Chapter 1

Unified Host and Network Data Set

Melissa J. M. Turcotte[*,‡], **Alexander D. Kent**[*] **and Curtis Hash**[†]

[*] *Los Alamos National Laboratory,*
Los Alamos, NM 87545, USA
[†] *Ernst & Young, New Mexico, USA*
[‡] *mturcotte@lanl.gov*

The lack of data sets derived from operational enterprise networks continues to be a critical deficiency in the cyber-security research community. Unfortunately, releasing viable data sets to the larger community is challenging for a number of reasons, primarily the difficulty of balancing security and privacy concerns against the fidelity and utility of the data. This chapter discusses the importance of cyber-security research data sets and introduces a large data set derived from the operational network environment at Los Alamos National Laboratory (LANL). The hope is that this data set and associated discussion will act as a catalyst for both new research in cyber-security as well as motivation for other organisations to release similar data sets to the community.

1. Introduction

The lack of diverse and useful data sets for cyber-security research continues to play a profound and limiting role within the relevant research communities and their resulting published research. Organisations are reticent to release data for security and privacy reasons. In addition, the data sets that are released are encumbered in a variety of ways, from being stripped of so much information that they no longer provide rich research and analytical opportunities, to being so constrained by access restrictions that key details are lacking and independent validation is difficult. In many cases, organisations do not collect relevant data in sufficient volumes or with high enough fidelity to provide cyber-research value. Unfortunately, there is generally little motivation for organisations to overcome these obstacles.

In an attempt to help stimulate a larger research effort focused on operational cyber-data as well as to motivate other organisations to release useful data sets, Los Alamos National Laboratory (LANL) has released two data sets for public use (Kent, 2014, 2016). A third, entitled the *Unified Host and Network Data Set*, is introduced in this chapter.

The Unified Host and Network Data Set is a subset of network flow and computer events collected from the LANL enterprise network over the course of approximately 90 days.[a] The host (computer) event logs originated from the majority of LANL's computers that run the Microsoft Windows operating system. The network flow data originated from many of the internal core routers within the LANL enterprise network and are derived from router netflow records. The two data sets include many of the same computers but are not fully inclusive; the network data set includes many non-Windows computers and other network devices.

Identifying values within the data sets have been de-identified (anonymised) to protect the security of LANL's operational IT environment and the privacy of individual users. The de-identified values match across both the host and network data allowing the two data elements to be used together for analysis and research. In some cases, the values were not de-identified, including well-known network ports, system-level usernames (not associated to people) and core enterprise hosts. In addition, a small set of hosts, users and processes were combined where they represented well-known, redundant entities. This consolidation was done for both normalisation and security purposes.

In order to transform the data into a format that is useful for researchers who are not domain experts, a significant effort was made to normalise the data while minimising the artefacts that such normalisation might introduce.

1.1. *Related public data sets*

A number of public, cyber-security relevant data sets currently are referenced in the literature (Glasser and Lindauer, 2013; Ma *et al.*, 2009) or

[a] The network flow data are only 89 days due to missing data on the first day.

are available online.[b] Some of these represent data collected from operational environments, while others capture specific, pseudo real-world events (for example, cyber-security training exercises). Many data sets are synthetic and created using models intended to represent specific phenomenon of relevance; for example, the Carnegie Melon Software Engineering Institute provides several insider threat data sets that are entirely synthetic (Glasser and Lindauer, 2013). In addition, many of the data sets commonly seen within the research community are egregiously dated. The DARPA cyber-security data sets (Cyber-Systems and Technology Group, 1998) published in the 1990s are still regularly used, even though the systems, networks and attacks they represent have almost no relevance to modern computing environments.

Another issue is that many of the available data sets have restrictive access and constraints on how they may be used. For example, the U.S. Department of Homeland Security provides the Information Marketplace for Policy and Analysis of Cyber-risk and Trust (IMPACT,[c] which is intended to facilitate information sharing. However, the use of any of the data hosted by IMPACT requires registration and vetting prior to access. In addition, data owners may (and often do) place limitations on how and where the data may be used.

Finally, many of the existing data sets are not adequately characterised for potential researchers. It is important that researchers have a thorough understanding of the context, normalisation processes, idiosyncrasies and other aspects of the data. Ideally, researchers should have sufficiently detailed information to avoid making false assumptions and to reproduce similar data. The need for such detailed discussion around published data sets is a primary purpose of this chapter.

The remainder of this chapter is organised as follows: a description of the Network Flow Data is given in Section 2 followed by the Windows Host Log Data in Section 3. Finally, a discussion of potential research directions is given in Section 4.

[b] https://www.ll.mit.edu/ideval/data/, http://malware-traffic-analysis.net/, http://www.unb.ca/cic/research/datasets/index.html.

[c] https://www.dhs.gov/csd-impact.

2. Network Flow Data

The network flow data set included in this release is comprised of records describing communication events between devices connected to the LANL enterprise network. Each *flow* is an aggregate summary of a (possibly) bi-directional network communication between two network devices. The data are derived from Cisco NetFlow Version 9 (Claise, 2004) flow records exported by the core routers. As such, the records lack the payload-level data upon which most commercial intrusion detection systems are based. However, research has shown that flow-based techniques have a number of advantages and are successful at detecting a variety of malicious network behaviours (Sperotto *et al.*, 2010). Furthermore, these techniques tend to be more robust against the vagaries of attackers, because they are not searching for specific signatures (for example, byte patterns) and they are encryption-agnostic. Finally, in comparison to full-packet data, collection, analysis and archival storage of flow data at enterprise scales is straightforward and requires minimal infrastructure.

2.1. *Collection and transformation*

As mentioned previously, the raw data consisted of NetFlow V9 records that were exported from the core network routers to a centralised collection server. While V9 records can contain many different fields, only the following are considered: *StartTime*, *EndTime*, *SrcIP*, *DstIP*, *Protocol*, *SrcPort*, *DstPort*, *Packets* and *Bytes*. The specifics of the hardware and flow export protocol are largely irrelevant, as these fields are common to all network flow formats of which the authors are aware.

This data can be quite challenging to model without a thorough understanding of its various idiosyncrasies. The following paragraphs discuss two of the most relevant issues with respect to modelling. For a comprehensive overview of these issues, among others, readers can refer to Hofstede *et al.* (2014).

Firstly, note that these flow records are uni-directional (*uniflows*): each record describes a stream of packets sent from one network device (*SrcIP*) to another (*DstIP*). Hence, an established TCP connection — bi-directional by definition — between two network devices, *A* and *B*, results in two flow records: one from *A* to *B* and another from *B* to *A*. It follows that there is no relationship between the direction of a flow and the initiator of a bi-directional connection (i.e., it is not known whether *A* or *B* connected first). This is the case for most netflow implementations as bi-directional flow (*biflow*) protocols such as Trammell and Boschi (2008) have yet to gain widespread adoption. Clearly, this presents a challenge for detection of attack behaviours, such as lateral movement, where directionality is of primary concern.

Secondly, significant duplication can occur due to flows encountering multiple netflow sensors in transit to their destination. Routers can be configured to track flows on ingress and egress, and, in more complex network topologies, a single flow can traverse multiple routers. More recently, the introduction of netflow-enabled switches and dedicated netflow appliances has exacerbated the issue. Ultimately, a single flow can result in many distinct flow records. To add further complexity, the flow records are not necessarily *exact* duplicates and their arrival times can vary considerably; these inconsistencies occur for many reasons, the particulars of which are too complex to discuss in this context.

In order to simplify the data for modelling, a transformation process known as *biflowing* or *stitching* was employed. This is a process intended to aggregate duplicates and marry the opposing uniflows of bi-directional connections into a single, *directed* biflow record (Table 1). Many approaches to this problem can be found in the literature (Barbosa, 2014; Berthier *et al.*, 2010; Minarik *et al.*, 2009; Nguyen *et al.*, 2017), all of them imperfect. A straightforward approach was used that relies on simple port heuristics to

Table 1: Bi-directional flow data.

Field Name	Description
Time	The start time of the event in epoch time format.
Duration	The duration of the event in seconds.
SrcDevice	The device that likely initiated the event.
DstDevice	The receiving device.
Protocol	The protocol number.
SrcPort	The port used by the *SrcDevice*.
DstPort	The port used by the *DstDevice*.
SrcPackets	The number of packets the *SrcDevice* sent during the event.
DstPackets	The number of packets the *DstDevice* sent during the event.
SrcBytes	The number of bytes the *SrcDevice* sent during the event.
DstBytes	The number of bytes the *DstDevice* sent during the event.

decide direction. These heuristics are based on the assumption that *SrcPort*s are generally *ephemeral* (i.e., they are selected from a predefined, high range by the operating system), while *DstPort*s tend to have lower numbers that correspond to established, shared network services and will therefore be observed more frequently than ephemeral ports. The heuristics are given below in order of precedence.

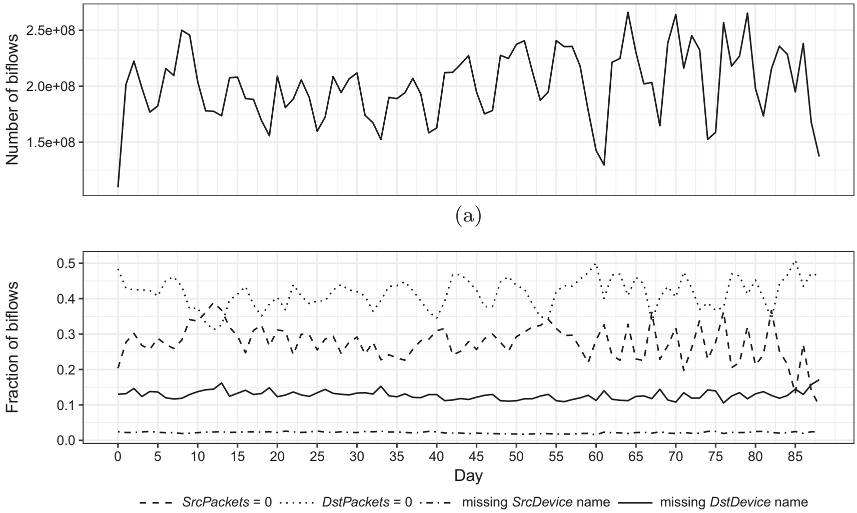
- Destination ports are less than 1024 and source ports are not.
- The top 90 most frequently observed ports are destination ports.
- The smaller of the two ports is the destination port.

Each uniflow was transformed into a biflow by renaming the *Packets* and *Bytes* fields to *SrcPackets* and *SrcBytes*, respectively. *DstPackets* and *DstBytes* fields were added with initial values of zero. Next, the port heuristics were considered and, if any were violated or ambiguous, the *Src* and *Dst* attributes were swapped, effectively reversing the direction. Finally, the *5-tuple* was extracted from each record and used as the key in a lookup table.

SrcIP, *DstIP*, *SrcPort*, *DstPort*, *Protocol*

If a match was found, the flows were aggregated by keeping the minimum *StartTime*, maximum *EndTime* and summing the other attributes. If no match was found, the flow was simply added to the table. This process was performed in a streaming fashion on all of the records in the order in which they were received by the collector. Flows were periodically evicted from the lookup table after 30 minutes of inactivity (i.e., failing to match with any incoming flows). Flows that remained active for long periods of time were reported approximately every 3 hours, but were *not* evicted from the table until inactive.

While biflowing the data mitigates the problems posed by duplicates and ambiguous directionality, it does not address another significant obstacle: the lack of stable identifiers upon which to build models. In some cases, IP addresses are transient (e.g., Dynamic Host Configuration Protocol (DHCP), Virtual Private Network (VPN)). In other cases, devices have multiple IP addresses (e.g., multihoming) or one IP address is shared by multiple devices (e.g., load-balancing, NAT). Whatever the case may be, modelling the behaviour of IP addresses on a typical network is clearly



(a)

(b)

Fig. 1: (a) Daily count of biflows by end time. (b) Fraction of biflows where *SrcPackets* = 0, *DstPackets* = 0, *SrcDevice* FQDN-mapping failed and *DstDevice* FQDN-mapping failed.

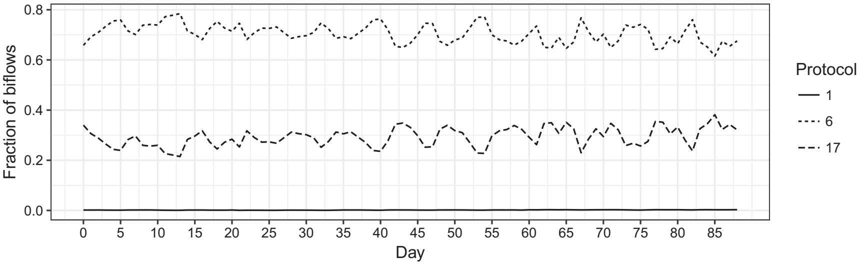


Fig. 2: Daily proportions of each *Protocol*.

error prone. Instead, one should endeavour to map IP addresses to more stable identifiers such as media access control (MAC) addresses or fully-qualified domain names (FQDN), interchangeably referred to as hostnames throughout the rest of the chapter. As with directionality, there is no perfect solution to this problem. The most appropriate identifier will depend greatly on the configuration of the target network, as well as the availability of auxiliary data sources from which a mapping can be constructed. An ideal solution will likely involve some combination of supplementary network data (e.g., Domain Name Service (DNS) logs, DNS zone transfers, DHCP logs, VPN logs, NAC logs), business rules and considerable trial and error.

For this data release, a combination of DNS and DHCP logs was used to construct a mapping of IP addresses to FQDNs over time. The IP addresses in each biflow were then replaced with their corresponding FQDNs at the time of the flow. Where a given IP address and timestamp mapped to multiple FQDNs, business rules were incorporated to give preference to the least-ephemeral name. IP addresses that failed to map to any FQDN were left as is. The resulting mix of names and IP addresses correspond to the *SrcDevice* and *DstDevice* fields in the final data.

Finally, the data were de-identified by mapping *SrcDevice*, *DstDevice*, *SrcPort* and *DstPort* to random identifiers. In the event that the IP-to-FQDN mapping failed, the random identifier was prepended with "IP". Well-known ports were not de-identified. Records with protocol numbers other than 6 (TCP), 17 (UDP) and (1) ICMP were removed entirely. The output from this process is provided in CSV format, one record per line, with fields in the order shown in Table 1.

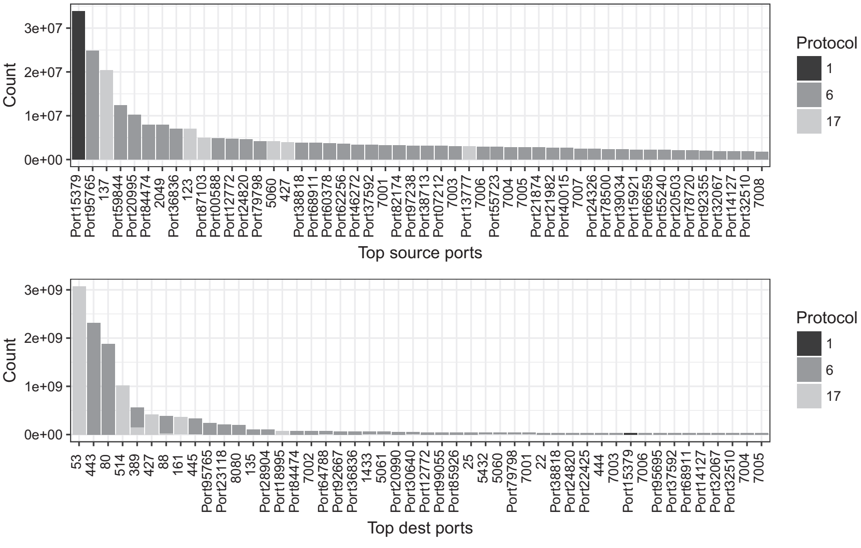


Fig. 3: Histogram of the top 50 *SrcPort*s and *DstPort*s.

2.2. *Data quality*

Several figures have been provided in order to assess the quality of the network flow data set. The top plot in Figure 1, which shows the number of biflows over time, demonstrates the periodicity that one would expect for data whose volume is driven by the comings and goings of employees during a typical 5-day workweek.

The bottom plot of Figure 1 is intended to measure the success rate of the biflowing and IP-to-FQDN mapping processes. TCP biflows where either *SrcPackets* or *DstPackets* is zero suggests a failure to find matching uniflows for both directions of the exchange. Fifty Seven percent of TCP and approximately 70% of all biflows fall within this category. This can largely be attributed to LANL's netflow sensor infrastructure, which has been specifically configured to export only one direction on many routes. In addition, some devices — namely vulnerability scanners and the like — attempt to connect to all possible IP addresses within a range; this results in a significant number of uniflows for which no response is possible. Likely for the same reason, IP-to-FQDN mapping failed for significantly more *DstDevice*s than *SrcDevice*s.

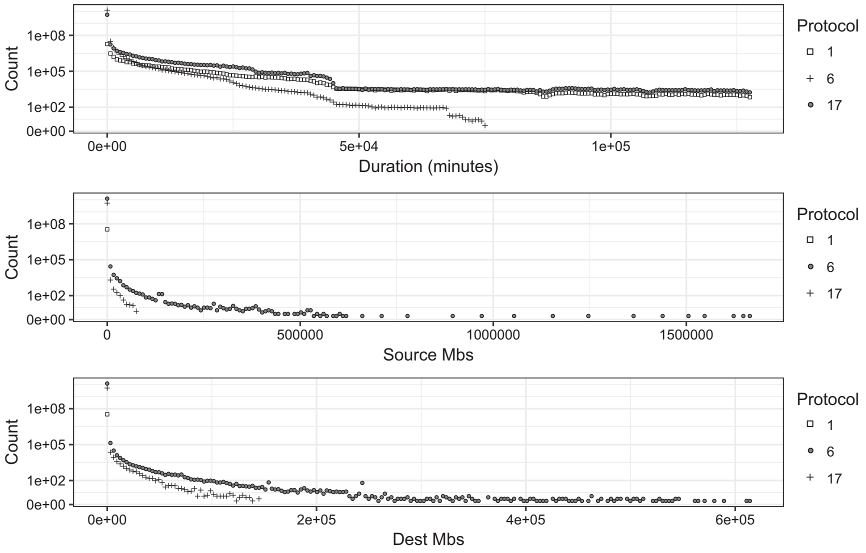


Fig. 4: Distribution of *Duration*, *SrcByte*s and *DstByte*s.

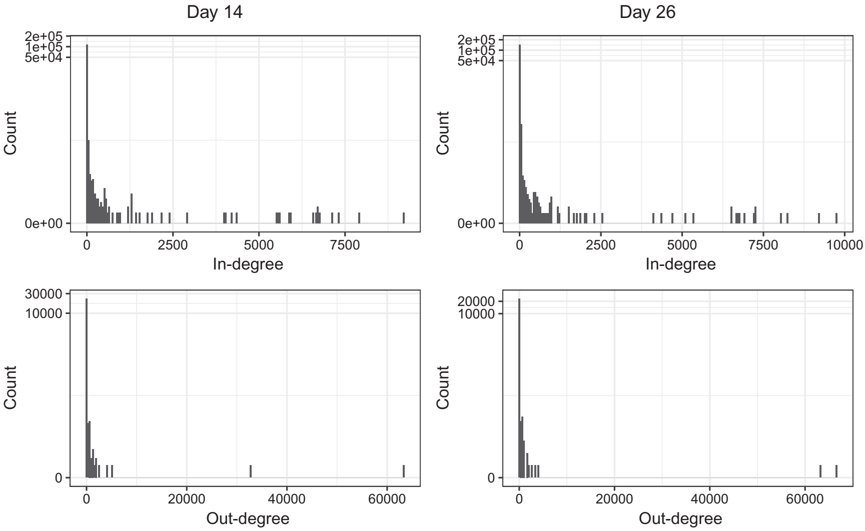


Fig. 5: In-degree and out-degree distribution for two randomly-selected days.

Figure 2 shows the daily proportion of biflows corresponding to each *Protocol*. Figure 3 contains two histograms of the top *SrcPort*s and *DstPort*s respectively. Note the non-uniformity in the *SrcPort* histogram; this illustrates either a consistent failure of the biflowing process to choose the appropriate direction or the presence of protocols that use non-ephemeral source ports. For example, the network time protocol (NTP) uses port 123 for both the source and destination ports per the specification.

Figure 4 shows the distribution of *Duration*, *SrcByte*s and *DstByte*s per *Protocol*. Of particular interest is the presence of many long-lived UDP and ICMP biflows in the data. This indicates frequent, persistent UDP and ICMP traffic sharing the same *5-tuple* and is an unfortunate side effect of not limiting the biflow transformation to TCP uniflows. Finally, Figure 5 shows exemplar in-degree and out-degree distributions for two randomly-selected days.

3. Windows Host Log Data

As remote attackers and malicious insiders increasingly use encryption, network-only detection mechanisms are becoming less effective, particularly those that require the inspection of payload data within the network traffic. As a result, cyber-defenders now rely heavily on endpoint agents and host event logs to detect and investigate incidents. Host event logs capture nuanced details for a wide range of activities; however, given the vast number of logged events and their specificity to an individual host, human analysts struggle to discover the few useful log entries amid the huge number of innocuous entries. Statistical analytics for host event data are in their infancy. Advanced analytical capabilities on this host data, including computer and user profiling, which move beyond signature-based methods, will increase network awareness and detection of advanced cyber-threats.

The host event data set is a subset of host event logs collected from all computers running the Microsoft Windows operating system on LANL's enterprise network. The host logs were collected with windows logging service (WLS), which is a Windows service that forwards event logs, along with administrator-defined contextual data to a set of collection servers.[d] The released data are in JSON format in order to preserve the structure of the original events, unlike the two previously released data sets based on this log source (Kent, 2014, 2016). The events from the host logs included in the data set are all related to authentication and process activity on each machine.

Table 2 contains the subset of *EventID*s included from the event logs in the released data set and a brief description of each; a more detailed description is available online.[e] Figure 6 shows the percentage of *EventID*s contained in the logs, as well as the *LogonType*s for *EventID*s 4624, 4625 and 4634.

Each record in the data set will have some of the event attributes listed in Appendix A and Table B.1 specifies which *EventID*s have each attribute. Note that not all events with a given *EventID* share the same set of attributes. If an expected attribute was missing from the original host log record, then the attribute was not included in the corresponding record in the de-identified data set.

All records will contain the attributes *EventID*, *LogHost* and *Time*. *LogHost* indicates the network host where the record was logged. For

[d] http://honeywell.com/sites/aero-kcp/SiteCollectionDocuments/WindowsLoggingServiceSummary.pdf.

[e] https://www.ultimatewindowssecurity.com/securitylog/encyclopedia/default.aspx.

Table 2: Host log *EventID*s.

EventID	Description
Authentication events	
4768	Kerberos authentication ticket was requested (TGT)
4769	Kerberos service ticket was requested (TGS)
4770	Kerberos service ticket was renewed
4774	An account was mapped for logon
4776	Domain controller attempted to validate credentials
4624	An account successfully logged on, see Logon Types
4625	An account failed to logon, see Logon Types
4634	An account was logged off, see Logon Types
4647	User initiated logoff
4648	A logon was attempted using explicit credentials
4672	Special privileges assigned to a new logon
4800	The workstation was locked
4801	The workstation was unlocked
4802	The screensaver was invoked
4803	The screensaver was dismissed
Process events	
4688	Process start
4689	Process end
System events	
4608	Windows is starting up
4609	Windows is shutting down
1100	Event logging service has shut down (often recorded instead of *EventID* 4609)

*LogonType*s (*EventID*s: 4624, 4625 and 4634)

2 — Interactive	5 — Service	9 — New Credentials
3 — Network	7 — Unlock	10 — Remote Interactive
4 — Batch	8 — Network Clear Text	11 — Cached Interactive
12 — Cached Remote-Interactive	0 — Used only by the system account	

directed authentication events, this attribute will always correspond to the computer to which the user is authenticating, and the source computer will be given by *Source*. For the user associated with the record, if the *UserName* ends in $ then it will correspond to the *computer account* for the specified computer. These computer accounts are host-specific accounts within the

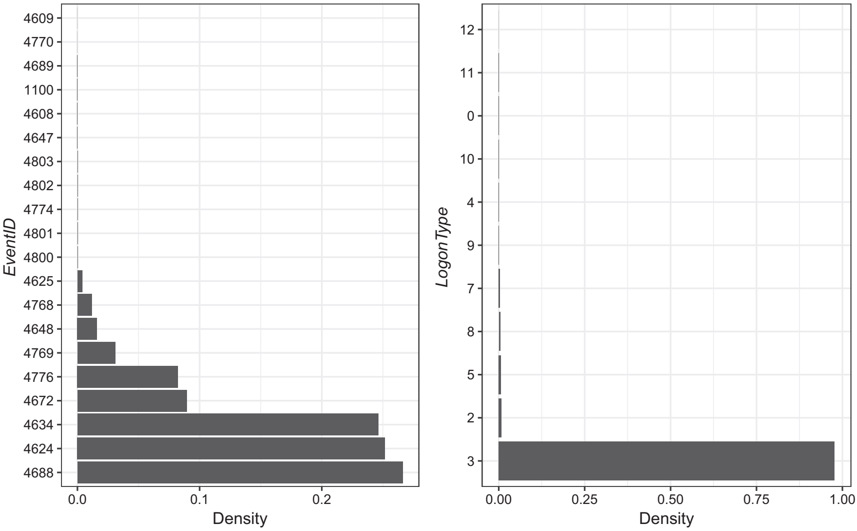


Fig. 6: Histogram of the *EventID*s and *LogonType*s.

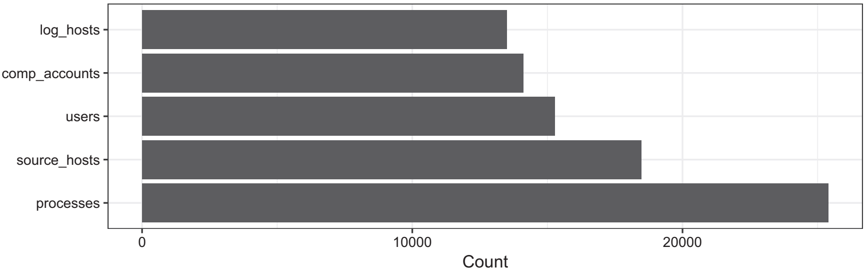


Fig. 7: Histogram of unique processes, usernames, log hosts (*LogHost*), source hosts (*Source*) and computer accounts for the whole time period.

Microsoft Active Directory domain that allow the computer to authenticate as a unique entity within the network. Figure 7 shows the count of unique processes, log hosts (*LogHost*), source hosts (*Source*), computer accounts (*UserName* ending in $) and users (*UserName* not ending in $) for the 90-day period. Note that the set of source hosts includes devices running non-Windows operating systems, hence there are more source hosts than log hosts. Figure 8 shows the number of wls records on a per-day basis, showing the diurnal patterns that one would expect and good collection

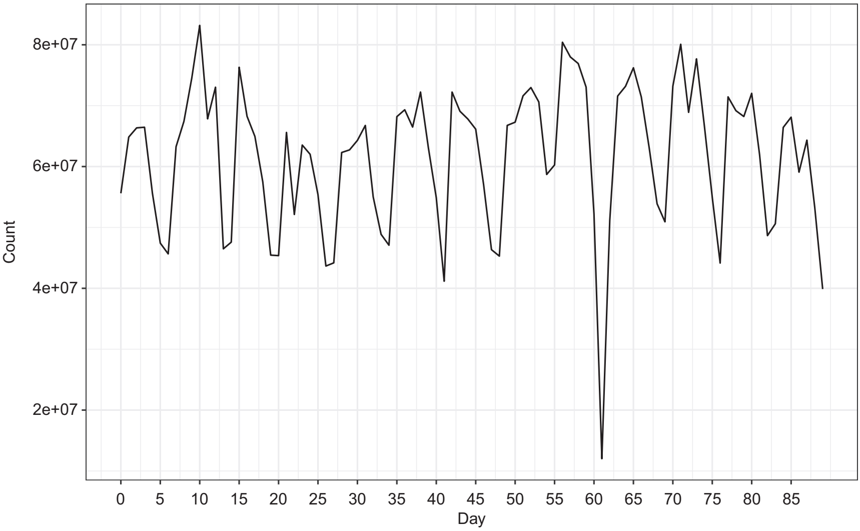


Fig. 8: Daily count of host log records.

throughout the 90 days minus a noticeable drop on day 61 similar to that of the netflow data set, Figure 1.

Requests to the Kerberos ticket granting service (TGS) (*EventID* 4769) correspond to a user requesting Kerberos authentication credentials from the Active Directory domain to a service or account name on a network computer. Hence, the *LogHost* attribute should always be an Active Directory machine and the service or account name the user is requesting access to will be given by *ServiceName*. The *ServiceName* often corresponds to a computer account on the target computer. Because this event only grants a credential, a subsequent network logon event (*EventID* 4624–*LogonType* 3) to the computer indicated by *ServiceName* is common. This differs from the previous data release (Kent, 2016), in which TGS events were assumed to be directed authentication events from the user's machine to the computer indicated by *ServiceName*, ignoring the Kerberos intermediary.

When de-identifying the process events, only the base process name was de-identified and the extension was left as is. Further, the parent process names (*ParentProcessName*) do not have file extensions unlike the child process names (*ProcessName*); this is a direct artefact of how the process information is logged within WLS. The missing extension can be obtained by using the *ParentProcessID* to identify the parent process start event.

Finally, many events include the *DomainName* attribute that indicates what Active Directory domain the event is associated with. The domain, combined with the *UserName*, should be considered a unique account identity. For example, user *u1* with domain *d1* is not necessarily user *u1* in domain *d2*. In addition, the domain may actually be a hostname, indicating the event does not involve a user or account associated with an Active Directory domain, but is instead a local account. Again, these accounts should be considered unique to the host indicated within the *DomainName* attribute. For example, the Administrator account on host *c1* likely does not have a relationship to the Administrator account on *c2* or the Administrator account in domain *d1*. The LANL data sets have a single primary domain, with a number of much smaller, secondary domains, and most computers have a small set of local accounts.

3.1. *Data parsing considerations*

While host logs can be an extremely valuable data resource for cybersecurity research, the formatting and content of the logs can vary drastically between enterprises depending upon the audit policy and technologies used to collect and forward the logs to a centralised server. Hence, parsing the data and extracting the relevant attributes is an important first step in analysing these data; see also Kent (2016).

Even though WLS provides more content and normalisation around the raw Windows logs, some challenges were still faced to provide the de-identified data.

Firstly, the semantics of attribute names are not necessarily the same for different *EventID*s and the attribute names themselves may differ according to what tool is being used to collect and forward the logs. For example, with WLS the *UserName* for *EventID* 4774 is *MappedName*, for *EventID* 4778 and 4779 it is *AccountName* and for most other events it is *TargetUserName*. When parsing the data, these names were all standardised to *UserName*.

As with the network flow data, an extremely important task is mapping IP addresses to FQDNs. Further, unlike netflow, each record may contain both IP addresses and hostnames. The machine where the event is recorded

(*LogHost* for the de-identified data) is provided as a hostname, whereas the *Source* computer for network logons is often given as an IP address.

Finally, both usernames and process names were standardised. In some records, usernames appear with the domain name or additional characters. These discrepancies were removed from the released data in order to ensure all usernames were in canonical form. In addition, some usernames, such as “Anonymous”, “Local Service” and “Network Service”, do not map to a computer or user account. For some analyses, one may want to remove these events. In the de-identified data these commonly-seen usernames were not anonymised. For the process names, dates, version numbers, operating systems and hexadecimal strings were removed where possible so that processes run on different operating systems or with different versions would map to the same process name. For example, *flashplayerplugin_20_0_0_286.exe* would be mapped to *flashplayer-plugin_VERSION.exe*.

4. Research Directions

Anomaly detection for the defensive cyber-domain is a major yet evolving research area, with much work still to be done in characterising and finding anomalies within complex cyber-data sets. Finding viable attack indicators and per computer, user and computer-to-computer models that enable anomaly detection and fingerprinting are all interesting and important research opportunities.

Although research on anomaly detection for cyber-defence spans more than two decades, operational tools are still almost exclusively rule- or signature-based. Two reasons that statistical methods have not been more widely adopted in practice are a high false-positive rate and un-interpretable alerts. Analysts are inundated with a large number of alerts and triaging them takes significant time and resources; this results in low tolerance for false alarms and alerts that provide no contextual information to guide investigation. Signature-based systems can be finely tuned to reduce false positives as they rely on very specific peculiarities that have been previously identified and documented as indicative of a cyber-attack. Further, they are interpretable as they refer to specific patterns within the data, such as weird domains, network protocols or process names.

However, despite their inherent challenges, anomaly detection methods have the advantage of being able to detect new variants of cyber-attacks and are able to keep pace with the rapidly changing cyber-attack landscape by dynamically learning patterns for normal behaviour and detecting deviations. Further, with the increasing level of encrypted network traffic, the importance of this research and the use of these methods can not be understated. Research into ways to reduce false-positives and providing interpretable anomalies will have significant impact in furthering the use of anomaly detection systems. In fact, providing interpretable anomalies can help overcome the false-positive issue as interpretability leads to quickly identifying alerts that are false positives in the same way it would enable understanding true positives. Research approaches to tackle these problems could include combining different data sets and signals, borrowing strength across entities that are similar by incorporating peer-based behaviour, community detection approaches and ways to provide meaningful context surrounding alerts to human analysts.

When using the host log data set for research, some notable characteristics of these data that need to be considered, especially if looking at the events as a time series, is periodicity and significant correlations between arrivals of different event types. This can be seen clearly in Figure 9, which shows the event times for various *EventID*s for User205265. Periodicity in the data is often an artefact of the computer regularly renewing credentials. This explains why *EventID* 4624–*LogonType* 3 (network logon) constitutes such a significant portion of the events as seen in Figure 6. For a given entity, extrapolating higher-level, interpretable actions from the sequence of low-level events would improve modelling efforts, understanding of these data, and would itself be very useful for security analysts. See Heard *et al.* (2014) and Price-Williams *et al.* (2017) for relevant research in this area.

Another area for research with the host logs is exploring the records related to process starts and stops in detail, in particular looking at process trees. To date, little has been done in this area. Computer systems operate hierarchically; an initial root process starts many other processes, which in turn start and run descendants. A process tree is the dynamic structure that results. In theory, any process can be traced, through its ancestors, to the root process. Unusual or atypical events in process trees could indicate potential cyber-security anomalies.

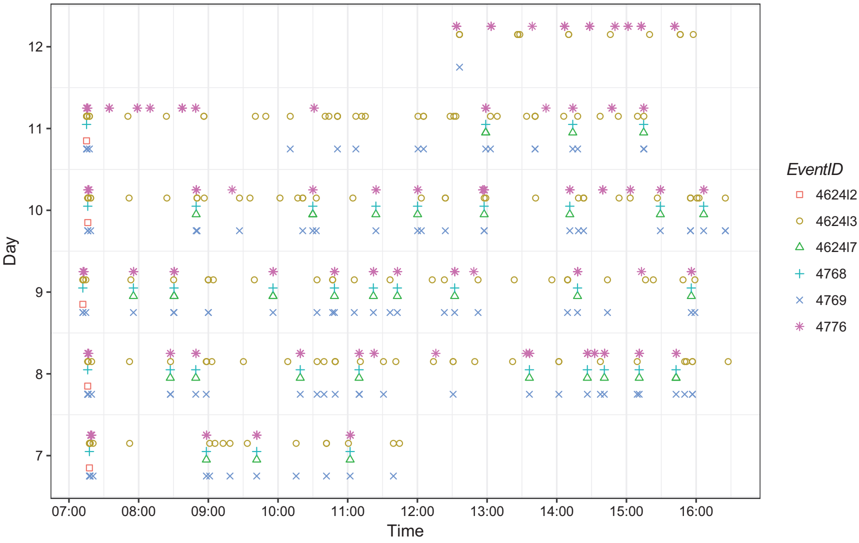


Fig. 9: Event times for User205265. 4624l2 corresponds to *EventID* 4624–*LogonType* 2.

Moving beyond anomaly detection, there are other important research directions for which these data could prove useful. For example, preliminary work has been done using similar data to model network segmentation and associated risk (Pope *et al.*, 2017). Using the data to build new, potential network topologies in order to reduce risk and improve security posture are viable directions. Another potential research problem is to quantify and understand data loss within cyber-data sets. The collection and normalisation processes in place for these data can result in information loss and understanding this data loss is an open problem both in general and specific to each element of the data. As most of the data elements represent people and their actions on computers, research on organisational and social behaviour is also viable using these data.

5. Conclusion

Operational cyber-security data sets are paramount to ensuring valuable and productive research continues to improve the state of cyber-defence. The network flow and host log event data discussed in this chapter are intended to enable such research as well as to provide an example for other potential data set providers. In particular, while there is a considerable amount of relevant work on network data, relatively little attention has been given to host log data in the literature. Host log data are becoming increasingly relevant as endpoint security tools gain popularity within the cyber-security ecosystem. It is important that researchers embrace both the opportunity and challenge that they present. Finally, even less consideration has been given to meaningful analyses that combine these and other data sets. This paradigm shift towards a holistic approach to cyber-security defence is critical to advancing the state of the art.

Acknowledgement

This work was supported by the US Department of Energy through the Los Alamos National Laboratory. Los Alamos National Laboratory is operated by Los Alamos National Security, LLC, for the National Nuclear Security Administration of U.S. Department of Energy (Contract DEAC52-06NA25396). The United States Government retains and the publisher, by accepting this work for publication, acknowledges that the United States Government retains a nonexclusive, paid-up, irrevocable, world-wide license to publish or reproduce this work, or allow others to do so for United States Government purposes.

Appendix A. Host Log Fields

- *Time*: The epoch time of the event in seconds.
- *EventID*: Four digit integer corresponding to the event id of the record.
- *LogHost*: The hostname of the computer that the event was recorded on. In the case of directed authentication events, the *LogHost* will correspond to the computer that the authentication event is terminating at (destination computer).
- *LogonType*: Integer corresponding to the type of logon, see Table 2.
- *LogonTypeDescription*: Description of the *LogonType*, see Table 2.
- *UserName*: The user account initiating the event. If the user ends in $, then it corresponds to a computer account for the specified computer.
- *DomainName*: Domain name of *UserName*.
- *LogonID*: A semi-unique (unique between current sessions and *LogHost*) number that identifies the logon session just initiated. Any events logged

subsequently during this logon session should report the same *LogonID* through to the logoff event.

- *SubjectUserName*: For authentication mapping events, the user account specified by this field is mapping to the user account in *UserName*.
- *SubjectDomainName*: Domain name of *SubjectUserName*.
- *SubjectLogonID*: See *LogonID*.
- *Status*: Status of the authentication request. "0 × 0" means success otherwise failure; failure codes for the appropriate *EventID* are available online.[f]
- *Source*: For authentication events, this will correspond to the the computer where the authentication originated (source computer), if it is a local logon event then this will be the same as the *LogHost*.
- *ServiceName*: The account name of the computer or service the user is requesting the ticket for.
- *Destination*: This is the server the mapped credential is accessing. This may indicate the local computer when starting another process with new account credentials on a local computer.
- *AuthenticationPackage*: The type of authentication occurring including Negotiate, Kerberos, NTLM plus a few more.
- *FailureReason*: The reason for a failed logon.
- *ProcessName*: The process executable name, for authentication events this is the process that processed the authentication event. *ProcessNames* may include the file type extensions (i.e., exe).
- *ProcessID*: A semi-unique (unique between currently running processes AND *LogHost*) value that identifies the process. *ProcessID* allows you to correlate other events logged in association with the same process through to the process end.
- *ParentProcessName*: The process executable that started the new process. *ParentProcessNames* often do not have file extensions like *ProcessName* but can be compared by removing file extensions from the name.
- *ParentProcessID*: Identifies the exact process that started the new process. Look for a preceding event 4688 with a *ProcessID* that matches this *ParentProcessID*.

[f] https://www.ultimatewindowssecurity.com/securitylog/encyclopedia/default.aspx.

Appendix B

Table B.1: Event attributes.

*EventID*s	Attribute
All	*Time*
All	*EventID*
All	*LogHost*
4624, 4625, 4634	*LogonType*
4624, 4625, 4634	*LogonTypeDescription*
All except System Events	*UserName*
All except System Events	*DomainName*
All except 4768, 4769, 4770, 4774, 4776	*LogonID*
4624 (*LogonType* 9), 4648, 4774	*SubjectUserName*
4624 (*LogonType* 9), 4648, 4774	*SubjectDomainName*
4624 (*LogonType* 9), 4648	*SubjectLogonID*
4768, 4769, 4776	*Status*
4624, 4625, 4648, 4768, 4769, 4770, 4776	*Source*
4769, 4770	*ServiceName*
4648	*Destination*
4624, 4625, 4776	*AuthenticationPackage*
4625	*FailureReason*
4624, 4625, 4648, 4688, 4689	*ProcessName*
4624, 4625, 4648, 4688, 4689	*ProcessID*
4688	*ParentProcessName*
4688	*ParentProcessID*

References

Barbosa, R. R. R. (2014). *Anomaly Detection in SCADA Systems — A Network Based Approach*, Ph.D. thesis, Centre for Telematics and Information Technology, University of Twente, Netherlands.

Berthier, R., Cukier, M., Hiltunen, M., Kormann, D., Vesonder, G. and Sheleheda, D. (2010). Nfsight: Netflow-based network awareness tool, in *Proceedings of LISA10: 24th Large Installation System Administration Conference*, p. 119.

Claise, B. (2004). Cisco systems NetFlow services export, Version 9, RFC 3954, Internet Engineering Task Force.

Cyber Systems and Technology Group (1998). DARPA intrusion detection data sets, URL: https://www.ll.mit.edu/ideval/data/.

Glasser, J. and Lindauer, B. (2013). Bridging the gap: A pragmatic approach to generating insider threat data, *2012 IEEE Symposium on Security and Privacy Workshops*, pp. 98–104.

Heard, N., Rubin-Delanchy, P. and Lawson, D. J. (2014). Filtering automated polling traffic in computer network flow data, in *2014 IEEE Joint Intelligence and Security Informatics Conference*, pp. 268–271.

Hofstede, R., Čeleda, P., Trammell, B., Drago, I., Sadre, R., Sperotto, A. and Pras, A. (2014). Flow monitoring explained: From packet capture to data analysis with NetFlow and IPFIX, *IEEE Communications Surveys & Tutorials* **16**, 4, pp. 2037–2064.

Kent, A. D. (2014). User-computer authentication associations in time, Los Alamos National Laboratory, doi:10.11578/1160076.

Kent, A. D. (2016). Cyber security data sources for dynamic network research, in N. Adams and N. Heard. eds., *Dynamic Networks and Cyber-Security*, Vol. 1, p. 37, World Scientific, UK.

Ma, J., Saul, L. K., Savage, S. and Voelker, G. M. (2009). Identifying suspicious URLs: An application of large-scale online learning, in *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 681–688.

Minarik, P., Vykopal, J. and Krmicek, V. (2009). Improving host profiling with bidirectional flows, in *International Conference on Computational Science and Engineering, 2009. CSE'09.*, Vol. 3, pp. 231–237.

Nguyen, K. V., Tyagi, N. K. and Lau, R. M. (2017). Flow de-duplication for network monitoring, US Patent 9,548,908.

Pope, A., Tauritz, D. and Kent, A. (2017). Evolving bipartite authentication graph partitions, *IEEE Transactions on Dependable and Secure Computing*, **99**, pp. 1–1.

Price-Williams, M., Heard, N. and Turcotte, M. (2017). Detecting periodic subsequences in cyber security data, in *IEEE European Intelligence and Security Informatics Conference (EISIC2017)*, pp. 84–90.

Sperotto, A., Schaffrath, G., Sadre, R., Morariu, C., Pras, A. and Stiller, B. (2010). An overview of IP flow-based intrusion detection, *IEEE Communications Surveys and Tutorials* **12**, 3, pp. 343–356.

Trammell, B. and Boschi, E. (2008). Bidirectional flow export using IP Flow Information Export (IPFIX), RFC 5103, Internet Engineering Task Force.