Challenges in Identifying Network Attacks Using Netflow Data

Edward Chuah[‡][*], Neeraj Suri[*][†], Arshad Jhumka[§], Samantha Alt[¶]
[‡]The University of Exeter, Stocker Road. EX4 4PY, UK. Email: t.chuah@exeter.ac.uk
[*]Lancaster University, Bailrigg, Lancaster LA1 4YW, UK. [†]Email: neeraj.suri@lancaster.ac.uk
[§]The University of Warwick, CV4 7AL. Email: H.A.Jhumka@warwick.ac.uk
[¶]Intel Corporation. Email: samantha.alt@intel.com

***Abstract*—Large networks often encounter attacks that can affect the network availability. While multiple techniques exist to detect network attacks, a comprehensive understanding of how an attack occurs considering the various layers and components of the network software stack, can be an important element to help improve network security. By performing correlation analysis on contemporary unlabeled Netflow data, this paper conducts a comprehensive study of network flow events to identify communication patterns that may precede an attack, thereby providing potentially useful attack signatures to network administrators.**

Our work shows that, surprisingly, the Netflow data is *not* strongly correlated to network attacks. We observe that while spoof requests trigger reflection attacks, only a small percentage of the network packets are associated with the attack. Furthermore, lead time enhancements are feasible for reflection attacks that show long dwell times. Our study on network event correlations highlights empirical observations that could facilitate better attack handling in large networks.

***Index Terms*—Large network; Temporal correlation; Spatial correlation; Network attacks; Netflow data**

I. Introduction

Exascale networks require high availability to run computer applications, enabling users from multiple application domains to address major challenges in sciences and engineering. As efficient network components are designed, current networking infrastructures require robust security mechanisms to keep up with the increasing rate of cyber attacks [1], [2]. Consequently, network attack prediction schemes that can indicate an impending attack are highly desired. To better support the processes of attack mitigation, it is helpful to first understand how an attack transpires in practice. Recent work that analyzed the network logs or process logs for anomaly detection [3]–[7] and malicious authentication detection [8]–[10] have revealed useful insights to address attacks in networks. The time elapsed between the precursor event and an attack is defined as the lead time. When proactive attack mitigation techniques [11], [12] are supported by anomaly analysis, it can improve the lead times and help effectively respond to an impending or manifested attack.

Analyzing attacks in any large-scale or complex network require awareness of the sequence of events that is encountered by the network components. While researchers have focused on specific components [7], [10] depending on their target problem, answering how attacks occur needs a more integrated approach towards correlation-based log-mining [13]. Our work is novel in that it considers the flow connection-specific events along with their inter-component relationships to increase the lead times to identification of an attack boosting network attack prediction schemes.

Background: State-of-the-art approaches in network attack detection lack in the following aspects for better security in large networks:

1. While few works consider the full network software stack to design attack detection frameworks that conform to the vision of cross-layer network security [14], various network protocols (e.g., SNMP [15], SMTP [16], FTP [17], HTTPS [18]), domain name service [19], and authentication protocols [8] are often studied in isolation without exploiting their correlations for network security.
2. A network is comprised of diverse components that affect each other, for example: FTP [17], HTTPS [18]. Focusing on a specific component separately provides a local view. However, it lacks the broader view. For example, analyzing the network routers [7] w.r.t. the hosts separately may miss anomalies in the event logs generated elsewhere. In the context of attack detection, knowledge of how much these components correlate in the manifestation of an attack can prevent such attacks from recurring and improve the effectiveness of network security protocols.
3. Once an attack occurs, security protocols need to be enhanced. An in-depth understanding of how attacks occur can help in selecting the appropriate mitigation technique for the network security [20].

On this background, the work presented in this paper analyzes real data taking into account the diverse components across the network, making recommendations for effective prediction of network attacks so that the attack predictors can be applied in practice.

Challenges: Given the availability of unlabeled logs such as netflow data, identification of an attack that leads to a breach in the CIA (Confidentiality, Integrity and Availability) triad or of network patterns leading to an attack is challenging but critical for large networks. Some reasons are:

1. The netflow data often contain a large amount of events in which only a few events are relevant. Furthermore, the data may contain missing information (e.g., time frames) or partial information (e.g., absence of certain event logs) as a result of discrepancies in logging. Deciphering the network and host events to identify an attack is a non-trivial task.
2. Attacks can occur at multiple layers of the network software stack as one may have to contend with attacks at the routers and hosts, different to single layer attacks [18]. Analysis of the former is important to assess predictable lead times.
3. The system administrator's knowledge may be needed to understand the implications of the low level network logs in order to identify the source of the attack accurately.

After the network attack detection techniques developed in [1], [2], [8]–[10], [15], this paper investigates how attacks occur with useful insights to their reasons. Specifically, while the use of attack signatures is well-known in intrusion detection, identification of multi-tiered attack signatures (and patterns leading to them) is not common. Our analysis is purely log-based with no human written reports.

Contributions: In this paper, we answer the following research questions to enhance the security of networks:

1. Using unlabeled netflow logs, are there spatial or temporal correlations between the network components or events that can be identified via log-analysis?
2. How much do the network events (e.g., packet counts, traffic type, protocols, etc.) influence an attack? If there exists a sequence of network events which indicate an attack, what is the lead time?
3. What is the probability of failing to detect an attack? Under what conditions?

Previous research have conducted analysis of network attacks with little consideration for the influence of multiple network events over simultaneous attacks. These works viewed attacks in isolation with little correlation analysis. In contrast, we take a global view. To summarize, we make the following contributions:

- We identify attacks on a large network, including the components or event correlations. We provide estimates of components or events not correlated with the attack.
- We analyze the network component attributes to drill down into their specific activities. Based on the insights we obtain from our correlation analysis, we discuss their implications for enhanced network security.
- We extract the sequence of network flow events which are associated with an attack and obtain their lead times.

II. System Model, Netflow Data & Approach

The client-server model is illustrated in Fig. 1. A network is protected by firewalls that monitor all the incoming and outgoing traffic. The routers and application servers that reside within the network perimeter may produce event data. In practice, the data is represented in logs such as netflow data [21]. The netflow data contain flow connection-specific events.

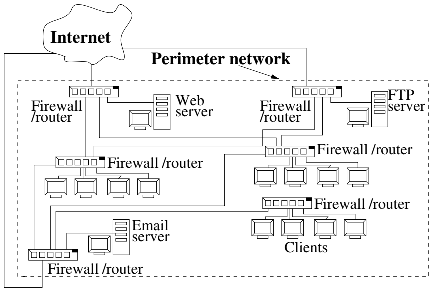


Fig. 1. Client-server model

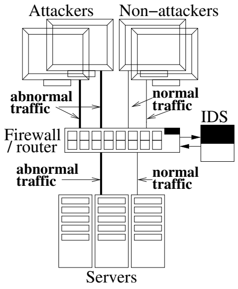


Fig. 2. Investigation approach

A. Netflow Data

The netflow data are monitored on most networks [22]. An example of a flow connection-specific event we called a *network flow event* is given as follows:

```
761, 4434, Comp132598, Comp817788, 6, Port12597, 22,
89159, 154950, 85257, 6976892
```

In this example, the event contains a start time (`761`), duration of the event in seconds (`4434`), source device number (`Comp132598`), destination device number (`Comp817788`), protocol number (`6`), source device port (`Port12597`), destination device port (`22`), number of packets (`89159`) and bytes (`154950`) sent by the source device, and number of packets (`85257`) and bytes (`6976892`) sent by the destination device. Port 22 is a secure shell communication (SSH) that provides remote administration access to a host. Thus, this is a SSH event.

B. Approach

We have consulted the literature on computer networks [23] and related security aspects [24]. Our investigation procedure is depicted in Fig. 2. Clients send requests to a server. The firewall monitors inbound and outbound traffic and forwards suspicious network packets to an IDS (Intrusion Detection System) for inspection. To identify an attack on the network, we perform correlation analysis in the following manner:

1. We trace the events from the source device to the destination device in the netflow data. These include the time of the network flow events. We correlate the flow connection-specific events to ascertain any network-wide influence evident over multiple dates.
2. We correlate the destination ports in the network flow events to ascertain any abnormal traffic over multiple dates. We investigate the requests on the network ports to identify an attack.

III. Identifying Network Attacks

The task to identify network attacks based on the investigation procedure described in Section II-B is difficult. For example, an attacker may try to compromise a host and a non-attacker may also attempt to access a host on the network. As these events take place on random hosts on the network, there are two possibilities: (a) if these events occur on the same host, then it is possible that some events do not lead to an attack, and consequently, it does not constitute an attack, or (b) if the events lead to an observable attack, then this is considered an attack on a single host.

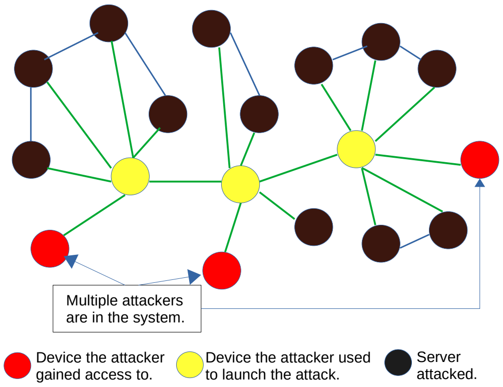

Multiple attackers are in the system.
Device the attacker gained access to.
Device the attacker used to launch the attack.
Server attacked.

Fig. 3. Intuition for simultaneous attacks.

However, if we assume that a network itself is composed of multiple hosts, then simultaneous attacks can be defined via correlated events. Consider a network to be composed of several hosts and client devices as illustrated in Fig. 3. Assume some devices to have the attacker's control. If there exists two devices which attacked two servers, then we call it a simultaneous attack if the devices that executed the attack results in one server to attack another server and vice versa.

The attacks on multiple servers are distinguishable, as they trigger different events from devices with well known source port numbers. For example, a firewall may be configured to allow port 22 (i.e., SSH) inbound/outbound traffic to allow the server to connect to other servers via SSH, and by using SSH as a source port, the attacker hopes to take advantage of such a rule to execute a reflection attack. These events can occur together in time and target servers in different locations. Thus, we define temporal correlation and spatial correlation as: (a) events which occur during the same time period and differ in location and (b) events which occur in the same location and differ in time respectively. Then, we analyze the netflow data in the following section.

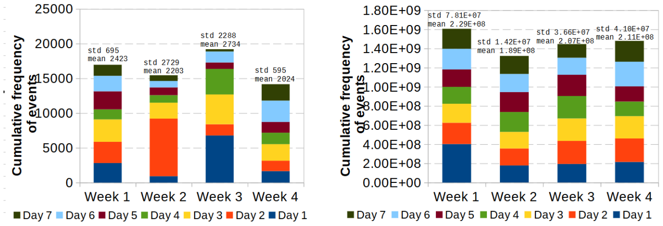
IV. Evaluation Results

Our study is carried out on a large network operated by Los Alamos National Labs. The network hosts 60,000 devices and provides user accounts and data storage services. Many networks also host a large number of devices. Our objective is to conduct detailed analysis of network attacks. In [25], it was reported that sample malicious events exist in the netflow data. However, we do not know which dates contain the malicious events. Therefore, we randomly selected four weeks worth of netflow data for our analysis.

A. *Temporal Correlation*

In the first phase, our goal is to identify the dates of a network attack. To achieve this, we need to address three issues: (a) the date of an attack is unknown, (b) the netflow data span multiple dates and (c) the data contains a large amount of network flow events. To address these issues, we: (a) obtain the time interval between the start time of the network flow events, (b) capture the change in the network flow events and (c) obtain the size and number of time-bins.

We obtain the time interval by grouping the network flow events according to their start time then subtract the start time of adjacent groups of network flow events to get the time interval. Then, we obtain the cumulative frequency of the network flow events per week and calculated their mean and standard deviation in Weeks 1, 2, 3 and 4.



(a) Time interval of 0 seconds.

(b) Time interval of 1 second.

Fig. 4. Frequency of network flow events.

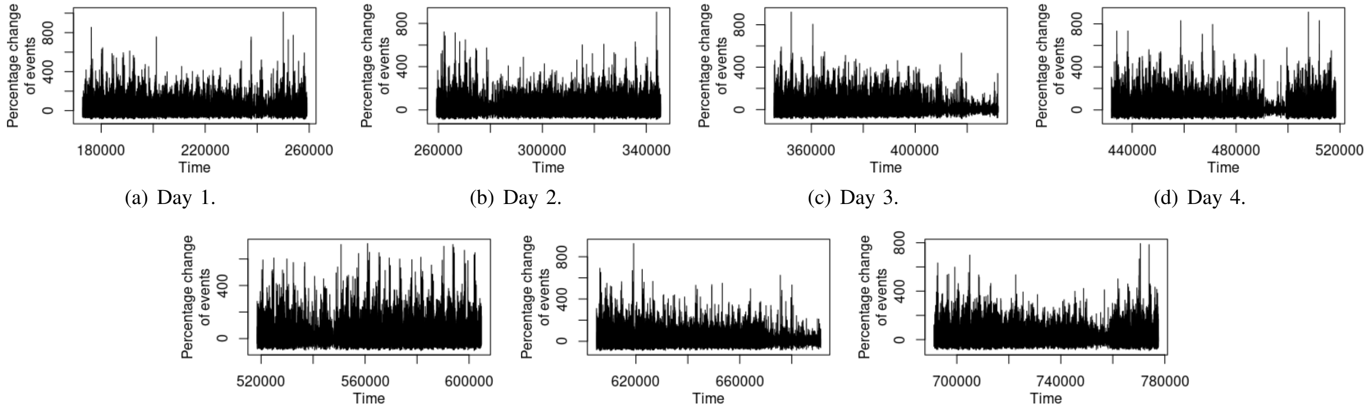
Fig. 4 shows the cumulative frequency of the network flow events. Events that started first during the day have a time interval of 0 seconds. From Fig. 4(a), we observed that the average number of events for Week 1 to Week 4 are 2423, 2203, 2734 and 2024 and the standard deviation is 695, 2729, 2288 and 595 events respectively. From Fig. 4(b), we observed that the average number of events for Week 1 to Week 4 are $2.29E^{+08}$, $1.89E^{+08}$, $2.07E^{+08}$ and $2.11E^{+08}$ and the standard deviation is $7.81E^{+07}$, $1.42E^{+07}$, $3.66E^{+07}$ and $4.10E^{+07}$ respectively. The mean and standard deviation for the network flow events changed over Week 1 and Week 4, indicating the need to investigate the change.

Next, we identify how the network flow events change over time. To understand the meaning between changes in the count of the network flow events at different times, we obtain the percentage change. In [21], it was reported that the volume of network flow events in the netflow data exhibited the periodicity of the traffic in a large network during a typical 5-day work-week. We calculated the percentage change by subtracting the number of events in two adjacent time groups then divide that change by the number of events in the preceding time group and convert that to a percentage [26].

Fig. 5 shows the percentage change for the network flow events in Week 1. We observed that: (a) the largest percentage increase occurred late on Days 1, 2, 4 and 7, (b) the largest percentage increase occurred early on Days 3 and 6 and (c) the largest percentage increase occurred midway and late on Day 5. This shows that the time of the largest percentage increase of network flow events changed over Day 1 and Day 7 in Week 1. We obtained the percentage change for all 7 days in Weeks 2, 3 and 4. The times of the largest percentage increase changed from Day 1 to Day 7 in Weeks 2, 3 and 4.

Next, we identify the size of the time-bins. Data binning is a preprocessing step for presenting the network flow events in a form on which analysis algorithms can be applied. Binning the network events have been applied as a preprocessing step to detect low volume and short duration attacks [7] and separate a set of network traffic measurements that correspond to normal and abnormal behaviour [27]. A time-bin is defined as a window of one fixed time interval. When presenting the data, setting up the time-bins is a decision that has to be made. However, the choice of the time-bin size will have a major effect on how the data can be interpreted. If the size of the time-bin is small, then more bins are needed. When there are too many bins, it can increase the error rate and it will be difficult to discern the signal from the noise. If the size of the time-bin is large, then fewer bins are needed. When there are too few bins, it will lack the details needed to discern any useful pattern in the data. Thus, we use multiple binning methods and applied the binning methods on the counts of the network flow events to obtain the size and number of time-bins. The binning methods are [28]: (a) data range, (b) standard deviation and (c) interquartile range.

Fig. 6 and Fig. 7 show the size and number of time-bins for the network flow events in Week 1. We observed that a long time-bin size and small number of time-bins were obtained on the data range binning method. We observed that short time-bin sizes and large number of time-bins were obtained on the standard deviation and interquartile range binning methods. Specifically, time-bins of 12 minutes to 14 minutes were obtained on the data range method, time-bins of 1.5 minutes to 2 minutes were obtained on the standard deviation method and



(a) Day 1. (b) Day 2. (c) Day 3. (d) Day 4.

(e) Day 5. (f) Day 6. (g) Day 7.

Fig. 5. Percentage change of network flow events in Week 1.

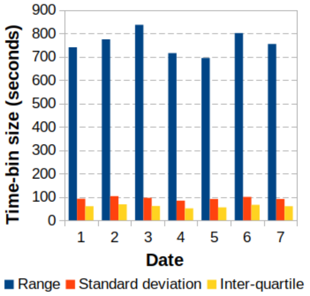


Fig. 6. Time-bin size

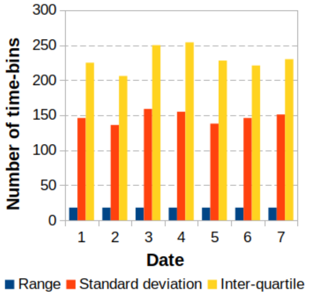
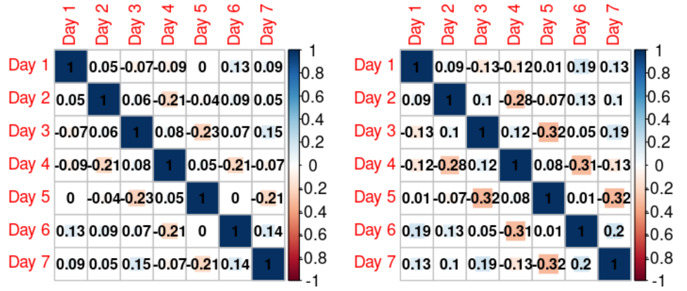


Fig. 7. Number of time-bins.

time-bins of 51 seconds to 68 seconds were obtained on the interquartile range method. Furthermore, we observed that the size and number of time-bins changed over Day 1 and Day 7. We obtained the size and number of time-bins for all 7 days in Weeks 2, 3 and 4. The size and number of time-bins changed over Day 1 and Day 7 in Weeks 2, 3 and 4.

> Time-bins of 51 seconds to 2 minutes were obtained on the standard deviation and interquartile range methods and time-bins of 12 to 14 minutes were obtained on the data range method, indicating the importance of using both long and short time-bins.

Next, we calculate the correlation score for the counts of network flow events on one date to the counts of the network flow events on another date. Pearson correlation is one of the most popular techniques for calculating the strength of the relationship between two variables. Pearson correlation requires that the variables contain the same number of time-bins. However, we showed that the number of time-bins changed on different dates (see Fig. 7). To solve this issue, we grouped the network flow events by their start time and count the number of events per time group. We obtained 86,400 time groups on all the days over Week 1 and Week 4. Pearson correlation assumes that: (a) when the value of one variable increases the value of another variable increases or (b) when the value of one variable decreases the value of another variable decreases. Spearman-Rank correlation measures a monotonic relationship between two variables: (a) when the value of one variable increase the value of another variable increase or (b) when the value of one variable remains the value of another variable remains [29]. The Pearson and Spearman-Rank correlation coefficients range from -1 to 1. We used the following rules of thumb to interpret the strength of the correlation coefficient [26]: (a) two days of network flow events are *strongly positive correlated* when the correlation coefficient lies between 0.8 and 1, (b) two days of network flow events are *moderately positive correlated* when the correlation coefficient lies between 0.3 and 0.79 and (c) two days of network flow events are *weakly positive correlated* when the correlation coefficient lies between 0 and 0.29.

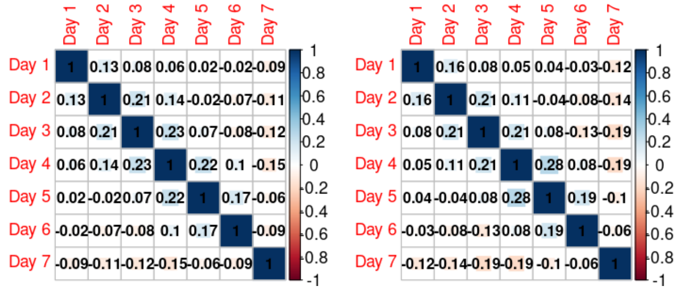


(a) Pearson correlation. (b) Spearman-Rank correlation.

Fig. 8. Correlation score for the days of network flow events in Week 1.

Fig. 8 and Fig. 9 show the correlation score for 7 days in Week 1 and Week 3 respectively. From Fig. 8(a), we observed that the Pearson correlation score range from -0.23 to 0.14. This shows that there is a weak linear relationship between the days of the network flow events. From Fig. 8(b), we observed that the Spearman-Rank correlation score range from -0.32 to 0.19. This show that all 7 days of network flow events are weakly correlated in Week 1.

From Fig. 9(a), we observed that the Pearson correlation score range from -0.15 to 0.23. This shows that there is a weak linear relationship between the dates of the network flow events. From Fig. 9(b), we observed that the Spearman-Rank correlation score range from -0.19 to 0.28. This shows that all the 7 days of network flow events are weakly correlated in Week 3. We applied the Pearson and Spearman-Rank correlation algorithms on all 7 days in Weeks 1, 2, 3 and 4. In Week 1, all 7 days are weakly correlated. In Week 2, all 7 days are weakly correlated.



(a) Pearson correlation.

(b) Spearman-Rank correlation.

Fig. 9. Correlation score for the days of network flow events in Week 3.

In Week 3, all 7 days are weakly correlated. In Week 4, all 7 days are weakly correlated.

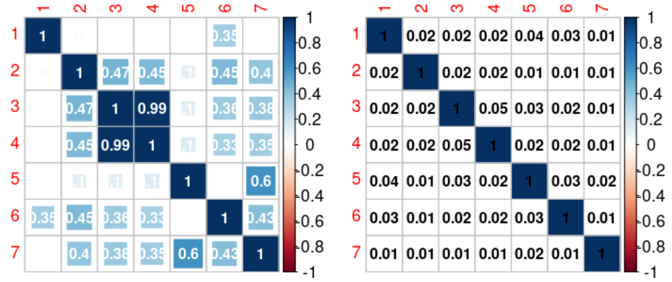
> The network flow events are weakly correlated between all seven days in Weeks 1, 2, 3 and 4, indicating that correlating the dates of the network flow events by time did not identify the date of an attack.

B. Spatial Correlation

The first phase of our analysis is characterized by the correlation of the days of the network flow events by time. We specifically observed that all the days of the network flow events are weakly correlated in Weeks 1, 2, 3 and 4. This confirms that the network flow events are not correlated by time. However, our objective is to identify the date of an attack. To achieve this, we determine the strength of the relationship between multiple pairs of days by their network ports.

Thus, in the second phase of our analysis, we obtain the correlation score between two days by the number of groups of destination ports. We grouped the destination ports according to their port number and count the number of destination ports to create a list of groups of destination ports by day. We applied Pearson and Spearman-Rank correlation algorithms on the lists of groups of destination ports for all 7 days in Weeks 1, 2, 3 and 4. In Week 1, 2 days are strongly positive correlated. In Week 2, all the 7 days are weakly correlated. In Week 3, 3 days are strongly positive correlated. In Week 4, all the 7 days are weakly correlated. Fig. 10 and Fig. 11 show the correlation score for 7 days in Weeks 1 and 3 respectively. From Fig. 10(a), we observed that there is a strong positive linear relationship between Day 3 and Day 4 in Week 1. The Pearson correlation score is 0.99. From Fig. 10(b), we observed that there is a weak monotonic relationship between all the days in Week 1. The Spearman-Rank correlation score range from 0 to 0.05.

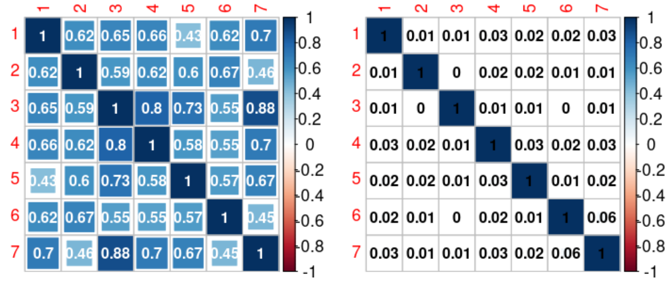
From Fig. 11(a), we observed that there is a strong positive linear relationship between (a) Day 3 and Day 4 and (b) Day 3 and Day 7 in Week 3. Their correlation scores are 0.8 and 0.88 respectively. From Fig. 11(b), we observed that there is a weak monotonic relationship between all the days in Week 3. The Spearman-Rank correlation score range from 0 to 0.06. The strongly positive correlated days were identified by Pearson correlation only. When Pearson correlation identifies all the days that are strongly positive correlated, it can be used as the primary method.



(a) Pearson correlation.

(b) Spearman-Rank correlation.

Fig. 10. Correlation score for the days of network flow events in Week 1.



(a) Pearson correlation.

(b) Spearman-Rank correlation.

Fig. 11. Correlation score for the days of network flow events in Week 3.

> A strong positive correlation was obtained for (a) two days in Week 1 and (b) three days in Week 3, indicating that it is likely that an attack occurred on these dates.

Validation: Next, we test the significance of the correlation coefficient. To test the significance of the correlation coefficient, we apply a standard technique called Fisher's z-score [29]. We define the null and alternate hypothesis using the following terminology. The null hypothesis is that a pair of dates are weakly positive correlated. The alternate hypothesis is that a pair of dates are strongly positive correlated. Then, we obtain the z-scores for all correlation coefficients. When the absolute value of z is large, e.g., $z = 2.94$ at 99% confidence level, we reject the null hypothesis in favour of the alternate hypothesis. We obtain the z-scores for all the dates that are *strongly positive* correlated. The z-scores range from 3.58 to 12.13. At 99% confidence level, under the null hypothesis, $z = 2.64$. Hence, we reject the null hypothesis in favour of the alternate hypothesis.

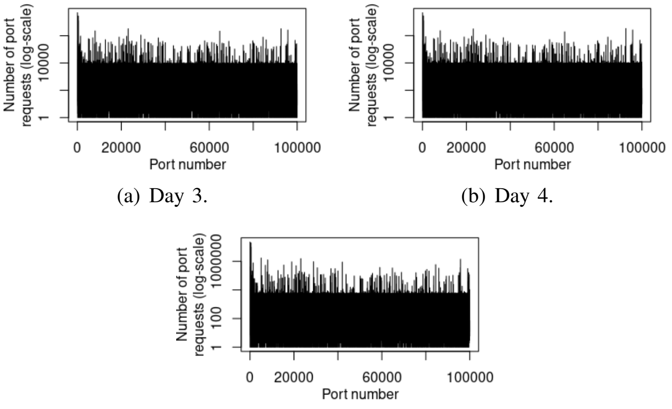
Handling False Positives: When multiple independent tests are being performed, the probability that there is at least one false positive due to chance increases. For example, if we have 20 hypotheses and obtained a P-value of 0.01 for each test, then the false positive rate is $1 - (1 - 0.01)^{20} = 1 - 0.99^{20} = 0.18$ or 18%. To solve the problem of inflation in false positives due to multiple hypothesis tests, we apply a standard technique called the Bonferroni Correction [30]. It obtains an adjusted P-value by multiplying the unadjusted P-value by the number of tests. To determine the probability of rejecting the null hypothesis when it is true, we apply a one-sided test and use the significance level, $\alpha = 0.01$ for all given hypothesis tests to obtain a P-value on Week 1 and Week 3. We obtained the z-scores for the dates in Week 1 and Week 3. The lowest z-score is 3.58. Since this is a one-sided test, the P-value is 0.00017. We obtained the adjusted P-value $0.00017 \times 14 = 0.0024$ where 14 is the number of dates. The adjusted P-value is less than 0.01, indicating that it is highly unlikely this result would be observed under the null hypothesis. The z-scores for all the correlation coefficients are greater than or equal to 3.58 and the adjusted P-values are less than 0.01, indicating it is highly unlikely these results would be observed under the null hypothesis.

The P-values for the correlation coefficients on Days 3 and 4 in Week 1, Days 3 and 4 in Week 3 and Days 3 and 7 in Week 3 are less than 0.01, indicating there is a low probability of identifying a false date of an attack.

C. Identify Network Attacks

The second phase of our analysis is characterized by the correlation of the days of network flow events by their destination ports. We specifically observed that: (a) Day 3 and Day 4 are strongly positive correlated in Week 1, (b) Day 3 and Day 4 are strongly positive correlated in Week 3 and (c) Day 3 and Day 7 are strongly positive correlated in Week 3. The strong positive correlation on those days indicate that an attack is likely to have occurred. However, our objective is to identify the attack on the network. To achieve this, we extract the destination ports to drill down into their specific activities.

Thus, in the third phase of our analysis, we extract the count of destination ports. We group the destination ports by their port numbers and count the destination ports. Fig. 12 shows the number of destination ports in Week 3. From Fig. 12(a), Fig. 12(b) and Fig. 12(c), we observed that ports 0 to 1024 received the highest number of requests. As was done with the destination ports in Fig. 12, we identify the destination ports that received the highest number of requests in Week 1. Fig. 13 shows the number of destination port requests. From Fig. 13(a) and Fig. 13(b), we observed that ports 0 to 1024 received the highest number of requests.



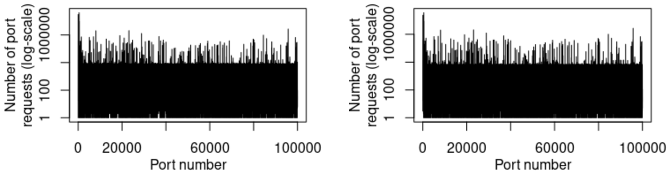
(a) Day 3.

(b) Day 4.

(c) Day 7.

Fig. 12. Destination port requests in Week 3.

Reflection attacks on large networks have been widely reported [31]. To identify a reflection attack, we extracted the source port that matched the destination port number in the network flow events on all the five dates. We identified many attacks though we focused on a subset of these attacks as reflection attacks on the SSH and DNS servers.



(a) Day 3.

(b) Day 4.

Fig. 13. Destination port requests in Week 1.

1) Reflection Attack on the SSH Server: When an attacker attempts to use a computer or device on the network to trick other computers by masquerading as a legitimate entity, one of the tools used to gain access to a computer is called spoofing. A spoofed attack involves replacing the source address in the IP header with the address of the target computer. This causes all the replies to go to the target computer. If the target computer receives a large number of replies, the computer becomes unresponsive. When a firewall is configured to allow port 22 inbound and outbound traffic, an attacker can take advantage of such a rule and use a spoofed IP address to connect to other SSH servers. We scanned the network flow events and identified several events that contain both source port 22 and destination port 22. The port requests are summarized in Table I. From Table I, we observed that spoofed requests were sent by the source devices on Days 3 and 4 in Week 1 and Days 3, 4 and 7 in Week 3.

TABLE I
PORT 22 REQUESTS.

<table>
<tr><th colspan="5">Week 1</th></tr>
<tr><th>Day</th><th>Source device</th><th>Destination device</th><th>Source packets</th><th>Destination packets</th></tr>
<tr><td>3</td><td>5</td><td>4</td><td>19</td><td>2427</td></tr>
<tr><td>4</td><td>6</td><td>3</td><td>74</td><td>9852</td></tr>
<tr><th colspan="5">Week 3</th></tr>
<tr><th>Day</th><th>Source device</th><th>Destination device</th><th>Source packets</th><th>Destination packets</th></tr>
<tr><td>3</td><td>4</td><td>4</td><td>65</td><td>8731</td></tr>
<tr><td>4</td><td>4</td><td>5</td><td>29</td><td>2468</td></tr>
<tr><td>7</td><td>1</td><td>1</td><td>6</td><td>302</td></tr>
</table>

Next, we obtain the percentage of the packets associated with the attack. We extracted the number of source packets and destination packets on Days 3 and 4 in Week 1 and Days 3, 4 and 7 in Week 3. On Day 3 of Week 1, there are 436,498,571 source packets and 1,366,801,370 destination packets. The port 22 source and destination packets make up 0.000006% and 0.00017% of the number of source and destination packets respectively. On Day 4 of Week 1, there are 75,187,450 source packets and 1,648,296,718 destination packets. The port 22 source and destination packets make up 0.00009% and 0.00059% of the number of source and destination packets respectively. On Day 3 of Week 3, there are 614,097,253 source packets and 1,646,568,662 destination packets. The port 22 source and destination packets make up 0.00001% and 0.0005% of the number of source and destination packets respectively. On Day 4 of Week 3, there are 1,756,318,108 source packets and 2,124,728,448 destination packets. The port 22 source and destination packets make up 0.000001% and 0.00011%

of the number of source and destination packets respectively. On Day 7 of Week 3, there are 464,206,047 source packets and 902,234,420 destination packets. The port 22 source and destination packets make up 0.000001% and 0.000033% of the number of source and destination packets respectively.

> The reflection attack on the SSH server comprises a small percentage of all the network packets on Days 3 and 4 in Week 1 and Days 3, 4 and 7 in Week 3, indicating that the traffic generated by the attack did not overwhelm the SSH server.

2) Reflection Attack on the DNS server: The Domain Name System (DNS) is a database that stores internet domain names and translates them into IP addresses. When an attacker executes an attack on a DNS server, they use a spoofed IP address to send a spoof request to the DNS server. The spoofed request contains the address of another DNS server. The DNS server replies to the request, creating an attack on the target server. We scanned the network flow events and identified several events that contain both source port 53 and destination port 53. The port requests are summarized in Table II. From Table II, we observed that spoofed requests were sent by the source devices on Days 3 and 4 in Week 1 and Days 3, 4 and 7 in Week 3.

TABLE II
PORT 53 REQUESTS.

Week 1				
Day	**Source device**	**Destination device**	**Source packets**	**Destination packets**
3	2	2	582	58449
4	2	3	65	5131
Week 3				
Day	**Source device**	**Destination device**	**Source packets**	**Destination packets**
3	4	4	90	6975
4	6	4	146	11735
7	3	3	28	3224

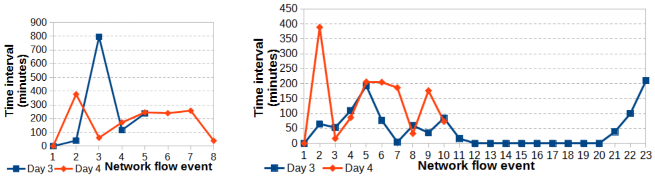
Next, we obtain the percentage of the packets associated with the DNS reflection attack. On Day 3 of Week 1, the port 53 source and destination packets make up 0.00013% and 0.004% of the number of source and destination packets respectively. On Day 4 of Week 1, the port 53 source and destination packets make up 0.00008% and 0.0003% of the number of source and destination packets respectively. On Day 3 of Week 3, the port 53 source and destination packets make up 0.000014% and 0.0004% of the number of source and destination packets respectively. On Day 4 of Week 3, the port 53 source and destination packets make up 0.000008% and 0.00055% of the number of source and destination packets respectively. On Day 7 of Week 3, the port 53 source and destination packets make up 0.000006% and 0.00034% of the number of source and destination packets respectively.

> The reflection attack on the DNS server comprises a small percentage of all the network packets on Days 3 and 4 in Week 1 and Days 3, 4 and 7 in Week 3, indicating that the traffic generated by the attack did not overwhelm the DNS server.

D. Lead Times of Reflection Attacks

The third phase of our analysis is characterized by the identification of reflection attacks on the SSH and DNS servers. We observed that these attacks occurred on multiple dates. This confirms that the network was attacked. However, our objective is to obtain the lead time of the attack. We define a lead time as the time interval between two network flow events. To achieve this, we extract the sequence of network flow events which are associated with the attack.

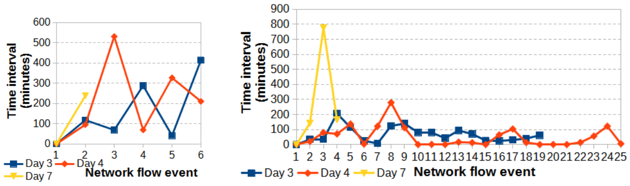
Thus, in the fourth phase of our analysis, we obtained the time interval between the network flow events by subtracting the start time of adjacent network flow events that are associated with the attack. Fig. 14 shows the time interval between the network flow events on Days 3 and 4 in Week 1. From Fig. 14(a), we observed that: (a) on Day 3 the shortest time interval is 40 minutes and the longest time interval is 795 minutes and (b) on Day 4 the shortest time interval is 39 minutes and the longest time interval is 377 minutes. From Fig. 14(b), we observed that: (a) on Day 3 the shortest time interval is 4 minutes and the longest time interval is 210 minutes and (b) on Day 4 the shortest time interval is 16 minutes and the longest time interval is 389 minutes.



(a) SSH server attack. (b) DNS server attack.

Fig. 14. Lead time of the network flow events on Days 3 and 4 in Week 1.

As was done with the network flow events in Fig. 14, we obtain the time interval between the network flow events in Week 3. Fig. 15 shows the time interval between the network flow events on Days 3, 4 and 7. From Fig. 15(a), we observed that: (a) on Day 3 the shortest time interval is 41 minutes and the longest time interval is 415 minutes, (b) on Day 4 the shortest time interval is 69 minutes and the longest time interval is 531 minutes and (c) on Day 7 the time interval is 239 minutes. From Fig. 15(b), we observed that: (a) on Day 3 the shortest time interval is 7 minutes and the longest time interval is 207 minutes, (b) on Day 4 the shortest time interval is 1 minute and the longest time interval is 279 minutes and (c) on Day 7 the shortest time interval is 143 minutes and the longest time interval is 777 minutes.



(a) SSH server attack. (b) DNS server attack.

Fig. 15. Lead time of the network flow events on Days 3, 4 and 7 in Week 3.

There is a minimum lead time for the network flow events that are associated with the reflection attacks. The minimum lead time range from 39 minutes to 239 minutes for the reflection attack on the SSH server. The minimum lead time range from 1 minute to 143 minutes for the reflection attack on the DNS server.

Next, we identify the start time of the network flow events associated with the SSH and DNS server attack in Week 1. For the network flow events associated with the SSH server attack, their start time range from (a) 353365 to 424737 on Day 3 of Week 1 and (b) 433647 to 517197 on Day 4 of Week 1. For the network flow events associated with the DNS server attack, their start time range from (a) 349522 to 412427 on Day 3 of Week 1 and (b) 434754 to 517123 on Day 4 of Week 1. In Fig. 5(c) and Fig. 5(d) in Section IV-A, we showed that the largest percentage increase in the network flow events occurred at different times on Days 3 and 4 in Week 1. The start times of the network flow events that are associated with the SSH server attack occurred at different times on Days 3 and 4. The time of some network flow events coincided with the time of the largest percentage increase in the network flow events. The time of other network flow events coincided with the time of smaller percentage increase in the network flow events.

As was done with the SSH and DNS servers attack in Week 1, we identify the start time of the network flow events associated with the attacks on the SSH and DNS servers in Week 3. For the network flow events associated with the SSH server attack, their start time range from (a) 4941046 to 4996800 on Day 3 of Week 3, (b) 5013002 to 5086915 on Day 4 of Week 3 and (c) 5305620 to 5319931 on Day 7 of Week 3. For the network flow events associated with the DNS server attack, their start time range from (a) 4934123 to 5008026 on Day 3 of Week 3, (b) 5014777 to 5088625 on Day 4 of Week 3 and (c) 5282726 to 5347824 on Day 7 of Week 3. The start time of the network flow events on Days 3, 4 and 7 range from (a) 4924800 to 5011199, (b) 5011199 to 5097599 and (c) 5270400 to 5356320 respectively. The attack on the SSH and DNS servers occurred at different times during the day. The time of some network flow events coincided with the time of the largest percentage increase in the network flow events. The time of other network flow events coincided with the time of smaller percentage increase in the network flow events.

While the time of some reflection attacks coincided with the time of the largest percentage increase in the network flow events, the time of other reflection attacks coincided with the time of smaller percentage increase in the network flow events. This implies that a large increase in the number of network flow events can reveal a reflection attack but it may also miss other attacks.

V. Discussion

From these results, we showed that a correlation analysis approach is unsuitable as a means to identify network attacks. Our analysis over the netflow data on a large network helps to become cognizant of the extent to which the network flow events are correlated to attacks. The fact that the majority of network flow events are not the primary indicators of an attack is not obvious, for example: increase in the network flow events does not necessarily indicate an attack. Besides, the netflow data containing the time of an attack and malicious events imply that a comprehensively labeled cyber-security dataset is important [32], [33]. We summarized our findings in Table III.

We observed that the traffic generated by the SSH and DNS reflection attacks did not overwhelm the servers. While network operators are less concerned with attacks which do not lead to a denial-of-service, it is better to equip the IDS and attack predictors to be aware of early signs of an attack to reduce service downtime. These recommendations are suitable for diverse networks as well, since complex network topologies, for example, Wide Area Networks can also benefit from netflow data analysis.

VI. Threats to Validity

We identified the following threats to validity: (a) the quality of the netflow data that can lead to variations of the network flow events over time, and thus could mislead our correlation analysis and (b) the selection of the target network.

As for the quality of the netflow data, the reference in [21] showed that the volume of network flow events matched what one would expect on a large network during a typical 5-day workweek. Thus, we have selected the dates that corresponded to a 5-day workweek (see Section IV-A). The limitation of our analysis is that, in the initial phase, we chose to focus on the times of the network flow events, before extending our approach to the destination ports in the netflow data using the correlation techniques discussed in Section IV-A.

Further considerations for user permission details [34], hardware performance counters [35] or behaviour logs analysis [36] is beyond the scope of this work. Since some network sites do not maintain incident reports [37] or collect detailed security logs and others may not release the data due to restrictions in their distribution policies [22], this makes our statistical inference difficult to confirm. Having said that, incident reports [38] and monitoring tools [39] are currently being integrated into these networks. Thus, validating our analyses has become practically attainable.

The conclusions we presented are based on the netflow data of a large network, and may not generalize to all types of networks. As the chosen network is widely used, we believe that the results are representative for a larger set of networks. Apart from whether or not the results are generalizable, we showed that reflection attacks on the network exist and that the network flow events are not strongly correlated to these attacks. The preprocessed data is available at https://tinyurl.com/ym58zwwv for supporting research into network attacks.

VII. Related Work

In [27], the authors proposed a general method for diagnosing network anomalies. Their method used Principal Component Analysis to identify normal and abnormal network conditions by using a set of network traffic measurements. In [7], the authors modified the cross-correlation function to improve the anomaly detection performance of the conventional cross-correlation function. In [4], the authors setup shallow Convolution Neural Network (CNN), moderate CNN and deep CNN to assess

TABLE III
FINDINGS AND RECOMMENDATIONS

Finding	Recommendation
Majority of the network flow events in the netflow data are not strongly correlated to reflection attacks.	Frequent occurrence of network flow events can be ignored unless major indicators are observed in the netflow data.
Spatial correlation of server attacks exist. Spoofed requests can trigger reflection attacks on SSH and DNS servers.	Network administrators can incorporate additional packet inspection tests in IDS to account for the servers attacking due to forged packets to track the source device besides closing the network port.
Minimum time intervals exist for certain spoofed request triggered reflection attacks helping in lead time improvements.	Network attack prediction schemes can incorporate these lead time enhancements for proactive attack handling.
The timing of some reflection attacks on the SSH and DNS servers did not coincide with the time of an increase in the percentage of network flow events.	Conducting a study of various anomaly detection algorithms on netflow data could improve the anomaly detector's accuracy in detecting network attacks.

TABLE IV
LARGE-SCALE NETWORK ATTACK STUDIES.

Paper	Focus	Finding
[27], [4], [5], [6], [7]	Network anomalies	A PCA-based method can diagnose large volume anomalies with very low false positive rate [27], a modified cross-correlation function can detect low volume and short duration attacks [7], deeper CNN structures did not improve the network anomaly detection performance [4], a hybrid feature selection CNN deep learning based model can improve the network anomaly accuracy and detection rates [5], significant deviations from a baseline network can be detected by a Bayesian hierarchical model [6].
[40], [41], [42] [43]	Botnets	Unique botnet infections can be related to IP addresses [40], new botnets such as Hajime [43] which targets many of the devices that Mirai [41] targets, can introduce new exploits thus increasing the resilience of botnets, a hybrid flow-based and graph-based analysis approach can detect botnets with high accuracy [42].
[15], [16], [31]	Network protocols	A multi-stage approach comprising of lightweight and focused anomaly detectors can improve DDoS detection accuracy with a low false negative rate [15], simple email messages sent to a SMTP server can overload it without consuming all the network bandwidth [16], well known amplification attack protocols exists and new amplification attack protocols are recently discovered [31].
Our work	Network protocols	Netflow data is not strongly correlated to network attacks, reflection attacks exist in the netflow data, lead time enhancements are feasible for reflection attacks that exhibit long dwell times.

the impact of the depth of the CNN on the performance of anomaly detection. They evaluated their models on the Kyoto-Honeypot, MAWILab and NSL-KDD datasets. In [6], a Bayesian hierarchical model was proposed to estimate the traffic rates and detect anomalous changes in the network. In [5], the authors proposed a hybrid data processing model for efficient network anomaly detection. In their model, an improved grey wolf optimization algorithm and CNN were developed. They compared their model to other state-of-the-art models used for network anomaly detection.

In [40], the authors detailed their efforts in taking control of the Torpig botnet and studied how the malware infected millions of IP addresses. In [41], the authors provided a detailed study on the types of devices that were infected by the Mirai Internet-of-Things (IoT) botnet and analyzed how Mirai emerged and infected vulnerable hosts. In [42], the authors proposed a hybrid flow-based and graph-based network traffic analysis approach to detect botnets on the network. In [43], the authors performed a detailed measurement of the scans executed by the Hajime botnet. They showed that Hajime can be used to understand how IoT botnets operate.

In [15], the authors designed and implemented a multi-stage approach called LADS to detect DDoS attacks. Their approach is comprised of a lightweight anomaly detector and a focused anomaly detector. In [16], the authors implemented a test environment that consisted of email servers and clients to test the performance of SMTP (Simple Main Transfer Protocol) servers against DDoS (Distributed Denial-of-Service) attacks. In [31], the authors analyzed multiple terabits of network traffic flows at a major Internet Exchange Point. They identified up to 2,608 DDoS amplification attacks on a single day. We summarized the findings in [4]–[7], [15], [16], [27], [31], [40]–[43] in Table IV. These studies have provided valuable insights into network attacks and our work complements them by identifying reflection attacks and extracting the lead times of the attack.

VIII. CONCLUSION

An approach based on correlation of netflow data is presented to identify network attacks. We showed that reflection attacks on the SSH and DNS servers exist in the netflow data and identified the lead times of those attacks. We determined that the netflow data is not strongly correlated to network attacks. Choosing an attack mitigation scheme with the understanding of the patterns of the network flow events when an attack is imminent can have long-term benefits in proactive attack handling.

ACKNOWLEDGEMENTS

This work is conducted, in part, under the auspices of the EU Horizon 2020 Research and Innovation program under Grant Agreement No. 830927 (CONCORDIA) and by Security Lancaster under the EPSRC grant EP/V026763/1.

REFERENCES

[1] A. Zimba, H. Chen, Z. Wang, and M. Chishimba, "Modeling and detection of the multi-stages of Advanced Persistent Threats attacks based on semi-supervised learning and complex networks characteristics," *Future Generation Computer Systems*, vol. 106, pp. 501 – 517, 2020.

[2] X. Wang, Q. Liu, Z. Pan, and G. Pang, "APT attack detection algorithm based on spatio-temporal association analysis in industrial network," *Journal of Ambient Intelligence and Humanized Computing*, 2020.

[3] D. S. Terzi, R. Terzi, and S. Sagiroglu, "Big data analytics for network anomaly detection from netflow data," in *Proceedings of International Conference on Computer Science and Engineering (UBMK)*, 2017, pp. 592–597.

[4] D. Kwon, K. Natarajan, S. C. Suh, H. Kim, and J. Kim, "An empirical study on network anomaly detection using convolutional neural networks," in *Proceedings of IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, 2018, pp. 1595–1598.

[5] S. Garg, K. Kaur, N. Kumar, G. Kaddoum, A. Y. Zomaya, and R. Ranjan, "A hybrid deep learning-based model for anomaly detection in cloud datacenter networks," *IEEE Transactions on Network and Service Management*, vol. 16, no. 3, pp. 924–935, 2019.

[6] E. Hou, Y. Yılmaz, and A. O. Hero, "Anomaly detection in partially observed traffic networks," *IEEE Transactions on Signal Processing*, vol. 67, no. 6, pp. 1461–1476, 2019.

[7] B. AsSadhan, A. Alzoghaiby, H. Binsalleeh, K. G. Kyriakopoulos, and S. Lambotharan, "Network anomaly detection using a cross-correlation-based long-range dependence analysis," *International Journal of Network Management*, vol. 30, no. 6, 2020.

[8] G. Kaiafas, G. Varisteas, S. Lagraa, R. State, C. D. Nguyen, T. Ries, and M. Ourdane, "Detecting malicious authentication events trustfully," in *Proceedings of IEEE/IFIP Network Operations and Management Symposium (NOMS)*, 2018, pp. 1–6.

[9] F. Amrouche, S. Lagraa, G. Kaiafas, and R. State, "Graph-based malicious login events investigation," in *Proceedings of IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, 2019, pp. 63–66.

[10] H. Bian, T. Bai, M. A. Salahuddin, N. Limam, A. A. Daya, and R. Boutaba, "Host in danger? detecting network intrusions from authentication logs," in *Proceedings of 15th International Conference on Network and Service Management (CNSM)*, 2019, pp. 1–9.

[11] Z. Liu, H. Jin, Y. Hu, and M. Bailey, "Practical proactive DDoS-attack mitigation via endpoint-driven in-network traffic control," *IEEE/ACM Transactions on Networking*, vol. 26, no. 4, pp. 1948–1961, 2018.

[12] N. Agrawal and S. Tapaswi, "A proactive defense method for the stealthy EDoS attacks in a cloud environment," *International Journal of Network Management*, vol. 30, no. 2, p. e2094, 2020.

[13] I. Friedberg, F. Skopik, G. Settanni, and R. Fiedler, "Combating advanced persistent threats: From network event correlation to incident detection," *Computers and Security*, vol. 48, pp. 35 – 57, 2015.

[14] R. Kumari and K. Sharma, "Cross-layer based intrusion detection and prevention for network," *Handbook of Research on Network Forensics and Analysis Techniques*, pp. 1–19, 2018.

[15] V. Sekar, N. Duffield, O. Spatscheck, J. van der Merwe, and H. Zhang, "LADS: Large-scale automated DDOS detection system," in *Proceedings of the Annual Conference on USENIX '06 Annual Technical Conference*, ser. ATEC '06. USA: USENIX Association, 2006, p. 16.

[16] B. Bencsath and M. A. Ronai, "Empirical analysis of denial of service attack against SMTP servers," in *Proceedings of International Symposium on Collaborative Technologies and Systems*, 2007, pp. 72–79.

[17] P. Manadhata, J. Wing, M. Flynn, and M. McQueen, "Measuring the attack surfaces of two FTP daemons," in *Proceedings of the 2nd ACM Workshop on Quality of Protection*, ser. QoP '06. New York, NY, USA: Association for Computing Machinery, 2006, p. 3–10.

[18] A. R. Chordiya, S. Majumder, and A. Y. Javaid, "Man-in-the-middle (MITM) attack based hijacking of HTTP traffic using open source tools," in *Proceedings of IEEE International Conference on Electro/Information Technology (EIT)*, 2018, pp. 0438–0443.

[19] G. Yan, Q. Li, D. Guo, and X. Meng, "Discovering suspicious APT behaviors by analyzing DNS activities," *Sensors*, vol. 20, no. 3, 2020.

[20] E. Schultz, "A framework for understanding and predicting insider attacks," *Computers & Security*, vol. 21, no. 6, pp. 526–531, 2002.

[21] M. M. Turcotte, A. D. Kent, and C. Hash, "Unified host and network data set," *Data Science for Cyber Security*, p. 16, 2018.

[22] D. Zhou, Z. Yan, Y. Fu, and Z. Yao, "A survey on network data collection," *Journal of Network and Computer Applications*, vol. 116, pp. 9–23, 2018.

[23] F. Halsall, *Data Communications, Computer Networks and Open Systems*. Addison-Wesley, 1996.

[24] M. Osborne, *How to cheat at managing information security*. Computer Security, 1st Edition, Elsevier, 2006.

[25] H. Zhenzheng, Q. He, E. Chuah, and et. al., "Developing data science tools for improving enterprise cyber-security," in *The Alan Turing Institute Data Study Group Final Report*, 2018. [Online]. Available: http://doi.org/10.5281/zenodo.3558251

[26] A. Agresti and C. Franklin, *Statistics: The Art and Science of Learning From Data*. Prentice Hall International, 2009.

[27] L. Anukool, C. Mark, and D. Christophe, "Diagnosing network-wide traffic anomalies," in *Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, ser. SIGCOMM '04. New York, NY, USA: Association for Computing Machinery, 2004, p. 219–230.

[28] M. P. Wand, "Data-based choice of histogram bin width," *The American Statistician*, vol. 51, no. 1, pp. 59–64, 1997.

[29] R. E. Walpole, R. H. Myers, and S. L. Myers, *Probability and Statistics for Engineers and Scientists*. Prentice Hall International, 1998.

[30] J. P. Shaffer, "Multiple hypothesis testing," *Annual Review of Psychology*, vol. 46, no. 1, pp. 561–584, 1995.

[31] K. Daniel, D. Christoph, and H. Oliver, "DDoS never dies? An IXP perspective on DDoS amplification attacks," in *Passive and Active Measurement*, H. Oliver, L. Andra, and L. Dave, Eds. Springer International Publishing, 2021, pp. 284–301.

[32] M. Sarhan, S. Layeghy, N. Moustafa, and M. Portmann, "Netflow datasets for machine learning-based network intrusion detection systems," in *Big Data Technologies and Applications*, Z. Deze, H. Huang, R. Hou, S. Rho, and N. Chilamkurti, Eds. Springer International Publishing, 2021, pp. 117–135.

[33] L. Dias, S. Valente, and M. Correia, "Go with the flow: Clustering dynamically-defined netflow features for network intrusion detection with DynIDS," in *2020 IEEE 19th International Symposium on Network Computing and Applications (NCA)*, 2020, pp. 1–10.

[34] J. Li, L. Sun, Q. Yan, Z. Li, W. Srisa-an, and H. Ye, "Significant permission identification for machine-learning-based android malware detection," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 7, pp. 3216–3225, 2018.

[35] S. Das, K. James, J. Werner, M. Antonakakis, M. Polychronakis, and F. Monrose, "A flexible framework for expediting bug finding by leveraging past (mis-)behavior to discover new bugs," in *Proceedings of Annual Computer Security Applications Conference (ACSAC)*, 2020, pp. 1–15.

[36] Q. Chen and R. A. Bridges, "Automated behavioral analysis of malware: A case study of wannacry ransomware," in *Proceedings of IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2017, pp. 454–460.

[37] M. Ahmed, A. Naser Mahmood, and J. Hu, "A survey of network anomaly detection techniques," *Journal of Network and Computer Applications*, vol. 60, pp. 19–31, 2016.

[38] A. Vázquez-Ingelmo, Á. M. Moreno-Montero, and F. J. García-Peñalvo, *Threats Behind Default Configurations of Network Devices: Wired Local Network Attacks and Their Countermeasures*. Cham: Springer International Publishing, 2020, pp. 133–172.

[39] S. Troia, L. M. M. Zorello, A. J. Maralit, and G. Maier, "Sd-wan: An open-source implementation for enterprise networking services," in *Proceedings of International Conference on Transparent Optical Networks (ICTON)*, 2020, pp. 1–4.

[40] B. Stone-Gross, M. Cova, L. Cavallaro, B. Gilbert, M. Szydlowski, R. Kemmerer, C. Kruegel, and G. Vigna, "Your botnet is my botnet: Analysis of a botnet takeover," in *Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS)*, ser. CCS '09. New York, NY, USA: Association for Computing Machinery, 2009, p. 635–647.

[41] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, C. Lever, Z. Ma, J. Mason, D. Menscher, C. Seaman, N. Sullivan, K. Thomas, and Y. Zhou, "Understanding the mirai botnet," in *Proceedings of 26th USENIX Security Symposium*. Vancouver, BC: USENIX Association, Aug. 2017, pp. 1093–1110.

[42] W. Wang, Y. Shang, Y. He, Y. Li, and J. Liu, "BotMark: Automated botnet detection with hybrid analysis of flow-based and graph-based traffic behaviors," *Information Sciences*, vol. 511, pp. 284–296, 2020.

[43] S. Herwig, K. Harvey, G. Hughey, R. Roberts, and D. Levin, "Measurement and analysis of hajime, a peer-to-peer IoT botnet," in *Proceedings of Network and Distributed Systems Security (NDSS) Symposium*, 2019, pp. 1–15.