Unraveling Network-Based Pivoting Maneuvers: Empirical Insights and Challenges

Martin Husák[1,2(✉)], Shanchieh Jay Yang[3], Joseph Khoury[2,4], Đorđe Klisura[2,4], and Elias Bou-Harb[2,4]

[1] Institute of Computer Science, Masaryk University, Brno, Czech Republic
husakm@ics.muni.cz
[2] The Cyber Center for Security and Analytics, The University of Texas at San Antonio, San Antonio, TX, USA
[3] Department of Computer Engineering, Rochester Institute of Technology, Rochester, NY, USA
jay.yang@rit.edu
[4] Division of Computer Science and Engineering, Louisiana State University, Baton Rouge, LA, USA
{jkhour5,dklisu1,ebouharb}@lsu.edu

Abstract. Pivoting is a sophisticated strategy employed by modern malware and Advanced Persistent Threats (APT) to complicate attack tracing and attribution. Detecting pivoting activities is of utmost importance in order to counter these threats effectively. In this study, we examined the detection of pivoting by analyzing network traffic data collected over a period of *10* days in a campus network. Through NetFlow monitoring, we initially identified potential pivoting candidates, which are traces in the network traffic that match known patterns. Subsequently, we conducted an in-depth analysis of these candidates and uncovered a significant number of false positives and benign pivoting-like patterns. To enhance investigation and understanding, we introduced a novel graph representation called a pivoting graph, which provides comprehensive visualization capabilities. Unfortunately, investigating pivoting candidates is highly dependent on the specific context and necessitates a strong understanding of the local environment. To address this challenge, we applied principal component analysis and clustering techniques to a diverse range of features. This allowed us to identify the most meaningful features for automated pivoting detection, eliminating the need for prior knowledge of the local environment.

Keywords: pivoting · lateral movement · monitoring · NetFlow

1 Introduction

Lateral movement has become a major research topic in network security [22]. Adversaries are always finding new ways of breaching systems and avoiding detection, often by moving laterally in the target network. The most valuable targets

The original version of the chapter has been revised. A correction to this chapter can be found at https://doi.org/10.1007/978-3-031-56583-0_23

S. Goel and P. R. Nunes de Souza (Eds.): ICDF2C 2023, LNICST 571, pp. 132–151, 2024.
https://doi.org/10.1007/978-3-031-56583-0_9

are often not accessible from the Internet, or the network is protected by intrusion detection systems (IDS) on the perimeter. The goal of the adversary in such cases is to get a foothold elsewhere in the network, such as on a common workstation exploited by social engineering attack (e.g., a phishing email) or a weakly secured IoT device. Creating a backdoor to such a device allows the adversary to use it as a *pivot* and connect to other targets in the network from within. The term *pivoting* [2,22] refers to such a scenario and can also be referred to as *island hopping* [23], *stepping stone attack* [33] or *command propagation* in the literature. Pivoting is no longer an advanced attack technique reserved for Advanced Persistent Threats (APT) and other advanced adversaries [9,22] but is more and more often seen adopted by novel malware [8,26].

Although pivoting or lateral movement detection, in general, has gained much attention in recent years [22], the state-of-the-art in the field is limited by several factors. First, the existing pivoting detection methods (e.g., [5,7,18]) are mostly host-based, meaning they can only detect the pivoting on (or with access to the data from) the machine that acts as a pivot. While such methods achieve high accuracy, their scope is limited to machines that have the necessary software equipment or those that can forward their logs elsewhere, which is often infeasible in large and heterogeneous networks. Moreover, the attacker may exploit a common workstation or IoT device as a pivot, where such devices would likely not be equipped with proper detection mechanisms. Thus, a network-based approach is vital and could play a key complementary role in such a context. Second, the related works mostly evaluated the detection capability using datasets or in environments with an insufficient amount of background traffic. Thus, existing approaches may achieve a high true positive rate but also a high false positive rate because it is not clear what false positive or benign events can be detected. Attempts to approach this problem were made [15] but needed to deliver long-term measurements or a detailed analysis of the false positives. We aim herein to fill this gap by detecting pivoting in real-world settings while differentiating between benign and suspicious events.

The contributions of this work to the state-of-the-art can be summarized as follows. First, we employ a modified state-of-the-art network-based pivoting detection algorithm [2] to detect pivoting and pivoting-like events in a campus network, focusing on SSH communication. Following the observations in related work [15], we employ a two-layer detection tactic starting with pivoting *candidate* detection with a high true positive rate followed by a second analytical phase aiming at false positive reduction leading to the selection of true positive candidates. The scope of this measurement vastly exceeds any experiments in related work. Second, we empirically analyze the measurement results, identify true and false positives, and investigate the benignity or maliciousness of the detected events. To this end, we (*i*) extract a list of heuristics based on knowledge of the local environment and convert them into rules for automatic annotation. Consequently, we (*ii*) devise a novel graph-based representation of pivoting activities, which provides comprehensive visualization and additional contextual features. Further, (*iii*) we study the evolution of pivoting-like events over time. Third,

we perform principle component analysis and clustering in order to identify the most meaningful features and feature sets that would allow for the design and development of a (semi-)automated pivoting detection tool without relying on local knowledge.

The remainder of this work is structured as follows. Section 2 comprehensively summarizes the background and relevant related work. Section 3 inititally presents the scenario and experiment setup for pivoting detection in the campus network. Subsequently, the pivoting detection algorithm and measurement results are described. Section 4 presents a detailed analysis of the measurement results using three approaches, heuristic filtering, graph-based representation, and timing analysis. Section 5 presents the approaches taken toward for the automation of the analysis. Section 6 discusses the measurement findings and their implication towards in-practice, pragmatic usages. Section 7 concludes the paper and paves the way for future work.

2 Background and Related Work

In this section, we first define the pivoting maneuver and its characteristics. Then, we provide the necessary background on network measurements. We also provide an overview of the related work on pivoting detection.

2.1 Pivoting Maneuver and its Characteristics

Pivoting, also known as island hopping or stepping stone attack, is gaining more and more popularity among attackers. The documented cases of cyber attacks involving pivoting include the events of Operation Aurora [8], in which the attackers gained control over the system of large corporations and exfiltrated business secrets. In 2015, the Ukrainian power grid faced a complex attack involving pivoting [12]. Such an attack caused a blackout in hundreds of thousands of households. The MEDJACK attack [26] abused hospital equipment, such as X-ray scanners, to exfiltrate data on patients. The report by TrapX Security [28] comments on other pivoting-based attacks in the healthcare domain. SamSam is an example of ransomware leveraging pivoting activities [1], while the Archimedes tool [31] uses pivoting and forwards network traffic to fake websites to steal authentication credentials. In line with the strategic pivoting tactics, amid the Russo-Ukrainian conflict, an attacker laterally maneuvered within the trusted management network of KA-SAT which eventually allowed him to execute legitimate, targeted management commands on a large number of residential modems simultaneously [30].

Here, we define pivoting as a pair of network connections involving three actors. First, the *Source* initiates a connection to the *Pivot*. Subsequently, either immediately or up to ϵ seconds later, the *Pivot* initiates a connection to the *Target*. The *Target* is different from the *Source*. The scenario is depicted in Fig. 1.

A malicious case of pivoting may involve an attacker located anywhere on the Internet as the *Source*. The attacker aims at a *Target* that is behind a firewall or in a private network segment. Thus, the attacker exploits a *Pivot* first since

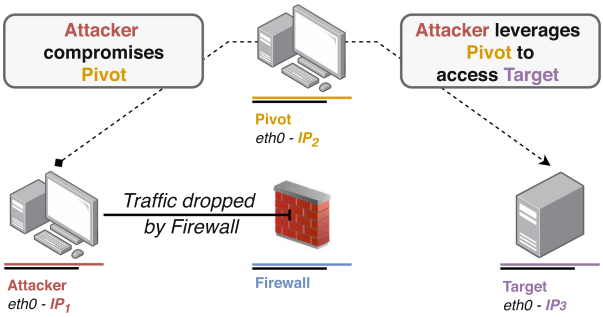

Fig. 1. An illustrative depiction of a pivoting maneuver through SSH.

it has both a public IP address and can reach the *Target*. However, pivoting can also be benign and even part of a typical daily workflow. For example, a user working from home connects to a publicly accessible SSH server in the employer's network and uses it to connect to another device in the network that is not directly accessible. This is essentially pivoting but conducted by a legitimate user and, thus, is benign unless it violates internal security policy that would, for example, prohibit connecting to SSH from outside of the network.

2.2 Related Work on Pivoting Detection

Despite the rising frequency of attacks involving pivoting, the research on detecting such events it is still scarce or limited in its applicability [22]. Earlier works [3,27] conceptualized pivoting attacks without proposing a detection method. Since pivoting and lateral movement is often a part of APT attacks as carefully studied by Gonzales et al. [13] according to the MITRE ATT&CK framework, it was mostly studied in terms of detecting and preventing APT [9,22]. The earliest works focused on alert correlation, not processing raw data. The foundational work by Valeur et al. [29] illustrates the correlation of alerts raised by IDS (Intrusion Detection Systems). However, the malicious activity may avoid being detected or would not trigger an alert, which complicates this approach [21].

The topic of lateral movement spans intrusion detection as well as forensics. Liu et al. [16] proposed Latte, a lateral movement detection system based on graphs with computers and users as nodes and connection and logon events as edges. Their approach is host-based and bridges forensic analysis and detection. Wilkens et al. [32] researched the reconstruction of lateral movement. Their contribution is a detection method where, using indicators of compromises, suggests the path of the attacker's lateral movement and narrows down the set of nodes to analyze to only 5% of all network hosts.

Host-based approaches are the most common in the proposed pivoting detection research. Here, Bai et al. [4,5] proposed an approach to detect lateral movement in RDP logs; the work is limited only to Windows-based hosts. Their

approach utilizes an ML classifier which yielded high performance illustrated on several datasets while also being robust against adversarial attacks [5]. Bian et al. [7] further elaborated on the topic, scrutinizing graph-based features and conducting dimensionality reduction. However, the low quality of network flow data in the used dataset prevented the authors from including such data in their approach. Recent approaches to lateral movement detection do not rely solely on system logs but combine multiple data sources, including monitoring network traffic. APIVADS [18] is a privacy-preserving approach to pivoting detection that can be used in complex networks. The proposed approach relies on NetFlow data collected on the pivot and, thus, it is a de-facto host-based method, even though network-based data are used. Powell [20] proposed a role-based lateral movement detection using unsupervised learning, utilizing systems calls and network connections alike, leveraging earlier work on graph-based anomaly detection in authentication logs [19]. Smiliotopoulos et al. [24] propose a Sysmon log-based lateral movement detection technique encompassing the labelling and pre-processing of the data, as well as the classification through a supervised machine learning approach.

The first attempts at characterizing stepping stones dated back to 1995 [25] and detecting it to 2000 [33]. Even back then, the authors mention the vast false positive rates. Since then, the dynamics of network traffic and the threat landscape have fundamentally changed, and research has mostly focused on host-based methods. The work of Apruzzese et al. [2] is the state-of-the-art network-based pivoting detection algorithm. The authors proposed an algorithm that correlates NetFlow data [14] and is capable of detecting sequences of pivoting activity of arbitrary length. Such an approach is well suited to private networks with not much background traffic. However, Husák et al. [15] recently evaluated an algorithm by Apruzzese et al. [2] in operational settings and pinpointed the challenges related to that, given that they have achieved a high rate of false positive detections and too few true detections of pivoting activity involving relevant services (e.g., SSH, RDP, Telnet). The authors proposed Principal Component Analysis (PCA) to infer characteristics of true pivoting events to enable further development of ML-based detection. Another angle was considered by Dong et al. [10,11] in which they identified lateral movement traces in enterprise network by performing behavior deviation measurement.

With respect to the state-of-the-art, the proposed work herein is scoped towards the detection of pivoting activities based on network traffic analysis (namely using NetFlow); hence it aims at detecting pivoting occurring anywhere in the network. Such a goal is highly ambitious and will require long-term empirical measurements and evaluation. In this vein, we start with the setup of the pivoting detection pipeline that would enable the latter. With the help of contextual information, heuristics, and machine learning, our goal is to devise an approach which reduces the false positive rate while distinguishing between benign and suspicious (or even malicious) pivoting activities.

3 Pivoting Candidate Detection

We describe in this section the experimental environment and the measurement setup (see Fig. 2) followed by the pivoting detection algorithm and the generated results. By pivoting candidates, we refer to the outputs of the detection algorithm as is, without any post-processing, which is discussed in the subsequent sections.

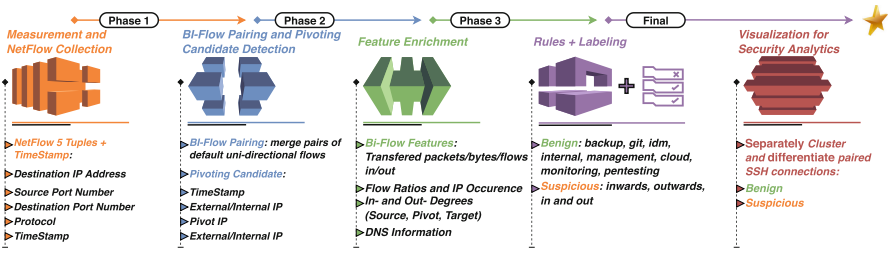

Fig. 2. Pivoting detection pipeline.

3.1 Environment and Experiment Setup

We executed a longitudinal measurement in March 2023 in the campus network of Masaryk University[1]. The campus network serves more than 30,000 students and 6,000 employees. Over 15,000 unique IPv4 addresses in the /16 IPv4 address range can be seen on a daily basis. The campus network is to be open and restrict only malicious network traffic and hazardous services rather than blocking everything and allowing only certain services. This makes the campus network an excellent environment to study benign pivoting-like activities.

The university operates a cybersecurity team CSIRT-MU[2] to manage cybersecurity in the network and operate a network monitoring infrastructure based on the NetFlow technology [14]. NetFlow monitoring enables the CSIRT-MU to perform intrusion detection and network forensics. The NetFlow probes are located at strategic locations to monitor network traffic flowing through major links in the campus network. No measurement is conducted on the routers or other active network devices. Two probes are located on the perimeter, monitoring any inbound and outbound traffic, and six are located inside the network, monitoring the most important internal links. The probes on the perimeter observe the highest traffic rates but do not have any visibility into internal network traffic, while the internal probes are capable of observing the majority of network connections within the campus network, including any connection inbound or outbound to the campus network. Thus, we chose to use only the data from the internal probes. As such, all the data from the probes are sent to two NetFlow collectors, where they are retained. Since the NetFlow measurement is primarily

[1] https://www.muni.cz/en.
[2] https://csirt.muni.cz/?lang=en.

used for security purposes, no traffic sampling is applied, even on high-speed links, to allow for precise monitoring. The active time-out is set to 30 s for the same reason, diverting from traditional settings of 5 min [14].

Notably, only SSH communication (i.e., identified by TCP destination port *22*) is considered during our measurement. The reasons are two-fold; *(i)* considering multiple protocols or protocol-agnostic detection would explode in complexity and would complicate the analysis (see the discussion); and *(ii)* SSH is widely used in the campus network and is less strictly regulated than other considerable protocols, including Telnet and/or RDP. The amount of Telnet traffic in the campus network is negligible, and RDP is strictly regulated by firewalls, which also leads to negligible amounts of observed traffic.

3.2 Candidate Detection Algorithm

The pivoting detection in NetFlow data follows two fundamental related works. First, we adapted the algorithm proposed by Apruzzese et al. [2]. Second, we enhanced the two-level approach proposed by Husák et al. [15], i.e., detecting candidates first and then using other approaches to classify the candidates as true and false positives (and benign and suspicious ones).

The algorithm by Apruzzese et al. [2] extracts the bi-flows (i.e., bi-directional network flows created by merging pairs of default uni-directional flows in opposite directions [14]) and then finds the paths of arbitrary lengths in which the next flow's source is the previous flow's destination. In addition, the new biflow has to start immediately or up to 30 s after the previous one. The algorithm is capable of detecting pivoting of arbitrary length.

We made several changes to the algorithm. First, we process data from multiple probes. Thus, all the biflows from all probes are merged into one list and sorted by timestamp. Duplicate biflows (e.g., connections observed by two or more probes) are removed. Second, we detect only simple pivoting consisting of two network connections (source to pivot, pivot to target). Pivoting over several pivots would still appear in the results as several candidates. Moreover, candidates of fixed form are easier to post-process. Finally, the time limit of 30 s was kept as a default but implemented as an optional parameter. The default time window is one day (midnight to midnight), but it is also subject to settings. The final form of the algorithm is summarized in pseudocode in Algorithm 1.

3.3 Results

The measurement and pivoting candidate detection spanned ten days. The results are summarized in Table 1. Processing the NetFlow data collected throughout the day by six probes took around 30 min on average on commodity hardware.

The pivoting detection algorithm has one parameter, namely, the time propagation delay (ϵ), i.e., a maximal time difference between the source-to-pivot and pivot-to-target communication [2]. We were interested in how various settings of ϵ influence the overall results. The values of 2, 10, 30 (default), and 60 s were

Algorithm 1. Pivoting candidate detection algorithm.

```
f ← list of flows on the input
ϵ ← 30
len ← size of f
for i in [0, len] do
    for j in [i+1, len] do
        if f_i.dstIP == f_2.srcIP then
            if f_1.ts < f_2.ts < f_1.ts + ϵ then
                candidates ← (f_i, f_j)
            end if
        end if
    end for
end for
```

Table 1. Pivoting candidate detection.

Measurement Artifacts	Min	Max	Total
Biflows	*3,416,328*	*6,412,670*	*39,399,832*
Candidates	*17,026*	*75,116*	*313,193*
Unique Sources (S)	*297*	*646*	*3,410*
Unique Pivots (P)	*64*	*112*	*238*
Unique Targets (T)	*76*	*227*	*468*
Unique Triplets (S, P, T)	*695*	*6,956*	*22,655*
Pivoting Graph Components	*12*	*21*	*14*

considered. The higher the ϵ, the higher the number of detected candidates and unique actors, but also highly increased processing time (from several minutes at lowest ϵ to close to one hour with the highest ϵ). However, the increases were observed mostly in false positive detections (see the following section). The number of suspicious candidates (i.e., those we aim to detect, see the next section) changed only marginally with various ϵ. It is also worth noting that higher fragmentation of NetFlows in time due to active timeouts is an influential factor allowing the use of low ϵ values.

4 Manual Pivoting Candidate Analysis

The second phase of the experiment is the analysis of pivoting candidates, i.e., the output of the pivoting detection algorithm. The algorithm is relatively simple but provides a solid true positive detection rate. However, it is prone to a high false positive rate, which needs to be addressed. Thus, the second phase primarily addresses the false positive rate reduction manner. Three approaches were taken and are discussed in the sequel. First, we leverage the knowledge of the environment and manually classify the candidates. Second, we employ a novel

approach based on graph-based visualization. Finally, we analyze the evolution of pivoting candidates over time.

4.1 Empirical Analysis and Heuristic Filtering

The empirical analysis of the results was conducted in collaboration with administrators of the campus network. The administrators have detailed knowledge of the environment and the roles of most of the deployed devices. The goal of this analysis was to either confirm that the detected candidate is some sort of pivoting-like activity or a false positive. Simultaneously, the empirical analysis served as a basis for the construction of a heuristic filter that can be used to filter the candidates, disregard false positives, and mark benign events. The empirical analysis was conducted manually. All the IP addresses were translated to domain names for increased comprehension. The analysts iterated the list starting with the most frequently appearing sources, pivots, targets, and their combinations. Several frequent patterns became apparent and a heuristic filter was filled with entries. A detailed breakdown of its results is displayed in Table 2.

Table 2. Rule-based annotation of pivoting candidates.

Class	Rule	Candidates
Benign and False Positives	Monitoring	*288,161*
	(Anonymized Services)	*15,761*
	Git & Backup	*5,404*
	Management & Cloud	*1,288*
	Pentesting	*1,627*
Unclassified and Suspicious	Internal	*29*
	Inwards	*338*
	Outwards	*19*
	In and Out	*566*
Total	-	*313,193*

First, a large number of candidates included one of the network measurement nodes. Tools like *Nagios* and *Icinga* deployed in the network use SSH to connect to or receive connections from other hosts in the network to update the status of the devices and services running on them. If the monitored nodes open another network connection, a candidate is detected. Either the monitored host received an SSH connection from elsewhere and then updated its status, or the monitoring node queried a monitored host, which then connected elsewhere. At one department, the administrators operate two monitoring hosts, one actively probing the devices and the second receiving updates. Simultaneous connections to and from

both of them resulted in a number of pivoting candidates with the same source and target but different pivots. All the cases can be declared as false positives, and the following filtering rule was established: if a known network monitoring node is involved in pivoting-like activity, the candidate is marked as *monitoring* and dismissed. The list of monitoring hosts can be provided by network administrators. Alternatively, if a common naming scheme is used, a rule can be stated as: if the actor's domain name is *{monitor,icinga,nagios}*.domain*, then dismiss this candidate. A network-wide penetration test from a known dedicated host taking the role of a source was also observed and was marked as *pentesting*.

Second, a similar observation was made with hosts belonging to the cloud or network management infrastructure. However, in such cases, the behavior is rather true benign pivoting than false positive. For example, a cloud management device receives a command via SSH connection from a controller and propagates it to one or more hosts in the cloud. Other network infrastructures, such as centralized identity management systems or various APIs to services in the private network display similar behavior. Again, a rule can be set: if a pivot is one of the hosts providing pivoting-like service, it is marked as *management* and considered benign or dismissed. Again, the list of hosts is provided by network administrators or derived by its domain name (e.g., *cloud-management*.domain*).

Third, some candidates involved the IP address of the Git repository or a backup device as a target. Typically, a user connected to a server via SSH did some development work, and committed the code to the Git repository via SSH, thus generating a pivoting-like event. This is a benign scenario and could be filtered by a rule checking for a domain name of a target containing strings, such as **.github.com*, *gitlab*.domain*, or *backup*.domain*. One distributed system in the network used three hosts constantly updating each other via SSH, generating many pivoting-like events (without interacting with other hosts in the network). This was marked separately, and the rule contained three distinct domain names.

The heuristic filtering marked the vast majority of candidates as false positives or benign events. The remaining candidates (less than 1%) were sorted by the location of the actors. A small number of candidates had all three actors inside the network. Since no interaction with the Internet was observed, we may assume these candidates as benign.

Other unclassified candidates involved an actor outside the network, which was either the source (*inwards* scenario), target (*outwards* scenario), or source and target *in and out scenario*. In such scenarios, actors outside the network may pose a danger. Moreover, the vast majority of such actors and candidates were unique in the sense that they appeared only once during the measurement.

We were mostly interested in the *inwards* scenario since it corresponds the most to the attack model. A few detected candidates were attributed to users working from home and pivoting through their own SSH servers to internal services, which turned out to be the most problematic observed behavior (the use of VPN is recommended instead). Although no violation of security was observed, the few observed *inwards* candidates served as true positive samples.

Algorithm 2. Pivoting graph construction.

```
G ← new empty directed graph
for each candidate do
    for N in S, P, T do
        if N not in G then:
            insert node X
        end if
        if (S,P) not in G then:
            insert edge (S,P)
        end if
        if (P,T) not in G then:
            insert edge (P,T)
        end if
    end for
end for
```

The remaining candidates in the *in and out* and *outwards* scenarios were found to be associated with cloud computing, which was apparent from the domain names of the actors. The communication patterns involved hosts in cloud environments of the campus network, clouds of other institutions (collaborating universities), or public cloud service providers. Although we cannot tell what happened in those events (the cloud services are outside of our scope, and data sharing and distributed computing are expected there), we did not observe anything highly suspicious or clearly malicious. Setting a new location (*cloud*) should be considered for future work.

4.2 Visual Analysis via Pivoting Graph and Its Decomposition

In the second part of the manual analysis, we investigated an approach based on graph-based visualization. We composed the *pivoting graph* and subsequently decomposed it into components that are easier to evaluate than a list of candidates.

The pivoting graph is a directed graph that represents the network's hosts as nodes and the connections between them as edges. However, it is important to note that the construction of the pivoting graph is based solely on pivoting candidate events rather than capturing all network (or SSH) traffic within the network. The process of constructing the graph is outlined in Algorithm 2.

The pivoting graph contains all the network hosts that were detected to be involved in pivoting maneuvers at any role. The motivation for constructing the graph is correlating all the pivoting candidates that share some of the actors or (recalling the limitation to only one pivot per candidate) merging pivoting events with more pivots. Indeed, the pivoting graph represents various situations well. Moreover, it is also a graph with many unconnected components, which allows for approaching each component individually, which turned out to be highly valuable.

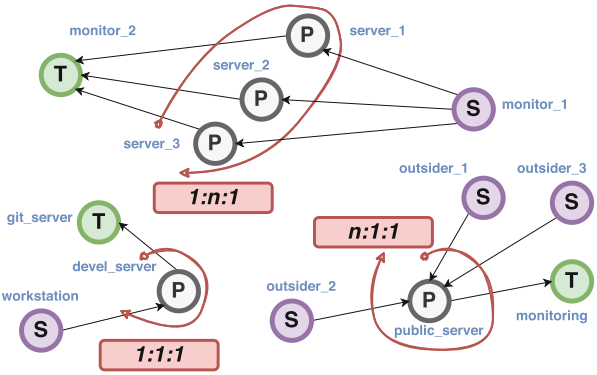

Fig. 3. An excerpt from the pivoting graph displaying the components of three different patterns.

The graph decomposition partitions the graph into components, precisely weakly connected components since the pivoting graph is directed. Standard decomposition algorithms from the NetworkX library [17] were used in this work. During the experiment, we found 12–21 components in the pivoting graph for every day of measurement and 14 components in the graph constructed from all the data. An excerpt from such a graph can be found in Fig. 3. Several patterns became imminent and are closely described next. For simplicity, we refer to the common component by the number of actors (sources, pivots, and targets) they contain, let the number be either 1 or n for 1 or more actors.

The $n:1:1$ patterns represent a component with numerous sources but only one pivot and one target. Pivots in these components were typically Internet-facing SSH servers, which receive incoming connections all the time, while the target is a monitoring node, to which the pivot reports its status. Indeed, all of them were found to be related to monitoring in manual analysis. The $1:n:1$ was observed at one department due to the unique setting of their host monitoring infrastructure, as we discussed in the manual analysis. The $1:1:n$ pattern with multiple targets usually indicates a cloud orchestration or host monitoring, which was confirmed in the manual analysis. Although it may seem that any pattern with n is benign, we are aware that an attacker may compromise one pivot and use it to exploit multiple targets. Thus, unless the pattern is labeled as benign by the local rules, it should be investigated.

The $1:1:1$ pattern was the most common; each component represented an isolated pivoting event involving actors not involved in other patterns. While some of these were evaluated as benign, it is rather a good indicator of suspicious activity that is worth investigating.

4.3 Pivoting over Time

Figure 4 shows the frequency of how often each pivot was detected throughout the measurement. Out of 238 unique pivots detected over 10 days, 120 were

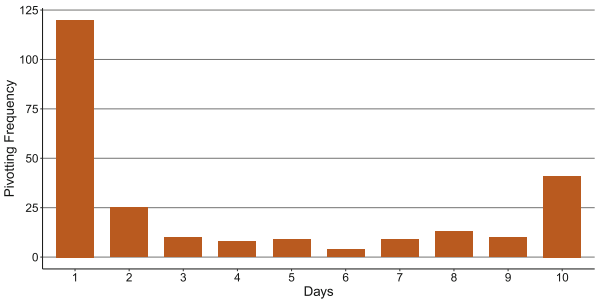

Fig. 4. Temporal analysis of pivot presence: observing candidate events across measurement days.

seen in only one day, while 41 were seen every day. It is not surprising that benign pivoting-like activities related to network management and monitoring are happening almost every day, while suspicious connections are more likely to happen in only one day.

Nevertheless, it would be a false assumption to automatically consider one-time events as suspicious and repeated as benign. We assume the attackers adopting the pivoting technique are also considerate of timing and, thus, perform "low and slow". With this in mind, a single event may not be significant, but a repeated, long-lasting activity involving unknown actors may be a sign of an advanced attacker, which we aim to expose.

5 Towards Automated Candidate Filtering

Herein, we present the third phase of our experiment. In the previous section, we presented the results of manual analysis, in which we outlined what frequent false positive and benign candidates were detected. Since the analysis was mostly based on the knowledge of the local environment, we could infer the contextual features of pivoting that could be leveraged for automated filtering of pivoting candidates.

5.1 Features

A total number of 39 features was selected for the experiment, along with the labeling provided by rule-based filtering. The first set of 18 features is derived from the NetFlow data and represents the duration and number of flows between the actors and the number of transferred packets and bytes in both directions. Moreover, the ratios of the features between the two communications are added. These features were already used in related work [15] and may signalize the similarity between the two flows.

The other features are contextual and describe the location of the actors and their relation to the remainder of the dataset. Three features designate the

location of the actor: 0 for an external IP (anywhere on the Internet outside the campus network), 1 for public IP in the campus network, and 2 for an IP in a private address range. Subsequently, we assume seven features for each combination of the actors, i.e., Source (S), Pivot (P), Target (T), and their combinations (SP, PT, ST, SPT). These features designate how many times was the combination of actors observed in the candidate list. Four features are inspired by the degrees of a corresponding node in the pivoting graph, i.e., the out-degree of S, the in- and out-degrees of P, and the in-degree of T. The last seven features represent the timing. For each combination of actors, a feature indicates whether this combination was observed on the previous day or not. For practical reasons, we assume only the previous day. However, we can generalize this feature to indicate the number of occurrences in any number of previous days. It is worth noting that the candidates from the first day were excluded from this experiment since their history was not observed.

5.2 Analysis and Results

We used Principal Component Analyses (PCA) to find the relations between the 39 features and their labels. We limited the number of principal components to 2 so that we could plot them in the graph. This was done with different feature sets, with all features and with contextual features only. The results are presented in Fig. 5.

Unfortunately, the obtained results were inconclusive in providing a definitive answer to the questions we were most interested in. Specifically, we were unable to determine the most significant features or their combinations, as well as whether it is feasible to cluster the pivoting candidates in a manner that aligns with the given labels. Even though the true positive (i.e., suspicious) candidates can be found in certain small areas of the graph, they are still mixed with false positive and benign candidates. However, the enrichment of the pivoting candidate with additional contextual features seems to help. Employing only the NetFlow-based features is the least illustrative while using only the contextual features results in clearer clusters.

Despite making several attempts at clustering and visualization throughout the experiment, the ones presented here are considered the most compelling. However, the results are rather negative as they did not reveal any key features that could effectively differentiate between benign and suspicious pivoting activities.

6 Discussion

The discussion is structured in three areas. First, we list the limitations of our approach. Second, we comment on the security implications, such as avoiding detection. Finally, we put forward a few recommendations for pivoting detection using our approach in practice.

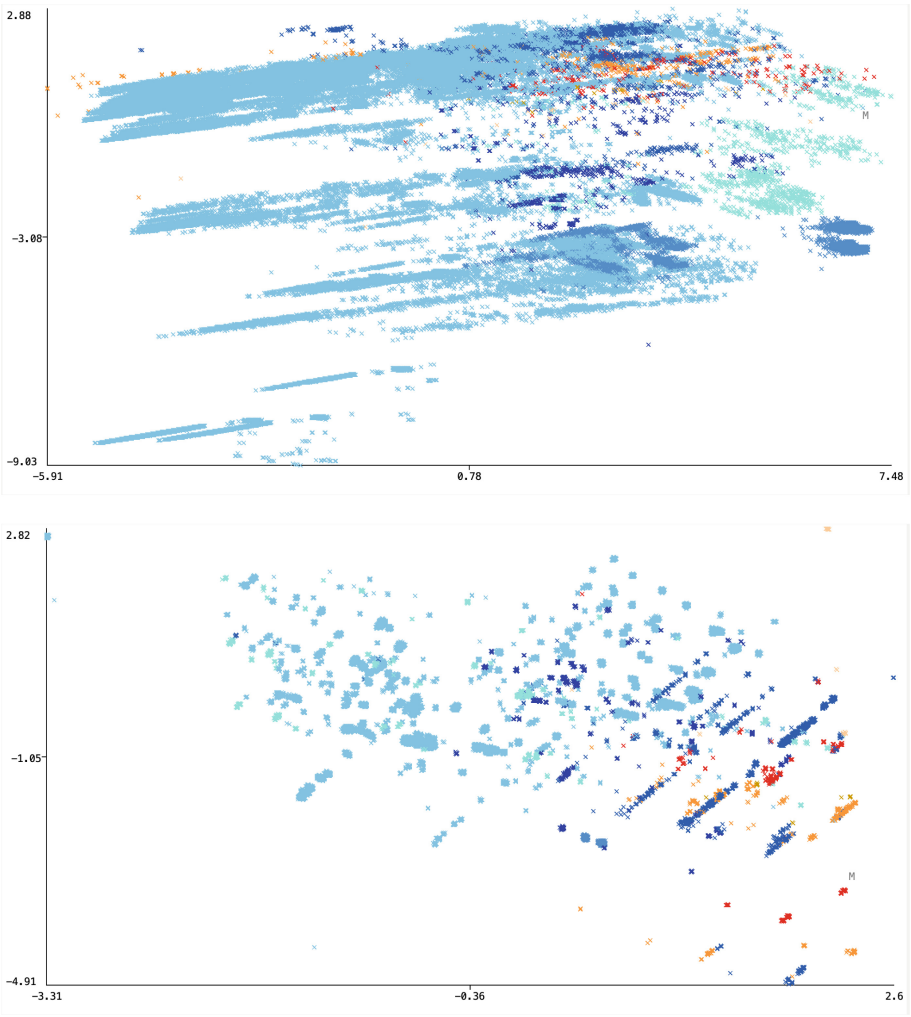

Fig. 5. Clustering analysis. The top figure shows clustering with all features, the bottom figure shows clustering with contextual features only. Colors are assigned as follows: blue for benign and false positive candidates, orange for in-and-out and outwards scenarios, red for inwards scenarios. (Color figure online)

6.1 Limitations

We are aware pivoting can be conducted using any suitable protocol, not only SSH. The other viable options would be RDP, Telnet, or even protocols associated with network printers (e.g., LPD, LPR, IPP). The attackers could even switch protocols and use one to access the pivot and another to contact the target. Nevertheless, it is not common in related work to use protocol-agnostic detections. At this stage of research, we have to first understand the command

propagation before devising detection algorithms. Moreover, protocol-agnostic detections suffer from false positive detections far more and explode in complexity [15]. We estimate that if a measurement similar to this one is conducted with a focus on other protocols, we may gather enough insights to propose a detection method that would reduce the false positive rates across the protocols.

As for the second limitation of our work, we do not reflect the situation in which the pivot may use two different IP addresses (e.g., public and private). We have observed pivoting-like traffic from public to private IP address ranges and vice versa via single-interface pivots. However, pivoting through multi-interface pivots could have been missed and would be worth investigating, even though that would mean additional complexity.

The final limitation is the lack of ground truth and extreme imbalance given by the very few true positives. No attacks were observed nor confirmed, only a small number of suspicious behavior samples, which, compared to the tremendous amount of false positive and benign samples, look negligible. Indeed, it is extremely difficult to refine reasonable data mining outputs or machine learning models. On the contrary, we documented a number of pitfalls for pivoting detection in real-world network traffic that could not be observed in laboratory experiments or in the available datasets.

6.2 Security Implications

The fact that we did not observe any clearly malicious activity is good news, but the ground truth is missing. Thus, we had to label the suspicious events as true positives. We assume the attacker would gain a foothold in the network by exploiting unsecured network host, such as a common workstation or IoT device. Then, the attacker would use the exploited device to access services available only to hosts within the campus network. This corresponds to the *inwards* scenario. Subsequently, the attacker would instruct the pivot to access the internal resources and exfiltrate data. Then, it could also fit the *outwards* scenario.

We argue that a potential attacker would be detected using the proposed method. Malware or an attacker with no knowledge of the environment would explore the surrounding of the exploited device by network scanning and performing brute-force password attacks, which could be detected by common IDS, assuming it is deployed within the network and not only on the perimeter. Since perimeter protection is often a priority and IDS in the internal network can be costly, solely NetFlow-based detection might be key.

However, an advanced attacker, such as in the case of APT, would target specific services and remain unnoticed unless pivoting detection is in place. Considering we were able to detect benign pivoting conducted by personnel working from home, we assume the advanced attacker would be detected, too. They would have three options to hide:

1. switch protocols or port numbers to avoid detection, which is certainly possible but could be approached in the detection by further measurements, development, and combining with related work,

2. exploit hosts on the whitelist or move laterally in a way that avoids vantage points, but that would require excellent knowledge of the environment,
3. set large command propagation delays to disassociate connection to and from pivot; the attacker would then have to, for example, connect to pivot, instruct it to perform an action after 5 min, disconnect, wait for 10 min, and connect again to see the results.

6.3 Recommendation for Pivoting Detection in Practice

Unfortunately, fully automated precise network-based pivoting maneuver detection was not yet achieved. However, a semi-automated solution is achievable under these conditions.

First, the algorithm by Apruzzese et al. [2] is robust and efficient for the first stage of detection. The second stage may use the knowledge presented in this paper to filter the vast majority of unwanted results and comprehensively visualize the remaining ones, thus helping the users in the investigation of the detected patterns.

Second, while deploying the pivoting detector, the users should check for the monitoring and cloud management infrastructure and write up filtering rules, preferably with the detection running for several days to get more samples. Setting more *zones* or *locations* is advisable to filter benign events like pivoting within the network or across clouds of collaborating institutions.

Third, the filtered results should be presented in graphical form as components of the pivoting graph (preferably with domain names and with actors from different locations in different colors) so that the user may promptly comprehend what the actors are and if such traffic is benign or suspicious.

An important issue to mention is the number of results. Our experiment shows tens or hundreds of candidates per day with mostly tens of unique combinations of actors and a low number of pivoting graph components. After careful filtering, these numbers can be reduced to under 10, which is a fair number that could be processed even by analysts under a heavy workload. Additional filters may be used to highlight or alert pivoting candidates with interesting parameters, such as an unknown $1:1:n$ pattern, *inwards* pivoting from an external IP address with low reputation [6], pivot or target is vulnerable or has been compromised recently, or pivoting involving valuable network assets, such as a part of critical infrastructure. Such events are not expected to be very rare and would definitely trigger further investigation, thus justifying the deployment of the presented pivoting detection procedures.

7 Conclusion

We presented an empirical study in pivoting maneuver detection in network traffic. Building upon a modified algorithm from related work [2], we performed experiments to identify real-world detection patterns. Although no clear malicious event was detected, our analysis yielded valuable insights into the network

traffic landscape, revealing a significant number of false positives and benign pivoting-like events. The scope of the experiment exceeds previous works [15] and complements results achieved in laboratory settings [7] and host-based methods [18].

We discovered that distinguishing between benign and suspicious pivoting events heavily relies on contextual factors. Consequently, we explored several contextual features that enhance the understanding of automated detection outcomes, reduce false positive rates, and dismiss benign results. While achieving precise pivoting detection in real-world settings remains an open challenge, our study offers critical insights, paving the way for the development of an automated pivoting detection tool that minimizes the burden on human analysts. Implementing such a tool and conducting long-term evaluations are proposed as future research directions.

Acknowledgment. This research was supported by project "MSCAfellow5_MUNI" (No. CZ.02.01.01/00/22_010/0003229). The authors would like to thank CSIRT-MU for providing access to real-world data.

References

1. Agency, C.I.S.: SamSam Ransomware. https://us-cert.cisa.gov/ncas/alerts/AA18-337A (2018). Accessed 14 Sept 2023
2. Apruzzese, G., Pierazzi, F., Colajanni, M., Marchetti, M.: Detection and threat prioritization of pivoting attacks in large networks. IEEE Trans. Emerg. Top. Comput. **8**(2), 404–415 (2020)
3. Ayala, L.: Active medical device cyber-attacks. In: Cybersecurity for Hospitals and Healthcare Facilities: A Guide to Detection and Prevention, pp. 19–37. Apress, Berkeley, CA (2016)
4. Bai, T., Bian, H., Daya, A.A., Salahuddin, M.A., Limam, N., Boutaba, R.: A machine learning approach for RDP-based lateral movement detection. In: 2019 IEEE 44th Conference on Local Computer Networks (LCN), pp. 242–245. IEEE, New York, NY, USA (2019)
5. Bai, T., Bian, H., Salahuddin, M.A., Abou Daya, A., Limam, N., Boutaba, R.: RDP-based lateral movement detection using machine learning. Comput. Commun. **165**, 9–19 (2021)
6. Bartos, V., Zadnik, M., Habib, S.M., Vasilomanolakis, E.: Network entity characterization and attack prediction. Futur. Gener. Comput. Syst. **97**, 674–686 (2019)
7. Bian, H., Bai, T., Salahuddin, M.A., Limam, N., Daya, A.A., Boutaba, R.: Uncovering lateral movement using authentication logs. IEEE Trans. Netw. Serv. Manage. **18**(1), 1049–1063 (2021)
8. Binde, B., McRee, R., O'Connor, T.: Assessing outbound traffic to uncover advanced persistent threat (2011). SANS Institute
9. Bowman, B., Laprade, C., Ji, Y., Huang, H.H.: Detecting lateral movement in enterprise computer networks with unsupervised graph AI. In: 23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020), pp. 257–268. USENIX Association, San Sebastian (2020)
10. Dong, C., et al.: Bedim: lateral movement detection in enterprise network through behavior deviation measurement. In: 2021 IEEE 23rd International Conference on

High Performance Computing & Communications; 7th International Conference on Data Science & Systems; 19th International Conference on Smart City; 7th International Conference on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys), pp. 391–398. IEEE (2021)
11. Dong, C., Yang, J., Liu, S., Wang, Z., Liu, Y., Lu, Z.: C-bedim and s-bedim: lateral movement detection in enterprise network through behavior deviation measurement. Comput. Secur. **130**, 103267 (2023)
12. E-ISAC: Analysis of the cyber attack on the ukrainian power grid (2016). https://media.kasperskycontenthub.com/wp-content/uploads/sites/43/2016/05/20081514/E-ISAC_SANS_Ukraine_DUC_5.pdf
13. González-Manzano, L., de Fuentes, J.M., Lombardi, F., Ramos, C.: A technical characterization of APTs by leveraging public resources. Int. J. Inf. Secur. **22**, 1–18 (2023)
14. Hofstede, R., et al.: Flow monitoring explained: from packet capture to data analysis with NetFlow and IPFIX. Commun. Surv. Tutorials **16**(4), 2037–2064 (2014)
15. Husák, M., Apruzzese, G., Yang, S.J., Werner, G.: Towards an efficient detection of pivoting activity. In: 2021 IFIP/IEEE International Symposium on Integrated Network Management (IM), pp. 980–985. IEEE, New York, NY, USA (2021)
16. Liu, Q., et al.: Latte: large-scale lateral movement detection. In: MILCOM 2018–2018 IEEE Military Communications Conference (MILCOM). IEEE, New York, NY, USA (2018)
17. Los Alamos National Laboratory. https://networkx.org. Accessed 14 Sept 2023
18. Marques, R.S., Al-Khateeb, H., Epiphaniou, G., Maple, C.: Apivads: a novel privacy-preserving pivot attack detection scheme based on statistical pattern recognition. IEEE Trans. Inf. Forensics Secur. **17**, 700–715 (2022)
19. Powell, B.A.: Detecting malicious logins as graph anomalies. J. Inf. Secur. Appl. **54**, 102557 (2020)
20. Powell, B.A.: Role-based lateral movement detection with unsupervised learning. Intell. Syst. Appl. **16**, 200106 (2022)
21. Ramaki, A.A., Rasoolzadegan, A., Bafghi, A.G.: A systematic mapping study on intrusion alert analysis in intrusion detection systems. ACM Comput. Surv. **51**(3), 1–41 (2018)
22. Salema Marques, R., Al Khateeb, H., Epiphaniou, G., Maple, C.: Pivot attack classification for cyber threat intelligence. J. Inf. Secur. Cybercrimes Res. **5**(2), 91–103 (2022)
23. Sarafijanovic-Djukic, N., Pidrkowski, M., Grossglauser, M.: Island hopping: efficient mobility-assisted forwarding in partitioned networks. In: 2006 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks, vol. 1, pp. 226–235. IEEE (2006)
24. Smiliotopoulos, C., Kambourakis, G., Barbatsalou, K.: On the detection of lateral movement through supervised machine learning and an open-source tool to create turnkey datasets from sysmon logs. Int. J. Inf. Secur. **22**, 1893–1919 (2023)
25. Staniford-Chen, S., Heberlein, L.: Holding intruders accountable on the internet. In: Proceedings 1995 IEEE Symposium on Security and Privacy, pp. 39–49 (1995)
26. Storm, D.: MEDJACK: hackers hijacking medical devices to create backdoors in hospital networks. https://www.computerworld.com/article/2932371/medjack-hackers-hijacking-medical-devices-to-create-backdoors-in-hospital-networks.html (2015). Accessed 14 Sept 2023
27. Tankard, C.: Advanced persistent threats and how to monitor and deter them. Netw. Secur. **2011**(8), 16–19 (2011)

28. TrapX Labs. https://securityledger.com/wp-content/uploads/2015/06/AOA_MEDJACK_LAYOUT_6-0_6-3-2015-1.pdf (2015). Accessed 14 Sept 2023
29. Valeur, F., Vigna, G., Kruegel, C., Kemmerer, R.A.: Comprehensive approach to intrusion detection alert correlation. IEEE Trans. Dependable Secure Comput. **1**(3), 146–169 (2004)
30. ViaSat: KA-SAT Network cyber attack overview. https://news.viasat.com/blog/corporate/ka-sat-network-cyber-attack-overview (2022). Accessed 14 Sept 2023
31. WikiLeaks: Vault7: Archimedes documentation. https://wikileaks.org/vault7/#Archimedes (2017). Accessed 14 Sept 2023
32. Wilkens, F., Haas, S., Kaaser, D., Kling, P., Fischer, M.: Towards efficient reconstruction of attacker lateral movement. In: Proceedings of the 14th International Conference on Availability, Reliability and Security. ARES 2019, ACM, New York, NY, USA (2019)
33. Zhang, Y., Paxson, V.: Detecting stepping stones. In: Proceedings of the 9th Conference on USENIX Security Symposium, Vol. 9. p. 13. SSYM 2000, USENIX Association, USA (2000)