Cyber security data sources for dynamic network research

Alexander D. Kent

Los Alamos National Laboratory, Los Alamos, NM, 87545, U.S.A
alex@lanl.gov

The importance of using real-world data to enable and validate dynamic network research for the purposes of cyber security cannot be understated. Unfortunately, the use of real-world data in most cyber security research either for motivation or validation is rare. The majority of useful information technology (IT) data sources were intended for operational monitoring and not for cyber security purposes. Access for research purposes is even more problematic. From a dynamic network point of view, they are often lacking in comprehensive coverage, difficult to integrate across disparate sources, likely have significant noise in various forms, and generally lack any form of normalisation. Nonetheless, there is a rich and abundant potential for useful cyber security data sets. We will discuss a variety of data source opportunities, usefulness and value, along with potential problems. In addition, we will provide an overview of a newly released, comprehensive, real-world cyber security data set that is now openly available to the research community.

1. Introduction

Given the fact that cyber security is security within the computing realm, a world built of data, the idea that these data sources are difficult to access and use for cyber security analytical purposes seems oxymoronic. Nonetheless, in real-world environments, comprehensive data for intended cyber security analysis and research analysis is scarce at best — in many cases it is completely non-existent. This scarcity exists for several reasons. The primary reason is that most IT data sources are not intended or collected for cyber security purposes. For example, the majority of computer

and computer network event logs were intended for operational awareness monitoring. Additionally, in most cases, these data sources are formatted for human inspection and not for easy automated parsing of the event attributes that enable data analytics.

The elements of data analysis techniques for enterprise cyber defense include the discovery of data sources, assessing their likely value, developing code for parsing the relevant event attributes into normalised forms and finally transforming and combining the normalised events into actionable analysis. Each of these steps require research and development efforts of varying significance. Unfortunately, the collection and curation aspects of cyber security data analytics are significantly under-represented in current research.

In this chapter, we primarily consider and discuss two publicly available data sets: a nine-month data set of simplified authentication events[1] and a new, more comprehensive enterprise cyber security data set spanning 58 days released in conjunction with this chapter, as discussed in Section 4. Throughout the chapter we will refer to the nine-month data set as "α data set" and the new, 58-day data set as "β data set".

2. Enterprise Cyber Security Data Sources

Sufficiently sized event data sets are necessary to ensure ample sample sizes for analytical approaches and also help avoid short-term, real-world data anomalies stemming from unintended data loss and other transient events. Data relevant to cyber security can come from a large variety of sources. Valuable internal sources of data we have focused on include:

(1) Event logs from Windows desktops, servers and Active Directory servers. An example event entry is shown in Figure 1. This type of

Apr 1 12:49:09 aserver.domain 1 2013-04-01T12:48:36-07:00 4769 Microsoft-Windows-Security-Auditing U1@domain Success Audit Kerberos Service Ticket Operations Account Information: Account Name: U1@domain Account Domain: domain Service Information: Service Name: C2.domain Client Address: 192.168.0.1 Client Port: 62201 Additional Information: Ticket Options: 0x40810000 Ticket Encryption Type: 0x12 Failure Code: 0x0

Fig. 1. An example authentication event log entry showing computer 192.168.0.1 requesting a service ticket (TGS) for computer C2. This data looks similar from both the central Windows servers and the desktop computers.

events is configurable and can include everything from user authentication activity to process starts and stops to various system configuration change events. Collecting event log information requires configuration control and local agents to be installed on each computer that events are collected from.

(2) Event logs from all enterprise-wide Windows-based desktop and server computers. These are very similar in format to the domain controller logs (logging mechanisms and collection infrastructure are identical).

(3) Network flow event logs from central routers within the enterprise network.[2] These records indicate network connection events between computers in the network. Details include the time of the connection, duration of the connection, the amount of information moved between the computers and protocol information. These flow records are generally *simplex* in nature, meaning a single flow record represents only one direction of a connection between two computers; a second record would present the other direction. Network flow data is generally sampled from routers and thus often does not include data about all flows seen by the router. In addition, network connections that do not cross a router are not recorded.

(4) Domain name service (DNS) lookup records from internal, enterprise DNS servers. These event records indicate lookups of computer names and IP addresses within the enterprise network. These events can be used to augment network flow data and represent network connections between computers in the network. However, connection events using DNS lookups can be missing and noisy due some connections not using DNS (direct IP connection with no DNS lookup), local system and application DNS caching and lookups that do not result in a network connection.
(5) Web proxy log events for Internet-bound web surfing activity from nearly all desktop and server computers. An example event entry is shown in Figure 2.
(6) Antivirus log events from nearly all Windows-based desktop computers. An example event entry is shown in Figure 3.
(7) Cyber incident response (IR) tickets generated by IR analysts and automated processes leading to cyber intrusion investigations.

#Fields: date time time-taken c-ip cs(X-Forwarded-For) sc-status s-action sc-bytes cs-bytes cs-method cs-uri-scheme cs-host cs-ip cs-uri-stem c-uri-query cs-username cs-auth-group s-hierarchy r-ip rs(Content-Type) cs(User-Agent) cs(Referer) sc-filter-result sc-filter-category x-virus-id s-ip s-sitename r-port
2013-04-01 12:45:46 171 192.168.1.1 192.168.2.1 200 TCP_CLIENT_REFRESH 50428 987 GET http news.google.com 192.168.1.1 http://news.google.com/?output=rss - - DIRECT 74.125.225.98 application/xml;%20charset=UTF-8 "Apple-PubSub/65.28" - OBSERVED News/Media – 192.168.1.1 SG-HTTP-Service 80

Fig. 2. An example web proxy request log entry for an internal computer accessing the website http://news.google.com. This data is generated for all Internet web activity from an enterprise network using an Internet-bound web proxy.

Apr 1 12:31:59 SymantecServer srv-a: Virus found,Computer name: c1,Source: Real Time Scan,Risk name: Trojan.Webkit!html,Occurrences: 3,****SUMMARIZED DATA****,,Actual action: Deleted,Requested action: Left alone,Secondary action: Left alone,Event time: 2013-04-01 13:00:00,Inserted: 2013-04-01 13:00:00,End: 2013-04-01 13:59:59,Domain: domain.com,Group: My Company \Base\Workstation\MAC,Server: srv-a,User: ,Source computer: ,Source IP: 0.0.0.0,Disposition: Good,Download site: null,Web domain: null,Downloaded by: null,Prevalence: Reputation was not used in this detection.,Confidence: Reputation was not used in this detection.,URL Tracking Status: Off,,First Seen: Reputation was not used in this detection.,Sensitivity: Low,0,Application hash: ,Hash type: SHA1,Company name: ,Application name: ,Application version: ,Application type: -1,File size (bytes): 0

Fig. 3. An example antivirus log entry generated by individual Windows desktop computers throughout the enterprise network when malicious software is detected on the computer. These log events are recorded on a central system as part of Symantec's Enterprise Endpoint Protection (SEP) system.

The first four data sources are used as the basis for data set β to be discussed in Section 4. The latter three data sets (web proxy events, antivirus events and incident response records) have been used for various existing data analytics research. For example, these three data sets have been used together to examine risky web surfing behaviour and web activity in similar timeframes to known compromise events.[3,4] There are also a number of other, more traditional data sources such as intrusion detection and prevention systems that provide large-scale data analysis opportunities.[5]

These data sources and many others are collected centrally and used for a variety of operational purposes by IT personnel — though generally not explicitly for cyber security and almost never as aggregate analytics. The file and data formats used for logging operational IT events are extremely

diverse. Various time periods and sources of data drive distinct event log files. For example, in some data sources, individual log files represent one hour of contiguously collected data from a single data source (e.g. one web proxy server) while for other data sources a single log file represents 24 hours of contiguously collected data from multiple data sources (e.g. all Windows desktop computers).

We have found that a valuable method of ensuring availability for research purposes is to make a continuously updated, read-only copy of these operational data collections to a secondary data store that is used for research purposes.[a] This secondary data store also serves as the primary backup and disaster recovery system for the operational IT collection systems as required by various regulatory and security policies. It is important to note that this backup capability provides an *incentive* for the operational IT department in facilitating a copy of their data. Enabling this bidirectional value for both research and operational IT is critical in facilitating access to often elusive data for the research community.

2.1. *Data constraints*

Three fundamental considerations must be addressed in the collection of enterprise security data: privacy, practicality and protection.

There is obvious value in collecting cyber defense-relevant event data that reflects the behaviour of users and computers within an enterprise computing environment. Nonetheless, as is legally required and ethically appropriate, the privacy of individual users must be respected. For example, the interception and decryption of encrypted network traffic for the purposes of cyber defense is fraught with the undesirable potential to intercept legally protected information. Likewise, the content of email messages or similar communication formats are other potential data sets which should be avoided for privacy reasons whenever possible.[7]

In terms of practicality, data collection must not unduly impact the operation of IT systems. Users should not be inconvenienced. The current reality is that any data collection infrastructure must adapt to existing operational IT systems and not the other way around. Operational effort must be considered and minimised in most instances.

---

[a]The continuous copy is created using the `rsync` tool and protocol.[6]

Finally, data collected for cyber security purposes usually have security sensitivities and require significant security protections. These protections must extend from the collection mechanisms through the systems conducting analysis and research. One key consideration is how to implement these security protection requirements and still enable the publication of research results. To this end there are a variety of data anonymisation approaches within the research community.[8] The complexities of attribute anonymisation and de-identification are non-trivial with many trade-offs in terms of content preservation and security for the data originating organisation.[9] For the data sets we have released and associated analysis, we have considered the following guidelines to help facilitate data value while minimising security concerns:

- De-identify and/or remove IP addresses, hostnames and usernames.
- De-identify and/or remove fidelity of timestamps and/or time frames.
- Make it difficult to associate or correlate data within released data sets with other external data sets. More specifically, remove associations with the Internet whenever possible.[b]

- Avoid details of operational IT system specifics in terms of architecture and configuration.
- Avoid the publication of specific, operational cyber intrusion detection signatures, methods, procedures and thresholds. This does not include generalised algorithms and approaches but would include the operational threshold used within an algorithm to differentiate a false positive from a true positive.
- Generalise and aggregate analysis results over the data sets for publication, but avoid specific details relating to intrusions, potential intrusions and vulnerabilities.

3. Data Parsing and Normalisation

Computer and network event logging is common across many IT services and systems. However, the formatting, content and parsability varies

[b] While counterintuitive, this consideration is why we believe it is easier to release internal, enterprise data sets than organisation-specific data sets associated with the Internet.

greatly from one system to the next. Figures 1–3 show differing examples of log formats. Extracting the relevant attributes or fields from each of these log formats is an important step towards any data analysis activity. In some cases, this parsing is straightforward as seen with the web proxy data shown in Figure 2, where fields are well defined and well separated. In other cases, like the Windows event log data in Figure 1, fields have poor separation and require computationally expensive, iterative approaches to determining field values.

Once fields of relevance have been parsed from the raw data sources, field normalisation becomes necessary.

3.1. *Data field normalisation*

Cyber defense-relevant data sources have a variety of fields that require normalisation due to multiple ways of expressing and defining attributes of interest.

For example, time fields (sometimes called event timestamps) must be parsed and normalised based on different string representations and potentially different time zone representations (including daylight saving time differences). Normalising time fields to standardised UNIX epoch time[c]

enables easy subsequent parsing, time differentiation and other operations.

Another primary normalisation requirement involves computer identification. Unfortunately, computers on enterprise networks are often identified by multiple unique (or semi-unique) identifiers. In some cases they are identified as a hostname with or without a domain suffix (for example, C1 or C1.domain). In addition, computers are often also identified as an IP address on the network (e.g. 192.168.0.1). As seen in Figure 1, event log entries can include both IP addresses and hostnames within the same entry. These must be normalised to a single identification type to be used effectively. Computers may also be identified as a network media access control (MAC) address in enterprise environments that use dynamic addressing.[10]

There are other relevant fields that often also have multiple unique identifiers. For example, in our source data, two different unique usernames

[c]Unix epoch time is defined as the number of seconds since 1 January 1970 in the Coordinated Universal Time (UTC) timezone.

are used by various enterprise computer systems. In some log entries, a user may be identified as `user1` while in other entries from other systems the user may be identified as an employee number such as `123456`. Other instances may likely exist in many enterprise IT environments. Normalisation between these various unique identifiers to a single identifier is an important aspect of parsing and preparing relevant data sets for analysis. Within both α and β data sets, we normalise computers to their IP addresses and all users to a single employee number identifier.

3.2. *Historical field normalisation*

When performing the normalisation mapping on historical data sets, it is also important to ensure that the mapping process uses mapping data that is relevant to the time frame of the specific event being normalised. For example, a computer named C1 may have an IP address of 192.168.0.1 today but last month the same computer with the name C1 may have had a different IP address of 192.168.99.1. Thus, the historical mapping data becomes an important cyber security data source that must be collected and archived in similar ways to the primary data sets. To this end, we have developed a

system that takes daily snapshots of all enterprise-wide host to IP address mappings (DNS data) and user name mappings and archives this data as another important research data set. This data is used by the normalisation processes to ensure a time-accurate mapping for both computers and users.

3.3. *Map-reduce parallel data processing*

The quantity of data available and complexity of data parsing and normalisation, particularly for Windows event data, requires significant processing time. For example, parsing and normalising one year of Windows event log data can take over 26 weeks of continuous, serial processing time. However, this processing is generally done on a per-entry basis and thus inherently parallelisable.

A naïve parallelisation approach involves the simple use of multiple processors on a single system. Normalisation is sped up by a factor associated with the number of processors within the parallel system. Unfortunately, due to the significant parsing complexity of the Windows event log data, more substantial parallelisation is required.

To enable distributed processing across a cluster of many computers, a Map-Reduce approach[11] using FileMap[12] is employed. FileMap enables the use of a generalised map function to parse all data across a compute cluster and then use a final reduce function to output the normalised data in user- and time-sorted order.

FileMap was successfully used to parse and normalise a year's worth of Windows event logs from both the central servers and all desktops for a previously used data set[13] and the α data set.[1] The raw input involved 33.9 B verbose log messages (1.4 TB compressed). The final, normalised output was about 4.5 B events (20 GB compressed). The work was performed by a 320 CPU cluster, 1 TB of aggregate memory, 5 GB/s of aggregate bandwidth and 283 TB of aggregate disk space using 2001 distributed map processes in 10.4 hours. This is a 423-fold increase over what traditional serial processing would have required. Figure 4 shows the performance statistics including computation, idle and reduce times of the cluster's individual systems while processing the Windows event log data.

FileMap and similar Map-Reduce systems are valuable and relevant to the parsing, normalisation and analysis needs of large-scale enterprise cyber defense. While improvements in how relevant cyber data is recorded

(logged) and stored to facilitate easier parsing and normalisation is desirable, the reality of most enterprise IT data sources is that it will continue to require significant effort to clean.

4. A Comprehensive, Real-World Cyber Security Data Set

In an attempt to foster increased cyber security research that is focused on real-world problems and data, we have released a new, substantial cyber security data set; referred to as the β data set throughout this chapter. It is freely available for download and use at `http://csr.lanl.gov/data/cyber1`.[14] This β data set is intended to be well-suited towards the dynamic graph research community but should also have broad applicability across many research areas. We believe there is opportunity for significant characterisation and dynamic modelling across this new data that will positively impact the cyber security research community. The β data set, while shorter in time span, substantially expands on the previously released α data set that only provides a time-ordered list of authentication

Fig. 4. FileMap run statistics for processing one year of Windows event log data. The run took 10.4 hours using FileMap on 320 CPU cluster, but would have required over 26 weeks of processing if done as a traditional serial process. The circled area shows that the final merge sort into a single, time-ordered file (the *reduce* event) accounts for roughly a third of the 10.4 hours. It can also be observed that nodes 8, 11 and 12 finished early since this was a computationally complex job and these nodes have relatively little storage per CPU compared to the other nodes (1–2 TB drives compared to 4 TB drives elsewhere within the FileMap cluster). Because of the smaller disks, the nodes spent more time transferring in data for processing in comparison to other nodes. Similarly the g2 and g3 nodes are 48-way machines that have more CPU than disk.

associations between users and computers.[1] Nonetheless, the α data set was successfully used by Hagberg *et al.* as the basis to explore user authentication random graph models.[15]

The β data set represents five different elements of data over 58 consecutive days of event data collected from four data sources within Los

Alamos National Laboratory's (LANL) corporate, internal computer network. The de-identification process for computers and users is uniform across the five data elements allowing correlation and data integration — this means that computer C1 in one data element is the same computer C1 in all other data elements. Excluding time skew differences, event times also align between the different data elements.

All data starts with a time epoch of 1 using a time resolution of one second. The specific time frame collected from the operational network is not disclosed for security purposes. In addition, no data that allows association outside of the internal enterprise network is included (there are no associations to the Internet).

The specific data elements include:

Authentication Windows-based authentication events from both individual computers and centralised Active Directory domain controller servers.

Processes process start and stop events from individual Windows computers.

DNS DNS lookups as collected on internal DNS servers.

Network Flows network flow data as collected at several key router locations.

Red Team a set of well-defined red teaming events that present bad behaviour.

In total, the β data set is $\sim$12 GB compressed across the five data elements and presents 1,648,275,307 events in total for 12,425 users, 17,684 computers and 62,974 processes. A breakdown of data elements are shown in Figure 5. Attributes and caveats of the data elements are discussed in the following subsections.

4.1. *User and computer authentication events*

Authentication events are the most significant data element of the β data set. In comparison to the α data set, which contains simplified events representing a user successfully authenticating to a computer, this new authentication data element has a rich set of attributes for each event.

Data Element	Event Count	User Count	Computer Count	Process Count
Authentication	1,051,430,459	12,418	17,666	N/A
Processes	426,045,096	10,097	11,960	62,974
DNS	40,821,591	N/A	15,296	N/A
Network Flows	129,977,412	N/A	12,027	N/A
Red Team	749	98	305	N/A
Total	1,648,275,307	12,425	17,684	62,974

Fig. 5. An overview of the newly released β data set in terms of size and volume for each of the five data elements.

More specifically, each authentication event contains the following attributes:

- *Time*: The time of the authentication event.
- *Source User@Domain*: The user initiating the authentication event. If the user ends in "$", then it is a computer account for the specified computer. Domain is either DOMx, indicating an Active Directory domain or Cx, indicating a local computer account on computer Cx.

- *Destination User@Domain*: The user that the authentication event is mapping to. In many cases, this user will be the same as the Source User, but in other cases this is an authentication mapping event where Source User becomes (or attempts to become) the Destination User. User and domain details are the same as discussed for Source Users.
- *Source Computer*: The computer originating the authentication event.
- *Destination Computer*: The computer that the authentication event is terminating at. In many cases the Destination Computer is the same as the Source Computer, indicating a local authentication event on the computer. If they are different, it indicates an authentication event where the user is authenticating from one computer to another; in other words the user is *moving* through the network. Destination Computer in some instances is the special cases of a ticket granting ticket TGT or a user identity. These are specific to Kerberos service ticket (TGS) requests where the Source User is requesting either a new TGT or a TGS specific to a user account for various services that may be owned (run) by that user account (generally used by automated accounts and services).

- *Authentication Type*: This indicates the type of authentication occurring including Negotiate (a general type commonly seen), Kerberos and NTLM. Several other types occur as well. In some instances, the authentication type is not indicated and is represented with a question mark.
- *Logon Type*: The type of authentication occurring including across the Network, an Interactive keyboard session, a Batch event, a system Service, a screen saver Lock or Unlock and several others. In some instances, the Logon Type cannot be determined or specified and is represented with a question mark.
- *Authentication Orientation*: How the authentication event is being used. This includes indicating whether it is for granting a Kerberos TGT or TGS, a log on or log off event, or an authentication credential mapping. In some instances, the Authentication Orientation cannot be determined and is represented with a question mark.
- *Success or Failure*: Indicating whether the authentication event was successfully completed or failed. Failure could happen for several reasons including the wrong password was provided, a locked out account, or an authorisation failure.

Users that are well-known system-related accounts (SYSTEM, Local

Users that are well-known system-related accounts (SYSTEM, Local Service) were not de-identified. Note that any and all administrator accounts were de-identified. Failed authentication events are only included for users that had a successful authentication event somewhere within the data element. Any fields that did not have a valid entry or was not relevant to the event are represented with a question mark ("?"). Figure 6 shows the time series of the number of events, computers and users per day within the data element.

4.2. *Process start and stop events*

Process start and stop events are parsed and normalised from the aggregate Windows event data and, as a result, will have no time skew issues when compared to the authentication events. Process names are de-identified using just the base name of the process executable and not the path of the executable. For example, "\path1\install.exe" and "\path2\install.exe" are both treated as "install.exe" and map to the same de-identified process name. Due to automated updates happening with significant regularity

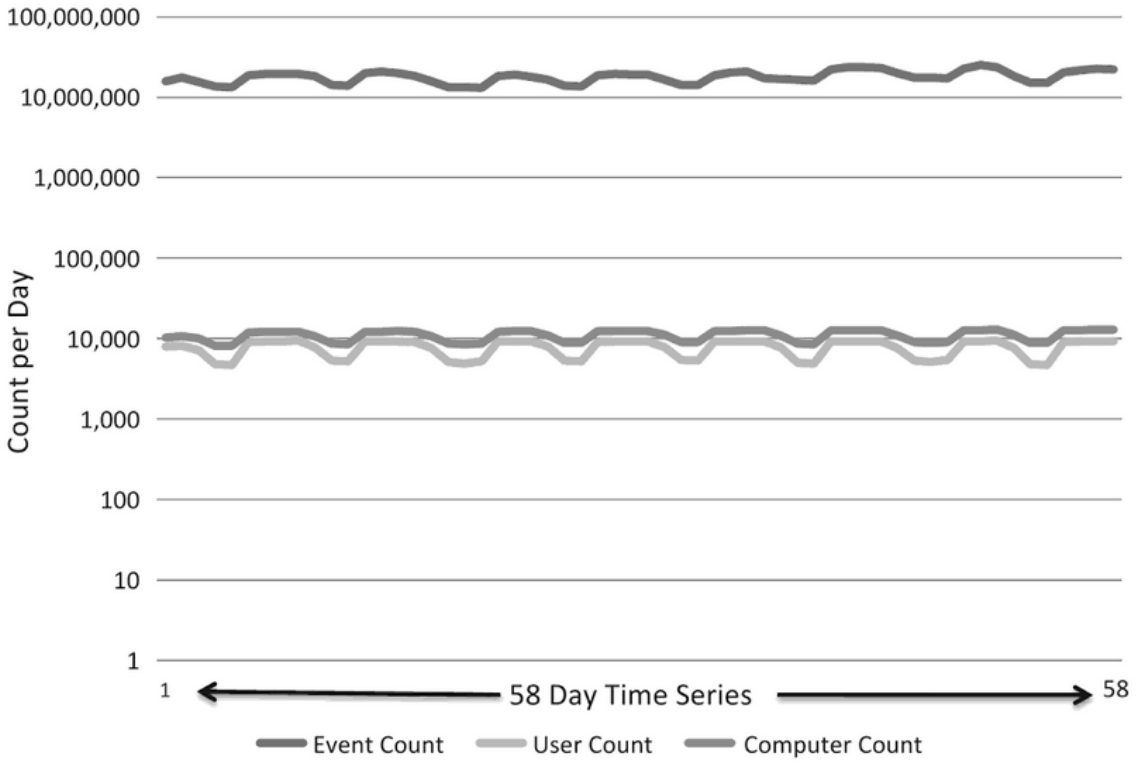

Fig. 6. Time series of authentication volume by event, computer and user count per day over the 58 days. User and computer counts show strong periodicity with non-work days showing lower counts. Collection volumes are consistent throughout the data element.

across the network, “install.exe” is the most common process in the data element and maps to the de-identified process name P16. Figure 7 shows the time series of the number of events, computers, users and processes per day within the data element. Each event includes a time, user (including domain as discussed in the previous authentication subsection), computer, process and whether the process is starting or stopping.

4.3. *DNS lookup events*

DNS lookup events provide an inferred connection event between two computers within the network. In combination with the network flow events, we find that these two data sources provide good coverage of many actual network connection events.

Collecting DNS lookup events is possible in most modern DNS servers, but may result in a performance impact. We have found that passive collection on the network connection to the DNS server is a practical way to collect this data element. Figure 8 shows the time series of the number of

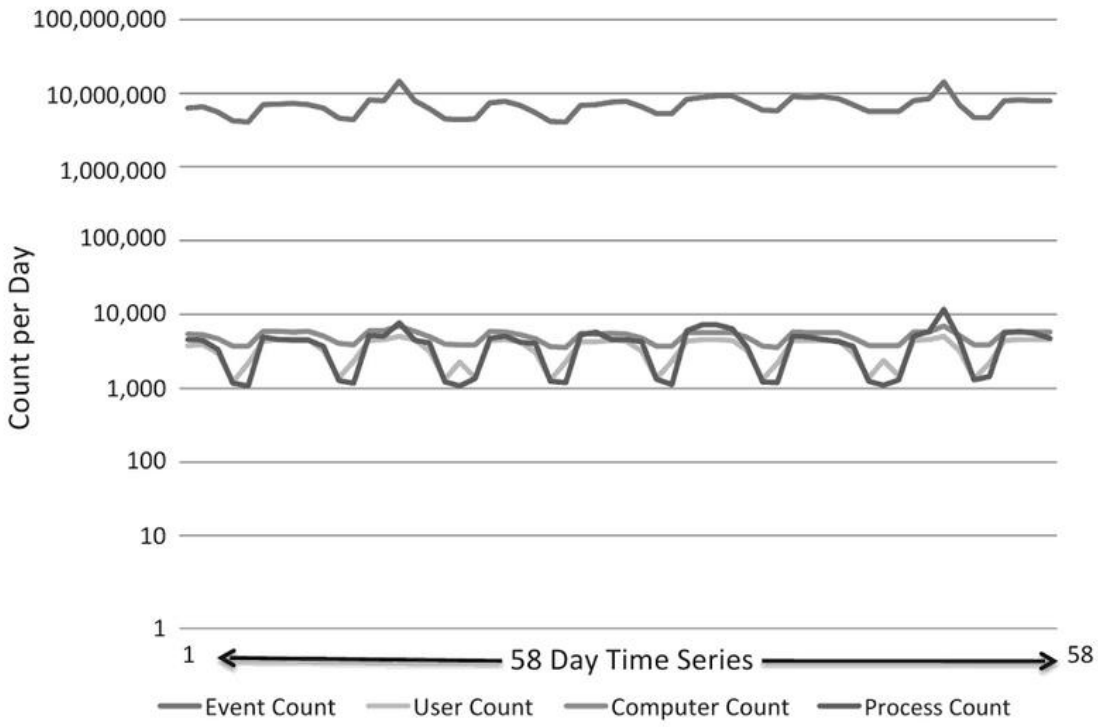

Fig. 7. Time series of process start and stop volume by event, computer, user and process count per day over the 58 days. Process and user counts, in particular, show strong periodicity with non-work days showing a lower counts. Collection volumes are consistent throughout the data element.

events and computers per day involved in DNS lookups within the data element. These events originate from three separate DNS servers. Unfortunately, in the first part of the data element, only one of the three servers had a valid collection configuration. As can be seen, this configuration problem is corrected on day 27. We believe this partial data loss is a common occurrence in real-world data.

The DNS data element includes a time, source computer and resolved computer in each event record. Included events are only to and for computers on the internal, enterprise network (no lookups to or from the Internet).

4.4. *Network flow events*

Network flow data theoretically should be great sources of information for cyber security analytics and in many ways are. Successful, cutting edge intrusion detection methods can take advantage of it.[16] However, in practice, we find this data source to be one of the most unstable and unreliable. In the real-world, flow collection is often considered an annoyance

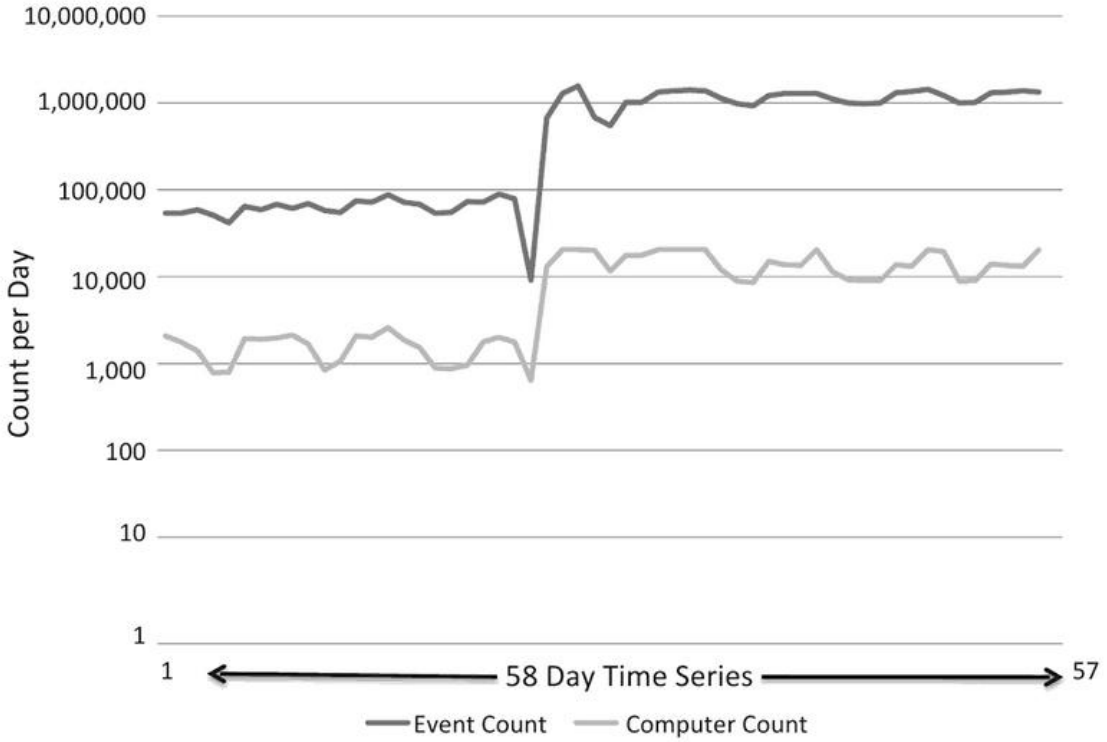

Fig. 8. Time series of DNS lookup volume by event and computer count per day over the 58 days. One of three DNS servers is reporting for the first 26 days. Some periodicity is seen within the data element, showing lower volume on non-work days.

to the IT personnel who run the network infrastructure as they rarely use it for troubleshooting purposes. It also has the potential to impact router and network performance if misconfigured. As a result, flow data is often lacking in consistency and quality. This can be seen explicitly in our network flow data element, where at day 29, a misconfiguration of the internal network routers completely stops the collection of network flow data. We assert that this is not an unlikely occurrence on most enterprise networks. In addition, the design of enterprise networks, the location of routers (and network switches, in some cases) and sampling rates all significantly impact the coverage of actual network flow events. As discussed in Section 8, our long-term hope is to rely on more data collected by individual computers and less by the network infrastructure.

Nonetheless, network flow data has value and is one of the data elements of the data set. A time series plot showing the number of events and computers per day is shown in Figure 9. The data element events contain a time, connection duration, source computer, source port, destination computer, destination port, protocol, packet count and byte count.

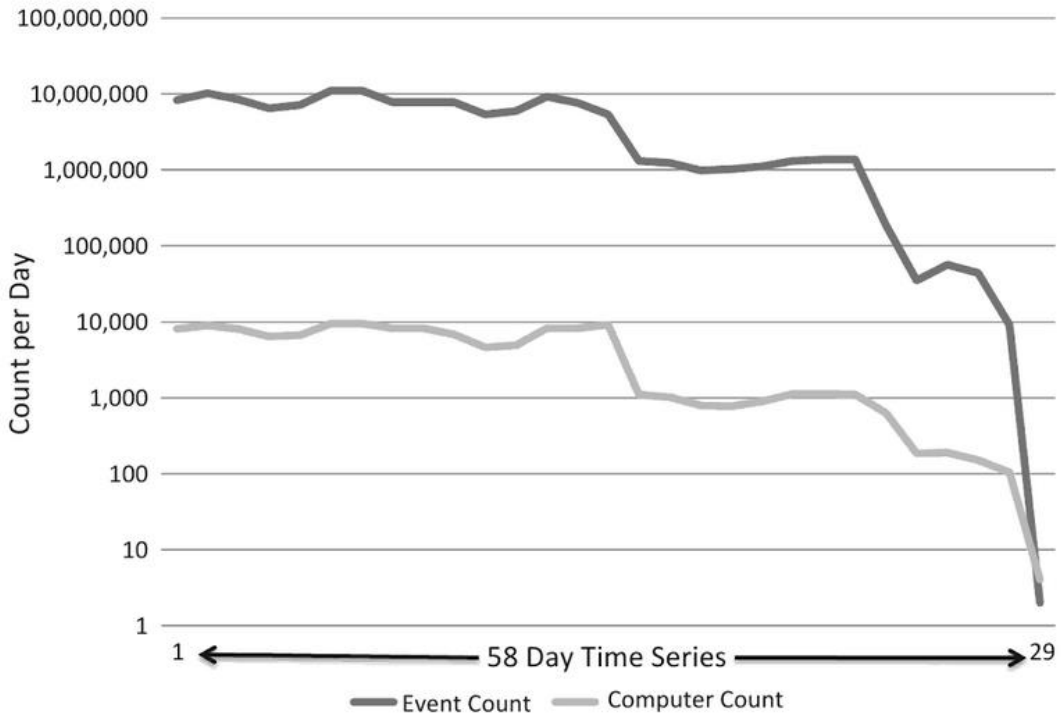

Fig. 9. Time series of network flow volume by event and computer count per day over the 58 days. However, due to a router configuration error, there is only network flow data for the first 29 days. Some periodicity is seen, but not to the extent exhibited by the other data elements.

Source and destination ports that are not well known (e.g. 80, 443, etc.) are de-identified.

4.5. *Red team compromise events*

During the 58 days, a number of compromise events exist. These events were intentionally created to test the security of the computers and user accounts within the enterprise network by a group of authorised attackers, commonly known as a red team. Based on the documented activity of the red team, 749 authentication events are known to have been performed by the red team using stolen user credentials. These events include time, user with domain, source computer and destination computer are included. Other indicators of compromise may exist throughout the β data set, but have not been validated or correlated. We believe this initial, yet incomplete, compromise labelling will provide significant opportunity for a variety of research outcomes. However, while there is important value in developing models for "*finding the bad guy*" in the data set, we hope that the

research community will use the β data set for broader research endeavours as well.

5. Data Quality

Data quality and errors within data sources are also a significant consideration when performing analysis. Relevant data parsing and normalisation processes must account for and manage these issues. We find that the best approach is to remove data that cannot be parsed due to it being relatively rare in comparison to the overall data set size and the importance of avoiding inaccurate data. Of course, such data can be saved to support parser extension and to account for the percentage of unaccounted traffic.

Log data is usually stored as compressed files, usually in the `gzip` format,[17] reducing space requirements by a factor of 10 or more due to the log text format and repetitive nature. Most compression mechanisms include a cyclic redundancy check (CRC) across individual data compression blocks (usually 4 KB) to help determine if data corruption exists within the data block. Within real-world data sets, we find between 0 and 4 corrupt data blocks over a year time period for each data set type. This corruption is the

result of either a failed compression process when the log file was being created or undetected read and write corruption from the file storage subsystem, as is possible within large-scale data stores.[18] When block corruption is detected within a compressed log data file, we stop processing the file and move to the next available data file. While this naïve approach results in a potential loss of data in the analysis processes, skipping the corrupt data blocks and determining proper field alignment would result in significantly more complex data parsing requirements and likely increase attribute inaccuracies.

Within the log data parsing process, other data corruption problems must also be considered. First, log entries can be incomplete or corrupt. Log entries may have only a partial set of fields or fields parse correctly yet still have corruption data within them. These situations are the result of a variety of factors that interrupt the event logging process. These factors range from computers being turned off suddenly to the loss and corruption occurring while distributed log data is transferred across the network. Most distributed log data is collected using the syslog network protocol, which

does not have message integrity or retransmission capabilities.[19] To avoid data inaccuracies within our analysis, we discard incomplete and corrupt log entries as part of the normalisation process.

5.1. *Missing data*

Another data quality consideration involves missing data. This problem exists in several forms including a loss of all data within a given time frame for a data set or a reduction in the number of data sources in a distributed data set (e.g. the number of computers reporting distributed event logs is undesirably reduced over some time period). Two examples of this were discussed in the previous section regarding DNS and network flow within the β data set.

One approach to detecting and mitigating this data loss problem is to instantiate a secondary monitoring system for evaluating the various data sources as they are captured and archived. We have found that providing data quality metrics to the operational IT staff allows them to fix the various problems that result in reduced or no data collection.[d]

We have found relevance in treating data collection loss as a time series of key quantification metrics over collected data sets. These metrics could

We have found relevance in treating data collection loss as a time series of key quantification metrics over collected data sets. These metrics could include the number of events, the number of computers, the number of users, or other similar quantifiers over a specific short time period. We find that one day (24 hour) time periods work well as time steps within the time series. Once the metric and time step is determined, we then use an exponential weighted moving average (EWMA) over the time series that is defined as:

$$\hat{\mu}_t = \alpha X_t + (1 - \alpha)\hat{\mu}_{t-1},$$

where X_t is the currently observed quantifier metric at time step t, α is the weighting factor, and $\hat{\mu}_{t-1}$ is the EWMA value observed at the previous time step. We find a constant α value of $\frac{1}{14}$ works well by providing a two week memory within the EWMA.

[d]While it seems intuitive that the operational IT systems themselves would monitor for log data collection problems, we find that the reality of modern IT system monitoring usually ensures that the systems are simply running, but does not consider data quality, quantity, or consistency; as is obviously seen in the DNS and network flow data elements released.

We also define a moving standard deviation in association to the EWMA as:

$$\hat{\sigma}_t = \sqrt{\alpha(X_t - \hat{\mu}_{t-1})(X_t - \hat{\mu}_t) + (1-\alpha)\hat{\sigma}_{t-1}^2},$$

where X_t is again the currently observed quantifier metric at time step t, α is the same weighting factor used by the EWMA, $\hat{\mu}_t$ is the current EWMA value at time step t, $\hat{\mu}_{t-1}$ is the EWMA from the previous time step and $\hat{\sigma}_{t-1}$ is the moving standard deviation from the previous time step. This moving standard deviation is derived from the work by Lambert and Liu.[20] Importantly, the moving standard deviation in combination with the EWMA allows us to construct a distance measure to track anomalous deviations from an expected norm within our time series.

Intuitively, the most obvious approach would be to track the number of logged events over some time period to ensure that a sufficient set was being recorded. Using event count over a year-long time period is shown in Figure 10. Note the high level of noise seen within the time series and the resulting difficulty in using an EWMA and moving standard deviation to determine data collection loss anomalies. We find that because of the variability in the number of events collected, other quantifiers are more

variability in the number of events collected, other quantifiers are more viable. For example, the number of computers logging on a daily basis provides an effective quantifier for many data sets. Using computer count over a year-long time period is shown in Figure 11. Note the lower noise content and the well-defined anomalies where computer reporting count drops below 4,000.

Another possibility is to track individual computers and their specific event reporting. Unfortunately, we find this difficult to do in practice due to the high variability of many desktops, laptops and other networked computers that come and go from the network. However, Figure 11 indicates that, in general, the same number of computers are consistently reporting each day. As a result, our primary assumption is that there is always a varying set of computers that are not connected to the network or not powered on. But it is also possible they are just not reporting data regularly for other, unknown reasons.

We also find that the consistency of individual computers and users varies across the β data set elements. Figure 12 shows that only approximately half of the computers are seen in all four primary elements of the β data set. Similar comparison for users is shown in Figure 13.

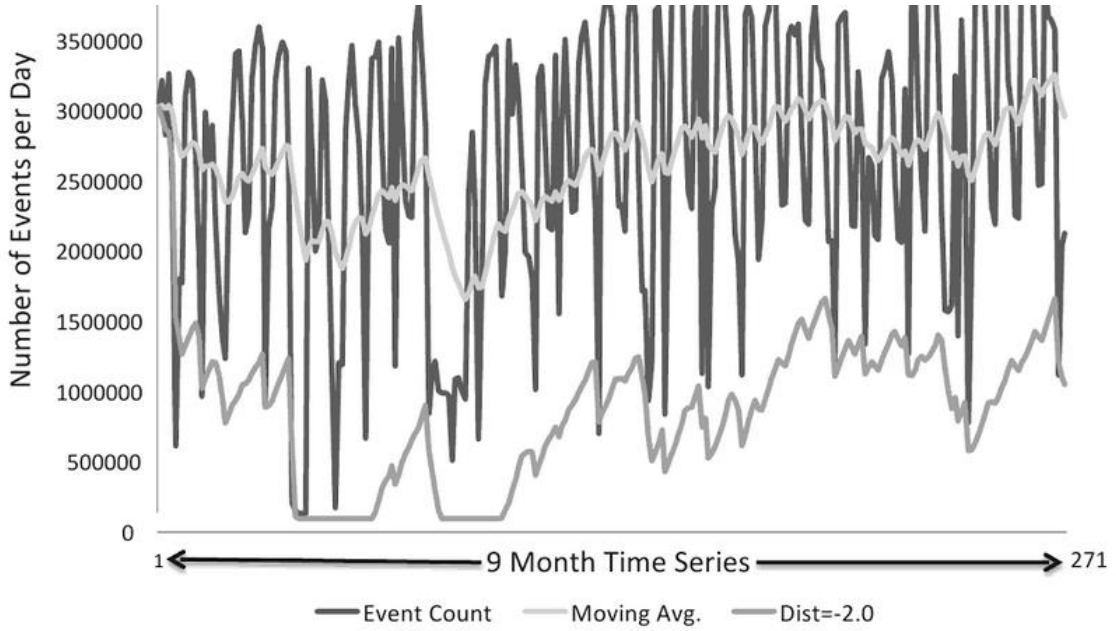

Fig. 10. The number of log events centrally collected per day from approximately 22,000 networked computers over the α data set.[1] Also displayed is an exponential weighted moving average (EWMA) and the associated moving standard deviation displayed at a Mahalanobis distance of 2.0 below the EWMA. The EWMA and moving standard deviation both use a weighted α value of $\frac{1}{14}$. Using log event count is a noisy indicator due to natural variation in volume. Figure 11 shows a much cleaner approach that shows obvious data loss events more effectively.

In general, within our data analysis processes, we do not directly address the loss of data other than to ensure a sufficiently large time window of data to reduce the impact of any small data loss events. Nonetheless, it is an area of data quality that must be considered, particularly as a function of the collection process. This is an area that may warrant additional future research.

5.2. *Time skew*

The last, and perhaps most significant, data quality issue that must be considered is time skew or timestamp inaccuracy within the data sets.

Data normalisation enables data integration across collected events and across disparate data sources. Time skew is a key concern when different sources of data are integrated. Computers, including servers, often have