RIT | Golisano College of Computing and Information Sciences
Department of Computing Security

Stepping Stone Detection: Data Collection

Authors
Christin Alex, Rayhan Baig, Saakshi Gupta, Salomi Rao

CSEC 499 - Undergraduate Capstone in Computing Security

Golisano College of Computing and Information Sciences

May 2023

I. ABSTRACT

This paper presents an innovative approach to detect stepping-stone intrusion attacks, which is achieved by creating a realistic hospital environment within a mini-competition infrastructure. The competition generated a significant amount of data that was analyzed using statistical analysis and programming to identify attack paths. The data was categorized as either malicious or benign based on daily analysis as the competition progressed. Additionally, we analyzed the average tunneling time in each subnet by several attackers, and SSH streams were extracted and analyzed to distinguish pivoting traffic from normal traffic. Our analysis of the collected data provided insights into the possible attack path used by the attackers to compromise the infrastructure. The findings of the study revealed that a future detection system could monitor incoming streams based on specific criteria and detect any outgoing connections.

II. INTRODUCTION AND MOTIVATION

Cyberattacks are a growing threat to organizations of all sizes. One common tactic attackers use is steppingstone intrusion (SSI), which involves moving laterally inside a network while concealing their identity. SSI can be used to gain access to sensitive data, disrupt operations, or launch further attacks.

Traditional methods for detecting SSI are often ineffective, as they rely on signatures or heuristics that can be easily evaded by attackers. In this paper, we propose a novel approach for detecting SSI by conducting a mini-competition where participants attempt to gain unauthorized access to a simulated network environment. We aim to collect data and differentiate between normal data traffic and stepping-stone attack data traffic.

To achieve this, we have designed a network topology that depicts an entire stepping-stone attack and planned out the attack so that it can be completed within a timeframe. We have listed the number of hosts, servers, routers, types of operating systems, and services that will be running on all the devices. Additionally, we have implemented vulnerabilities in systems that will result in pivoting attacks and tested the stepping stone attack to obtain a successful hop tunnel path.

We have made the following contributions to the domain of stepping-stone intrusion analysis:

- We have developed a novel approach for detecting SSI by conducting a mini-competition.
- We have collected a large dataset of stepping-stone attack data that can be used for further research.
- We have employed existing tools for analyzing the collected data to identify stepping-stone attacks.
- We have identified patterns from the collected data to distinguish between a stepping-stone attack and benign data.

We believe that our approach has the potential to improve the detection of SSI.

III. BACKGROUND

A. *What are Stepping Stone Intrusions?*

Stepping Stone Intrusions (SSIs) refer to pivoting attacks where an attacker uses intermediary hosts to obscure the origin of their malicious activity on a target. There are two main use cases for pivoting attacks: **outside perimeter pivoting and lateral movement within an internal network**. Outside perimeter pivoting involves using multiple intermediary hosts, such as attackers' proxies or compromised hosts on the internet, to avoid detection by obscuring the origin of the attack [1] [2]. In contrast, lateral movement pivoting attacks use compromised hosts within an internal network [3] to overcome connectivity restrictions imposed by firewalls or access control mechanisms and access different network segments [4].

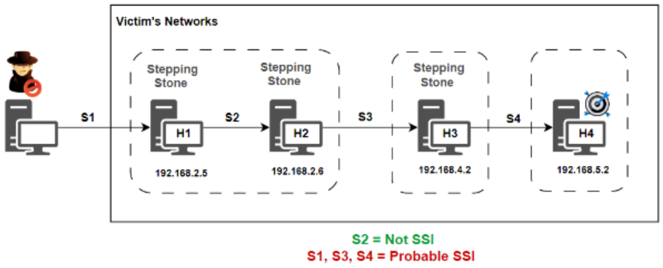

Fig. 1. Stepping-Stone Attack

In Fig. 1, a stepping stone attack is illustrated, the attacker initiates the attack by gaining access to one of the hosts (S1) within the victim's network, which serves as the first pivot point. While the attacker moves laterally within the network, this does not constitute a stepping-stone attack, so the second host (S2) is not considered a stepping-stone intrusion (SSI). However, the attacker subsequently leverages H3, which is on another subnet, to pivot further into the network, thus creating a stepping-stone intrusion.

A pivot host is a computer that relays communication between an attacker and a target that would otherwise be inaccessible to the attacker. This allows the attacker to indirectly connect to a normally non-routable network. There can be more than one pivot host in a chain, with each host relaying communication to the next [3] [4].

This sort of **lateral movement is at the core of an attack campaign on a large organizational network segmented into different subnets that also employ firewalls to deter access from outside attackers.** It allows an attacker to get closer to the original objective that prompted the network intrusion using compromised hosts. The use of outside perimeter pivots is an additional measure to obscure their activity, secondary to the primary use of accessing inaccessible hosts. This is reasonable to assume because it is unlikely that the first entry point achieved by an attacker has what they require to consider the attack successful [5]. This is especially true as pivot attacks are common techniques used in advanced persistent threat (APT) campaigns [6].

B. *Characteristics of Pivoting Traffic*

Pivoting traffic is characterized by several key features [1]. Firstly, all communication between the attacker and target goes through the pivot. Secondly, the pivot establishes bidirectional connections between pivots, sources, or targets to control a specific target. This traffic generally involves small packet sizes [2] and can be measured by the number of packets transferred, number of bytes, and duration of flow [3]. Additionally, each incoming connection should have at least one pair of bidirectional network flow to the corresponding outgoing connection pair. Requests and replies occur sequentially with small time differences between packets [4], and upstream connections begin first and end last, resulting in correlated timestamps [2]. Finally, it is essential that connection pairs do not share the same port on the same pivot host, as this would suggest access to the same server on the pivot host. These features are relevant for identifying pivot attacks, which are commonly used in APTs [2] [3] [4].

C. *Detecting Stepping Stone Intrusion*

In this section. we will discuss the two categories of solutions for detecting Stepping Stone Intrusion: host-based detection and network-based detection.

Host-based detection aims to determine if a given host machine is being used as a pivot by an attacker by comparing incoming connections with outgoing connections to find a relayed connection pair. This method could produce false positives and should be context aware [1].

Network-based detection estimates the length of the connection chain and identifies a pivoting instance if the number of connections passes a threshold beyond what would be used for legitimate purposes. However, this method is limited as only the downstream of the connection chain from the sensor is visible, and it is only applicable in the context of outside perimeter detection [1]. Both types of solutions use content or metadata characteristics of traffic for detection.

Many proposed solutions employ analysis of NetFlow records to mitigate encrypted connections used by attackers to evade detection [2] [3] [4]. NetFlow captures **metadata of the network communications** between two hosts, such as the start, duration, and size of the transmissions [7].

The above stepping-stone intrusion detection solutions can be further divided into techniques such as packet manipulation, real-time analysis, number of sensors, and content-based analysis [2].

It is important to consider the baseline behavior of a service when looking for suspicious behavior to prevent false positives. SSH is an example of a service that should be investigated when displaying pivoting characteristics in network traffic [3] [8]. It is also essential to consider whether the service's normal activities are similar in characteristics to malicious pivoting. For example, SMTP, VoIP, NTP, Instant messaging, P2P, Streaming, and DNS are examples of services whose traffic can generate false positives.

Detecting pivot activity in enterprise networks can be challenging due to the diversity of methods used by attackers [9]. Some detection schemes agnostic to content or protocol-specific calculation [10] have been developed, which focus on **metadata characteristics for detection** [4].

One such scheme addressed the limitations of previous schemes and employed an **agent-based pivot detection approach** focused on scalability with options for aggregation of results to give a holistic view of agents in a CTI Framework [4] [9]. It is agnostic to transport and application layers, addresses privacy issues in data processing, and is capable of outside network pivot detection.

D. *Evasion Techniques*

Evasion techniques are used by attackers to defeat existing intrusion detection systems. Two common techniques are:

- **Time jittering** involves holding a packet for a while before releasing it, in order to change the time gap between TCP/IP packets and evade time-based SSID.
- **Chaff perturbation**, on the other hand, involves injecting meaningless packets into a TCP/IP session to change the time gap and the number of packets in a certain time period, making it difficult for IDS to detect intrusion.

While packet matching can be used to filter chaffed packets and resist intruder evasion, there is no guarantee that all chaffed packets will be filtered out [1].

It is important for organizations to be aware of these evasion techniques and to have appropriate countermeasures in place. This could include:

- Implementing more advanced IDS that can detect and filter out chaffed packets
- Using encryption and other security measures to protect against time-jittering attacks [10].
- Regular monitoring and updating of security measures.

IV. PROJECT OUTLINE

The purpose of this project is to design and conduct a competition for collecting data on pivoting attacks with high hop counts, generating ground truth information for training machine learning models to detect pivoting attacks, and encouraging participants to exploit vulnerabilities and pivot through networks.

The following outcome objectives were developed for this project:

- Collect data on pivoting attacks.
- Encourage participants to exploit vulnerabilities and pivot through networks.
- Evaluate the success of the competition.

A. *High Level Requirements for Data Collection*

To ensure the accuracy and representativeness of the dataset for pivoting attacks, high-level requirements must be considered during data collection. Currently, there are limited datasets available for pivoting attacks, so it's important to obtain **raw packet captures** from the router and hosts for both host-based and network-based stepping-stone detection. Raw packet captures allow for the creation of flow summaries and metadata extrapolation, ensuring reproducibility [8].

The collected data should represent all significant load conditions and trends, including peak loads and services that run on specific days. The sample space should be long enough to capture the temporal characteristics of normal and threat events. Additionally, the cyberinfrastructure deployment should be representative and maintain performance during the capture period.

The dataset should include metadata such as flow summaries, and nuances in flow timing like session time and burst characteristics must be captured accurately. It is important to consider good quality timing and geo-spatial supporting data as accurate timestamps are essential for many IDS techniques.

Clear documentation should be provided regarding any constraints and limitations in the design or data. The dataset origin, publication date, and deployment context should be clearly labeled to avoid incorrect assumptions about the scope of threats.

The dataset must exhibit pivoting attack behaviors, including hop counts greater than 2. Ground truth information should be available to assist in the labeling process of pivoting and non-pivoting traffic.

B. *Competition Design*

Our competition was designed to meet the following requirements:

- Collect data on pivoting attacks with high hop counts.
- Generate ground truth information for training machine learning models to detect pivoting attacks.
- Encourage participants to exploit vulnerabilities and pivot through networks.

We selected a **King of the Hill style competition** design because it is well-suited for collecting data on pivoting attacks. In a King of the Hill competition, each team starts with a single machine and must attack and gain control over as many other machines as possible. Teams can expand their territory by using compromised machines to **pivot onto their next targets**. This design encourages participants to exploit vulnerabilities and pivot through networks in order to gain an advantage.

We also incorporated elements of **Capture the Flag (CTF) competitions** into our design. In CTF competitions, participants must exploit vulnerabilities or solve challenges on hosts in order to reveal flags. We hid flags behind vulnerabilities in our competition, which gave participants two incentives: to exploit vulnerabilities for points and to expand their reach in the competition.

We believe that our competition design was successful in meeting our requirements. We collected a large dataset of pivoting attacks with high hop counts, and we generated ground truth information that can be used to train machine learning models to detect pivoting attacks. We also believe that our competition design encouraged participants to exploit vulnerabilities and pivot through networks.

V. METHODOLOGY

A. *Competition Design and Topology*

The total number of participants competing was 24 students, 13 blue teamers, and 11 red teamers as a mandatory part of their course requirements.

After we had the participants and the Red vs. Blue team competition in mind, we began seeing further challenges with the number of networks and the load that our router would handle. Due to restrictions on the VASE platform with the number of networks available, we were restricted to a total of 3 hops (from each network subnet) that we could obtain for one complete stepping-stone attack. This limitation also made it infeasible to incorporate multiple routers since we wouldn't have distinct networks to assign to the interfaces, which was initially a cause for concern due to the high traffic we were expecting. After researching more on this issue, we found out that PfSense is no longer single-threaded. This meant that it could use multiple CPU cores, giving it the ability to handle all the traffic flow. Once we had finalized the network topology theoretically after all the limitations and modifications, we began getting more into the details of this infrastructure.

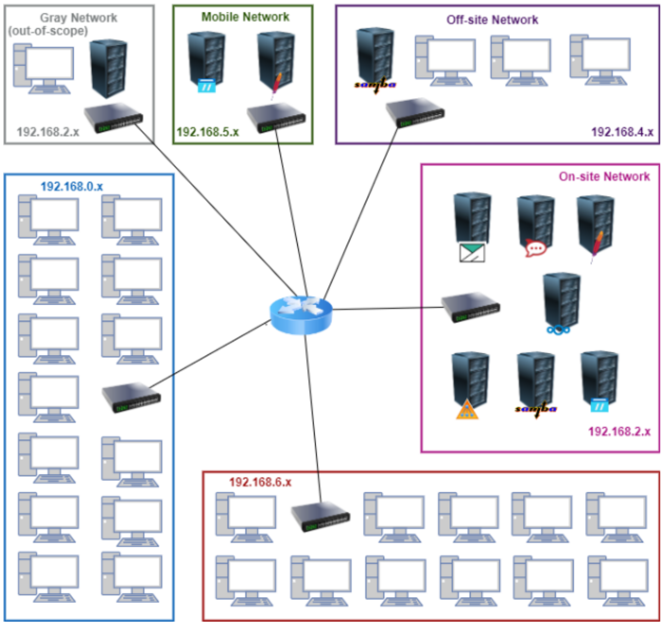

Fig. 2. Stepping-Stone Attack

*1) **Infrastructure**:* We finalized a mix of Windows-based and Linux-based hosts for our competition which included Windows AD, Windows systems, Linux servers, and systems, with the incorporation of various flavors of Linux such as Rocky Linux, Ubuntu, and Kali, and varied versions that were being run. Our infrastructure had around 10 servers running multiple services across the network which included Active Directory (AD), 2 SMB servers, 2 HTTP servers, NextCloud, RocketChat, 2 FTP servers, and an SMTP/IMAP server. We believed that having the router in-scope for the competition could easily give one team more strength than the other, so it was a unanimous decision to keep the router out-of-scope. The gray team and scoring were also out-of-scope.

2) ***Competition Theme:*** Our competition was designed around the theme that Bravo Hospital was testing its security controls and the strength of its defenders by contracting a set of skilled penetration testers to find weaknesses and exploit them within Bravo Hospital's network infrastructure.

3) ***Competition Design:*** For the stepping-stone attacks, we incorporated multiple pathways that the red team could follow. These were either in the form of concealed hints or credentials to the next network being stored as flags, or vulnerabilities found within servers that would help them pivot into the next network, some of which we can see in Table I.

TABLE I
TABLE OF VULNERABILITIES

Machine Name	IP Address	Exploitable Vulnerability
Offsite 1	192.168.4.2	Binary Code
Offsite 2	192.168.4.3	Samba
Offsite 3	192.168.4.4	Log4j
Offsite 4	192.168.4.5	Cowsay Escalation Privilege
Mobile 1	192.168.5.2	FTP Anonymous Login
Mobile 2	192.168.5.3	Local File Inclusion

These pathways of stepping-stone were enforced through firewall rules that we had set in place. The implemented firewall rules gave the blue team access to the entire in-scope network, which consisted of multiple subnets but restricted the red team's access to only the first network, restricting their access to the rest of the networks. In a similar manner, all traffic originating from the first network was only allowed to communicate ahead to the second network or the red, or blue. And similarly, with the third network, the third network was only allowed to communicate with the second network and the blue team directly.

The first and second networks consisted of various services that needed to be running for the blue team to score points. If the red team were able to take down these services, they would be scored on downtime. Additionally, they also contained flags that not only gave the red team useful clues but also granted them a high number of points. The red team was motivated to hop on to the next networks as they had services to bring down and finding the flag on the last network gave them an even higher number of points. The only possible way for the red team to get to the final flag was by pivoting through the various subnets in the competition infrastructure.

For the competition, we designed extensive packets for both teams outlining the competition's rules, scoring criteria, injects, pre-bake day access, and flags the links to which can be found in the Appendix.

Once we had the competition deployed, it was time to delve deeper into the data collection. We configured cron jobs on each machine and router to run tcpdump on startup to capture all network traffic. Before the start of the main competition, we ran an internal test-run competition among team members to analyze machine and router load capacity and performance.

We were able to successfully conduct 3 days of competition and captured roughly around 15GB of raw data that we extracted from the router and every machine on VASE.

B. Data Collection

Following the conclusion of a three-day competition, we collected 54 raw pcaps from all host machines and the router, resulting in almost 15GB of traffic data. The competition was designed such that attackers had to traverse through three networks, affording us the opportunity to investigate potential stepping-stone attacks through the identification of two-hop points. As part of facilitating reconnaissance during the competition, participants from the blue and red teams were given pre-bake time, which allowed them to install any necessary tools for exploitation and allow them to create multiple entry points, given that internet access had been taken down during the competition.

To analyze, Python was used to open packet captures and perform data analysis. Python is one of the most common machine learning tools used, hence we used Python to develop our data analysis codes. We implemented the Scapy library in Python which would open all the packet captures one by one and process them to extract data according to our requirements. Once raw data is collected from Python and stored in variables, we then use the Plotly library to convert this raw data into statistical graphs or better interpretation and understanding of stepping-stone attacks. Through those codes, we have generated a graph representation of how much traffic was generated from each of the subnets of the infrastructure.

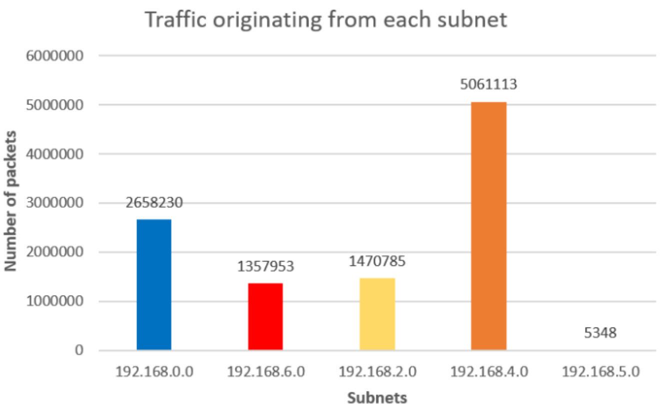

Fig. 3. Number of packets originating from each subnet.

To analyze the data collected, we divided it into day wise to faithfully show progress throughout each day and performed a thorough investigation. This allowed us to analyze traffic patterns and determine how attackers exploited vulnerabilities and compromised networks.

1) ***Day 1: 27th March:*** During the initial phase of the competition, we gathered 1.7 GB of data and conducted a thorough analysis of the raw pcaps. Our investigation revealed

TABLE II
NUMBER OF PACKETS GENERATED PER PROTOCOL

Day	Protocol	Source IP Addresses	Destination IP Addresses	Number of Packets
Day 1	ICMP	192.168.6.2 - 192.168.6.12	192.168.0.2 - 192.168.0.14	1030
	HTTP	192.168.6.12 , 192.168.6.5, 192.168.6.4	192.168.2.6	36
	SSH	192.168.6.12, 192.168.6.11, 192.168.6.9, 192.168.6.10, 192.168.6.4, 192.168.6.8, 192.168.6.7, 192.168.6.5	192.168.2.6, 192.168.2.5, 192.168.2.8	18627
	SMB	192.168.6.4, 192.168.6.13	192.168.2.2, 192.168.2.3	5
Day 2	ICMP	192.168.2.5,192.168.2.6, 192.168.2.9	192.168.4.2, 192.168.4.3, 192.168.4.4, 192.168.4.5	110
	SSH	192.168.6.11, 192.168.6.10, 192.168.2.5, 192.168.2.6, 192.168.0.2 – 192.168.0.14	192.168.2.5, 192.168.2.6, 192.168.4.2, 192.168.4.3, 192.168.2.0 – 192.168.2.8	22045
	HTTP	192.168.6.11, 192.168.6.4	192.168.2.1, 192.168.2.3, 192.168.2.4, 192.168.2.6	165
Day 3	FTP	192.168.4.2, 192.168.4.4	192.168.5.2	38
	SSH	192.168.6.11, 192.168.6.10, 192.168.2.6, 192.168.2.5, 192.168.4.3	192.168.2.6, 192.168.2.5, 192.168.4.2, 192.168.4.3, 192.168.5.3	23145
	ICMP	192.168.4.2, 192.168.4.4	192.168.5.2, 192.168.5.2	408
	SMB	192.168.2.8, 192.168.2.3	192.168.4.3	45

that the red team was able to successfully penetrate the .2 network, achieving the first hop count. To gain a better understanding of the data, we separated the internal and external data received on the router interfaces. Since we had designated interface vmx3 to the red team, we expected that all malicious traffic would originate from that interface. To distinguish between malicious and benign data, we closely scrutinized the red team's pivot onto the next network, as well as their use of compromised machines within the internal network to move further. Additionally, we had prior knowledge of the IP ranges of our internal networks, which enabled us to filter the packets received from external subnets and identify the attacker's attack paths.

*2) **Day 2: 29th March**:* On day two of the competition, we analyzed the red team's progress from 6 GB of collected data. By the end of day one, some members of the red team had infiltrated our internal network using compromised hosts. This generated traffic from vmx3 (red team) and vmx5 (internal network 1). The red team conducted reconnaissance, moved laterally to identify vulnerabilities, and sought specific flags to exploit Offsites' weaknesses. Meanwhile, the blue team hardened services and changed passwords. The red team aimed to gain access to the final network and extract critical information.

*3) **Day 3: 3rd April**:* During the last day of the competition, we collected a substantial amount of data, amounting to 7 GB, and carefully analyzed the raw packet captures. Throughout this final day, the red team was highly active and engaged in performing extensive reconnaissance across all internal networks. However, their primary focus appeared to be centered on identifying vulnerabilities and locating flags within the Offsites network (192.168.4.0) that would facilitate their ability to pivot and gain access to the final network.

*4) **Overall Analysis**:* After analyzing Table II, it was discovered that the red team had targeted the blue team's IP ranges using several ICMP packets, with the aim of compromising machines that already had access to the internal network. Analysis of network traffic revealed that several IP addresses had accessed the hospital's web server, which led to HTTP protocol traffic. While some of the packets were not malicious, the protocols used suggested that some of the IP addresses may have conducted reconnaissance activities during their visits to the website. The red team also utilized tools such as Nmap to generate a lot of ICMP and TCP packets. The team's IP addresses recorded a lot of successful and unsuccessful SSH connections, indicating that they may have obtained credentials through reconnaissance. By the end of the first day of analysis, the traffic was classified into different types, as depicted in figure 4.

On Day 2, it was noticed that a significant amount of ICMP traffic was directed from the internal network's compromised machines to the next network (192.168.4.0/24) that the attackers aimed to target. However, the blue team took down the SSH service on the first internal network, leading to some red team members losing their access. Despite this setback, the red team persisted and attempted multiple SSH connections, eventually gaining access and pivoting onto the next network, which proved to be a crucial point in the attack path, leading to the second hop count.

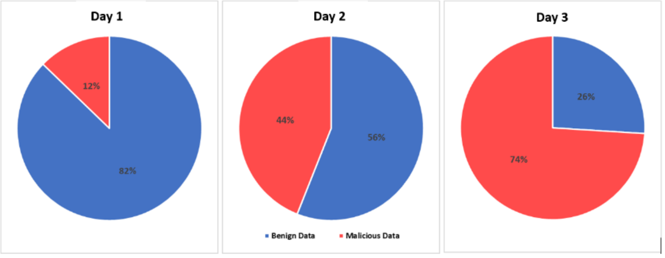

Fig. 4. Traffic Segregation - Benign Data vs Malicious Data

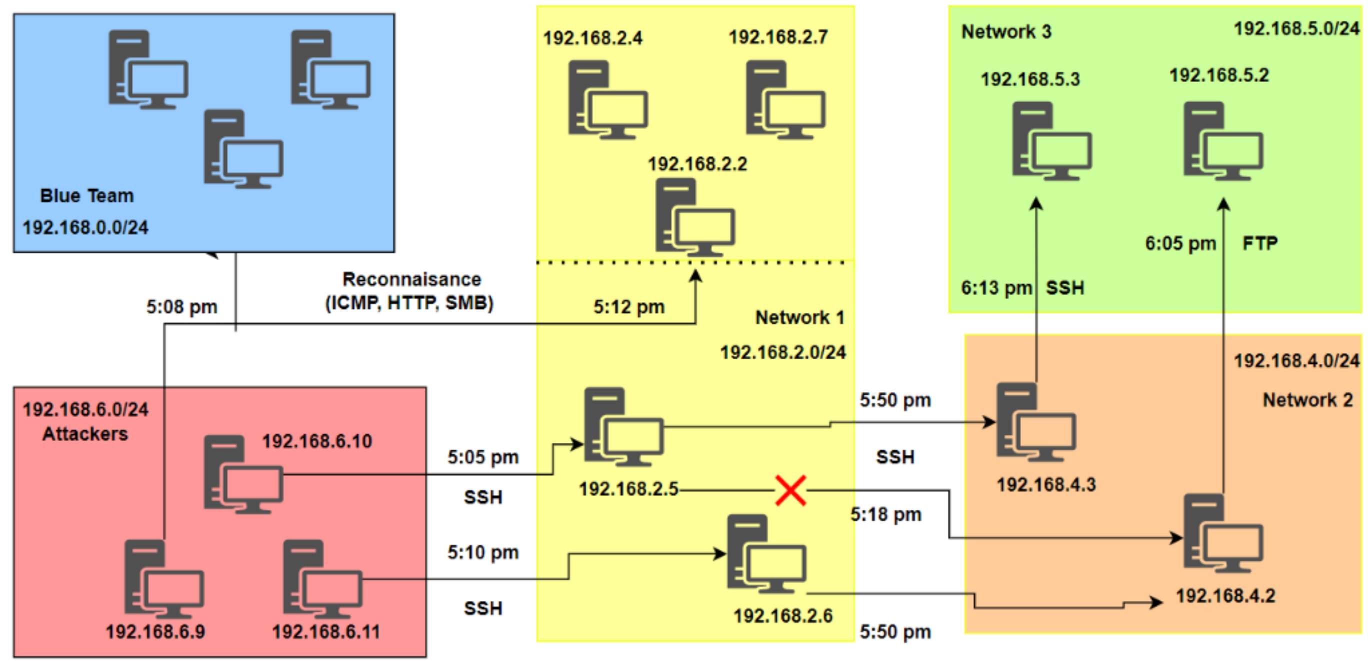

Fig. 5. Attack Path of the Attackers.

On Day 3, the red team used ICMP and SMB packets in the reconnaissance phase. The blue team's efforts to harden the SSH service on one of the machines did not deter the attackers. By exploiting an FTP server and anonymous login, the attackers discovered a flag that hinted at the existence of a Local File Inclusion (LFI) vulnerability on another machine, which they were able to exploit and gain SSH access to the final network. As a result, the attackers were able to pivot across the network, and a total of 3 hop counts were recorded. The final analysis indicated that most of the malicious data generated during the competition occurred on the third day, with the traffic progression depicted in figure 4.

VI. Results and Analysis

A. *Attack Path*

We have generated the attack path after closely analyzing all the network traffic data from all three days of the competition. This attack path in Figure 5 shows the full picture of a pivoting attack.

The attackers from the subnet 192.168.6.0 began their attack by identifying weak credentials on a machine in the first internal network, which allowed one of the members of the red team to quickly gain access. At the same time, attackers started performing reconnaissance on both the blue team and Network 1 using different scanning tools. Within 10 minutes, two attackers were able to pivot onto network 1 via an SSH connection, after which they continued their reconnaissance to identify further vulnerabilities to exploit. The attackers moved laterally along the same network, attempting to gain access wherever possible. One of the attackers, with the IP address 192.168.2.5, unsuccessfully attempted to SSH into Network 2. However, at 5:50 pm, both attackers were finally able to pivot onto Network 2 (192.168.4.0/24), after spending an average of 45 minutes on the first internal network. Within the next 15 minutes, one of the attackers was able to perform an FTP connection to 192.168.5.2 (Network 3) and retrieved flags that gave a hint to pivot to 192.168.5.3 host machine (Network 3), which they shared with the other attacker. Finally, at 6:13 pm, an SSH connection was made to 192.168.5.3, and the attackers were able to find the final flag by compromising Network 3.

B. *Tunneling Time*

In this paper, tunneling time is defined as the amount of time (seconds) spent by an attacker in a single machine to gain a remote connection to one of the machines in the next network. The tunneling time differs for each attacker as the competition progresses, for example, if an attacker is trying to establish a successful connection from one machine to another, then the amount of time spent to establish a successful connection will be considered as tunneling time generated to access the next machine. The average tunneling time between networks was calculated by analyzing the flow records of the SSH packets and specifically looking at the td column.

Table III provides a detailed breakdown of the tunneling time for three different attackers. Let's focus on the tunneling time for attacker 2, whose IP address is 192.168.6.11. To gain access to the first network (192.168.2.6), the attacker spent a total of 931.429 seconds. This time included reconnaissance, taking down services, and finding flags before finally being able to successfully gain access to one of the hosts in Network 1. During the 3-day competition, this attacker established six successful SSH connections to the first network (Figure 6).

TABLE III
TUNNELING TIME (TT) FOR ATTACKERS IN EACH NETWORK

Src IP Addr	**TT : 1st Hop**	**TT : 2nd Hop**	**TT : 3rd Hop**	**Total Time per Machine**
192.168.6.10	2747.35	185.354	1380	120
192.168.6.11	931.429	649.388	900.615	600.821
192.168.6.7	443.299	538.4	0	0

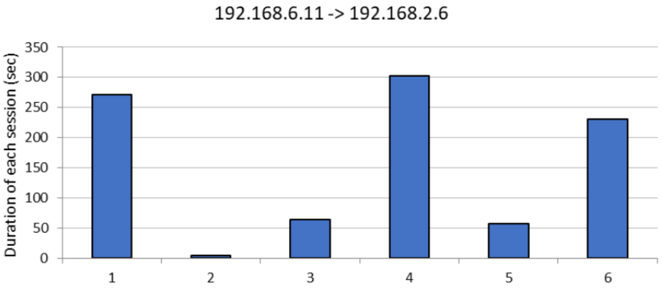

Fig. 6. SSH Connections from Attacker to Host in the First Network

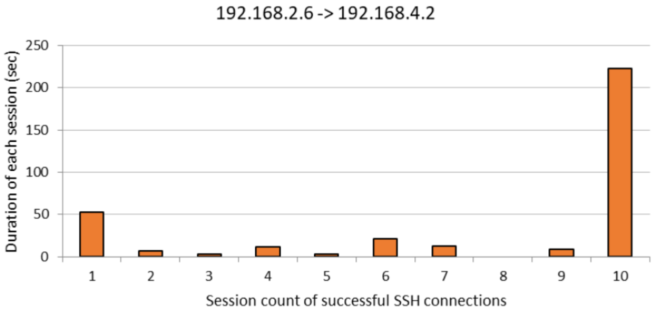

Fig. 7. SSH Connections from Attacker to Host in the Second Network

Once on Network 1, the attacker spent 649.388 seconds on the host (192.168.2.6) performing reconnaissance to identify vulnerabilities in Network 2. From this compromised machine the attacker then established ten successful connections during the entire competition duration (see Figure 7).

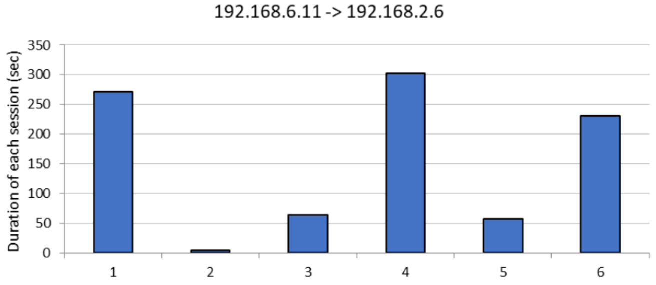

Fig. 8. SSH Connections from Attacker to Host in the Third Network

After pivoting to the host on Network 2, the attacker spent 900.615 seconds searching for flags that provided hints on how to gain access to Network 3, as SSH had been hardened and was not accessible. Finally, the attacker was able to perform an anonymous FTP login and spent 600.821 seconds navigating through directories to find the final flag. This flag aided another attacker in compromising another host in Network 3 and finding the ultimate flag. Thus finally to pivot to the final network, the attacker established three sessions (see Figure 8).

C. SSH Streams

In SSH connections, every character that the user types is transmitted to the destination machine. However, in the context of a pivoting attack, the attacker sends the characters to the pivot machine, which then forwards them to the destination machine. This characteristic presents an opportunity to identify potential pivoting attacks by examining the patterns of data transfer in the network.

We conducted an analysis of the pcaps that were collected from the host machines presented in the attack path. These pcaps were used to extract bidirectional SSH streams, which were then analyzed to identify the ingoing and outgoing streams that were present in the attack path, such as the connection between 192.168.6.11 and 192.168.2.6, as well as the connection between 192.168.2.6 and 192.168.4.2. The aim of this analysis was to identify any characteristics that could distinguish pivoting traffic from normal traffic.

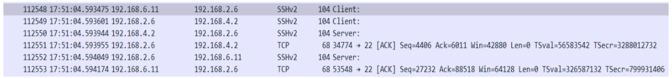
```
112548 17:51:04.593475 192.168.6.11     192.168.2.6     SSHv2    104 Client:
112549 17:51:04.593601 192.168.2.6      192.168.4.2     SSHv2    104 Client:
112550 17:51:04.593944 192.168.4.2      192.168.2.6     SSHv2    104 Server:
112551 17:51:04.593955 192.168.2.6      192.168.4.2     TCP       68 34774 → 22 [ACK] Seq=4406 Ack=6011 Win=42880 Len=0 TSval=56583542 TSecr=3288012732
112552 17:51:04.594049 192.168.2.6      192.168.6.11    SSHv2    104 Server:
112553 17:51:04.594174 192.168.6.11     192.168.2.6     TCP       68 53548 → 22 [ACK] Seq=27232 Ack=88518 Win=64128 Len=0 TSval=326587132 TSecr=799931406
```

Fig. 9. Pcaps Demonstrating SSH Connections from Attacker to Internal Networks

Figure 9 displays a snapshot of a network capture file (pcap) after an SSH connection has been established from the pivoting machine to the destination machine. This scenario involves two separate SSH connections: the first connection is from the attacker's machine (IP address: 192.168.6.11) to the pivoting machine (IP address: 192.168.2.6), and the second connection is from the pivoting machine to the destination machine (IP address: 192.168.4.2). The figure illustrates the pivoting technique, where the pivoting machine initiates a connection to the destination machine while an existing connection is already established between the attacker and the pivoting machine.

Netflow statistics were used to verify the validity of a connection chain by tracing it back from the destination machine. In this analysis, we focused on a single successful and long-term connection chain, specifically identifying the timestamp at which the pivoting behavior was observed. The network traffic from a host machine (192.168.2.6) acting as a pivot node for the attacker (192.168.6.11) was analyzed, including incoming and outgoing traffic measured in bytes and timestamps. Our analysis reveals that by examining the SSH connections from the perspective of pivot nodes, specific features unique to pivoting traffic can be identified. These characteristics can then be used to detect and prevent pivoting attacks, which can pose a significant threat to network security.

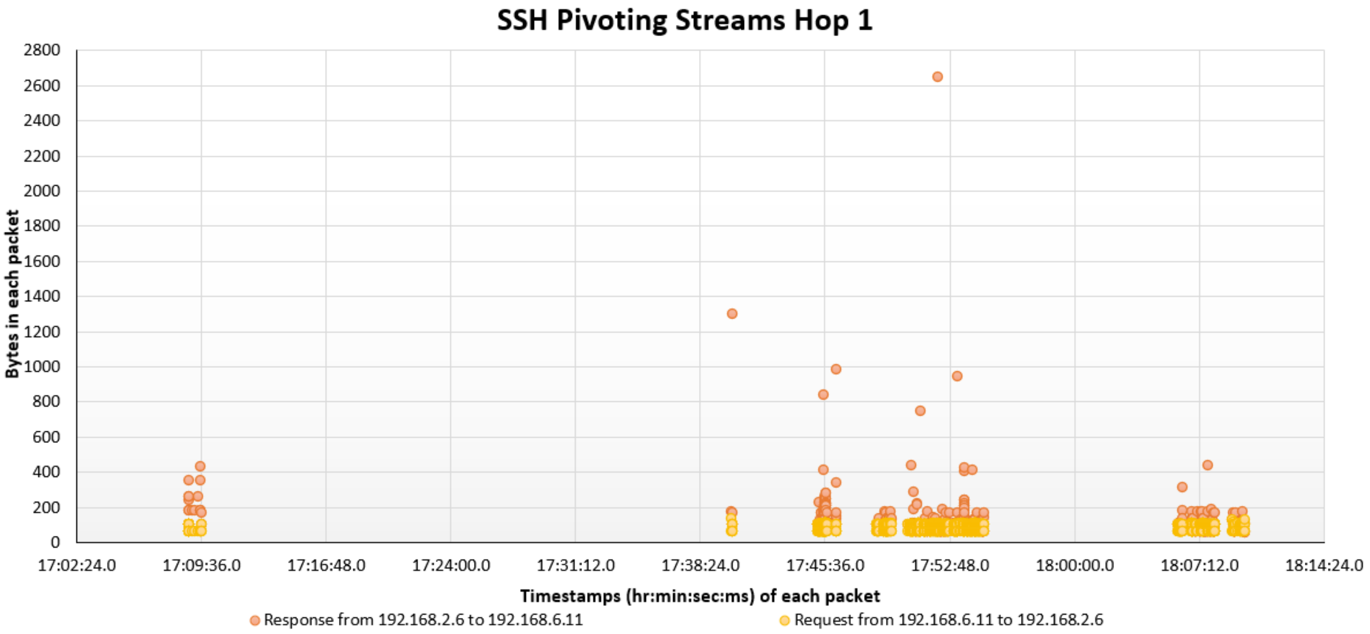

Fig. 10. SSH Pivoting Streams Hop 1

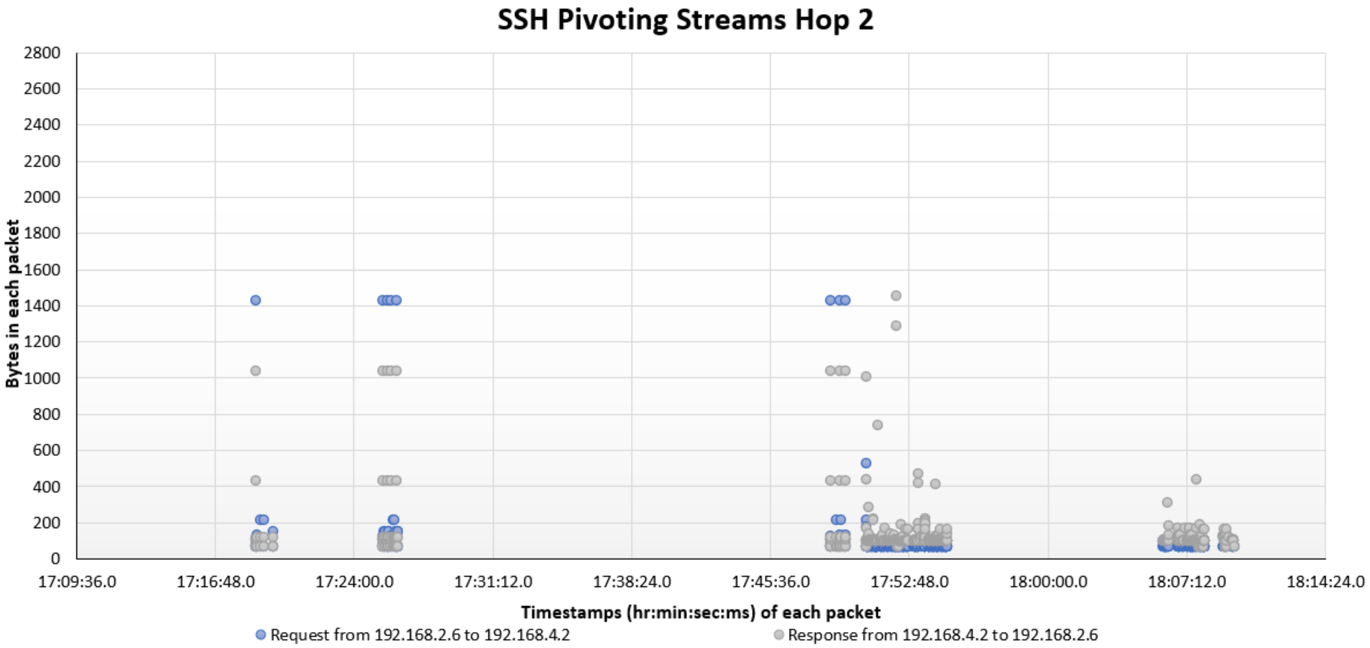

Fig. 11. SSH Pivoting Streams Hop 2

The presented graph depicts SSH traffic extracted from the pivoting machine, highlighting the attacker's access pattern during the competition. Initially, the attacker was able to access only the first machine, i.e., the pivoting machine (192.168.2.6), representing the first hop or pivot. From 17:09:36.0 until 17:45:36.0, the attacker sent numerous SSH commands to the pivoting machine, which are represented by the orange and yellow dots in the graph. These dots demonstrate the frequency of SSH requests and responses exchanged between the attacker and the pivoting machine during this time.

At 17:49:04.7, the attacker successfully established an SSH connection with the destination machine (192.168.4.2) via the pivoting machine. The graph clearly shows the SSH connections between the pivoting machine and the destination machine, represented by blue and gray dots. Interestingly, the yellow and orange dots overlap with the blue and gray dots, indicating that the responses to the attacker's requests sent to the pivoting machine were forwarded to the destination machine. This observation correlates with the timestamps captured in the Wireshark pcap file shown in Figure 9, indicating the establishment of a pivoting connection where both SSH connections were open simultaneously.

VII. Challenges and Lessons Learned

Over the course of our capstone, we faced several technical challenges. These are listed below:

- **Selecting a suitable deployment platform:** Initially, we considered RLES, but it had scalability and load-balancing issues. To address this, we collaborated with the Cyber Defense Techniques class and used the platform OpenVase instead.
- Expanding the Stepping Stones network: Our initial plan involved the implementation of multiple routers, but we soon realized that in order to ensure optimal performance, we would need additional networks to facilitate trunking between these routers. However, due to competition rules and limited resources, we were unable to allocate more networks and had to make do with the networks that were already assigned. This presented a challenge that we had to overcome by modifying our infrastructure and making the best use of the available resources.
- **Capturing and storing data:** Initially, we had concerns about the ability of a single pfsense to handle the competition traffic and capture data efficiently. To address this, we added a 100 GB hard drive to the pfsense router. This ensured we had enough storage space to capture and store all the required data without compromising router efficiency.
- **Establishing firewall rules:** Establishing firewall rules on the router became tremendously hard and the command line was not easy to work with. We had to establish firewall rules on the router which would accept only selected network traffic into a certain network.
- **Extracting data:** Extracting data was challenging due to hardened SSH and Telnet, as well as internet access issues on the VASE platform. We had to bring back services and it took time to extract all the raw pcaps.

VIII. Future Works

Our statistical analysis, which considered time delays and differences between routers and hosts, revealed that a future detection system could monitor incoming streams based on specific criteria and detect any outgoing connections. We could use machine learning to develop models that can automatically detect pivoting attacks.

We could also use the data to develop new methods for mitigating pivoting attacks. We suggest using k-means clustering analysis to differentiate between automated and manual attacker attempts. To enhance our project infrastructure, we recommend integrating a centralized logging solution such as Security Onion, using netflow capture with nfdump, visualizing traffic patterns with NfSen, and logging host activities with Sysmon and Auditd. Finally, generating alerts with Zeek for suspicious traffic can provide valuable context for detecting and mitigating future pivoting attacks. These proposed improvements have the potential to significantly enhance our ability to detect and respond to pivoting attacks.

In addition to the above, we believe that there are several other areas that could be improved in future competitions to improve our understanding of pivoting attacks. These include:

- **Increasing the number of participants:** The more participants there are, the more data will be collected, which will lead to a more robust dataset.
- **Increasing the complexity of the network:** A more complex network will provide a more realistic environment for attackers to operate in, which will lead to more realistic data.
- **Analyzing the types of vulnerabilities that are exploited in pivoting attacks:** This information could be used to develop more effective security solutions that can prevent these vulnerabilities from being exploited.
- **Analyzing the impact of pivoting attacks on organizations:** This information could be used to develop more effective mitigation strategies that can reduce the impact of pivoting attacks.

IX. Conclusion

In this report, we have presented a novel approach to collecting data on pivoting attacks. Our approach involved designing and conducting a simulated competition environment in which participants were encouraged to exploit vulnerabilities and pivot through networks. After conducting a simulated competition environment, we were able to generate a significant number of instances of pivoting attacks. By analyzing the vast amount of raw data collected, we successfully identified the path that the attackers may have taken to pivot through each network and ultimately reach the final network. Our analysis involved examining various characteristics, including byte count and packet count, between incoming and outgoing streams, which can be used to establish approximate limits. Overall, our findings provide valuable insights into the tactics and methods used by attackers during pivoting attacks.

References

[1] L. Wang and J. Yang, "A research survey in stepping-stone intrusion detection," *EURASIP Journal on Wireless Communications and Networking*, vol. 2018, pp. 1–15, 2018.

[2] Y. Zhang and V. Paxson, "Detecting stepping stones.," in *USENIX Security Symposium*, vol. 171, p. 184, 2000.

[3] M. Husák, G. Apruzzese, S. J. Yang, and G. Werner, "Towards an efficient detection of pivoting activity," in *2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pp. 980–985, IEEE, 2021.

[4] R. S. Marques, H. Al-Khateeb, G. Epiphaniou, and C. Maple, "Apivads: A novel privacy-preserving pivot attack detection scheme based on statistical pattern recognition," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 700–715, 2022.

[5] A. Greco, G. Pecoraro, A. Caponi, and G. Bianchi, "Advanced widespread behavioral probes against lateral movements," *Int. J. Inf. Secur. Res*, vol. 6, no. 2, pp. 651–659, 2016.

[6] R. Vera, A. F. Shehu, T. Dargahi, and A. Dehghantanha, “Cyber defence triage for multimedia data intelligence: Hellsing, desert falcons and lotus blossom apt campaigns as case studies,” *International Journal of Multimedia Intelligence and Security*, vol. 3, no. 3, pp. 221–243, 2019.

[7] M. Ring, D. Schlör, D. Landes, and A. Hotho, “Flow-based network traffic generation using generative adversarial networks,” *Computers & Security*, vol. 82, pp. 156–172, 2019.

[8] A. Kenyon, L. Deka, and D. Elizondo, “Are public intrusion datasets fit for purpose characterising the state of the art in intrusion event datasets,” *Computers & Security*, vol. 99, p. 102022, 2020.

[9] G. Apruzzese, F. Pierazzi, M. Colajanni, and M. Marchetti, “Detection and threat prioritization of pivoting attacks in large networks,” *IEEE transactions on emerging topics in computing*, vol. 8, no. 2, pp. 404–415, 2017.

[10] R. S. Marques, H. Al-Khateeb, G. Epiphaniou, and C. Maple, “Apivads: A novel privacy-preserving pivot attack detection scheme based on statistical pattern recognition,” *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 700–715, 2022.

X. APPENDIX

This is the Google Drive link to where we have hosted all our scripts and raw pcaps that we collected.

Link: Google Drive Link