Towards an Efficient Detection of Pivoting Activity

Martin Husák
Institute of Computer Science
Masaryk University
husakm@ics.muni.cz

Giovanni Apruzzese
Institute of Information Systems
University of Liechtenstein
giovanni.apruzzese@uni.li

Shanchieh Jay Yang
and Gordon Werner
Rochester Institute of Technology
{jay.yang, gxw9834}@rit.edu

***Abstract*—Pivoting is a technique used by cyber attackers to exploit the privileges of compromised hosts in order to reach their final target. Existing research on countering this menace is only effective for pivoting activities spanning within the internal network perimeter. When applying existing methods to include external traffic, the detection algorithm produces overwhelming entries, most of which unrelated to pivoting. We address this problem by identifying the major characteristics that are specific to potentially malicious pivoting. Our analysis combines human expertise with machine learning and is based on the inspection of real network traffic generated by a large organization. The final goal is the reduction of the unacceptable amounts of false positives generated by the state of the art methods. This paper paves the way for future researches aimed at countering the critical menace of illegitimate pivoting activities.**

***Index Terms*—Pivoting, Lateral Movement, Machine Learning, Flow Inspection, Intrusion Detection**

I. Introduction

Despite the many advances in cybersecurity, the reality is that existing systems are continuously breached. Motivated attackers increasingly refine their strategies, and no enterprise network can consider itself absolutely secure [1]. At the base of many offensive campaigns is the technique of *pivoting*, which allows attackers to exploit the privileges of compromised hosts to reach their final target. As an example, consider a network that can be accessed only through a VPN server: by compromising this server, it is possible for an external attacker to access every host in the given network. Detecting these occurrences would prevent the attackers from expanding their control in the target network (i.e., lateral movement) to accomplish their objective. Hence, stopping the malicious pivoting activities early in an attack campaign is of paramount importance for the security of modern organizations.

Despite the critical role of pivoting detection, there is limited prior work tackling this problem. To the best of our knowledge, the only existing approach [2] is effective only for pivoting activities that originate *within* a network. In fact, we empirically apply this method on traffic coming from external networks and found that over 99% of the "detected" pivoting samples are not pivoting at all. This outlines a crucial problem in state of the art, as pivoting activities can span over multiple networks—which is especially true in recent times with an increase of remote network accesses [3].

This paper addresses the issue of pivoting detection. We provide a detailed study that serves as a basis to decrease the amount of "false" pivoting activities detected by means of the existing approach. We do not make any assumption on the location or protocol involved for the pivoting activities, and the only requirement is NetFlow analysis. To formalize the scope of our work, we pose the following three research questions which we shall answer:

1) Which phenomena are similar to pivoting?
2) Which NetFlow features are intrinsic to pivoting?
3) How to automatically reduce false positives?

Our proposal combines human expertise with machine learning methods. By manually inspecting the output of prior work on real traffic data, we determine which 'candidates' represent true pivoting activities. Then, we infer which traits are more typical of true pivoting activities with respect to noisy results. Finally, we combine our findings with machine learning to identify features and feature combinations that are intrinsic to true pivoting and the weight they should have in true pivoting detection.

Our paper paves the way to more efficient and automatic methods that can reliably detect pivoting occurring within or originated beyond a given network. The remainder of this paper is structured as follows. Section II compares this paper with related works. Section III describes the application scenario. Section IV presents our method. Section V is devoted to the experimental evaluation, and conclusions are drawn in Section VI.

II. Related Work

Although many advanced cyberattacks exploit the technique of pivoting (e.g., [4], [5]), we highlight the scarcity of proposals that focus specifically on the detection of malicious pivoting activities. Among the first efforts is the seminal work by Valeur et al. [6], which leverages the correlation of alerts generated by an Intrusion Detection System (IDS): the problem of this approach is that it requires the triggering of alerts (as also done in [7]), whereas we operate on raw NetFlows [8]. A related area investigates malware propagation [9] by detecting anomalous communication spikes, but their effectiveness in large networks requires scenarios where hundreds of hosts are involved – whereas typical pivoting activities span over just a few selected devices. Some papers focus on pivoting *prevention* by means of game-theory [10]–[12], but require the complete restructuring of the entire network, and the attacker may just deviate from the designed model to easily evade the countermeasure. The proposal by Fawaz et al. [13] operates

978-3-903176-32-4 ©2021 IFIP

on host-based data (similarly to [14]), which are prone to manipulation by an expert attacker that is already controlling some devices in the target network. A relevant study is the one in [15] where graph-analytics techniques are applied to detect malicious logins on *individual* machines, which cannot be used to model pivoting activities because they span over *multiple* hosts. The works in [16], [17] conceptualize pivoting attacks but do not propose any original detection approach. Finally, pivoting attacks can be detected as a side effect, but only when performed through specific services, such as RDP [18].

In summary, with respect to state of the art, our paper focuses on the detection of pivoting activities occurring anywhere in a given network. Our proposal is based just on NetFlow inspection. By applying machine learning algorithms, we aim to reduce the (unacceptably high) rate of false positives.

III. Pivoting Scenarios

We describe the concept of pivoting activities and the intuition of existing pivoting detection methods, which form the base of our analysis.

A. Pivoting Activities

Pivoting leverages the idea of using machines with privileged characteristics (e.g., the entry point of a network, or dedicated ACL) to "connect" two hosts that would be otherwise unreachable. Pivoting activities involve three actors: the *source* (S), the *pivoter* (P), and the *target* (T).

We provide an example in Figure 1, representing a typical setting where an organization network is accessible from the Internet through an entry point host. In this context, remote users (S_b or S_m in Figure 1) can access the network by "pivoting" on such a host (P in Figure 1), which serves as a stepping stone to reach the organization's internal services (T_x in Figure 1). As a consequence, all communications between the remote users (the *source*) go through the *pivoter* host, which forwards them to the final *target*. This workflow is reversed when the target sends its responses back to the corresponding source. It is obvious that such activities are not malicious by definition. On the contrary, in such a situation, (legitimate) pivoting activities are "expected" to happen.

Pivoting activities can span over more than three hosts, which would represent scenarios with multiple pivoters. Without loss of generality, in the remainder of this paper, we focus only on pivoting activities involving three hosts, as all results can be easily extended to cover cases with longer pivoting chains.

B. Existing Pivoting Detection Algorithm

Apruzzese et al. [2] proposed an algorithm that leverage temporal graph-analytics to detect pivoting by analyzing NetFlow records. This data-type captures high-level information on the network communications between two hosts, such as the start of the transmissions and their duration and the amount of exchanged bytes or packets. At the base of this algorithm are two main intuitions that are used to model pivoting activities: (i) all communications between S and T *must* go through P;

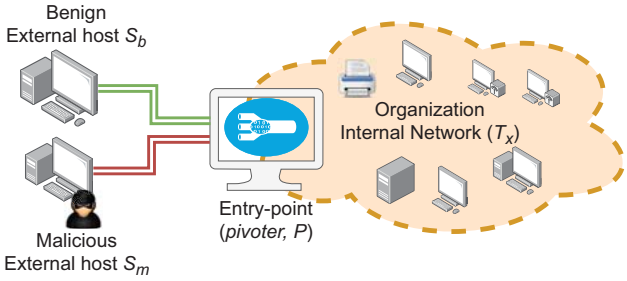

Figure 1. Depiction of pivoting.

and (ii) P forwards the communications to T *after* receiving the communications from S. Hence, by modeling this time constraint as a function of ε, it is possible to model pivoting activities as "pairs of NetFlows linking S with P, and P with T, that are separated by at most ε time units". By optimally tuning ε, such formalization was applied in [2] to detect pivoting activities occurring in an *internal* network, obtaining perfect accuracy as all results were indeed related to pivoting activities.

However, we claim that this approach is not feasible when considering external traffic. In such circumstances, the pivoter host would receive millions of incoming connections from thousands of external hosts in very short time frames. As a consequence, even by considering very small values of ε, the algorithm would output a considerable number of false pivoting activities.

C. Preliminary Findings

We validate the infeasibility of [2] by applying their algorithm to a real use case. We consider real network traffic data on the Masaryk University campus, consisting of 23 GB of NetFlow data collected throughout one working day. The traffic includes both external and internal hosts. We run the existing algorithm on this data by setting $\varepsilon = 1$ second, and we obtained $90\,774$ pivoting activities. After manual inspection, we determined that only 13 of these were *true* pivoting activities, with a false positive rate of over 99.99%.

Such result confirms the crucial problem at the base of our work, which we address in the next sections. Given the unreliability of the results, in the remainder of this paper, we will refer to the output of the detection algorithm as "pivoting candidates".

IV. Methodology

We aim to answer the three research questions posed in Section I. We do so by looking at the resulting pivoting candidates produced after the application of the algorithm in [2] and using expert knowledge to determine which of these pivoting candidates are actually pivoting. Then, we inspect the true pivoting activities, and we try to discern which characteristics captured by network flows could be used to differentiate them from the false positives. This protocol is typical in cybersecurity-related analyses [19]. Let us first summarize our implementation of the algorithm.

The NetFlow collector of the considered network is set to produce only unidirectional flows. Hence, we had to convert these data into their bidirectional version (which is an assumption of [2]). To accomplish this task we rely on *nfdump*[1]. Only the flows with the duration and the number of transferred packets and bytes in both directions bigger than 0 are considered. We consider, in this work, pivoting activities must have bidirectional communications, i.e., T has to respond to S. Finally, we reorder the flows according to their timestamp. We do not apply any additional filtering; therefore, communications over all protocols and port numbers are considered.

The pivoting candidate detection follows the algorithm from [2]. Each flow f with a timestamp t is checked against all the flows with the timestamp between t and $t + \epsilon$, where ϵ is the maximal command propagation delay tolerated by the detection algorithm, in our case 1 second. This produces a set of flow pairs. Then, each pair of flows (f_1, f_2) is considered as a pivoting candidate if:

- the destination IP address of f_1 is also the source IP address of f_2,
- destination IP address of f_2 is not the source IP address of f_1,
- protocol and destination ports are the same for both f_1 and f_2.

The last criteria on the same protocol and destination ports are motivated by the expectation that the same service, e.g., SSH, is used in S to P and P to T connections.

The execution of such implementation of the algorithm yields $90\,774$ pivoting candidates.

A. Discovery of phenomena similar to pivoting

To answer the first research question, we need to inspect the pivoting candidates and identify which candidates represent potential pivoting activity and which represent other events. We will use manual inspection of the candidates and expert knowledge of network traffic and the environment. The manual inspection is going to be highly situational and may rely on the insider's knowledge of the network.

The first features we are going to investigate are the destination ports and protocols, which are the same in both flows that form a pivoting candidate. This would allow us to associate pivoting candidates with network services and applications. We expect to see potentially interesting samples on TCP ports 22 (SSH), 23 (Telnet), and 3389 (RDP), which are mostly used for both benign and malicious remote access to network hosts. Some network applications may display similar behavior as pivoting, such as an SMTP during receiving and relaying an email. In case we encounter pivoting candidates using such ports and services, we will investigate if it is a suspicious behavior or a common behavior of such a service.

The second feature we are going to inspect is the IP addresses involved in a candidate event. Namely, we are interested in the location of the actors, e.g., if it is in the private or public network of an organization or elsewhere the Internet. Our assumption is the true pivoting does not involve all the actors from the same network. The prime example of true pivoting involves a source from the Internet, pivot in the public network of an organization, and target inside such a network, as presented in Section III. Thus, we will check the locations of the actors and inspect the directions of the suspected pivoting activities.

[1]https://github.com/phaag/nfdump

B. Manually identifying intrinsic features

The discovery of phenomena similar to pivoting should eliminate candidates that are explainable as a different activity than pivoting but does not improve our understanding of pivoting. In the next step, and to answer the second research question, we need to identify the features of pivoting activity that can be inferred from the candidates but are not reflected in the existing detection method.

We will investigate the features discussed in the previous subsection, this time with respect to the candidates appearing to be true pivoting activities, such as the candidates with the expected services and actor's location. Subsequently, we will count the occurrences of each IP address in the candidates to check for frequent talkers and isolated events. Frequent actors and their combinations may suggest a common non-pivoting activity, and, thus, true pivoting is expected to be more likely found in isolated events, i.e., candidates with rather a unique set of actors. Finally, an interesting feature to focus on is the volume and the duration of the transferred data. If a true pivoting propagates a command or data, it should display similar NetFlow features, such as the number of bytes and packets transferred and the duration of the flow.

Our experiences and expectation in identifying suspected pivoting activities suggest the following critical features:

- ***S*, *P*, *T* Count** are the numbers of pivoting candidates that share the same S, P, or T (e.g., the pivot count of a candidate X is the number of candidates in which the pivot's IP address is the same as in X, including X).
- ***S*, *P*, *T* Locations** is the label given to IP addresses that describes their location with respect to the monitored network. The 'Private' tag is assigned to privately routed IP addresses in the network (e.g., 10.0.0.0/8), the tag "Public" or "Internal" is given to publicly addressable IP addresses of the monitored network (e.g., 147.251.0.0/16 for Masaryk University network), and the tag "External" is given to all the other IP addresses.
- **Duration, In/Out Packets, In/Out Bytes, Flows ratios** are the ratios of the biflow feature values. The value of each feature in the first biflow (source to pivot) is divided by the value in the second flow (from pivot to target).

C. Automatically identifying intrinsic features

Our research here is to move the time-consuming practices towards a generalizable machine learning approach. The contextually meaningful features that we identify are being examined with statistical correlation and Principal Component Analysis (PCA) methods. Using the Kaiser criterion, we will

select the principal components with eigenvalue > 1 and set the number of principal components needed to identify true pivoting. The eigenvectors of the principal components will illustrate the weight of the candidate features in the process. Further, the Pearson linear correlation coefficient calculates the covariance of two vectors and divides them by the product of their standard deviations $\rho_{x,y} = \frac{cov(x,y)}{\sigma_x \sigma_y}$. The result is a number ranging from $-1 \leq \rho_{x,y} \leq 1$, which represents the strength and direction of the correlation between the two vectors. By determining the Pearson correlation between attributes and the label, we can better understand which individual features may be indicative of suspected true pivoting flows.

V. Experimental Results

Herein, we present the results of the experiment. First, we comment on the measurements and candidate detection. The findings of manual inspection of pivoting candidates follow. Subsequently, we present the results of principal component analysis and comment on feature correlation.

A. Pivoting Candidate Detection

The experiment took place in the campus network of Masaryk University that uses NetFlow probes located at various observation points. The hosts in the campus network use mostly public IP addresses (/16 IPv4 and /48 IPv6 range), but certain infrastructures are addressed by private IPv4 range accessible only from within the campus network. We used the NetFlow [8] data from a probe located near a server segment, where we expect to capture the most interesting examples of pivoting, namely the pivoting from public to private IP range and vice versa with servers as pivots. The network traffic of personal computers connected to the university's VPN and Wi-Fi is also visible to the NetFlow probe and, thus, we also have a chance to inspect the behavior of personal computers in the network.

B. Investigation of Pivoting Candidates

The HTTP(S) traffic is dominant and deserves further investigation. We found 133 pivoting candidates on SSH, which is one of the services we expected to display pivoting characteristics that could be malicious and is worth detailed investigation. Unfortunately, no pivoting candidates were found with the TCP port 3389, on which the RDP service is running. Probably there were no RDP servers active in the monitored network segment at the time of the measurement. The other destination ports indicate legitimate traffic that is similar to pivoting. The discussion on particular ports and services follows. We did not find any pivots that would enable pivoting on more ports simultaneously. We only found ten cases in which the pivoting candidates differed only in the protocol used; the actors used port 51413 over both TCP and UDP.

A large portion of pivoting candidates was observed on ports that would indicate explainable, benign network traffic that only resembles pivoting and, thus, could be considered as false positives. First, we observed a number of pivoting-like events in which known servers acted as pivots. Namely, in the DNS and SMTP traffic, we can see traffic patterns similar to pivoting. In DNS, a server receives a request from a client, and if an appropriate record is not found at the server, the server queries a different server. In SMTP, a server receives an email from a client and relays it to a different server. Both activities are very similar to pivoting and are correctly detected by the detection method. However, they are very common, and no suspicious candidates were found. Thus, we can proclaim such candidates as benign. Second, we found a number of candidates in which only the personal computers served as pivots. Most often, we found a pivoting-like activity on ports associated with BitTorrent and online gaming. In such cases, a host running a BitTorrent client or an online game initiates and receives a lot of connections at the same time, which may be detected as pivoting. The NetFlow characteristics of such activities are very similar to pivoting. However, no suspicious candidates were found, and the candidates may be proclaimed as false positives. The same also applies to VoIP, instant messaging, NTP, and other services based on p2p communication, which were also detected, but mostly with only a few candidates or pivots. The NAT traversal on UDP port 4500 enables the host connected to VPN to use p2p or VoIP and, thus, should be mentioned here as well. Finally, we identified a number of pivoting candidates and corresponding ports that are not easily explainable, mostly because the port number, e.g., 15001, is not well known. In such cases, often only one pivot and a few hosts in the campus network were involved in such traffic patterns, and, thus, it is most likely a legitimate network service. Some of the pivots were identified as back-up servers, cloud management servers, and other legitimate services. Such candidates can also be proclaimed as benign or false positives.

Table I
True and false positive pivoting candidates.

False positives	HTTP(S)	70,312
	other services - DNS, SMTP, NTP	12,714
	p2p - BitTorrent, VoIP, online gaming	7,615
	SSH	120
True pivoting	SSH	**13**
Total		**90,774**

When analyzing pivoting candidates that used TCP port 22 (SSH), we identified seven distinct pivots, 3 benign and 4 suspicious. Each benign pivot initiated a connection with only one other known host, e.g., a back-up machine or a GitHub repository. Such connections appeared throughout the day and were often long-term. Thus, any incoming connection (e.g., a one-time brute-force password attack from the Internet) to a suspected pivot could cause the detection algorithm to connect it to a long-term activity and regard both flows as pivoting. The true pivots either appeared to be pivoting a connection from the Internet via the campus network to another host on the Internet or from the campus network via a host on the Internet to the campus network. We found 13 pivoting candidates with 4 unique pivots like this and used them as examples of true

pivoting. We can infer a heuristic out of these observations - a true pivoting activity is the one in which the pivot and the remaining actors are from different parts of the network (e.g., internal network and the Internet).

The candidates using HTTP(S) seem to be mostly benign and can be explained as common benign behavior of the protocol. The actors, namely pivots and targets, are mostly well-known HTTP(S) servers from the campus network or world-wide. However, due to the prevalence of HTTP(S) traffic in today's network and its popularity among users and attackers, it is not reasonable to declare it all as benign. Malicious pivoting activity may use HTTP(S) ports either in relation to a web-based attack or as the primary choice of an attacker when bypassing firewalls. Thus, we recommend further investigation as future work, possibly using an automated method due to the number of candidates. Further, the extended NetFlow monitoring with HTTP header parsing [20] or TLS analysis [21] might be helpful in identifying true or even suspicious pivoting.

C. Feature Correlation for Suspicious Flows

The linear correlation between numerical features and the suspicious label was measured to investigate if any individual attributes are possibly related to whether a flow is suspicious or not. Figure 2 shows the absolute value of Pearson measurements for a selection of attributes, representing the strength of correlation between each individual attribute and the label. Target and pivot counts as well as Duration1[2] show a strong correlation, though these are the only notable examples. All other numerical attributes have very weak correlations similar to InPackets2[3] and OutBytesRatio. It must be noted, though, that it is not expected that the individual attributes would have strong linear relationships to the label. The categorical attributes may also play a critical role in predicting suspicious flows. There should also exist relationships between groups of attributes that more strongly relate to a flow being suspicious. PCA can give a deeper understanding of the overall feature space by finding such relationships.

D. Principal Component Analysis

We used the SSH pivoting candidates and processed them using the Principal Component Analysis (PCA). This is a known machine learning method that is appreciated for the analysis of network traffic [19], [22].

Out of all the port numbers and services, the SSH candidates were the most suitable for investigating which are false positives and which could be true pivoting (benign or malicious). Out of 133 candidates, 13 were marked as True and the remaining 120 as False, following the investigation discussed previously. We run PCA using the features listed in Section IV-B and the "suspicious" flag as the target.

Using the cumulative proportion, we determine the amount of variance that the principal components explain. With future analyses in mind, we want to have at least 90% of the variance

[2]Duration of the candidate's first flow.

[3]InPackets value of the candidate's second flow.

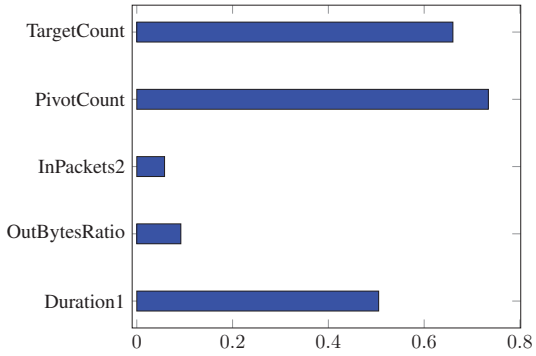

Figure 2. Bar Chart Showing Pearson Linear Correlation Coefficient Between Suspicious Label and Attributes.

explained, which corresponds to five principal components. Similarly, using the Kaiser criterion (selecting principal components with eigenvalues > 1), we get the same result. The eigenvalues and eigenvectors are presented in Table II.

PCA shows several interesting associations of principal components with features. The most apparent is the positive association with "Internal" S and T and "External" pivot in the first vector. The second vector shows negative associations for the combination of "External" S and T and "Internal" P, which suggests that both combinations are interesting. There is also a strong negative association with "count" features, namely with P count and T count. The "ratio" features did not show any strong associations except for a few values in the second and third vector. However, that might be caused by the low number of true positive candidates we were able to collect. Nevertheless, the "location" and "count" features turned out to be highly interesting.

VI. Conclusions

Accurate detection of pivoting activities is of paramount importance to protect modern network environments. Existing proposals work only under specific circumstances and cannot be reliably applied in realistic settings, generating overwhelming numbers of false positives. This paper addresses this issue. We perform a comprehensive analysis of the results obtained by applying state of the art methods, with the goal of identifying a practical and effective way to automatically reduce the number of false pivoting detections. We do so by formulating and answering three research questions.

First, we focus on networking activities that are similar to pivoting and may raise false positive alerts. It was shown above that various p2p networking (e.g., BitTorrent) and several common protocols (e.g., SMTP, DNS) display similar characteristics as pivoting and also represent the vast majority of the detected candidates. True pivoting occurs only with the SSH traffic. Unfortunately, we did not observe any Telnet or RDP traffic that could also include pivoting. Then, we focus on identifying the unique traits exhibited by true pivoting activities. We show that the most significant traits include the location (e.g., inside or outside the monitored network) and

Table II
EIGENVALUES AND EIGENVECTORS OF PRINCIPAL COMPONENTS.

Eigenvalues						
Eigenvalue	4.15755	2.36467	2.00138	1.36928	1.07673	0.86379
Proportion	0.34646	0.19706	0.16678	0.11411	0.08973	0.07198
Cummulative	0.34646	0.54352	0.7103	0.82441	0.91413	0.98612
Eigenvectors						
***S* location=Internal**	0.4271	-0.2369	0.0053	0.2051	0.1684	-0.155
***S* count**	-0.2547	-0.3806	-0.005	0.1136	0.3858	0.3973
***P* location=External**	0.4271	-0.2369	0.0053	0.2051	0.1684	-0.155
***P* count**	-0.4106	-0.0511	0.0033	0.3107	0.3059	-0.2268
***T* location=Internal**	0.4271	-0.2369	0.0053	0.2051	0.1684	-0.155
***T* count**	-0.3988	-0.0586	0.0035	0.3204	0.3281	-0.269
Duration ratio	-0.068	0.2496	0.0036	0.5616	-0.4721	-0.3759
In Packets ratio	-0.0811	-0.285	-0.6018	-0.0888	-0.1743	-0.113
Out Packets ratio	-0.145	-0.4729	0.342	-0.1523	-0.2855	-0.182
In Bytes ratio	-0.0787	-0.2516	-0.6247	-0.079	-0.1436	-0.1146
Out Bytes ratio	-0.1482	-0.461	0.3611	-0.1522	-0.2767	-0.1846
Flows ratio	0.0029	-0.1943	-0.0049	0.5346	-0.3646	0.6418

the uniqueness of the actors (i.e., how often they appear in the detected candidates). Finally, to facilitate the reduction of false positives, we apply PCA to automatically infer the true pivoting features.

Our findings pave the way to future work, where we aim to create an efficient detector of pivoting candidates. Such a detector would allow the development of more refined methods to counter the threat posed by malicious pivoting activities to modern organizations.

ACKNOWLEDGMENT

This research was supported by ERDF "CyberSecurity, CyberCrime and Critical Information Infrastructures Center of Excellence" (No. CZ.02.1.01/0.0/0.0/16_019/0000822).

REFERENCES

[1] "Mandiant M-Trends 2020," FireEye, Inc., Tech. Rep., 2020. [Online]. Available: https://content.fireeye.com/m-trends/rpt-m-trends-2020

[2] 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, 2020.

[3] F. Malecki, "Overcoming the security risks of remote working," *Computer Fraud & Security*, vol. 2020, no. 7, pp. 10–12, 2020.

[4] "SamSam ransomware," https://us-cert.cisa.gov/ncas/alerts/AA18-337A, 2020.

[5] "Wikileaks vault7: Archimedes documentation." https://wikileaks.org/vault7/#Archimedes, 1 2021.

[6] F. Valeur, G. Vigna, C. Kruegel, and R. A. Kemmerer, "Comprehensive approach to intrusion detection alert correlation," *IEEE Transactions on dependable and secure computing*, vol. 1, no. 3, pp. 146–169, 2004.

[7] A. A. Ramaki, A. Rasoolzadegan, and A. G. Bafghi, "A systematic mapping study on intrusion alert analysis in intrusion detection systems," *ACM Computing Surveys (CSUR)*, vol. 51, no. 3, pp. 1–41, 2018.

[8] R. Hofstede, P. Čeleda, B. Trammell, I. Drago, R. Sadre, A. Sperotto, and A. Pras, "Flow Monitoring Explained: From Packet Capture to Data Analysis with NetFlow and IPFIX," *Communications Surveys & Tutorials, IEEE*, vol. 16, no. 4, pp. 2037–2064, Fourthquarter 2014.

[9] L. Bilge, D. Balzarotti, W. Robertson, E. Kirda, and C. Kruegel, "Disclosure: detecting botnet command and control servers through large-scale netflow analysis," in *Proceedings of the 28th Annual Computer Security Applications Conference*, 2012, pp. 129–138.

[10] J. R. Johnson and E. A. Hogan, "A graph analytic metric for mitigating advanced persistent threat," in *2013 IEEE International Conference on Intelligence and Security Informatics*. IEEE, 2013, pp. 129–133.

[11] M. Chapman, G. Tyson, P. McBurney, M. Luck, and S. Parsons, "Playing hide-and-seek: An abstract game for cyber security," in *Proceedings of the 1st International Workshop on Agents and CyberSecurity*, ser. ACySE '14. Association for Computing Machinery, 2014.

[12] M. A. Noureddine, A. Fawaz, W. H. Sanders, and T. Başar, "A game-theoretic approach to respond to attacker lateral movement," in *International Conference on Decision and Game Theory for Security*. Springer, 2016, pp. 294–313.

[13] A. Fawaz, A. Bohara, C. Cheh, and W. H. Sanders, "Lateral movement detection using distributed data fusion," in *2016 IEEE 35th Symposium on Reliable Distributed Systems (SRDS)*, 2016, pp. 21–30.

[14] M. Cinque, R. Della Corte, and A. Pecchia, "Contextual filtering and prioritization of computer application logs for security situational awareness," *Future Generation Computer Systems*, vol. 111, pp. 668–680, 2020.

[15] B. A. Powell, "Detecting malicious logins as graph anomalies," *Journal of Information Security and Applications*, vol. 54, p. 102557, 2020.

[16] C. Tankard, "Advanced persistent threats and how to monitor and deter them," *Network security*, vol. 2011, no. 8, pp. 16–19, 2011.

[17] L. Ayala, "Active medical device cyber-attacks," in *Cybersecurity for Hospitals and Healthcare Facilities*. Springer, 2016, pp. 19–37.

[18] T. Bai, H. Bian, M. A. Salahuddin, A. Abou Daya, N. Limam, and R. Boutaba, "Rdp-based lateral movement detection using machine learning," *Computer Communications*, vol. 165, pp. 9–19, 2020.

[19] T.-F. Yen, A. Oprea, K. Onarlioglu, T. Leetham, W. Robertson, A. Juels, and E. Kirda, "Beehive: Large-scale log analysis for detecting suspicious activity in enterprise networks," in *Proceedings of the 29th Annual Computer Security Applications Conference*, 2013, pp. 199–208.

[20] O. van der Toorn, R. Hofstede, M. Jonker, and A. Sperotto, "A first look at http(s) intrusion detection using netflow/ipfix," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2015, pp. 862–865.

[21] P. Velan, M. Čermák, P. Čeleda, and M. Drašar, "A Survey of Methods for Encrypted Traffic Classification and Analysis," *Netw.*, vol. 25, no. 5, pp. 355–374, 2015.

[22] D. H. Hoang and H. D. Nguyen, "A PCA-based method for IoT network traffic anomaly detection," in *2018 20th International Conference on Advanced Communication Technology (ICACT)*. IEEE, 2018, pp. 381–386.