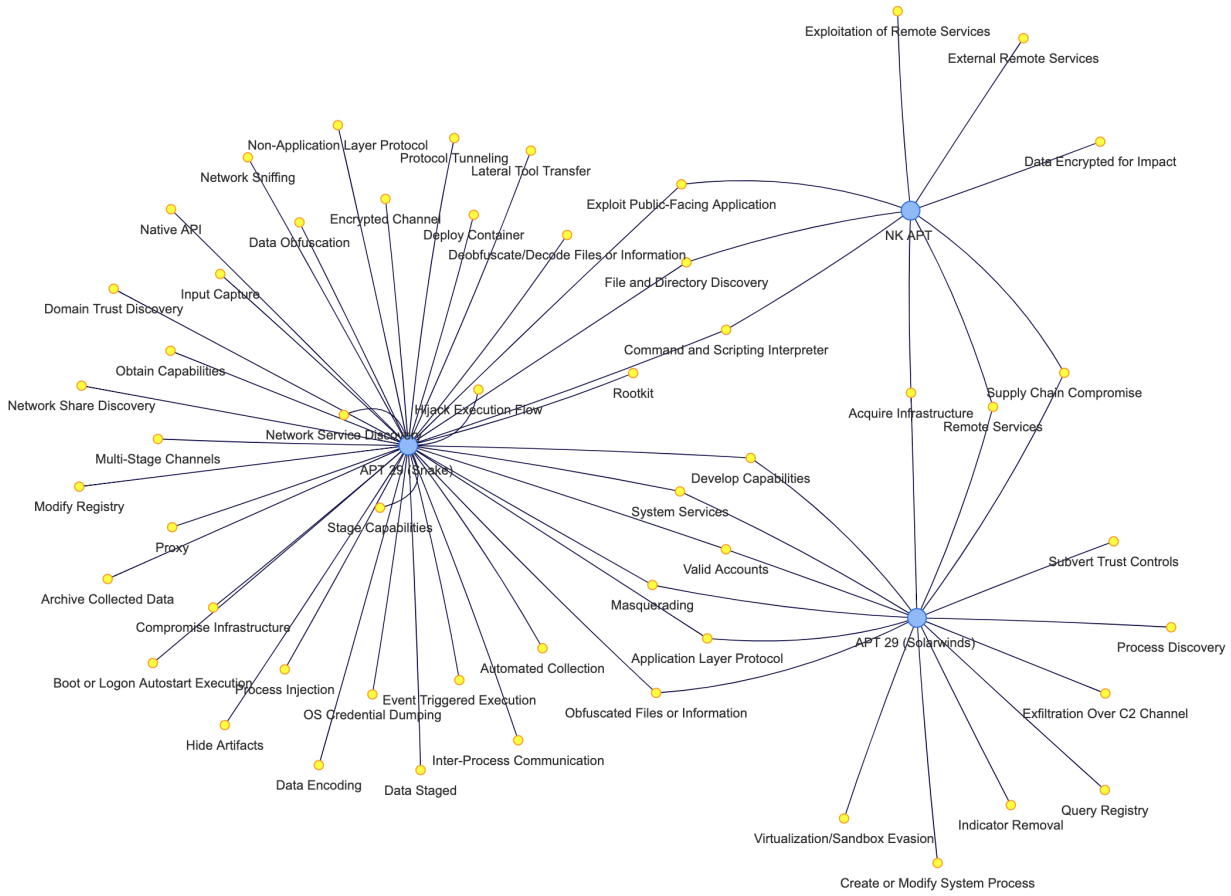

Figure 1. Common TTPs among three APT groups

3.1. Deception methods for enhancing defenses

Common tools in the defender's toolkit such as network monitoring, disk encryption, antivirus software, intrusion-detection systems, and firewalls have limitations. Network-monitoring tools often require much time by personnel. Encrypted disks primarily protect data not in use. Antivirus software and intrusion-detection systems are easily bypassed by changing data slightly. Attackers can still use protocols allowed by the firewall. Deception is unexpected on digital systems, and can give defenders additional time and space to engage the attacker by additional freedom of maneuver (Ferguson-Walter et al., 2019).

Table 2 shows that different methods of deception can interfere with an attacker's techniques, many of which can slow or stop an attacker. Most can be instrumented to provide defenders with better situational awareness. For example, network decoys can affect the reconnaissance phase of an attack. Fake shells or interpreters can impede intelligence collection during execution by filtering out important information or introducing deceptive content. Deceptive man-in-the-middle capabilities can interfere with command-and-control communications and data exfiltration by disrupting the flow of data, such as indicating nonexistent network congestion or failures, or introducing fake information. Honeytokens, or fake objects, can interfere with collection and data exfiltration while also offering a high-confidence tripwire alert when they are accessed (Shabtai et al., 2016).

MITRE has two matrices for defenders, D3FEND (MITRE, 2022a) and ENGAGE (MITRE, 2022b). D3FEND has more traditional passive defensive methods, whereas ENGAGE has more active methods of adversary engagement. For deception, ENGAGE includes lures (EAC0005) whereas D3FEND version 0.12.0-BETA-2 has 11 techniques under the "deceive" tactic (d3f:Deceive). It can be difficult for defenders to determine how to best use the defensive information

Table 2. Common TTPs used in MITRE APT data set (MITRE, 2023) and possible deception methods that can be used for each

Tactic	Technique	Deception Method
Reconnaissance	Gather Victim Identity Information (T1589)	Fake personas
	Active Scanning (T1595)	Network decoys to catch scanning attempts
Resource Development	Obtain Capabilities (T1588)	Network decoys running vulnerable services that do not exist
	Acquire Infrastructure (T1583)	Network decoys running vulnerable services that do not exist
Initial Access	Phishing (T1566)	Fake personas
	Valid Accounts (T1078)	Honeyuser accounts
Execution	Command and Scripting Interpreter (T1059)	Fake shell or interpreter
	User Execution (T1204)	Honeypot with fake automated user
Persistence	Boot or Logon Autostart Execution (T1547)	Fake service creation/viewing tools
	Scheduled Task/Job (T1053)	Fake job creation/viewing tools
Privilege Escalation	Boot or Logon Autostart Execution (T1547)	Fake service creation/viewing tools
	Scheduled Task/Job (T1053)	Fake job creation/viewing tools
Defense Evasion	Obfuscated Files or Information (T1027)	Compression or file handling tool that fails indicating success or filters information
	Indicator Removal (T1070)	Tools to move logs instead of delete, but stating otherwise
Credential Access	OS Credential Dumping (T1003)	Tools to filter out valid credentials or add fake credentials
	Brute Force (T1110)	Honeyusers
Discovery	File and Directory Discovery (T1083)	Tools to filer out valid files and directories
	System Information Discovery (T1082)	Tools to report fake or inconsistent system information
Lateral Movement	Remote Services (T1021)	Tools reporting fake information or concealing information
	Use Alternate Authentication Material (T1550)	Fake access tokens
Collection	Data from Local System (T1005)	Tools to filter out valid files and directories
	Archive Collected Data (T1560)	Tools to replace files with tracking honeytokens when used for compressed or encrypted archive creation
Command and Control	Application Layer Protocol (T1071)	Server in the middle rewriting packet payloads
	Ingress Tool Transfer (T1105)	Tools that fail to complete a download but indicate success
Exfiltration	Exfiltration Over Alternative Protocol (T1048)	Server in the middle rewriting packet payloads
	Exfiltration Over C2 Channel (T1041)	Server in the middle
Impact	Disk Wipe (T1561)	Tool seems to indicate success but fails
	Data Encrypted for Impact (T1486)	Tool moves files to a safe hidden location before encryption

in ENGAGE, D3FEND, and similar databases. Our tool offers a way to help summarize the ATT&CK information into a set of actionable defensive methods to counter the common attacking techniques.

3.2. Deception coverage of techniques in motivating scenarios

We used JSON representations of the techniques in sections 2.1 to 2.3 with our tool to analyze the motivating scenarios. We wanted to see how well the deception methods in Table 2 could retrospectively apply to techniques used in the three motivating-example attacks. Comparing to the set of 28 most common techniques across all major APT groups (Table 2), we found 7 of the 16 techniques used by APT 29 in the Solarwinds campaign were among the 28 most common. For the North Korean APT group, 4 of the 9 techniques were commonly used techniques; and for Snake, 9 of 40.

Deception capabilities such as those in Table 2 can interfere with techniques of APTs. This is easiest when techniques are frequently seen because then deceptions can be preplanned for efficiency. For example, if network decoys present allegedly vulnerable but nonexistent services, it could encourage an attacker to try to handle these services with new infrastructure, wasting their time and money. A fake shell or interpreter could interfere with the attacker after their initial exploitation of our system, learning their goals while also delaying, deterring, or denying them forward progress. Fake user accounts could be used as a high-confidence tripwire to impede an adversary's initial access or lateral movement within the environment. These methods are useful against the techniques in the three APT examples.

3.3. Multilayer defense planning

A multilayer approach for deception provides "deception in depth." This can be made more effective if each layer uses other active defense methods as well as deceptions. Active defense methods include ongoing modification of systems, such as a moving target with occasional random modification of Internet Protocol (IP) addresses (Dunlop et al., 2011), as well as automated tracking of attacks and attackers. Most attackers are familiar with such nondeceptive active defenses and can figure them out quickly, though they still will be impeded. However, deliberate defensive deceptions of a wide range of methods are available (Rowe and Rrushi, 2016) that are hard for attackers to anticipate. Different deceptions can be deployed at different stages of an APT attack. Table 2 shows

example deceptions for different stages.

Deceptions that are used early against an APT can affect what deceptions should be used later. Tracking deceptions used is important as defenders increase their use of defensive autonomous agents to stage complex interactions. Tracking is aided by using a data structure to record deceptions used and their effects. Data can include attacker action, attacker phase, deception layer, deception location, timestamp, and deceptive actions. Figure 2 shows an example scenario in which network masking is used to fool an attacker. The attacker scanned a machine for open ports but the real target machine is only running a Web server. During the port scan a deceptive router modified and routed packets destined for port 22 (Secure Shell) on the Web server instead to a honeypot, and also rewrote the response packets to trick the attacker into thinking that the machine is running a service that it is not. Other traffic was routed normally. In this scenario, the attacker was told that the target machine was also running the vulnerable Secure Shell (SSH) application when it was not.

With deception tracking enabled, this port-scanning attempt would be recorded (as in the JSON format in Figure 3). This could lead to enabling a policy for system tools on the Web server to indicate that SSH is running if an attacker can later access the system and examine running processes. Another policy could specify allowing deceptive system tools on the honeypot to modify networking information like IP addresses to mislead attackers. For example, the system tools could report a fake address of the Web server when queried. These policies could be limited to specific systems or durations.

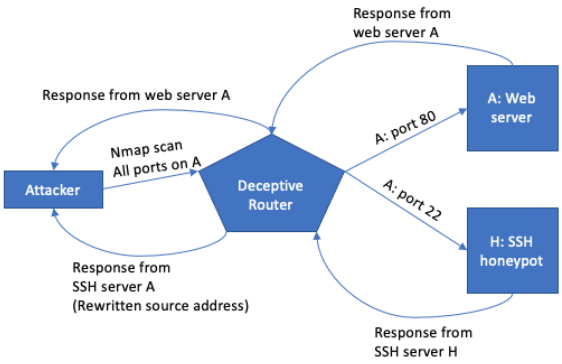

Figure 2. Network masking of system

Defenders can choose deception techniques appropriate for different stages of an attack. The basic layers for defense are network, system, and data, and they can be managed separately for subnetworks and separate volumes of storage. At the network layer, we

```
{
  "Deception Events": {
    "Event": [
      {
        "layer": "Data",
        "location": "Decoy-ftp",
        "timestamp": "Tue Mar 14
           22:37:21 EDT 2023",
        "action-type":
           "served-file"
        "actions":
           ["network-topology.png"]
      },
      {
        "layer": "Network",
        "location": "Deceptive
           Router",
        "timestamp": "Tue Mar 14
           22:40:34 EDT 2023",
        "action-type":
        "modified-packets",
        "actions": ["Modified
           and routed packets for
           A on port 22 to H",
        "Modified response packets
           from H on port 22 to
           appear from A"]
      }
    ]
  }
}
```

Figure 3. Deception tracking format

may deploy network decoys for reconnaissance and to influence resource development. Decoys can appear to run services like those of real machines, and can work as early-warning systems. At the system layer, we can use fake system tools to interfere with the tactics identified by ATT&CK such as code execution, persistence, privilege escalation, defense evasion, credential access, discovery, lateral movement, collection, command and control, exfiltration, or impact. An example fake system tool could be a modified shell program that modifies display output such as processes running, or lists nonexistent files and directories. At the data layer, we can use data collected about fake users as a form of tripwire for initial access and to identify any future credential access.

These methods are represented as hexagon nodes in a generic data exfiltration scenario shown in Figure 4. With the three methods (honeyusers, network decoys, and fake tools), there are four opportunities to interfere with the attacker during their attack.

3.4. Deception opportunities for the motivating scenarios

In the 2020 data breach and North Korean ransomware examples, network decoys running services

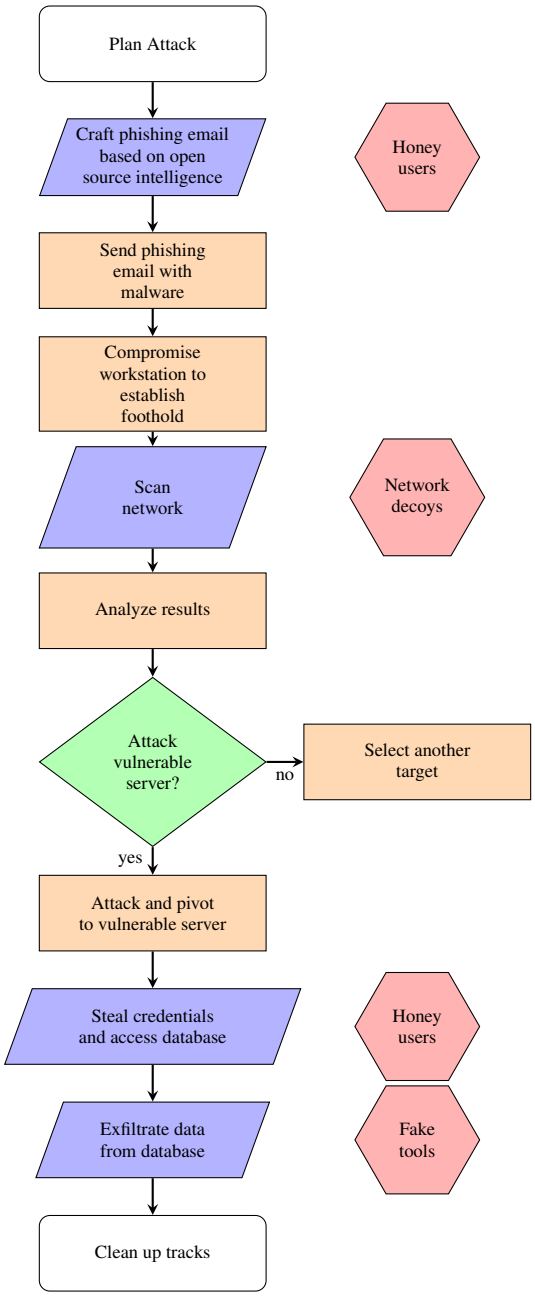

Figure 4. Generic data exfiltration scenario with deception

that mirrored the services on real machines could have alerted when an attacker attempted to use a remote service such as the Remote Desktop Protocol (RDP). Similarly, the decoys could have run a public-facing service that appeared to be vulnerable to attract attackers, which could have been enticing for operators of the Snake malware or the North Korean attackers. Advanced attackers may prefer easier targets like RDP with many known exploits rather than exposing their own tool.

A fake shell could protect against techniques in all three groups. It could have interfered with Snake operators or those involved in the 2020 data breach by disrupting the attacker's ability to create or modify system services. A fake shell could also protect against ransomware by identifying attempted accesses to fake files and directories while protecting real files. A fake user could have confused the Snake and Solarwinds attackers since both used valid accounts during their attack. Fake users could store files with additional fake data to waste an attacker's time.

Deception capabilities have limitations and weaknesses like all defensive tools. Nonetheless, they could alert defenders sooner of a breach and allow incident responders to deter, delay, or deny attacker progress, wasting attacker resources. These can all be advantages for defenders.

3.5. Coordinating deception layers with global variables

A challenge for multilayered deceptions is that the chances of detection by the attacker of the later defensive deceptions is higher. That is because once someone recognizes they have been deceived, they are on alert and more likely to detect other deceptions, and the subsequent deceptions will likely be ineffective. This effect can be reduced if deceptions are sufficiently different from one another that analogous situations are reduced. However, it requires careful planning because each layer has preferable deceptions and makes assumptions of system properties that could be invalidated due to inconsistencies when combining with other layers. Note that deception planning is still useful against automated attacks because when attacks fail due to deception, a human often examines the data to figure out what happened, and can modify the attack plan accordingly.

Connections between deceptions at different layers can be modeled by the conditional probability of a particular deception succeeding given some feature of the attacker or circumstances. Deception planning would thus seem to require estimating many such conditional probabilities. However, most of these can be derived from a few basic parameters starting with psychological theories. For instance, some attackers are naturally more suspicious than others, and will overreact to deceptions in a consistent way. Other attackers are more alert than others, and will be more able to notice deceptions like false error message. Yet other attackers are more inclined to be proactive in exploring a system, and will be inclined to search a

system for confirmation of a false error message. We have identified as key global variables the degrees of suspiciousness, alertness, skill, adaptability, patience, maliciousness, and reliability of the target system; nearly all other useful probabilities can be derived from these. We are exploring building decision trees for the possible attacker responses to each deception, estimate probabilities, costs, and benefits for each branch, and estimate the expected benefits of each possible deception as the APT proceeds through its phases.

As an example, consider an attacker trying to download a rootkit onto a target system after gaining access to an administrator account, where the attacker has been already been detected by their anomalous traffic. One way to interfere with the download is to give a false error message that the download failed. Live attackers may be discouraged then and if they are low on the "patience" and "adaptability" measures, they may not notice the error message and waste time trying the download repeatedly. Another possible response is to instead delete the download immediately after it has arrived without telling the attacker. They will try to unpack it and discover it is missing, then probably try to download a few more times. This will waste their time to a degree that is determined by the "patience" global variable. However, attackers with a high "suspiciousness" measure may find it odd that no error message was given, and this will further increase their suspiciousness and decrease their likelihood of being fooled by further similar deceptions. Of the two options, it will be better to give a false error message while failing to download. Then the "alertness" measure determines whether they notice the error message, and the "skill" measure determines whether they will try other means of downloading. Using decision trees, we can estimate these probabilities and combine them with costs and benefits to determine the best set of tactics to use. The global variables can be updated based on attacker behavior, and can be learned to be associated with times of day, ranges of network addresses, and favored methods to provide a simple form of attacker recognition.

4. Future work

This is ongoing work in deception planning. A useful feature that could be added to our APT analysis tool is the ability to filter the APT data for groups that a target organization would care about. For example, defenders working for a bank may only care about groups that threaten the financial industry, and defenders in a power plant may only care about threats to industrial control systems. The MITRE data currently does not have this information, but other information is available online about APT targets, goals, and countries of origin.

We manually identified deception methods that could be deployed to impede or counter common APT techniques. This could be automated to suggest the deception methods to use for a given set of techniques and prioritize them. Incorporating costs and an expected probability of success to plan defensive options, such as using decision trees, may enable a defender to plan responses intelligently.

Cyber deception has been validated in some experiments. For example, network decoys have been found effective against red-team participants even when they told deception might be used against them (Ferguson-Walter et al., 2021). Incorporating real machines in a deception strategy to make them look like honeypots seems promising (Rowe et al., 2007). Making real assets look fake and fake assets look real is also known as two-sided deception (Miah et al., 2020). Using a simulator with human participants to test one and two-sided deception showed participants scored better without deception than with it (Aggarwal et al., 2021).

5. Conclusion

Advanced persistent threats are serious cyber threats that are difficult to defend, and they require a wide range of countermeasures. We are building an APT analysis tool that can be used to design good deception methods to foil them. Real-world examples use many common techniques, so deception methods designed for APTs should start in planning defenses with tactics known to be effective against these common techniques. However, APTs are sufficiently challenging and persistent that they require multiple deceptions and other active measures for effective defenses. Attackers may discover they are being deceived and become much harder to deceive further. Thus interactions between the defensive measures should be studied more carefully to develop effective defensive plans.

6. Acknowledgements

The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government. This work was supported by Science, Mathematics And Research for Transformation (SMART) Scholarship for Service Program.

References

Aggarwal, P., Du, Y., Singh, K., & Gonzalez, C. (2021). Decoys in cybersecurity: An exploratory study to test the effectiveness of 2-sided deception. *CoRR*, *abs/2108.11037*. https://arxiv.org/abs/2108.11037

Bing, C. (2020). *Suspected russian hackers spied on u.s. treasury emails - sources*. Retrieved March 14, 2023, from https://www.reuters.com/world/us/suspected-russian-hackers-spied-us-treasury-emails-sources-2020-12-13/

Chacon, J., McKeown, S., & Macfarlane, R. (2020). Towards identifying human actions, intent, and severity of apt attacks applying deception techniques-an experiment. *2020 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*, 1–8.

Chainalysis. (2022). *Russian cybercriminals drive significant ransomware and cryptocurrency-based money laundering activity*. Retrieved June 10, 2023, from https://blog.chainalysis.com/reports/2022-crypto-crime-report-preview-russia-ransomware-money-laundering/

Cimpanu, C. (2021). *Third malware strain discovered in solarwinds supply chain attack*. Retrieved March 14, 2023, from https://www.zdnet.com/article/third-malware-strain-discovered-in-solarwinds-supply-chain-attack/

CISA. (2022). *North korean state-sponsored cyber actors use maui ransomware to target the healthcare and public health sector*. Retrieved March 10, 2023, from https://www.cisa.gov/news-events/cybersecurity-advisories/aa22-187a

CISA. (2023a). *#Stopransomware: Ransomware attacks on critical infrastructure fund dprk malicious cyber activities*. Retrieved March 10, 2023, from https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-040a

CISA. (2023b). *Hunting russian intelligence "snake" malware*. Retrieved May 9, 2023, from https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-129a

CISA. (2023c). *North korea cyber threat overview and advisories*. Retrieved March 10, 2023, from https://www.cisa.gov/northkorea

Cole, E. (2012). *Advanced persistent threat: Understanding the danger and how to protect your organization*. Newnes.

Crowdstrike. (2022). 2022 global threat report.

Dunlop, M., Groat, S., Urbanski, W., Marchany, R., & Tront, J. (2011). Mt6d: A moving target ipv6 defense. *2011-MILCOM 2011 Military Communications Conference*, 1321–1326.

ESET. (2023). *Eset research: Russian apt groups, including sandworm, continue their attacks against ukraine with wipers and ransomware*. Retrieved March 13, 2023, from https://www.eset.com/int/about/newsroom/press-releases/research/eset-research-russian-apt-groups-including-sandworm-continue-their-attacks-against-ukraine-with-wipe/

FBI & CISA. (2022). *Iranian state actors conduct cyber operations against the government of albania*. Retrieved June 10, 2023, from https://www.cisa.gov/news-events/cybersecurity-advisories/aa22-187a

Ferguson-Walter, K., Fugate, S., Mauger, J., & Major, M. (2019). Game theory for adaptive defensive cyber deception. *Proceedings of the 6th Annual Symposium on Hot Topics in the Science of Security*, 1–8.

Ferguson-Walter, K., Major, M., Johnson, C., & Muhleman, D. H. (2021). Examining the efficacy of decoy-based and psychological cyber deception. *30th USENIX Security Symposium (USENIX Security 21)*, 1127–1144. https://www.usenix.org/conference/usenixsecurity21/presentation/ferguson-walter

Ferguson-Walter, K., Shade, T., Rogers, A., Trumbo, M. C. S., Nauer, K. S., Divis, K. M., Jones, A., Combs, A., & Abbott, R. G. (2018). *The tularosa study: An experimental design and implementation to quantify the effectiveness of cyber deception.* (tech. rep.). Sandia National Lab.(SNL-NM), Albuquerque, NM (United States).

Goodin, D. (2021). *Ios zero-day let solarwinds hackers compromise fully updated iphones*. Retrieved April 25, 2023, from https://arstechnica.com/gadgets/2021/07/solarwinds-hackers-used-an-ios-0-day-to-steal-google-and-microsoft-credentials/

Kovacs, E. (2020). *Solarwinds likely hacked at least one year before breach discovery*. Retrieved March 14, 2023, from https://www.securityweek.com/solarwinds-likely-hacked-least-one-year-breach-discovery/

Landsborough, J., Carpenter, L., Coronado, B., Fugate, S., Ferguson-Walter, K., & Van Bruggen, D. (2021). Towards self-adaptive cyber deception for defense. *HICSS*, 1–10.

Mandiant. (2020). *Highly evasive attacker leverages solarwinds supply chain to compromise multiple global victims with sunburst backdoor*. Retrieved June 10, 2023, from https://www.mandiant.com/resources/blog/evasive-attacker-leverages-solarwinds-supply-chain-compromises-with-sunburst-backdoor

Mandiant. (2022a). *Assembling the russian nesting doll: Unc2452 merged into apt29*. Retrieved April 24, 2023, from https://www.mandiant.com/resources/blog/unc2452-merged-into-apt29

Mandiant. (2022b). *M-trends 2022 report*. https://mandiant.widen.net/s/bjhnhps2mt/m-trends-2022-report

Miah, M. S., Gutierrez, M., Veliz, O., Thakoor, O., & Kiekintveld, C. (2020). Concealing cyber-decoys using two-sided feature deception games. *HICSS*, 1–10.

Michael, J. B., & Riehle, R. (2001). Intelligent software decoys. *Engineering Automation for Reliable Software-Interim Progress Report (10/01/2000-9/30/2001)*, 80.

MITRE. (2022a). *D3fend*. Retrieved June 12, 2023, from https://d3fend.mitre.org/

MITRE. (2022b). *Engage*. Retrieved February 6, 2023, from https://d3fend.mitre.org/

MITRE. (2022c). *Enterprise tactics*. Retrieved February 6, 2023, from https://attack.MITRE.org/tactics/enterprise/

MITRE. (2023). *Groups*. Retrieved April 28, 2023, from https://attack.mitre.org/groups/

Newman, L. H. (2020). *Russia's fireeye hack is a statement—but not a catastrophe*. Retrieved April 24, 2023, from https://www.wired.com/story/russia-fireeye-hack-statement-not-catastrophe/

NIST. (2020). *Security and privacy controls for information systems and organizations* (tech. rep. NIST Special Publication (SP) 800-53r5). Gaithersburg, MD. https://doi.org/10.6028/NIST.SP.800-53r5

NSA. (2009). Defense in depth. https://web.archive.org/web/20121002051613/https://www.nsa.gov/ia/_files/support/defenseindepth.pdf

Ozarslan, S. (2020). *Tactics, techniques, and procedures (ttps) used in the solarwinds breach*. Retrieved April 7, 2023, from https://www.picussecurity.com/resource/blog/ttps-used-in-the-solarwinds-breach

Rowe, N. C., Custy, E. J., & Duong, B. T. (2007). Defending cyberspace with fake honeypots. *J. Comput.*, *2*(2), 25–36.

Rowe, N. C., & Rrushi, J. (2016). *Introduction to cyberdeception*. Springer.

Schneier, B. (2012). *Forever-day bugs*. Retrieved February 3, 2023, from https://www.schneier.com/blog/archives/2012/04/forever-day_bug.html

Shabtai, A., Bercovitch, M., Rokach, L., Gal, Y., Elovici, Y., & Shmueli, E. (2016). Behavioral study of users when interacting with active honeytokens. *ACM Transactions on Information and System Security (TISSEC)*, *18*(3), 1–21.

Shade, T., Rogers, A., Ferguson-Walter, K., Elsen, S. B., Fayette, D., & Heckman, K. E. (2020). The moonraker study: An experimental evaluation of host-based deception. *HICSS*, 1–10.

Sophos. (2023). *Attacker dwell time increased by 36%, sophos' active adversary playbook 2022 reveals*. https://www.sophos.com/en-us/press/press-releases/2022/06/attacker-dwell-time-increased-by-36-percent-sophos-active-adversary-playbook-2022-reveals

Strom, B. E., Applebaum, A., Miller, D. P., Nickels, K. C., Pennington, A. G., & Thomas, C. B. (2018). Mitre att&ck: Design and philosophy. In *Technical report*. The MITRE Corporation.

Stubbs, J., Satter, R., & Menn, J. (2020). *U. s. homeland security, thousands of businesses scramble after suspected russian hack*. Retrieved April 7, 2023, from https://www.reuters.com/article/global-cyber-idUSKBN28O1Z3

Tadeusz, M. B., & Kipp, J. (2020). *Trend data on the solarwinds orion compromise*. Retrieved March 10, 2023, from https://blog.cloudflare.com/solarwinds-orion-compromise-trend-data/

Vavra, S. (2020). *Microsoft alerts crowdstrike of hackers' attempted break-in*. Retrieved April 24, 2023, from https://cyberscoop.com/crowdstrike-solarwinds-targeted-microsoft/