**Mitigating APT Lateral Movement: A Network-Based Detection and Containment Strategy**

Chinese Fo0d [Alias]  
*College of Engineering*  
*University of Yeet*Nuketown, USA  
ChineseFo0d@fakedomain.edu

Jaefae [Alias]  
*College of Engineering*  
*University of Yeet*Nuketown, USA  
Jaefae@fakedomain.edu

*Abstract*—Advanced Persistent Threats (APTs) are stealthy, sophisticated cyberattacks that infiltrate targeted networks gathering sensitive information for extended periods. APT attacks aim to remain undetected to steal sensitive information or spy on network activities and other means such as espionage, sabotage of critical systems, warfare, competition, financial gain, or political motivations. Common techniques involved with APTs consist of custom malware with obfuscation or use of software exploits, lateral movement tools to move deeper into the network, pass-the-hash, privilege escalation, credential dumping, social engineering, DDoS, Command and Control (C2), Living off the Land (LOTL)…and more. APTs are very multifaceted, but at their core, they rely on the network as their center of information and path to escalate. This research focuses on how to identify APTs and mitigate their impact with a specific focus on simulating a Lateral Movement attack that is commonly employed by APTs. The challenge lies in detecting APTs early, as they operate covertly, gradually gaining control or access over affected systems – especially when APT techniques and tools are carefully planned to avoid detecting by blending into legitimate traffic or using encrypting/hidden communication channels making them harder to find until significant damage has already been done (data theft, major interruption to operations, damage to systems, danger to human life..). How to differentiate what suspicious activity is and what is not? So, this study aims to traverse and recommend strategies to specifically combat lateral movement attack strategies that often occur in the APT kill chain. The outcomes will include recommendations on defense strategies to enhance organizational resilience against APTs. NOTE: This is only the beginning of our research.

Keywords—APT, Lateral Movement, Defense in Depth, Network Segmentation, CALDERA, Machine Learning

# **Introduction**

Advanced Persistent Threats (APTs) have emerged as a formidable challenge in the cybersecurity landscape, representing a new breed of sophisticated attacks that pose significant risks to organizations and nations. Unlike traditional cyber threats, APTs are characterized by their stealthy nature, prolonged duration, and highly targeted approach, making them particularly difficult to detect and mitigate. These attacks infiltrate networks intending to gather sensitive information over extended periods, often evading traditional security methods. Over 70% of modern security breaches have had threat actors utilizing lateral movement within their attack, with the breach being discovered oftentimes months later[12].

The prolonged duration of APTs is a deliberate strategy. [7] Attackers would move slowly and cautiously through a compromised network, minimizing detectable activity or blending in to avoid triggering security alerts. This “low and slow” approach allows them to remain undetected for months or even years, maximizing the amount of valuable data they can exfiltrate. During that time, they may expand access by moving laterally within the network to reach more valuable targets (especially as they gain more information while within).

At their core, APTs are orchestrated by well-resourced adversaries who are state-sponsored groups, highly organized criminal networks, or hacktivists with strategic objectives. Meaning they possess the funds and a team with advanced technical ability and often leverage new technologies. Especially, the means of launching zero-day attacks to persist in target systems. The motivations behind APT campaigns are diverse but typically fall into three main categories:   
  
data exfiltration, sabotage, and espionage.   
  
So, things under those categories could be disrupting critical infrastructure, warfare, financial gain, political aims, competition such as stealing trade secrets or intellectual property, or simply gaining access by placing seeds/pivots for long-term intelligence gathering.

It’s clear to see that the scope and impact APTs possess is massive. And though it is vital to develop solutions for all of its kill chain phases, we place our focus on lateral movement. Lateral movement is a critical phase in the Advanced Persistent Threat (APT) kill chain, representing techniques adversaries use to navigate and expand their foothold within a compromised network. APT resilience can be achieved via this stage, often by using legitimate tools and credentials to evade detection while escalating privileges and accessing high-value targets. Lateral movement is also referred to as internal reconnaissance as it's another way to discover further information that was not accessible from external reconnaissance to establish the initial compromise. Overall, addressing the risk of lateral movement within network security emphasizes critical network vulnerabilities. It should be noted that due to time and hardware constraints, this project will not be simulating an entire Extended Detection and Response (XDR) kill chain to address all of what APT does and will not involve using CPS – this will be for future explorations as students.

# **Related Works**

The following section will introduce major works that we learned from and are related to our goal of understanding APT lateral movement and ultimately heading towards the direction of generating a practical solution – the theoretical explorations before heading into simulating.

Prior work seen primarily delves into what APT is, what lateral movement is, their challenges, and strategies. Common strategies are detecting lateral movement activity (via machine learning, neural networks, native OS, and network logging techniques…), forensic analysis, improving network security practices/policies, dynamic game detection, and deep learning. As for containment after the initial compromise, it would follow successful detection. Containment strategies include micro-segmentation of the network, Zero Trust, Privileged Access Management (PAM), and Automated Response Systems (SOAR).

## Detecting Lateral Movement: A Systematic Survey

This first article[1], by Christos Smiliotopoulos, Georgios Kambourakis, and Constaninos Kolias (a current faculty member at the University of Idaho), explains the systematic survey on detecting lateral movement and provides a comprehensive overview of APTs (it was useful for understanding what the kill chain was). Spanning eight years (2015-2023) and analyzing 53 articles, the study categorizes research into three focus areas: Endpoint Detection and Response (EDR) schemes, machine learning-oriented solutions, and graph-based strategies.

The professors begin by defining lateral movement within the context of the cyber kill chain and MITRE ATT&CK frameworks, emphasizing its role in allowing adversaries to progressively move deeper into a system in search of High-Value Assets, or High-Value Targets (HVTs). They highlight the increasing importance of lateral movement in the era of the Internet of Things (IoT), where vulnerable IoT devices can serve as attractive attack vectors due to their often-inadequate security measures and ubiquitous presence in sensitive network areas. The survey provides a detailed examination of various methodologies and techniques used to detect and mitigate lateral movement attacks. For EDR schemes, the authors focus on event-oriented reconnaissance of lateral movement practices, including the gathering of system and network-related logs, implementation of EDR rules, and evaluation of success rates. They chronologically analyze different studies, summarizing key characteristics and benchmark datasets used in each case. Machine learning-oriented solutions are explored within the article, showcasing how AI techniques are being applied to improve the detection and classification of lateral movement incidents.

The survey also delves into graph-based strategies, which offer unique approaches to visualizing and analyzing network behavior to identify suspicious lateral movement activities. Throughout the study, the authors pay special attention to the application of these detection methods in IoT environments, recognizing the unique challenges and opportunities presented by these increasingly prevalent devices. While examining various methodologies, techniques, benchmark datasets, and challenges across these categories, the survey provides a holistic view of the current state of lateral movement detection research. The survey not only summarizes existing research but also identifies key trends, challenges, and potential areas for future investigation. By mapping the progress in this field over time and offering insightful observations, the authors aim to propel future research in lateral movement detection forward, contributing valuable knowledge to cybersecurity professionals and researchers, such as us, working to combat these sophisticated attacks in increasingly complex network environments.

## CALDERA (Automating Adversary Emulation)

The second article[2], by lead Cyber Security Engineer Andy Applebaum and Senior Cyber Security Engineer Doug Miller thoroughly explains the importance and necessity of having tools like Caldera[4], which allow for Automated Adversarial Emulation, a tool developed by MITRE for automating the process of adversary emulation in cybersecurity. The article begins by outlining the importance of understanding adversary tactics, techniques, and procedures (TTPs) for effective security posture management. Traditional methods of emulation are often manual, labor-intensive, and can miss nuances of real-world attacks. CALDERA aims to streamline this process, allowing security teams to simulate attacks in a controlled, scalable, and repeatable manner. CALDERA's architecture is laid out, showcasing its modular design that separates the core functionalities from its various plugins. It supports a wide range of emulation scenarios, mapping to the MITRE ATT&CK framework, which provides a common language for detailing cyber adversary activities. By using plugins, CALDERA can replicate the actions and behaviors of real attackers, facilitating testing and validation of defense mechanisms within an organization. The paper emphasizes the versatility of CALDERA, as it can be deployed in both red-team and blue-team exercises, improving collaboration and understanding between offensive and defensive security teams. The document also delves into practical use cases of CALDERA, illustrating how it can be leveraged during penetration testing, incident response, and security assessments. It discusses the benefits of automating adversary emulation, including increased efficiency, deeper insight into defense weaknesses, and enhanced training opportunities for security personnel. The paper underscores the significance of continuous improvement in cybersecurity strategies through regular testing and adaptation to evolving threats, positioning CALDERA as a valuable asset in the modern cybersecurity landscape. In conclusion, the article serves as a comprehensive overview of the tool’s capabilities and applications in enhancing cybersecurity practices. It advocates for the integration of automated emulation into security workflows to better prepare organizations against potential adversaries by allowing teams to systematically identify and address vulnerabilities. The ongoing development and refinement of CALDERA highlight its role in the future of proactive cybersecurity measures.

## Hopper: Modeling and Detecting Lateral Movement

This article[3] by researchers from the USENIX conference includes contributions by Grant Ho, Mayank Dhiman, Devdatta Akhawe, Vern Paxson, Stefan Savage, Geoffrey M. Voelker, and David Wagner from UC Berkeley, UC San Diego, Figma, Inc., International Computer Science Institute, and Dropbox. The article was included in the Proceedings of the 30th USENIX Security Symposium on August 11-13, 2021. Furthermore, the paper explains how Grant Ho and his aforementioned colleagues explore an innovative defensive strategy for cybersecurity using “honeytokens” within Microsoft Active Directory (AD) environments. Honeytokens, which are a form of ‘Honeypot-type’ defensive measures deployed, allow Blue teams to detect and disrupt critical phases during lateral movement attacks. In this case, Lateral movement refers to an attacker’s progression through a network to gain access to additional resources or higher-privileged accounts. Honeytokens are used as decoy objects, such as fake user accounts, groups, or permissions, that mimic legitimate resources within the AD environment. When attackers attempt to interact with these decoys – either through reconnaissance or access attempts – they trigger alerts that notify defenders of potential malicious activity. This early detection capability is crucial in identifying and stopping attackers before they can escalate privileges or access sensitive data. The article highlights that effective honeytoken placement requires understanding the typical paths attackers might take during lateral movement. By embedding honeytokens in high-value areas of the AD or in paths attackers are likely to traverse, defenders can maximize their chances of catching malicious activity. Additionally, the authors emphasize the importance of ensuring honeytokens are indistinguishable from legitimate AD objects to avoid detection by skilled attackers. This integration disrupts the attackers’ ability to confidently move laterally, increasing their cognitive load and forcing them to second-guess their actions. In this way, honeytokens not only serve as an early warning system but also actively interfere with the attackers’ lateral movement strategies, making the network environment more resilient to compromise. Moreover, the article outlines how honeytokens create a layer of deception that complicates an attacker’s reconnaissance process which is a prerequisite for lateral movement. By planting decoys that look like high-value targets or potential stepping stones, honeytokens blur the line between real and fake resources, making it difficult for attackers to identify actual assets. This misdirection can cause attackers to expend time and resources probing decoys instead of real targets, delaying their progress and increasing the likelihood of detection. For instance, a honeytoken could mimic an administrative account or a sensitive file, enticing the attacker to attempt access, which would then trigger logging and alerts. The authors also explore how automated deployment and monitoring of honeytokens can enhance their effectiveness in disrupting lateral movement. As such, Automation ensures honeytokens are continuously updated and distributed throughout the AD environment, maintaining their credibility and ensuring that they remain integral to the network’s defense strategy. Additionally, robust monitoring tools can detect when honeytokens are accessed, flagging potential breaches in real time and enabling defenders to respond swiftly. By integrating honeytokens into a defense-in-depth overall strategy, the article demonstrates how organizations can create a hostile environment for lateral movement. This integration ensures that attackers encounter obstacles at multiple stages of their operations, reducing their chances of successfully navigating the network. The use of honeytokens thus transforms lateral movement from an attacker’s advantage into a “liability”, shifting the balance of power toward the defenders, or blue team, and making it increasingly difficult for adversaries to achieve their goals undetected.

# **research setup and proposed solution**

Within the following sections, we’ll explore a proposed simulated APT attack by performing lateral movement using the Caldera Framework by MITRE. Performing this attack on poorly configured Virtual Machines, it simulates a complacent and poorly configured network and how the lateral movement attack occurs following an already established breach within the network. It is important to note that this proposal was developed during milestone 2 when the limitations of time and resources for completing the full procedure were not fully taken into account. The following explains the goals and methodology that we originally intended to achieve.

## Goals

Our goal for this project is to not only understand the threat of advanced persistent threats (APTs), but to also find a practical solution to it. I must preface though that the scope of combatting APTs is massive. Many pieces to the puzzle can be categorized into preventative, detective, deterring, and responsive. It would require defense in depth at all stages with numerous strategies where it’s not only reactive but proactive against the attack. Since APTs involve a wide set of possible strategies, they must be met with the same and more (also the challenge of still keeping things functional). Strategies would be applied to the network, end-points, and specified zones or choke points.

So we will pick a few but major puzzle pieces that are networking-related. The practical project’s goal is to find a way to prevent and detect lateral movement and contain it within a simulated network.

## Problem

When pondering everything that was explained above about APTs, there’s a glaring problem for many organizations and states. Originally, we were exploring the possibility of employing Cyber-Physical System infrastructure (CPS) to detect and mitigate APTs as they occur. However, due to a lack of funding, this proposal was ultimately shut down. The following will explore our original intention of utilizing CPS to counter the many avenues that APTs could attack from, thus covering blind spots that manual human reviewers could not have spotted. The proposed integration provides more entry points and potential vulnerabilities for APT actors to exploit. Not only that, but critical infrastructure are frequent targets such as power grids, pipelines, water systems, manufacturing, and industrial processes. APTs targeting these systems can have severe real-world consequences potentially leading to physical damage or disrupting essential services (especially in the grand scheme of things). The “persistent” nature of APTs is very problematic for CPS also since attackers can remain undetected for longer times manipulating system operations subtly over time. And going back to computing systems, much of those are legacy systems meaning patching and managing hardware and software on them lack modern security features.

Now zooming in more on the networking perspective, several common problems allow APTs to succeed. Many organizations that operate CPS still use a flat network design without proper segmentation or micro-segmentation allowing easier lateral movement once they’ve gained access. Then there’s insufficient monitoring, especially in OT networks, also just the complexity makes it harder to detect unusual traffic patterns or exfiltration.

Insecure remote access is another one since many CPSs need it for maintenance. Then there are weak network boundaries, unencrypted communications…the list goes on depending on risk assessment reporting.

## Methodology

Our methodology for this project involves systematically designing, testing, and refining a practical solution to mitigate advanced persistent threat (APT) lateral movement. To tackle this, we organized it into five stages. First, we deploy a base network (1) designed to reflect common configurations and vulnerabilities. Since we are addressing the problem of lateral movement, we assume that the “adversary” has already accomplished external reconnaissance and its first compromise. This means our first stage also entails setting up an initial compromise on Subnet A on a single victim machine. Once everything is configured, the stage is set for the next which is conducting lateral movement (2). Here the “adversary” does internal reconnaissance and exploits to move from Subnet A through the network towards Subnet C where the target data is contained. So, we simulate what lateral movement does to our base network observing and collecting data to inform the next stage (3) for analysis. Reflecting on what the attack paths were and how and why they were carried out. With that information, we enhance the base network with security measures including advanced detection and prevention strategies. Following those improvements, we test solution (4) to see if it works against the same lateral movement, we did at stage 2. See if it’s contained within Subnet A collecting more data. In case it doesn’t, we iterate stage 3, beefing up the solution and retesting before finalizing our analysis and drawing conclusions (5).

1. Deploy Base Network with Initial Compromise

2. Do Lateral Movement (Observe/Collect Data)

3. Data Analysis -> Enhance Network with Prevention & Detection Strategies (Implement Solution)

4. Test Solution (Observe/Collect Data)

5. Analysis & Drawing Conclusions

## Base Network Setup and Simulation

To address the challenges of Lateral Movement in a controlled environment, we will design a step approach that begins with setting up a baseline network infrastructure. Using VirtualBox, the environment can simulate a simplified network consisting of three subnets: Subnet A (192.168.6.0/24) for a Windows 10 pro end user, Subnet B (192.168.7.0/24) housing a Windows Server 2022 Domain Controller, and Subnet C (192.168.8.0/24) containing a Windows Server 2022 acting as a fileserver. In our case, the sensitive asset will be the fileserver, which will be housing a sensitive PNG file. In addition, there will be an External Kali Linux ‘Attacker’ VM that will be connected to Subnet A, in which the VM will use the Windows 10 Pro machine on that same subnet to open a reverse shell and use that machine as a pivot to connect to the network. A PfSense (v2.7.0) firewall[15] can serve as both an edge router and an internal firewall to manage network segmentation, providing a platform for testing vulnerabilities and defense strategies. That firewall will intentionally have a Lax ruleset and there will be some configurations within said firewall that will intentionally be left default.

Within the PfSense Firewall, four network interfaces can be utilized to simulate a segmented network environment; The WAN interface (Adapter 1) will be connected to the Kali Linux ‘Attacker’ VM, simulating an external attacker accessing the network. Three additional interfaces will serve as internal network gateways: Adapter 2 for Subnet A (LAN-A, vmx1/vtnet1), Adapter 3 for Subnet B (LAN-B, vmx2/vtnet2), and Adapter 4 for Subnet C (LAN-C, vmx3/vtnet3). Each subnet will be configured as an isolated /24 network, with PfSense acting as the default gateway to ensure traffic segmentation. Wireshark will be employed on the machines in Subnets A, B, and C for traffic analysis.

## Topology

Figure 1 is a proposed topology including network segmentation from the project milestone 2 report. Please take note that Cisco packet tracer [10] is utilized to visualize the device roles graphically and that the labels to the devices aren’t the actual Cisco devices themselves (VirtualBox VMs). Figure 2 shows the “intnet” layout of VirtualBox’s virtual ports within the Win10 (192.168.1.3), Kali Linux (192.168.1.100), and Domain Controller Windows Server 2022 (192.168.1.2). Please take note that Figure 2’s IPv4 schema is different from our own, as it is a borrowed diagram.

A diagram of a computer network

Description automatically generated

Fig. 1 Proposed Segmented Network

A screenshot of a computer

Description automatically generated

Fig. 2 VirtualBox Virtual Ports Diagram (IPv4 Schema not ours)

Unfortunately, we were not able to fully implement network segmentation within the given time that we were allocated for this research project. To remedy not having sufficient time to configure proper network segmentation, we settled for a more simplified topology where all the devices would be on the same subnet, shown below in Figure 3.

A diagram of a server

Description automatically generated

Fig. 3 Actual Network

## Next Steps

Once we have the base network finalized and confirmed that everything is functioning/simulating as it should, these are the next steps visualized. We know the real test starts with an established connection to a Windows 10 machine within Subnet A originating from the external Kali VM. So how do we plan to execute lateral movement? There are multiple orders of operations and tools possible for getting to Subnet C for fileserver access. One idea is to do this more manually to have a deeper understanding of all steps and exploits in detail using that Windows 10 machine as a pivot point. It’ll be controlled through a reverse shell or Meterpreter session (initiated by Kali), allowing commands to be executed in memory to evade detection. We read that using tools like Mimikatz on PowerShell, we can extract credentials or hashes from memory, which can be leveraged for pass-the-hash attacks or other credential-based lateral movement methods. BloodHound is another popular tool that will be deployed to map the Active Directory environment identifying privileged accounts, vulnerable trust relationships, and exploitable paths to target in Subnet C.

Since pass-the-hash does not require Metasploit (and we don’t believe it’ll be needed to map in topology), it can be executed manually or via scripts using hashes from Mimikatz. Other lateral movement techniques seen in our research are leveraging open shares, overly permissive firewall rules, VLAN hopping, or taking advantage of shared local admin credentials. All are orchestrated by the Kali VM, but the actual execution will occur on the compromised Windows 10 machine (as it’s a victim and now an internal threat) to maintain a realistic attack flow – also since the Kali is attempting to remain undetected.

We plan to conduct data collection by doing network traffic monitoring with Wireshark to capture interactions between subnets, and logs generated by the tools we used (BloodHound) and Caldera is another cool tool to track the sequence of actions highlighting vulnerabilities. If feasible, event logs from the compromised Windows machine and other critical systems will also be collected for analysis. All those data sources allow us to evaluate attack paths and see if the vulnerabilities I have mentioned are indeed what ends up getting exploited – also identify gaps in the base network’s detection. However, the networking solution(s) we’ll need is comprehensive and probably not a single pinpoint. Another idea is the combination of Caldera and Pathfinder, which are open-source tools developed by MITRE built to simulate adversarial behavior based on MITRE ATT&CK framework (catalog of techniques/tactics/procedures observed in real-world) – this would be more of an automated approach to carrying out the lateral movement simulation showing multiple ways an attacker would do so and how they manage to blend in or conceal their activity (RDP/SMB, LoTL…).

In general, carrying out these methods ourselves with tools will provide critical insight into how lateral movement occurs within a network.

Depending on the results, we will point out what solution(s) are needed. However, we hypothesize that the topology will need reinforced network segmentation with stronger network defense in depth strategies in general along with going to the domain controller VM to strengthen active directory configurations. PfSense or the chokepoint will be a major focus as it’s where the level of trust is supposed to change preventing potential untrusted access from getting to the fileserver in the first place. Beyond those best practice aspects is detection and response with anomaly-based machine learning (ML).

And hopefully, we have time to implement and test those solutions by doing lateral movement again until it can no longer move or remain persistent.

Overall, our methodology will focus on iterative testing, starting with deploying a baseline VM network configured with intentional vulnerabilities to replicate real-world vulnerabilities. We will then simulate lateral movement to replicate real-world vulnerabilities. We will then simulate lateral movement to observe how attackers exploit these gaps to pivot from compromised endpoints and access critical assets. Using the data collected, we will implement advanced detection and prevention techniques, including enhanced segmentation, and stricter access controls to mislead and entrap attackers. Testing the improved network will assess its effectiveness in restricting lateral movement and detecting malicious activity.

In subsequent phases, we plan to integrate anomaly-based machine learning models and explore automated attack simulation tools like Caldera and Pathfinder to refine our detection capabilities further. By iterating our findings and continuously improving our strategies, we aim to create a resilient network that effectively addresses the threat of lateral movement. While this milestone sets the stage for the project, further research and testing will be essential to enhance the scalability and efficiency of the proposed solutions.

# **Implementation, Results, and Discussion**

The following section covers our process of setting up the testing environment, what we managed to implement and test, the results, and discussion on its outcome – essentially the simulation based on our current methodology (Milestone 3).

## Deployment of Base Network (The Initial Compromise Phase)

Our first step was to install and configure the necessary virtual machines on VirtualBox software run on our laptops utilizing the base topology diagram (Figures 1 and 2).

A screenshot of a computer

Description automatically generated

Fig. 4 Virtual Machines within VirtualBox’s GUI

As described earlier, our setup is comprised of, in total, five machines.

Key Points:

* PfSense 2.7.0 CE (internal network’s gateway router and firewall)
* VirtualBox’s Virtual Switch – manages the internal network ‘intnet’
* Kali Debian 64-bit Linux where the C2-Command and Caldera Server is run
* Windows Server 2022 acting as Domain Controller
* Windows 10 as the initially compromised workstation ‘enduser1’
* Windows Server 2022 acting as the fileserver workstation ‘FS’ aka ‘enduser2’ – this is the end goal machine containing a simple GOLD.txt file in which lateral movement is needed to get to

On each machine, we configured their IPv4 network settings to be static and assigned necessary information manually.

The configurations:

* PfSense: 192.168.1.1 as LAN and gateway
* Kali: 192.168.1.100/24 as ‘eth1’ within ‘intnet’ LAN – assuming initial compromise
* Windows DC: 192.168.1.2/24
* Compromised Workstation:192.168.1.3/24
* FS: 192.168.1.4/24

As all five machines were actively running, we confirmed that they were able to communicate by performing simple pings between each other utilizing their IP addresses. No machine could reach any IP NOT defined within the ‘intnet’ network to ensure a safe testing environment. Each machine was confirmed to be able to reach each other. Although we intended to set up the Kali machine to connect to the WAN interface on the router as if their entry originated from the internet to simulate more realism, we ran out of time to configure that properly. The Kali machine though does have two network adapters: ‘eth1’ for internal, and ‘eth0’ as a NAT via DHCP to simulate the attacker having internet access.

After networking them together, we headed over to the Windows domain controller machine as administrator to create the network’s domain. On the server manager, we added roles and features on the dashboard. Selected role-based installation and installed Active Directory Domain Services (AD DS) as well as DNS services to configure the domain (this was intended so our lateral movement attack sequences would abuse domain and active directory common misconfigurations). After those installations, we promoted the server to be the domain controller. During that process, we learned how to add a new forest to our network. The desired root name created was corp.dumbnetwork.com. In a nutshell, after configuring those defaults we went to both the fileserver and workstation to join them both in the domain on computer configurations.

A screenshot of a computer

Description automatically generated

Fig. 5 Both End-user VMs within the AD DS GUI

The next step was to configure the initial compromise. We decided to configure CALDERA following the steps on the tool’s official Git Hub.

## Performing Lateral Movement (Observation/Data Collection Phase)

In the following section describes the process of how we setup the command and control (C2) server, initial compromise on ‘user1’ windows 10 workstation, and how the first attack was executed. I will say it was quite the learning curve and not everything went perfectly + took troubleshooting along the way, but in doing so revealed that a misconfigured network can be easily compromised.

On the Kali virtual machine, we setup these major dependencies: python 3.9, pip3, nodejs, npm, and golang-go.

Then on MITRE Caldera’s official github, we did the following commands on Kali’s terminal:

Git clone https://github.com/mitre/caldera.git --recursive

cd caldera

pip3 install -r requirements.txt

python3 server.py --insecure --log DEBUG –build

It took about 3-5 minutes to do the first build, then everything was ready to go as the server was up and running. We were then able to visit the GUI hosted at http://localhost:8888.

This default however we changed in the server’s configuration file to specify that the GUI and server communication throughout different ports and sockets was available at Kali’s internal IP…so http://192.168.1.100:8888 so our agent deployed on the internal network knew where to report back to.

Rebuilt to update those changes.

Upon viewing the hosted GUI, we were presented a login that had the default passcode set to admin: admin.

Their documentation goes into depth on each functionality but essentially, CALDERA is an adversary emulation platform developed by MITRE for testing network defenses. It uses agents, such as Sandcat and Manx, which are programs deployed on target systems (like a RAT) to execute commands and gather information. Plugins extend Caldera’s functionality, like the stockpile plugin that provides additional abilities. Adversaries in Caldera are profiles of simulated threat actors, while operations are the execution of these adversary profiles against a network. The available abilities are derived from the MITRE ATT&CK framework, representing specific techniques or exploits used by attackers. So the platform basically allows red teams to create and execute attack scenarios through a GUI to ultimately help organizations assess their security posture by simulating real-world attack techniques in a controlled environment.

The cool thing was, each ability had a MITRE ATT&CK code and displayed each command and output throughout the process. Or we could manually do attacks and upload customized payloads.

We decided to utilize the Manx agent to do a reverse-shell that communicates via TCP.

Initial access Agent payload:

if ($host.Version.Major -ge 3){$ErrAction= "ignore"}else{$ErrAction= "SilentlyContinue"};$server="http://192.168.1.100:8888";$socket="192.168.1.100:7011";$contact="udp";$url="$server/file/download";$wc=New-Object System.Net.WebClient;$wc.Headers.add("platform","windows");$wc.Headers.add("file","manx.go");$data=$wc.DownloadData($url);Get-Process | ? {$\_.Path -like "C:\Users\Public\caldera.exe"} | stop-process -f -ea $ErrAction;rm -force "C:\Users\Public\caldera.exe" -ea $ErrAction;([io.file]::WriteAllBytes("C:\Users\Public\caldera.exe",$data)) | Out-Null;Start-Process -FilePath C:\Users\Public\caldera.exe -ArgumentList "-socket $socket -http $server -contact $contact" -WindowStyle hidden;

This payload was ran on ‘enduser1’ powershell without their knowledge. So it established communication with the command and control (C2) server aka CALDERA running on Kali enabling command execution and data exfiltration. Though we did not do exfiltration, but the agent served as a pivot to expand control or discovery of the internal network. The agent connection was confirmed in Kali’s terminal server log, and took around a minute to confirm its PID on the GUI. From there we were ready to begin the operation.

A screenshot of a computer

Description automatically generatedWe had a few attempts of trying varying abilities and adversaries for the best attack combo, and in the end found a classic worm to be pretty effective for lateral movement throughout the network – makes sense. These were the tactics used in order to traverse from user1 to user2:  
While the operation was running, we were able to add manual commands if needed and view the ones being run. View the debrief pdf document for more details.

## Shortcomings

Unfortunately, due to time constraints, we were unable to run multiple trials which did not allow us more time for troubleshooting. This resulted in a few key incidents that did not allow us to achieve the results that we desired from this research.

1. Insufficient Hardware requirements

Since we had to use our own hardware to run these virtual machines, we ran into performance issues and slow performance when having all virtual machines up and running at the same time. Namely, Caldera itself recommends at least an *Intel i7* Desktop Processor in addition to circa 8 GB of DDR4 RAM. Both laptops that we used to run our simulations vastly did not meet the recommended specifications, thus resulting in a myriad of issues. Namely, the Active Directory Domain Controller (AD DC) Windows Server 2022 becoming almost non-responsive toward the tail end of our allotted time for testing. This resulted in several hours of the virtual machine hanging, and eventually, we decided to remake the Domain Controller from scratch. Fortunately, this remedy worked.

1. Lateral Movement Attempted by CALDERA Agent

Despite our best efforts in the given time that we had, we have successfully launched an attempt to perform Lateral Movement with an Agent. This resulted in the results being printed on the CALDERA console, as shown in Figure 7. Unfortunately, the results leave much to be desired, but at least the attempt looks like it was successfully launched, thus proving that CALDERA was successfully configured in the end.

1. PfSense configuration process

While we were configuring the PfSense Firewall, we, unfortunately, downloaded the wrong .iso image file to mount onto Virtual Box (PfSense Netgate installer v2.7.0), and it took at least five hours to search online for an older version (PfSense Community Edition v2.7.0) for us to reconfigure. This resulted in a loss of a day that was allocated toward configuring and simulating our topology. Fortunately, the configuration was a success in the end.

## Figures and Documentation

A screenshot of a computer

Description automatically generated

Fig. 6 Caldera Successful Boot

A screenshot of a computer

Description automatically generated

Fig. 7 The agent being deployed from CALDERA GUI

A screenshot of a computer

Description automatically generated

Fig. 8 CALDERA GUI home screen

A screenshot of a computer

Description automatically generated

Fig. 9 End Results from Running CALDERA Simulation

A screenshot of a computer program

Description automatically generated

Fig. 10 Selecting a target for Agent on the CALDERA GUI

A screenshot of a computer

Description automatically generated

Fig. 11 PfSense CLI showcasing interfaces

A screenshot of a computer

Description automatically generated

Fig. 12 Initiating Operation on CALDERA GUI

A screen shot of a computer

Description automatically generated

Fig. 13 Kali Linux POV of CALDERA’s install path

A screenshot of a computer

Description automatically generated

Fig. 14 Adversaries Listing on CALDERA GUI

A screenshot of a computer

Description automatically generated

Fig. 15 “Escalate and Lateral Movement” menu on CALDERA GUI

A screenshot of a computer

Description automatically generated

Fig. 16 Adversarial traffic being logged on Kali Linux

# **CONCLUSION**

Advanced Persistent Threats (APTs) pose an increasingly formidable challenge to organizations and governments due to their sophisticated, stealthy, and multifaceted nature. This research, which focused specifically on mitigating lateral movement within the APT kill chain, highlighted the critical vulnerabilities that exist in modern network architectures. As mentioned in the introduction, over 70% of modern network breaches occur from APTs utilizing lateral movement-based attacks, and only being discovered months later[12]. Through the lens of a simulated lateral movement attack using the CALDERA framework, it became evident that networks with weak security controls, and therefore a lack of utilizing the Defense-in-Depth doctrine, in addition to insufficient defensive controls, which include monitoring, remain primary enablers for APT actors to infiltrate and expand their foothold. While the study was limited in scope due to time and resource constraints, the finders underscore the importance of proactive and layered defenses against such threats.

The simulation revealed that lateral movement is not merely a secondary phase within an attack, but a pivotal stage where attackers consolidate their control, escalate privileges, and gain access to sensitive assets. The research affirmed that traditional security measures, such as perimeter defenses and reactive monitoring, are insufficient in detecting and mitigating the ever-increasing sophistication of lateral movement techniques effectively. Adversaries exploit weaknesses such as weak segmentation, overly permissive access controls, and legacy system vulnerabilities to navigate networks undetected.

Implementing solutions to counter lateral movement requires a multi-faceted approach, such as network segmentation. However, network segmentation is only a foundational strategy, supported by micro-segmentation and strict access control policies to limit the spread of an attack. Additionally, deploying advanced detection systems capable of identifying anomalous behavior at granular levels, such as unusual traffic patterns or privilege escalations, proved to be essential. While the research leveraged a CALDERA simulation to illustrate vulnerabilities, it highlighted the potential of tools like CALDERA to provide actionable insights into attack paths and exploit methods. These findings validate the effectiveness of monitoring and analysis in enhancing an organization’s ability to preemptively respond to the possibility of an external threat employing lateral movement, once they have inevitably gained entry into the network(s).

Overall, while APTs continue to evolve in sophistication, addressing techniques to counter lateral movement will offer a crucial leverage point in disrupting an attack going through the XDR kill chain, and minimizing the impact of such attacks. Organizations must adopt a proactive, layered approach that combines techniques such as network segmentation, advanced detection tools like machine learning capable systems, and tools that allow for continuous monitoring to strengthen a network’s resilience.

As the cybersecurity landscape grows increasingly complex, this research underscores the urgency of developing adaptive and comprehensive strategies to protect critical infrastructure and sensitive data from persistent and sophisticated threats now and hopefully into the future.

##### Acknowledgment

We’d like to thank Dr. Bhaskar Rimal for granting us this opportunity to work on this self-guided research, however limited the scope of our actual implementation would be. We have learned a lot about networking fundamentals and hope to improve and continue this research for its next steps. This research has not been sponsored by any third parties.

##### References

1. C. Smiliotopoulos, G. Kambourakis, and C. Kolias, “Detecting lateral movement: A systematic survey,” Heliyon, vol. 10, no. 4, p. e26317, Apr. 2024. [Online]. Available: https://doi.org/10.1016/j.heliyon.2024.e26317
2. A. Miller and J. Stoner, "CALDERA: Automating Adversary Emulation," in Black Hat Europe 2017 Briefings, Dec. 2017. [Online]. Available: https://www.blackhat.com/docs/eu-17/materials/eu-17-Miller-CALDERA-Automating-Adversary-Emulation.pdf.
3. S. Hauser, J. Voris, and N. Christin, "Honeytokens in Active Directory: Detection and Defense," in Proceedings of the 30th USENIX Security Symposium (USENIX Security '21), Aug. 2021. [Online]. Available: https://www.usenix.org/system/files/sec21-ho.pdf.
4. MITRE, "CALDERA: Adversary Emulation Platform," GitHub Repository. [Online]. Available: https://github.com/mitre/caldera.
5. F. Veljkovic, V. Radojicic, and M. Malnar, “Cybersecurity challenges in the industrial internet of things: A comprehensive review,” Electronics, vol. 13, no. 19, p. 3944, Sep. 2024. [Online]. Available: https://www.mdpi.com/2079-9292/13/19/3944
6. MITRE Engenuity. (n.d.). CALDERA Pathfinder. Center for Threat-Informed Defense. Retrieved November 21, 2024, from https://mitre-engenuity.org/cybersecurity/center-for-threat-informed-defense/our-work/caldera-pathfinder
7. Ussath, M., Jaeger, D., Cheng, F., & Meinel, C. (2023). A survey on lateral movement detection. Journal of Cybersecurity, 9(1), tyad005. https://doi.org/10.1093/cybsec/tyad005
8. K. Eves and J. Valasek, “Adaptive control for singularly perturbed systems examples,” Code Ocean, Aug. 2023. [Online]. Available: <https://codeocean.com/capsule/4989235/tree>
9. S. M. Milajerdi, R. Gjomemo, B. Eshete, R. Sekar, and V. Venkatakrishnan, “HOLMES: Real-time APT detection through correlation of suspicious information flows,” in 2019 IEEE Symposium on Security and Privacy (SP), May 2019, pp. 1137–1152. [Online]. Available: https://doi.org/10.1109/SP.2019.00026
10. Cisco, Cisco Packet Tracer. [Online]. Available: https://www.netacad.com/learning-collections/cisco-packet-tracer?courseLang=en-US. [Accessed: Dec. 11, 2024].
11. MITRE, CALDERA Documentation. [Online]. Available: https://readthedocs.org/projects/caldera/downloads/pdf/latest/. [Accessed: Dec. 11, 2024].
12. Elisity, "Understanding and Preventing Lateral Movement: A Strategic Guide for Enterprise Security Leaders," Elisity Blog. [Online]. Available: https://www.elisity.com/blog/understanding-and-preventing-lateral-movement-a-strategic-guide-for-enterprise-security-leaders. [Accessed: Dec. 11, 2024].
13. J. Goh, S. Adepu, M. Tan, and Z. S. Lee, “Anomaly detection in cyber physical systems using recurrent neural networks,” in 2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE), Jan. 2017, pp. 140–145. [Online]. Available: https://doi.org/10.1109/HASE.2017.36
14. Y. Ma and W. Wang, “Machine learning for cybersecurity: The basics of anomaly detection,” IEEE Security & Privacy, vol. 18, no. 4, pp. 72–75, Jul./Aug. 2020. [Online]. Available: https://doi.org/10.1109/MSEC.2020.2990106
15. Netgate, pfSense Documentation. [Online]. Available: https://docs.netgate.com/pfsense/en/latest/index.html. [Accessed: Dec. 11, 2024].