Deconstructing Information Security Analysis

GIAC (GCIA) Gold Certification

Author: Daniel Severance, d.S3VERANCE@gmail.com
Advisor: *Bryan Simon*

Accepted: *4/30/21*

Abstract

Security resources often simplify the analysis cycle into a single atomic unit or a series of highly contextual steps rather than outline the process of building relationships in security data. Analysis workflows may be generalized for anomaly detection in positive security models. This paper explores common analytical techniques to deconstruct and reconstruct security data for analyst benefit. These techniques identify outliers and edges in behaviors and highlight contexts where these edges may occur. Methods are introduced to measure the effectiveness and efficiency of these pivots in investigation while covert channels are introduced to contrast practical application. Methods in this paper should supplement analyst workflows rather than replace existing processes.

1. Introduction

Information security analysis is the culmination of techniques used to identify and process the relationships in data through a security context. Within this context, analysts describe data through its respective confidentiality, integrity, and availability. This analytical process deconstructs information technologies and their standards, providing an active role in identifying adversaries and their methods.

Security analysts and researchers derive modern detection frameworks such as MITRE ATT&CK from common techniques and indicators utilized within red team or offensive playbooks (Strom et al., 2018). In this manner, security programs may often operate under a negative security model by attempting to block known bad methods and indicators within an offensive campaign. While this approach is effective for known adversaries and techniques, negative security models have no innate method to detect novel attacks. At scale, this model centralizes analysis techniques and provides tools for the dissemination of these indicators or signatures through technologies and protocols such as STIX and TAXII (Barnum, 2012) (Connolly et al., 2012). With the rise of data science and machine learning, mathematically or behaviorally derived detections were able to scale more effectively. However, while able to ingest data to determine abnormal behaviors in an environment, these approaches are often opaque box in nature, either through the nature of the model or through their nature as a proprietary detection technique for security companies.

This paper aims to facilitate a positive security model through identifying known acceptable behaviors and codifying fundamental searching and pivoting techniques utilized in hunting and proactive detection. Rather than known signatures, this paper will cover data exploration and anomaly detection through data deconstruction and reconstruction within new relationships. Pivots are reduced to analysis methods which produce edges to behavioral surfaces in the form of outliers and uncommonalities. These surfaces are placed in the context of environmental constraints. Security tooling and investigative context are highlighted to outline the identification of perspectives where edges may exist. Methods are introduced to measure and communicate the effectiveness and efficiency of information retrieval during this process.

Daniel Severance, d.S3VERANCE@gmail.com

This analytical workflow is applied in sum through the analysis of common covert channels. For the purpose of this paper, the definition of covert channels has been expanded beyond the original definition by Butler Lampson from channels "not intended for information transfer" to include side channels not intended for transfer by their operators in addition to their developers or creators (1973). In this way, this paper includes channels such as social media wherein data is transmitted tangent to its original role. This abstracted definition covers hidden channels used for both data exfiltration and command and control within lateral movement and asset management in offensive operations. A series of descriptive and inferential analysis techniques are proposed across these channel types to ascribe their benefits and limitations within analytical workflows.

2. Related Works

While this paper is scoped for a security audience, the content contained herein is primarily derived from other established scholastic domains such as statistics, discrete math, and criminal justice. The novelty and usefulness of this paper are within the application rather than the techniques themselves. Those accustomed to these domains may find some concepts familiar. The analytical techniques contained herein cover only a sample of techniques from the fields of discrete math and statistics. In the interest of accessibility, this paper will only cover descriptive statistics such as mean and standard deviation as well as a high-level selection of techniques in categories such as inference and time-series analysis. More advanced coverage of these topics is available in written and online resources (Bruce & Bruce, 2017) (VanderPlas, 2016). Study design has been indirectly applied to the security analysis pipeline as well. Some deviations and simplifications have been applied (Ranganathan & Aggarwal, 2018).

Additionally, the criminal justice system has established investigative patterns, including those for cybercrime. Investigations within the criminal justice system may include additional data collection techniques like interviewing and geo-spatial analysis that may not be present in all information security hunting processes (Criminal Intelligence, 2011). Despite differences in data collection processes, procedural overlap occurs in analysis.

Daniel Severance, d.S3VERANCE@gmail.com

Lastly, the incident response process within the information security domain inherently shares an academic scope. The detection process within the NIST response framework and the identification process contained within the SANS incident response resources both refer to the same analysis process contained herein (Cichonski et al., 2012) (Northcutt, 1995). Data points used for experimentation and analysis relate back to atomic indicators within threat sharing models (Johnson et al., 2016). The incident response process is primarily differentiated through its reactive detection facilitating analysis from known pivots in contrast to proactive threat hunting which utilizes arbitrary differentiated points. An incident response pipeline will vertically encompass impacted stages of the kill chain for a single investigation. Threat hunting will typically focus on a breadth-first analysis spanning all potential instances of a technique due to the absence of pivotable features. While confidence in pivoting differs between processes, the pivoting techniques within remain largely interchangeable.

3. Basis of Analysis

Analysis is a cyclical process. From some initial observation, analysts ask a question and construct a hypothesis. The analyst then tests this hypothesis, collecting and processing data from the environment until enough data are present to begin analysis: to describe the data, and infer relationships between datum. At this junction, if the results align with the hypothesis, the analyst can communicate these results in the affirmative. However, if the results don't align, or only partially align, with the hypothesis, these results inform the background research and observations within the next question and experimentation process. This analytical application of the scientific method is depicted in Figure 1 below.

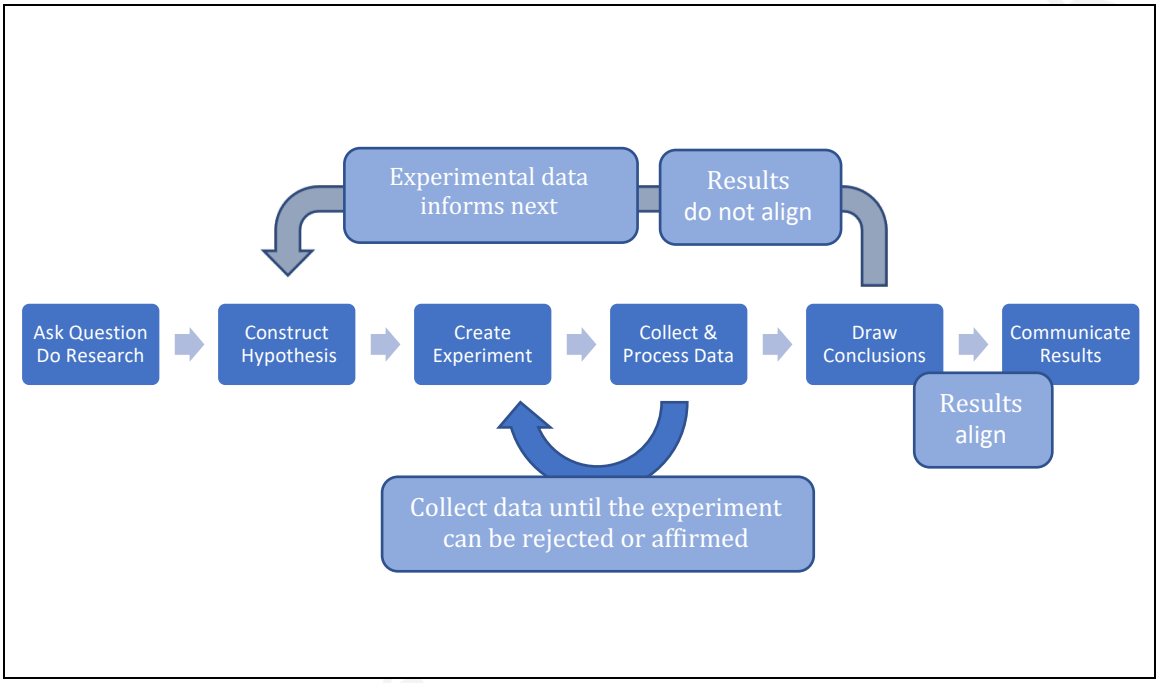

Figure 1
Analytical Application of Scientific Method

For example, within a security context, analysts could observe the bounds of sensitive regulatory information in the environment and ask if proprietary information exists outside those bounds. The analyst then collects logs over the last seven days from a set of web application servers in the environment. The analyst then filters through these logs for any data containing the format of a credit card number.

Upon finding no results, the analyst may reasonably conclude that for those servers on those days no credit cards are present. However, absence of evidence is not evidence of absence. The analyst has only confirmed that the result cannot be concluded within the given data collected. The analyst may pivot to a wider timeframe or other servers not included within initial data collection. Alternatively, during the testing phase, the analyst may manipulate or filter this data through other means, such as looking for the United States Social Security Number format, another form of potential regulatory data.

Analysis, like statistics, can be split into two separate paradigms: descriptive and inferential. Descriptive analysis aims to summarize. This analysis technique describes the data but does not draw any particular conclusions, nor does it extrapolate any relationships within.

Daniel Severance, d.S3VERANCE@gmail.com

Alternative to descriptive methods, inferential analysis is a form of reasoning. In this paradigm, analysts query data such that they can infer, or carry forward, some reasonable information. This paper predominantly utilizes inductive reasoning whereby a likely conclusion is inferred from multiple observations. Analysts will identify patterns, the limitations and bounds to those patterns, and any exceptions, outliers, or variations to those patterns. Additional categories of analysis such as predictive analysis are not covered in this paper, while prescriptive analysis may be derived from the application of these techniques.

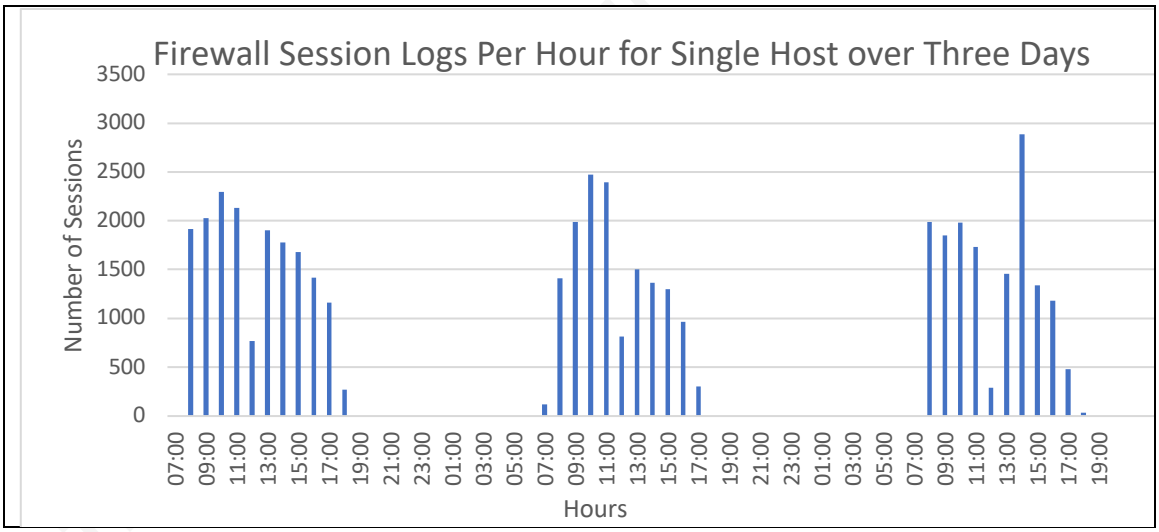

Figure 2
Time-series of internet traffic for single host across three days

Figure 2 above depicts total internet traffic events on a user workstation across three days. From this chart, an analyst can reasonably infer when the user starts and stops for the day and determine some common variations like seasonality, patterns occurring over regular intervals, such as the day-to-day cycle shown here. Some variations or exceptions define limited bounds where the behavior doesn't hold, for example, at approximately noon. The analyst may also determine outliers, or unusual data points in the set, such as the abnormal amount of traffic shown in the afternoon of the third day in the series.

During this inference and extrapolation, the analyst may form tentative narratives or explanations for the findings. In Figure 2 above, the analyst could potentially say that this user starts work around 0800 and works more in the morning than in the afternoon

but takes a break around 1200 for lunch every day. The analyst with less confidence may extrapolate that the target user's interest in work wanes through the afternoon, as traffic declines downwards over time each day.

These narratives are distinct from the results and conclusion of the experiment. While the analyst can conclude through descriptive analysis that there is more internet traffic in the morning than in the afternoon, the narrative is merely the beginning of the hypothesis for the analyst's next experiment.

Within this first experiment in Figure 2, all internet traffic has been atomized into a single sum. The analyst cannot directly determine if the target user is working, as both leisure and professional internet traffic may potentially source from this workstation. Given this secondary narrative, the user may process this data, filtering by website category to test the new hypothesis.

In the security domain, threat hunting involves proactively looking for abnormal behavior. During this process, the analyst is indirectly reviewing normal behavior for these variations, exceptions, and outliers. With a narrative informed by institutional knowledge, an analyst can review data and assets as they move through the environment.

When hunting, the analyst can also work from narratives from an adversary perspective through models such as the ATT&CK framework (Strom et al., 2018). In Figure 3 below, there is a naïve interpretation of logs showing an initial access attempt through an interactive logon followed by an escalation of privilege and an attempt to gain persistence on the server.

<86>Mar 3 23:58:02 widgetprint142 systemd[106956]: pam_unix(systemd-user:session): session opened for user devcms by (uid=1000)
<37>Mar 4 01:02:13 Message forwarded from widgetprint142: su: from devcms to root at /dev/pts/0
<78>Mar 4 02:51:01 widgetprint142 info CROND[15232]: (root) CMD (/usr/bin/cchannel)

Figure 3
Event sequence within a kill chain narrative

Within this inference, narratives may provide additional pivots in a positive security model through indicators in a negative security context. Through extrapolation in deduction or induction, the analyst may derive possible relationships from this data, whether they may be causal or correlative. To determine these relationships, analysts rely

on descriptive analysis to qualify and atomize this data. In Figure 2 all events are grouped as internet traffic. This atom prevented inference of qualities within that total but facilitated a simple summary of traffic across the three days. Distinct atoms facilitate comparison between these units at the cost of complexity in summarizing the data as a whole. In the case of internet traffic, website categories by Uniform Resource Locator (URL) may classify or categorize the individual elements of traffic further.

The adversary sequence in Figure 3 is atomized through the host context. The analyst loses visibility to behavior on other hosts but reduces noise in adversary sequencing. The cyclical analysis process facilitates inward and outward mobility in atomizing this information at multiple levels of abstraction. Analysts may maximize efforts through improving the information to data ratio, discussed further in Section 4, Environmental Constraints and Measures.

In Figure 4 below, URLs were extracted from the internet traffic of the user workstation in Figure 2. Reviewing all internet traffic indicated, there was one afternoon where more than normal data was transferred. This initial atomization quickly depicts the outlier over time but does not give additional context to where the traffic originated. The data below can be atomized, or grouped, at multiple levels of abstraction. Atomizing by unique URL may provide the most direct depiction of a covert Hypertext Transfer Protocol (HTTP) channel, however URLs can be repeated or may contain relatively unique pseudorandom variables within locations such as the resource path. This number of unique resources yields a low effective covert channel to legitimate channel ratio. In Figure 4 below, the ratio for our covert channels within unique URLs is 1/8. This grouping also requires the analyst to manually identify neighbors, such as the recon command that preceded the exfiltration.

Conversely, grouping by primary domain name yields a 1/4 ratio for the channel. While this atomization does not differentiate between reconnaissance and exfiltration for the channel, it correctly groups the commands for the next iteration of analysis.

URLs
https://github.com/
https://ko.wikipedia.org/
https://en.wikipedia.org/
https://c2.malware.kom/channel.php?cmd=recon
https://c2.malware.kom/channel.php?cmd=exfiltrate
https://www.google.com/
https://github.com/about
https://github.com/about

Figure 4
Traffic presenting logical abstraction of URL hierarchies

Information technologies often utilize structured languages and formats such as comma-separated values (CSV) or JavaScript Object Notation (JSON) such that the analyst can derive classifications, or groups, based on attributes like key-value pairs, static separators, or structures like subdomain.

Within security analysis, the focus is upon the confidentiality, integrity, and availability of information and information systems. For the covert channel depicted in the series of Figures 2 and 4, analysis is presented within the context of an application flow. Data was transferred from an internal workstation to some external asset. During this transaction, technologies such as endpoint detection and response (EDR), segmentation firewalls, web proxies, and intrusion detection systems may have seen or interacted with this traffic. Each log source in this transaction may provide a different context to the sequence of events that occurred. While an EDR platform may inform the analyst of the local processes connecting to the malicious domain, process level logging may present the lowest information ratio absent precise filters or experimentation.

This traffic can also be organized through logical layers of abstraction through models such as the Open Systems Interconnection (OSI) model. EDR may provide in-depth process information at the application level, however this information may be summarized with a higher information ratio when encapsulated at a transport level through technologies like Netflow due to the absence of application information not relevant to the hypothesis. Limitations such as sampling may impact data collection in some technologies (Cisco Systems, 2004). This application or technology-centric pivoting relies on classifying assets by traits of the data carrier. For data in rest and

transit, information can be grouped by traits such as Operating System, server role, or protocol version.

Operating perpendicular to the data carrier and an abstracted connectivity stack, the analyst can review the data itself contained within the carrier. Security analysis may also follow this data as it is transported across the environment, much like the review of carriers at junction points. Data-centric pivoting relies on classifying assets by the information contained within the channels, such as locations containing regulatory data. This data-centric analysis also bridges carrier relationships in the form of data custodianship and intangible relationships with entities such as business units.

Lastly, these two methods imply a stasis in the environment. Not only may data change, but changes in technology may occur as well. Security analysis can pivot on change occurring within the environment. Change flow can be represented across the delta of two states between any two arbitrary points in time. Often this change is represented in explicit logging within the change control platform in the environment, but it may also be derived independently as the range of the delta itself. For instance, given four nodes in a release pipeline deployment group, some measurable change occurs during release on all servers in the group. One server of the deployment group presented a different behavior of the four and therefore is an outlier and an opportunity for analysis.

Analytical methods within this framework can be primarily split into two distinct groups: qualitative and quantitative. Qualitative analysis deals with non-numerical data, primarily text. This analytical field relies on quality: the distinct attributes possessed by the measured entity. Accordingly, qualitative analysis is heavily dependent upon the observational context of the analyst. While structured languages explicitly differentiate attributes for qualitative analysis, the relevance and meaning of these values are dependent upon the analyst's background knowledge. This knowledge may be domain-specific, such as cryptographic protocols commonly present in Kerberoasting attacks, or it may be institutional, such as externally accessible internet protocol (IP) address ranges (Steal, 2020). Qualitative analysis is contingent upon the understanding of these values in context such as the MITRE ATT&CK framework. Absent inferred meaning, these

qualities still provide some context within descriptive analysis in classifying the data for numerical analysis.

In a numerical form, quantitative analysis facilitates the description of data through the measurement of these numbers, such as in magnitude and multitude. Unlike qualitative analysis, which may require the translation of attributes into some comparable quality, quantitative analysis allows for the immediate and direct measurement of numeric data for comparison.

Numerical data allows analysts to identify relationships in the form of outliers, variations, and exceptions without requiring an implicit understanding of background context. Instead, context is required only when beginning another iterative qualitative analysis or when deriving understanding of the results in conclusion. Analysts can rely on data from the population to define behaviors passively.

This qualitative classification for descriptive analysis can be sourced through multiple means. Previously stated, the classifications can be explicitly defined through sources such as key-value pairs in syslog, or a JSON payload. Where not explicitly defined, this data may still be implicitly defined, such as the hostname in syslog, which has a value but no explicit key.

Quantitative analysis, too, may implicitly classify objects for further descriptive analysis. For instance, a group of workstations with over 100 MB of SSH File Transfer Protocol (SFTP) uploads each may be considered SFTP users.

Some classifiers require a form of inference before cyclically beginning descriptive analysis. Given the following tcpdump output in Figure 5 below, the source and destination addresses are defined, however network hierarchy for these addresses is not present. When attempting to categorize the local source addresses for analysis, the analyst requires external context such as the Classless Inter-Domain Routing (CIDR) information to determine the subnet. For external destinations, information such as whois data can be processed to determine network scope and features such as the autonomous system number (ASN) to look for all networks associated with the company or provider. Alternatively, some intelligence or data feeds, such as those provided by Microsoft for

address ranges, may inform the analysis cycle (Office 365, 2021). In these instances, some sort of inference or derived context is required before beginning the original descriptive analysis and classification.

```
4:39:56.586949 IP 172.28.5.53.514 > 10.31.18.20.53306: Flags [.], ack 196315125, win 366, length 0

14:39:56.586952 IP 10.31.18.20.53306 > 172.28.5.53.514: Flags [.], seq 196315125:196316585, ack
1495174656, win 8212, length 1460

14:39:56.586955 IP 10.31.18.20.53306 > 172.28.5.53.514: Flags [P.], seq 196316585:196317306, ack
1495174656, win 8212, length 721
```

Figure 5
Inferential requirements in collected data sets

Descriptive analysis within quantitative techniques is predominantly tied to the four fundamental arithmetic operations. One of the most basic techniques is the sum of any series of events over some discrete interval or sample. Likewise, the quotient of two numbers provided through division can represent events as a fraction or ratio of occurrences within data sets.

Descriptive techniques continue through methods such as the arithmetic mean, median, or mode. These methods present the data related to itself, describing traits such as central tendency or frequency. These fundamental operations are built into most query languages such as the examples shown in Figure 6 below.

```
SELECT COUNT(username) FROM events

SELECT SUM(bytes_in) FROM events
WHERE protocol = 'https'

SELECT source_address, AVG(COUNT(destination_address)) FROM events
WHERE traffic_category = 'entertainment' GROUP BY source_address
```

Figure 6
Descriptive operations built into query languages

Applying this concept, the assets in Figure 7 below are the sample output of the grouped sum in Figure 6 above. With the context that the source address is a central update server, the sum of the bytes downloaded potentially indicates some level of update transfer. From this data, there is a likely grouping of the Boston assets in comparison to the Seattle offices. On the contrary, if the analyst groups by team function, such as marketing, there is a visible discrepancy in the result, indicating less of a relationship.

Asset Name	Megabytes Downloaded
BOS-MKTG-12	233
BOS-MKTG-09	232
BOS-ACCT-22	232
SEA-MKTG-14	1
SEA-ADVT-07	0

Figure 7
Sum of traffic to update server for collection of hosts

However, real-world data are far from as clean as the example in Figure 7. Analysts may address this quantitative data in a qualitative manner for quick pivoting, however this method has no distinct precision. When grouping data, the analyst can use quantitative approaches to measure or describe the results. Data with greater levels of volatility are inherently more difficult to predict and therefore more difficult to define or infer some relationship.

A numerical comparison between these data points can measure this volatility. One of the most straightforward measures is the range, calculated as the difference between the maximum and minimum in a series of data. An example of this query type is provided in Figure 8.

```
SELECT (MAX(bytes_in) – MIN(bytes_in)) AS RANGE
FROM events
WHERE destination_address = 'update-east-01' AND traffic_type = 'OS_UPDATE'
GROUP BY BUSINESSUNIT(hostname)
```

Figure 8
Numerical comparison of data points in a set

Much like the sum, the range may promptly indicate outliers due to the reduction and inclusion of extremes within the data set. However, while quick and efficient, this method provides no innate relative context to the series as a whole, only the bounds, or the upper and lower absolute limits of the data. Relative scale may differ dependent upon the measured data. Some enterprises may see an average of 1 Gigabyte (GB) of traffic for some arbitrary pipeline, while another enterprise may see 1 Terabyte (TB). A 1 GB range for the former enterprise is a variance equal to the 100% of the average itself, yielding low predictably with the current data set. The same range variance for the 1 TB pipeline has 1/1000th of the impact. For this reason, flat methods such as range comparison are best maintained within data sets of some relative context.

Other common methods such as the variation ratio, mean deviation, or standard deviation provide the analyst some level of additional context for these comparisons. Like the arithmetic operations before, most query languages have functions to calculate these comparison methods.

```
SELECT CalendarYear AS Year, CalendarMonth AS Month, COUNT(destination_address) AS
DestAddrCount,
    STDEV(DestAddrCount) OVER (ORDER BY CalendarYear, CalendarMonth) AS StdDeviation
FROM events
WHERE hostname = 'REG-DATA-01' AND network_zone = 'remote'
ORDER BY CalendarMonth
```

Figure 9
Measuring deviance from central tendencies in query languages

Standard deviation measures dispersion – the spread of the data from its central tendency or average. Unlike the range, which is determined by the bounds, the measure of central tendency through methods like the arithmetic mean facilitates a range from a representation or average of the entire data set. Low values indicate smaller variance in the data set – which indicates less dispersion.

For highly predictable data sets, such as those found in automation, the mode, or the value appearing most frequently, may quickly identify central tendencies. In technologies, such as Internet Control Message Protocol (ICMP) pings or Simple Network Management Protocol (SNMP) traps, the variation ratio may best highlight deviations from the default instance of the repeated task. Numeric variations such as within the payload of the ICMP packet can determine the similarity in these tasks. For Figure 10 below, two sets of ICMP traffic are provided in the context of a 32-byte field. In these sets, the variation ratio can be calculated as $1 - (\text{Frequency}_{\text{Mode}} / \text{Total })$. In the top example, the duplicated traffic yields a variation ratio of 1-(40/80) = 0.5. However, in the bottom example the unique payloads derived from the transport of an EXE file yields a variation ratio of 1 – (1/80) = 0.9875.

```
Address   0 1 2 3 4 5 6 7 8 9 a b c d e f Dump
00000000  414243444546474849 4a4b4c4d4e4f50 ABCDEFGHIJKLMNOP
00000010  515253545556575859 5a4142434445464748 QRSTUVWXYZABCDEFGH
00000020  414243444546474849 4a4b4c4d4e4f50 ABCDEFGHIJKLMNOP
00000030  515253545556575859 5a4142434445464748 QRSTUVWXYZABCDEFGH
00000040  414243444546474849 4a4b4c4d4e4f50 ABCDEFGHIJKLMNOP
00000050  515253545556575859 5a4142434445464748 QRSTUVWXYZABCDEFGH
00000060  414243444546474849 4a4b4c4d4e4f50 ABCDEFGHIJKLMNOP

Address   0 1 2 3 4 5 6 7 8 9 a b c d e f Dump
00000000  4d5a9000030000000400000 0ffff0000 MZ.........ÿÿ..
00000010  b800000000000000040000 000000000 ¸.......@.......
00000020  000000000000000000000 000000000 .................
00000030  000000000000000000000 00f8000000 .............ø...
00000040  0e1fba0e00b409cd21b80 14ccd215468 ..º..´.Í!¸.LÍ!Th
00000050  697320070726f6772616d 2063616e6e6f is program canno
00000060  742062652072756e20696 e20444f5320 t be run in DOS
```

Figure 10
Use of mode in finding deviance in highly regular data sets

This proportion represents the number of entries in the data set that do not belong to the mode. In statistics, this technique belongs to the field of qualitative variation. Covering an extensive scope, this category includes indices through concepts like entropy, or the randomness of data, as well.

Data sets such as the English language contain their own frequency distributions for atomic units like letters or words. In a raw form, entropy represents the disorder, randomness, or uncertainty of an entity. Within information entropy, this uncertainty is used to describe the possible states of an entity (Shannon, 1948). A series of two binary numbers can represent 2^2 or 4 possible values – [00,01,11,10] meaning that this data set can convey four unique meanings without any external context.

Some technological processes such as encryption provide this randomness as a means to prevent guessing or approximation. While successfully preventing these guessing attacks, encryption naturally presents randomness, a visible quality.

Techniques in qualitative variation and analysis not only compare data against another datum but against other truths or paradigms. Shannon calculated this entropy relative to possible state in a method now known as Shannon Entropy (1948). While not

natively available in most platforms outside a data science role, this entropy can still be calculated in query languages such as the examples provided in Figure 11.

```
SELECT CEILING(LOG2(COUNT(DISTINCT username))) FROM events
```

```
SELECT -SUM(cu*LOG2(rr)) FROM (
SELECT RATIO_TO_REPORT(cu) OVER() AS rr FROM (
SELECT COUNT(usernames) AS cu FROM authevents))
```

Figure 11
Measuring entropy of data sets in query languages

The first example provides the amount of uncertainty, or total bits of information entropy, required to encode all of the values in the column. Increased randomness in a field such as username may indicate a blind spray, while entropy in payloads may indicate encryption or distinct data types may be present.

This concept is expanded in the second example of Figure 11 above. Measured through the Shannon entropy, the randomness is calculated not only over the series but for the individual data points as well. Further expansion can occur when these uncertainties are displayed as probabilities.

The analytical techniques discussed thus far have framed the data sets irrespective of time. The data structure is viewed in an unordered list which supports sorting and manipulation by arbitrary classifiers or attributes. The first instance suggesting ordering occurs in Figure 3 where the events are viewed in an ordered linear data structure depicting a sequence of events in a kill chain.

While datetime data are included in the original figure, the rudimentary model only defines the sequence of the events relative to each other without indicating an absolute timeline. In addition to the qualitative and quantitative methods outlined previously, data sets can be modeled within these unique data structures. Abstract data types such as graphs facilitate the depiction of relationships between atomic units within the data sets. Applying the probability of events provides a stochastic model depicting the likelihood of events or a sequence of events. Presented within finite state machines, state changes can be recorded through space and time.

The flat authentication events for a user in the environment are provided in Figure 12 below where the events have been depicted in a finite state machine. Within this model – common behaviors are visually apparent. The sample user has a distributed one-to-many spread of destinations from the source workstation. Also visible is the privileged access workstation (PAW) schema. The target user authenticated into the PAW before pivoting to the regulatory server in 100% of the recorded events. Conversely, the nonregulatory server was accessed both through the PAW and the nonprivileged workstation with a 50% equal spread.

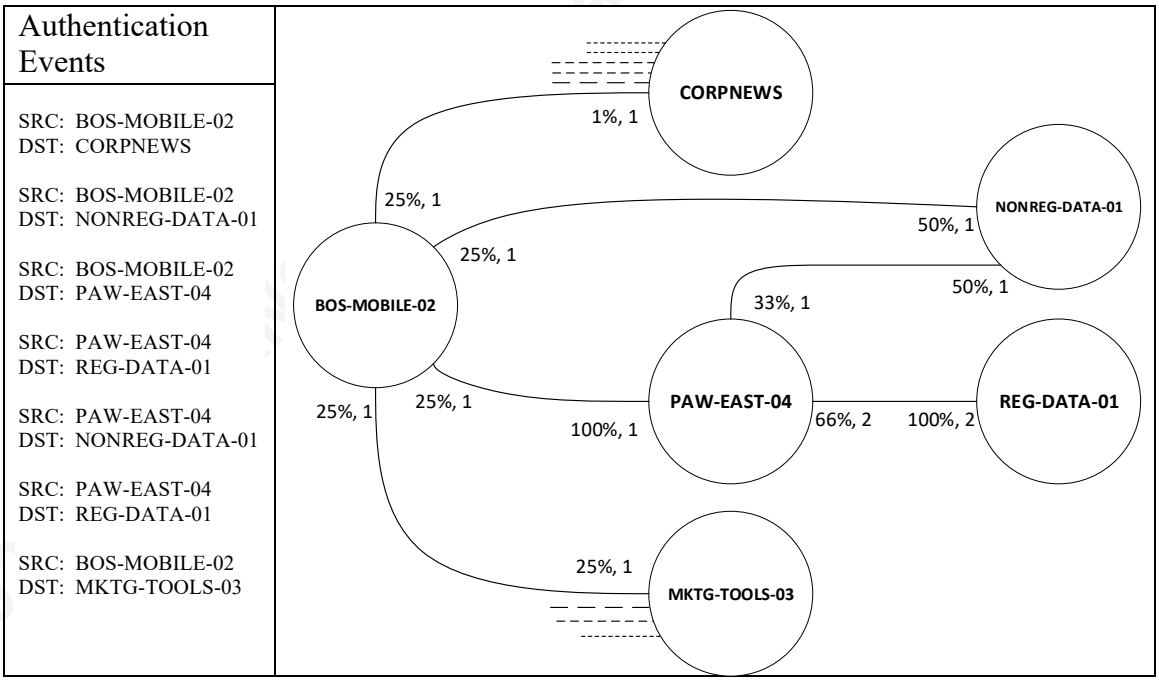

Figure 12
Variations in data depictions over multiple dimensions

These models can be used in conjunction with the techniques previously discussed, such as comparing access trends for the regulatory server between users of the same role through classifiers, or potentially as a summary through a summation of access events for all users in this role, rather than the individual.

While the analyst can use this combination of data states over time to create useful models of the environment, isolating context to time alone may provide as much benefit as isolating within a state context. Depicted in Figure 2, a time-series is a sequence of data points indexed within a timed order. This indexing is performed in

discrete time, often as equally spaced finite intervals of time. This analysis technique facilitates a visual comparison of magnitudes for some series over time and is natively present in many SIEM and log analytics tools (Krause-Harder, 2019).

In these tools, the magnitude and interval are the malleable features used for analysis. Discussed previously in this section, the magnitude is dependent upon the value measured, which may be grouped by some classifier. Provided some classifier, such as IP addresses, analysts may move up or down hierarchical groupings for both summary and explication.

The selection of hierarchical level in classification should be informed by both the hypothesis and observations. If, for example, during the third afternoon depicted in Figure 2 there was network congestion and the indicated user was observed watching videos from the internet, the time-series graph can align the internet traffic. However, this pivot is done in the context of meeting some arbitrary bandwidth cap. If this cap is not met by the indicated user alone, or absent an original observation of this user, moving up the network hierarchy to include the user network would best summarize data for the hypothesis. The bounds of the network congestion and these known observations are utilized in determining the most useful scope, such as an entire corporate level, a single office location, or a localized team. This hierarchical grouping extends to the bounds in time based on applicable knowledge. Some server lifespans may be measured in years, while containerized infrastructure may be measured in days and some more ephemeral serverless infrastructure measures transactions in milliseconds or seconds.

Lastly, it is within the context of these bounds that data are manipulated by the analyst during the experimentation phase where the analyst derives some results. These manipulation techniques do not actively provide results, but instead, passively facilitate analysis through organizing the data in translation. Methods such as sorting translate unordered lists into ordered lists required for sequencing while filtering modifies the values present for categorical analysis. Value truncation, limiting results, and aliasing fields may assist not only in calculation for analysis but may improve the ability of the analyst in perceiving qualitative results.

Daniel Severance, d.S3VERANCE@gmail.com

Techniques such as long-tail analysis are a compound form of this type, relying on sorted ordering to facilitate analyst focus on uncommon results in the 'tail' of a chart (Conrad, 2015). Such methods rely on the distribution of behaviors or activity that occur at scale and frequency distributions that naturally occur in many human interactions (Adamic & Huberman, 2002). These distributions result as a product of the environment where the data collection and analysis occur.

4. Environmental Constraints and Measures

Data collection and processing are inherently tied to the maturity of the security program alongside the architecture of the environment. This data can be presented in a multidimensional state. Highlighted previously, width is present in the horizontal direction through the location of data across hosts, processes, and other data receptacles. This width ultimately ties back to some storage mechanism such as hard drives (HDD) or the more ephemeral random-access memory (RAM). Following this horizontal presentation, height is then present through levels of logical abstraction. These abstractions can be identified across information technologies from programming languages, character encoding, and more nuanced circumstances such as hashes, or one-way-functions. For example, within the framework of networked hosts in an environment, the OSI model logically differentiates levels of abstraction. In addition to the where and how the data are stored, there is a question of when the data are stored. Depth is present in the concept of time. Technologies such as routers and switches at a network level or registers at a process level work with data in a highly contextual temporary environment.

Security analysis entails collecting and processing the data most relevant and applicable to the current investigation or experimentation within the cyclical analysis process. In this analysis process, the analyst should work within the constraints to provide the greatest efficiency and effectiveness. Effectiveness is derived from the analyst's ability to provide security benefit. In analysis, this benefit occurs in the translation of data to information or intelligence, which can be used to inform other security processes such

as engineering, architecture, operations, or execution. Effectiveness is built upon asking questions that provide value and the analyst's ability to answer those questions.

Efficiency then is the ability to avoid waste in that translation. Typically, efficiency is measured as the amount of useful output compared to the input of the process. Input contains not only the collected data but resources such as analyst time and computational resources as well. Data science, much like security analysis, can consider this success within a binary true or false classification model. This classification occurs at all levels of abstraction, whether the analyst is determining if fraud occurred or when selecting addresses belonging to a specific network from a list. Analysts can use concepts such as accuracy, precision, and recall to measure the relation of successes to failures.

The analyst builds upon the familiar confusion matrix consisting of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) comparing the actual results from the predicted results. Accuracy presents the number of correct predictions out of the total number of predictions. This concept was seen through the information ratio presented in Section 3, Basis of Analysis. Accurate analysis indicates that predictions, such as the occurrence of fraud, are correct. However, accuracy pertains to the result, not the process. A philosophical perfect data collection indicates that all of the required data points are present—no necessary data points are missing, and no unnecessary data points have been collected. Precision and recall are an extension of this concept. In information retrieval, precision is the proportion of retrieved data that are relevant to the query. Precision provides the context of waste in relevance – false positives. Conversely, recall is the proportion of relevant documents that were successfully retrieved. Recall provides the context of error in retrieval – false negatives. These three concepts are applied in Figure 13 below. In an environment of one-hundred servers, management identified ten servers during a possible ransomware incident. During debrief, it was determined that thirty total servers were ransomed, however only six of those servers were on the original list of ten. Six true positives out of ten selected indicates four false positives. Six true positives out of thirty actual cases likewise indicates twenty-four false negatives: missed cases. Of the not selected servers, indicated by the false negatives and true negatives, there are sixty-six true negatives.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{6 + 66}{100} = 0.72$$
$$Precision = \frac{TP}{TP + FP} = \frac{6}{6 + 4} = 0.60$$
$$Recall = \frac{TP}{TP + FN} = \frac{6}{6 + 24} = 0.20$$
$$F_1\ Score = \frac{2(Precision \cdot Recall)}{Precision + Recall} = \frac{2\ (0.60\ \cdot 0.20)}{0.60 + 0.20} = \frac{0.24}{0.80} = 0.30$$

Figure 13
Calculating efficiency

The analyst can achieve 100% recall by merely returning all documents, thereby reducing the false negatives, or missed results, down to 0. Conversely, the analyst can achieve 100% precision by returning only known true positives, thereby reducing the false positive detections at the cost of missing any unknowns, creating false negatives. It is then the combination of these concepts in the F1 score, typically the harmonic mean of the precision and recall, that presents a summarized depiction of our result's efficiency and effectiveness. In the above example of Figure 13, the analysts were relatively precise, only incorrectly identifying four servers. However, the cost of the precision was found in recall such that twenty-four servers were never polled. This efficiency is found in information retrieval through that information ratio.

However, analysis does not occur in a vacuum. The analyst may achieve 100% effectiveness by waiting 7-14 business days to have magnetic tape backups transported, or 2-4 business days to retrieve data from a separate business unit, however this delay in action may be inefficient. Furthermore, efficient results may lose their effectiveness due to environmental constraints like management mandates or business release timetables. Additionally, these metrics may be relative, as what may be efficient for external business units or the business as a whole may be inefficient for the individual analyst.

External resources such as analyst time and compute time must be considered in determining waste and efficiency in the analysis cycle. Similar to the iterative qualitative and quantitative cycles depicted previously, the analyst may allow for qualitative confidence levels, measuring efficiency through qualitative means instead of quantitative means some portion of the time. Additionally, environmental constraints present themselves through the same tenets of security. Detective and protective controls must be

available for analysis to occur, and the integrity of the resulting data determines the confidence of the investigation.

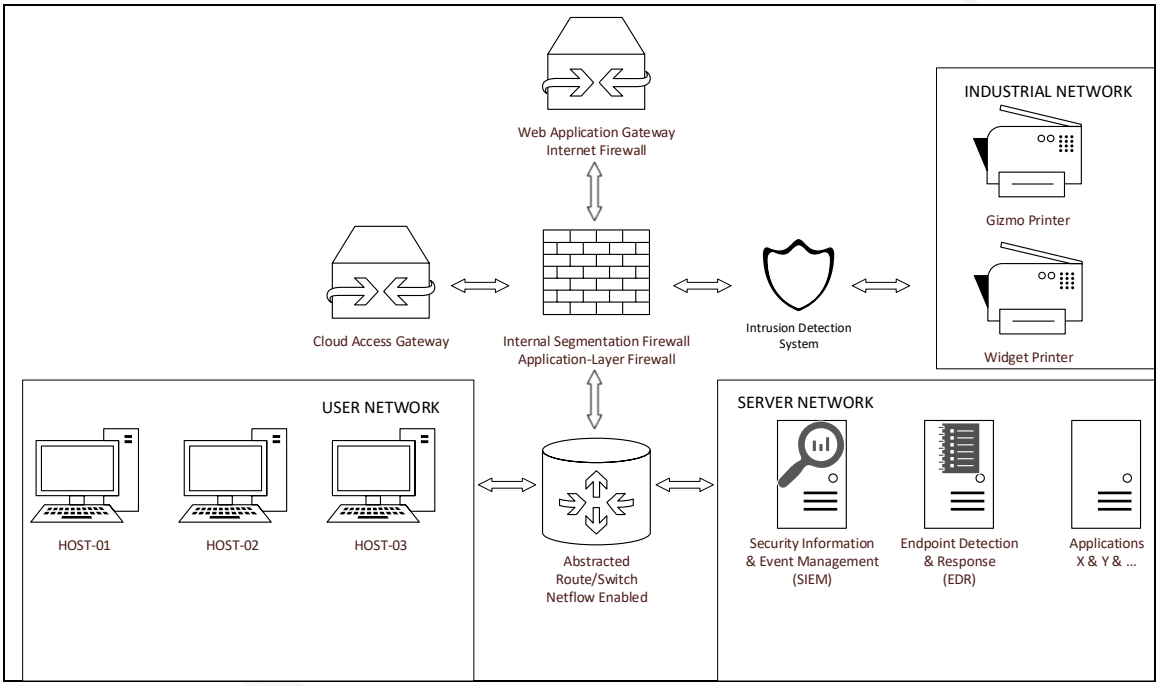

Figure 14
Sample logical network architecture for Widget Corp

An example environment for Widget Corp has been outlined in Figure 14 above to depict these constraints in action. In this given scenario, an insider threat has been secretly printing threatening widgets from the user network. Absent any differentiation in the quality of measurements, the closer the analyst is to the destination, the closer the analyst is to the object being acted upon, thereby identifying what behavior occurred. Conversely, the closer the analyst is to the source of the change agent, the closer the analyst is to the person(s) or thing(s) enacting the change.

In the provided environment, a centralized SIEM is collecting all local enterprise logs with the exception of host OS logging which remains on the user workstations. This collection equalizes analyst efforts during the collection and processing tasks within those processes. The application logs on the widget printers indicate that threatening widgets were printed but have no user information. This limitation spawns the recursive investigative cycle. The analyst must derive this information actively through user authentication events, or passively through behavioral byproducts such as network

activity. Visible in the tools network, the EDR platform available will have user information along with the process and network information of the source workstation. This device will have high recall, guaranteeing that the user pulls some related event, but low precision due to the number of unrelated events pulled. This precision provides the analyst low efficiency in an initial investigation unless more filters or constraints can be used to limit the retrieved data. Similarly, host OS logs may have a high recall rate and provide effectiveness through source user information, however architectural constraints here require the analyst's time to pull in the distributed logs. Due to the high volume of events, the retention periods for these devices may also be reduced for an equivalent total volume.

Passive network logging remains a useful byproduct. The route/switch equipment in this environment provides Netflow. Containing primarily information at layer 3 and 4 of the OSI model, this source may provide a better information ratio given the absence of application-level logging at the cost of prohibiting any further deconstruction of that communication as an atomic unit. Once success is derived at the network level, providing an IP address and port, the analyst must pivot to another collected data set where the information ratio is beneficial for determining the user, given the new IP address pivot.

Given the network topology of Figure 14, the segmentation firewalls may provide a similar role to Netflow. Technological constraints such as logging profiles and sampling may motivate analyst decisions for effectiveness and efficiency. Additionally, tools such as the IDS pictured within may provide negative security models through known signatures. In instances where there is a known signature, this negative security model will traditionally surpass positive models in recall due to the alert-driven activity that is provided instead of the collection of all source events used to derive the alert.

Selection of toolsets in the data collection process may include other features at analyst discretion. Logging at session start or session end may impact the effectiveness of some logging solutions. This example has also assumed relative levels of effectiveness. Purpose-built tools such as the web gateway depicted may provide additional context for web traffic over a generalized solution such as the segmentation firewall. Concepts such

as centralization in logging impact not only the availability of logs, but potentiality the integrity of logs as log manipulation is simplified on discrete hosts.

This concept of relative efficiency and effectiveness is expanded for verticality or logical abstraction, as briefly mentioned for Netflow. Covert channels at the application level may not present themselves in lower-level logging. Likewise, channels utilizing the underlying transport or presentation layers may be abstracted to an unrecognizable level in application-level logs. Architectures such as Software-as-a-Service (SaaS) may also limit detection and protection capabilities in some capacity through limiting abstraction.

Environmental efficiencies are further derived from changes over time. Information assets may change in the environment. These assets include tangibles such as end-user and Internet-of-Things (IoT) devices but likewise include intangibles such as proprietary information used for intelligence gathering or financial gain. Understanding change motivators is critical to understanding where change may originate in the environment. External motivators such as regulatory bodies may source change alongside internal security or audit teams rather than traditional business units. For subsidiaries, some manner of change may be controlled in part by the parent company. In smaller or less regulated industries, change may originate directly from high-level personnel outside normal business cycles.

For companies with an Information Technology Service Management (ITSM) platform and processes, these changes may occur through some level of change management. In-band changes to information services should occur through the change management platform. In following responsibility matrices, informed security personnel may use this tooling as an alternative data collection point for changes in depth or time.

Analysis encompasses description and inference derived from relationships and comparisons. Much like the actual and predicted state utilized within efficiency, logged changes can be differentiated between change platforms, ideal templates such as source control and secure gold configurations, and the resulting deployments in infrastructure and distributed clients.

Change can be audited between these interrelated change platforms, much like the comparison of exclusive horizontal platforms in two disparate network technologies such as network flows and application firewall logs. Intra-related change can still be derived directly through active change markers, such as those provided in Microsoft Active Directory Security Groups, or passively through a discrete change in values for some unique identifier (Audit Security, 2019). This latter example can be depicted through a recreation of Dynamic Host Configuration Protocol (DHCP) logs providing an IP address for an asset over time given unifying keys like hostname or username.

In-band changes provide some level of procedural guarantee for certification and testing even if that guaranteed level is no validation. Accordingly, there is some level of confidence between predicted and actual state. In-band changes therefore prioritize interrelated change platforms, which can assist foremost in determining the difference between deployed states, such as within deployed source code in a Continuous Integration/Continuous Deployment (CI/CD) pipeline.

Out-of-band changes, such as ad hoc requests provided by high-level personnel, may circumnavigate traditional change management processes, and forgo customary certification and testing procedures. In such circumstances, there is an unknown delta. Intra-related change is prioritized to determine the change target before proceeding.

Lastly, these architectural limitations may operate in conjunction with external principles such as the entropy presented above. The presence of decryption technologies may modify analysis opportunities available for data within the carrier method. Not only is the data made accessible for negative security models within detection technologies, but the randomness presented by encryption provides differing levels of entropy in pseudo-random and deterministic data which provide context for identification.

Decryption of transport layer security (TLS) in network stacks offers view into the underlying application data. Data independently encrypted at the application level, such as within malware command channels, display entropy not found within the now deterministic decrypted traffic.

Daniel Severance, d.S3VERANCE@gmail.com

Likewise, absent decryption technologies, differentiation in contextual data fields is inhibited without external context. An example of such method and its indicators is found within domain name, Server Name Indication (SNI) and HTTPS host headers used in methods such as domain fronting (Fifield et al., 2015).

It is through the limitations of security architecture, collection platforms, and business processes that the analyst ultimately selects data for collection and processing within the analysis lifecycle. The horizontal and vertical levels of physical and logical abstraction assist the analyst in providing both efficiency and effectiveness to the analysis process. By utilizing iterative and recursive hypothesis testing within these environmental constraints, analysts may better use the data available.

5. Discourse on Analysis Methods

Practical security analysis occurs within the framework of this analytical cycle and its constraints. Covert channels are introduced into the environment depicted previously in Figure 14. Timing channels utilize the modulation of some resource over time to convey meaning. Analysts can derive meaning through both relative and absolute timing. Relative channels measure the channel to itself, measuring the state at t_1, t_2, t_3, to t_n for some interval. By extension, the interval can be modulated such that the frequency is used to convey meaning. Absolute timing channels operate on a specific timescale, such as some event occurring on Friday the 2nd, or events occurring on odd or even seconds.

Storage channels modulate the state of the data itself to convey meaning. The definition of traditional covert channels identifies information transfer through media in a way not intended by the designers of the media. This definition often depicts data carving or padding. Carved data represents the modification, or substitution, of data with new content of an ascribed meaning. Carving includes substitution in fields such as ICMP payloads, or least significant bits (LSB) in steganography (Gupta, Goyal, & Bhushan, 2012). Padded data represents the addition of new content with no meaningful modification in the original channel content. Modern covert channels may utilize legitimate channels with padded content, such as within social media or the abuse of

legitimate connections utilizing stolen or impersonated credentials. Other methods such as subtraction also hold.

Lastly, these channels can be deconstructed in the same manner as the security analysis. Storage channels represent the change in an axis over space while timing channels represent the change in an axis over time. Like the relationship between acceleration and velocity, behavioral channels present themselves as an integral relating to the change in behavior over time. Behavioral channels occur as changes in the slope, or sequences of events over time.

During analysis, the three channel categories can be applied against themselves, against neighbor classifiers in a social context, against natural laws in the form of protocols and other legislature, and lastly against physical or scientific laws in the form of entropy, communication speeds, and other boundaries.

Manipulative techniques are included throughout the paper. These techniques encompass the data processing that occurs between data collection and analysis. Data manipulation may occur during experimentation as data is added, subtracted, and substituted through functions such as joining and splitting. Processes like thresholding filter the collected data by analyst criteria thereby assisting in the translation to reactive detective measures by means of prescriptive analysis.

Data manipulation is a necessary but not sufficient condition for identification of all three covert channel categories. The data must be described or examined in some means beyond processing for analysis to occur. Some manipulation techniques are explicated within the context of qualitative-variation where inference may occur.

Qualitative-Descriptive techniques summarize the quality, or attributes, of security data. Derived from the features of the data, quality is innately reliant on perception. Qualitative techniques require the identification and interpretation of input to achieve awareness. Value interpretation often requires external knowledge to provide effectiveness. It is also due to perception that qualitative methods are not readily scalable. Values are not innately comparable due to externally derived meaning. Examples of qualitative-descriptive techniques are provided in Figure 15 below.

```
SELECT DISTINCT user_agent FROM events WHERE hostname= 'BOS-MKTG-14'
SELECT datetime, username FROM events WHERE hostname= 'BOS-MKTG-14'
```

Figure 15
Example qualitative-descriptive techniques

Qualitative-Descriptive methods are effective in negative security models where collected data can be preprocessed in some manner for known strings or formats. Derived from quality, these methods describe content features. These descriptive methods will identify but not interpret the data of a storage channel. Analysis of quality is primarily limited to the data in the channel due to lack of quality differentiation in the carrier.

Storage channels reliant on padding are differentiated by their addition of content yielding numerical differences in quantities such as the sum of bytes transferred by the host. Numerical differences facilitate quantitative comparison.

Carved storage channels reliant upon data modification present no such quantitative differentiation in an external context. Instead, carved channels are best differentiated by the quality of transported data. This comparison can be executed within the complexity of the carving and detective measures present. Given a steganographic example of hiding data within an image file, application-layer firewalls may differentiate qualities in images such as file extension and internal metadata like file headers and magic numbers. Data loss prevention utilities may further differentiate traffic through comparisons in encoding mechanisms, file structure, or derived artifacts from factors such as compression. Said qualitative detection techniques require external context such as encryption and encoding algorithms for proper interpretation. It is for this reason that qualitative-descriptive techniques may be conditionally effective and efficient in identifying storage channels through data.

Timing channels rely on the timing of events rather than the value of events to convey some meaning. Due to the qualitative and descriptive nature of events, timing channels are not differentiated in any capacity. Qualitative-Descriptive methods are both inefficient and ineffective at identifying timing channels.

Behavioral channels rely on both qualitative values as well as time. Qualitative-Descriptive methods display and summarize the elements of a behavioral channel in a flat

structure absent any relationship. While behavioral elements may be identified, interpretation of the data during analysis requires external observation to properly interpret. It is for this reason that qualitative-descriptive methods may be inefficient but effective at identifying behavioral channels under ideal circumstances.

Qualitative-Variation techniques introduce a union of **qualitative-inference** and **quantitative-descriptive** methods. These techniques translate between qualitative and quantitative data and are an extension of the data processing techniques highlighted within data manipulation. Defined formally, qualitative-variation operates on nominal data. However, the ability to operate on both unordered and ordered data across space and time facilitates its role as versatile and widely scoped.

Flat techniques such as sum and range are complemented by weighted techniques like mean, median, and standard deviation. Through the ability to find both central tendencies and outliers, qualitative-variation is efficient and effective at identifying all three channel types.

Summary techniques facilitate pivoting by isolating contexts that differentiate outliers for the hypothesis. This pivoting was presented in the scenarios around the time series depicted in Figure 2. In both covert channel and congestion scenarios, the sum of traffic in the outlier was utilized to pivot either in the classification of the content transferred, or within the context of the data transferred by neighbors. In this manner, quantity facilitates efficient pivoting between positive outlier-outlier contexts. Analysts may introduce inefficiencies when translating outlier results into qualitative conclusions or when pivoting to contexts without outliers through inference.

Due to the inherent reliance on the data set in determining central tendencies, these techniques are dependent upon proper selection in data collection rather than the experimentation found in qualitative techniques. Errors in collection modify the central tendencies of the collected data set.

For storage channels, the quantization of data requires proper classification to be effective. If the analyst has already properly classified and structured the channel's traffic, other forms of qualitative and quantitative analysis may better infer meaning over

a descriptive method. While inefficient at comparing the channel to itself or neighbors, comparison against natural and physical laws is effective due to differences in tendency at scale in concepts such as entropy.

Due to its nature as a descriptor of quality in numeric data sets, use of qualitative-variation techniques is effective along the time axis in behavioral and time channels. Operating on nominal data sets, some limitations apply, such as within ordered time data. As a descriptive quantitative method, the capacity to summarize data facilitates efficient and relatively effective pivoting in timing and behavioral channels.

Quantitative methods have been deconstructed into two categories to depict relationships of multitude and magnitude over space and time.

Quantitative-Space techniques are structured on values or states of data. These techniques focus on quantizing collected data primarily on the basis of some qualitative value. This category includes fields such as categorical analysis. Due to quantization, this category of analytical techniques is highly scalable. The dependence on quantity rather than quality removes the interpretation requirement that occurred in qualification. Likewise, reliance on state-space for organization yields flexible iterative classification in conjunction with descriptive analytical techniques such as within Figure 16 below.

```
SELECT username, COUNT(username) FROM events
GROUP BY hostname ORDER BY hostname
```

```
SELECT bytes_out FROM events
WHERE deploy_group = 'prod-web-widget'  AND log_source = 'web-proxy-gw'
GROUP by hostname LAST 7 DAYS
```

```
SELECT DISTINCT destination_address FROM events
WHERE log_source='network-firewall'
AND NOT IN (SELECT destination_address FROM events
      WHERE log_source='host_firewall')
```

Figure 16
Example quantitative techniques comparing state values

Quantization facilitates innate comparison between classifications in neighbors providing highly effective social context. However, due to reduction of context in time to a single discrete time interval, this method is ineffective in comparing channels against previous states.

Provided proper flat and weighted context, Quantitative-Space methods are efficient and effective in identifying storage channels. However, the removal of quality in the analysis of classifiers increases the emphasis on quantity. Improper use of flat and weighted measure introduces inefficiencies in pivoting.

Timing channels rely on the change of state over time rather than the intrinsic meaning of state to derive meaning. Quantification of state space is inefficient and ineffective at identifying relative timing channels. Due to the presence of explicit indicators such as timestamps as classifiers, absolute timing channels may be inefficiently but effectively identified through quantitative patterns.

It is due to the abstraction in the time axis described above that behavioral channels are also inefficiently but effectively identified. Relative context of events is natively removed however the analyst may still succeed in directly identifying patterns within the quantified classifications, or indirectly through inference in a qualitative translation of classifiers in the data set.

Quantitative-Time techniques are structured upon temporal data and focus on quantizing data in reference to time rather than state-space. This category includes fields such as time-series analysis. The focus on when events occurred abstracts away native comparison of classifiers. Time-series analysis reflects the change in a single or grouped value over time.

Simple time-series may reflect the unique count or sum of events such as firewall sessions or total bytes transferred over some interval. Methods such as the mean or median may be used to measure the central tendencies of the data per interval. In a quantitative form, times-series are effective in determining quantitative changes over time not only in excess, but in absence as well, through situations such as modification of logging profiles on endpoints.

```
SELECT *, AVG(authentication) OVER(PARTITION BY application ORDER BY date ROWS
BETWEEN 6 PRECEDING AND CURRENT ROW) as moving_average_7day
FROM authentication_events ;
```

Figure 17
Example quantitative techniques comparing values over time

Inference for these time series has been described only within the context of qualitative analysis of the resulting time series. More precise methods such as regression analysis may be used to quantitively describe the relationship between two time series, such as within internet traffic for multiple hosts on a network over time.

Within the scope of this paper, quantitative time methods are inefficient in identification of covert channels since the analyst is required to select an interesting slice of the measured data points. Quantitative-time methods require some key indicators or observations, such as the intelligence provided the analyst in the analysis of Figure 2. Time-series may provide interpretation for qualitative results by providing a temporal context to an investigation

The reliance on quantization over time natively facilitates effective identification of timing channels. This same quantization over time yields ineffective results for carved storage channels where the modification presents no outward changes in quantity. Padded channels presenting an addition of content are still present, however, the selection of data points within the time series provides only additional context used in observations for the next hypothesis cycle, and provides no additional information on the classified data point itself, due to atomization of the data point.

This same limitation occurs through the atomization of the data point in behavioral covert channels yielding relatively ineffective and inefficient identification.

Discrete-Abstract techniques introduce an additional field into the analysis process in the form of modeling. Due to the presence of dimensionality in data structures, abstracted models may represent both qualitative and quantitative data over space and time. However, this translation must occur at the cost of analyst time as flat collected data is organized during processing and provided structure.

For identifying storage or timing channels, methods such as stochastic modeling may be effective but inefficient due to the inclusion of the additional axis not related to the channel type Visual modeling presents opportunities for analysis of visual qualities that may not be readily apparent in flat data structures, such the probabilities depicting asset relationships in Figure 12.

Modeling may present changes in state over time depicting an additional dimension that required external inference from the analyst in other techniques. Due to the ability to identify likely behaviors and activity, the use of structures and methods like stochastic modeling are the most efficient and effective at identifying and interpreting behavioral channels.

6. Future Considerations

No mention is provided on data distributions and their optimal analysis techniques. Any further applications of statistical principles should begin to rely on these foundations for proper analysis. As with all data, the confidentiality, integrity, and availability of this data are also inherently relevant. This paper does not review these features in a comprehensive capacity.

Of these features, confidentiality provides the most nuance to the analysis pipeline. Analysis through inference is innately derived from the data available to the analysts. While the product of the analysis may have some level of confidentiality, the results of the analysis are derived from information and observations that may be confidential to some parties. By deriving certain conclusions, it is possible to ascertain information that may have been present in that determination.

Availability of data derives similar concerns in the analysis pipeline. Information or intelligence has timeliness. While displayed as a cyclical process, this paper has assumed the available data are static. In real-world scenarios, analysts may have to supplement data through alternative tools and sources until more relevant data are available. Contrarily, the analyst may need to revise previous models and hypotheses to account for new data collected that was unavailable during previous experimentation cycles.

Lastly, the integrity and reliability of the data may be a concern. This paper assumes collected data to be truthful unless otherwise stated. While multiple data sources are compared and contrasted, real-world data may need to be evaluated alongside regulated confidence intervals. Only briefly discussed were incongruencies in the data and potential causes to the logging configuration.

Daniel Severance, d.S3VERANCE@gmail.com

These tenets may directly impact the analysis pipeline in determinant ways. This paper also provides no large context for where these detections are performed at scale in modern security toolsets. Likewise, purpose-built tools, such as machine learning algorithms, may do pattern matching better at scale than naïve human matching. The independent techniques displayed in this paper may be used alongside these technologies for mutual benefit outweighing individual performance.

Knowledge or understanding of these information systems is considered largely static for the purpose of this paper. Real-world data collection may require additional context from third parties to ensure meaning has been inferred properly. This process would occur in parallel alongside data collection and processing and may include additional techniques such as interviewing referenced in investigative criminal justice proceedings.

The culmination of these considerations impacts primarily the interpretation of data in the results phase. Future work should more explicitly handle these interpretations to better facilitate the translation of data by the analyst.

7. Conclusion

Despite the weaknesses in interpretation discussed in Section 6, Future Considerations, analytical procedures were successfully deconstructed within a cyclical analysis model. Proactive threat hunting was placed within a positive security context reliant on pivoting through abnormalities. These security analytical processes were then placed within the context of security concepts and environmental constraints, where means were provided to measure the effectiveness and efficiency. The techniques were then successfully classified and applied against varying covert channel types where common effectiveness and efficiency was derived. The techniques and processes provided within this paper can be integrated into security analysts' toolkits to proactively hunt within positive security models alongside the traditional negative security models.

References

Adamic, L. A., Huberman, B. A. (2002). Zipf's Law and the Internet. Glottometrics, 3:143-150

Audit security group Management (Windows 10) - Windows security. (2019, February 28). Retrieved March 1, 2021, from https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/audit-security-group-management

Barnum, S. (2012). Standardizing cyber threat intelligence information with the structured threat information expression (stix). Mitre Corporation, 11, 1-22.

Bruce, P., & Bruce, A. (2017). Practical statistics for data scientists. Sebastopol, CA: O'Reilly Media.

Cichonski, P., Millar, T., Grance, T., & Scarfone, K. (2012). Computer Security Incident Handling Guide : Recommendations of the National Institute of Standards and Technology. National Institute of Standards and Technology. https://doi.org/10.6028/nist.sp.800-61r2

Cisco Systems NetFlow Services Export Version 9. (2004). doi: 10.17487/rfc3954

Connolly, J., Davidson, M., Richard, M., & Skorupka, C. (2012). The Trusted Automated eXchange of Indicator Information (TAXII trademark). MITRE CORP MCLEAN VA MCLEAN.

Conrad, E. (2015, January 22). Long tail analysis of windows event logs. Retrieved March 10, 2021, from https://www.ericconrad.com/2015/01/long-tail-analysis-with-eric-conrad.html

Criminal Intelligence: Manual for Analysts (V.10-58435-April 2011-100 ed.). (n.d.). United Nations Publications.

Daniel Severance, d.S3VERANCE@gmail.com

Fifield, D., Lan, C., Hynes, R., Wegmann, P., & Paxson, V. (2015). Blocking-resistant communication through domain fronting. Proceedings on Privacy Enhancing Technologies, 2015(2), 46–64. doi: 10.1515/popets-2015-0009

Gupta, S., Goyal, A., & Bhushan, B. (2012). Information Hiding Using Least Significant Bit Steganography and Cryptography. International Journal of Modern Education and Computer Science, 4(6), 27–34. doi: 10.5815/ijmecs.2012.06.04

Johnson, C. S., Badger, M. L., Waltermire, D. A., Snyder, J., & Skorupka, C. (2016). Guide to Cyber Threat Information Sharing. https://doi.org/10.6028/nist.sp.800-150

Krause-Harder, S. (2019, August 29). Querying and aggregating time series data in elasticsearch. Retrieved March 5, 2021, from https://www.elastic.co/blog/querying-and-aggregating-time-series-data-in-elasticsearch

Lampson, B. W. (1973). A note on the confinement problem. Communications of the ACM, 16(10), 613-615.

Northcutt, S. (1995). Computer Security Incident Handling Step-By-Step (Tech.). SANS Institute.

Office 365 URLs and IP address ranges - Microsoft 365 Enterprise. (2021, March 1). Retrieved March 1, 2021, from https://docs.microsoft.com/en-us/microsoft-365/enterprise/urls-and-ip-address-ranges?view=o365-worldwide

Ranganathan, P., & Aggarwal, R. (2018). Study designs: Part 1 - An overview and classification. Perspectives in clinical research, 9(4), 184–186. https://doi.org/10.4103/picr.PICR_124_18

Daniel Severance, d.S3VERANCE@gmail.com

Shannon, C. E. (1948). A Mathematical Theory of Communication. *Bell System Technical Journal*, *27*(3), 379–423. doi: 10.1002/j.1538-7305.1948.tb01338.x

Steal or forge Kerberos Tickets: Kerberoasting. (2020, October 20). Retrieved March 10, 2021, from https://attack.mitre.org/techniques/T1558/003/

Strom, B. E., Applebaum, A., Miller, D. P., Nickels, K. C., Pennington, A. G., &

Thomas, C. B. (2018). Mitre att&ck: Design and philosophy. Technical report.

VanderPlas, J. (2016). Python Data Science Handbook. Sebastopol, CA: O'Reilly Media.