Steps for Improving Data Comprehension for Digital Security and Forensics

Robert F. Erbacher
Department of Computer Science, UMC 4205
Utah State University
Robert.Erbacher@usu.edu

Abstract – *Currently, there is an enormous need to better understand the underlying data being examined (or not examined) by intrusion and attack detection techniques. This lack of understanding leads to enormous amounts of extraneous information being collected, resulting in data sets in the gigabyte to terabyte range and creating scalability issues in terms of how to collect, collate, store, and analyze the collected data. Additionally, much of the available data with respect to the internal network is not yet being collected, resulting in enormous vulnerabilities, especially with respect to insider threat. This will exacerbate the scalability problem with the need to collect and collate the available information. In this paper, we discuss the fundamental issues relating to data collection and data comprehension. Additionally, we propose directions for improving attack analysis and the advantages of our modified data collection strategies.*

Keywords **–**Intrusion Detection, Data Comprehension

1 Introduction

Intrusion and attack detection systems are intrinsically dependent on our understanding of the underlying data. Currently, our understanding of the underlying data is limited and this reduces our ability to develop effective techniques for dealing with this data. These limitations apply not only to the analysis techniques but also to the collection of data. Given our poor understanding of the data we are in essence collecting a lot of the wrong data and likely not enough of the data that is needed.

This results in an inability to effectively identify attacks, especially insider threats, and the collection of extensive data sets which are un-analyzable. By insider threats we are referring to direct attempts at compromise from internal systems, data exfiltration, data extrusion, etc. Thus the question arises, what data should be collected? This need is compounded by the volume of network data collected and the general lack of system log information being collected. Generally, network data will be collected on perimeter systems and select system log information from servers. Network data is not collected internally, limiting detection ability of insider threats. Alternatively, many system administrators have begun only collecting network flows, as opposed to network traffic data. This alone limits the analyzability of the available data. System log information is not collected from desktop systems, especially Apple and Microsoft systems, which adds additional limits to the detection of many types of attacks; making insider threat detection particularly problematic. Additionally, collating network wide information allows for identification of correlations not otherwise identifiable.

Collecting system log information and internal network data is clearly critical for insider threat detection and general improved attack detection. However, collecting arbitrary information will only exacerbate the scale issue currently presented to intrusion detection systems. Thus the critical challenge is what data needs to be collected and what data is not.

2 Problem Domain

Theorem 1: The inadequacy of current intrusion and attack detection techniques derive from a lack of understanding of the underlying data and that more fundamental research into the nature of said data is needed.

2.1 Intrusion detection techniques

Enormous effort has been applied towards intrusion and attack detection. These efforts have thrown all manner of techniques at the problem, including: visualization, statistical data mining, machine learning, pattern matching, change auditing, etc. These techniques fall into six main categories: pattern matching, database techniques, supervised learning techniques, unsupervised learning techniques, state transition techniques, and visualization techniques [9] [10].

- **Pattern Matching** – Compares current data against known patterns of activity known to be part of an attack or compromise data stream. These patterns can be identified through explicit signatures or through rule systems. This is exemplified by snort [18].
- **Change Auditing Techniques** – Compare current system information against that stored in a database. Essentially provide a database of auditing information. Tripwire provides an example of this class of technique.
- **Supervised Learning** – Data mining techniques fall into one of two categories. With supervised learning data mining techniques are trained against known activity. This training data may be known good or known bad behavior, though finding real-world data that is verified to be benign is very difficult. When the training data is known bad behavior the supervised learning techniques becomes similar to a pattern matching technique. When the training data is known good behavior then the technique essentially attempts to identify activity not matching known good behavior, i.e., anomaly detection. Here

we lump together the disparate data mining techniques, including: statistical techniques, machine learning techniques, distance techniques, etc.

- **Unsupervised Learning** – Unsupervised learning attempts to classify current data without the use of training data. Anomalous activity is identified by assuming the largest class is normal data and the remaining activity is anomalous activity.
- **State Transition** – These tools create state machine representations of attack models and attempt to identify attacks by identifying when a sequence of events puts the state machine into a known attack state. These tools can also model approved state transitions and flag any behavior that deviates from them.
- **Visualization** – The goal of visualization is to create a visual representation of the data, either in part or as a whole, which will allow an analyst to use visual cognition and domain knowledge to identify normal vs. abnormal activity [17]. An extensive visualization capability will also provide extensive forensic, i.e. analysis, capability. Visualization techniques are generally designed to work with other intrusion detection techniques such that anomalies identified by intrusion detection systems will be identified within the visualization as an additional data source, providing information as to where an analyst may wish to focus.

We have listed the main techniques being applied; though other techniques clearly exist, there are too many techniques to list individually. Additionally, we have grouped all of the data mining techniques into supervised and unsupervised. These rather large groups encompass many techniques based on distances, profiling, modeling, etc. Lazarevic et al. [10] provide a detailed discussion of the different techniques.

2.2 Limitations of intrusion detection systems

The effectiveness of these techniques is completely dependent on the data available for analysis as well as our comprehension of said data and the effectiveness to which we can apply the said techniques. These techniques can be applied to network traffic data, system log files, process accounting logs, combinations of data sources, etc. These techniques have given limitations:

- **Slight deviations can avoid detection** – This is particularly problematic with pattern matching techniques as slight variations made by an attacker can prevent matching.
- **Zero day attacks** – Novel attacks can completely avoid detection by pattern matching techniques.
- **Data integration** – Attackers aware of potential data mining techniques can more tightly tie their attack to normal network data, making detection that much more difficult.
- **Naïve attacks can be used to hide sophisticated attacks** – Sophisticated attackers will sometimes initiate a noisy naïve attack from one system while initiating a quieter and more sophisticated attack from a separate host, or set of hosts. In this way, the sophisticated attack will be obfuscated. Naïve attacks can not be completely ignored as they can be an important indicator of threats. Yet in general a well configured network has no concern for naïve attacks.
- **Noisy data** – Network traffic data is intrinsically noisy and chaotic in nature. This adds a general layer of complexity to the analysis of network traffic data. This can be further complicated by encrypted data and the added difficulty of detecting attacks in such scenarios.
- **Explaining alerts** – Anomaly detection systems are poor at explaining why they reached an alert, which limits their utility in operational environments; the analyst must manually discover the context that triggered the alert.

Ptacek et al. [12] discuss extensively of how many intrusion detection techniques can be evaded.

2.3 Known problems of intrusion detection systems

The limitations of these techniques result in known problems:

- **False positive and false negative rates** – This is the classic problem associated with IDS systems. Current techniques are inaccurate and result in many missed attacks and many wrongly identified connection sequences. This is an exhibition of the ever-changing nature of individuals' use of network data in both the short and long term.
- **Data collection rate** – The volume of data generated by most sensors is itself problematic. Many organizations have gone beyond collecting megabytes or gigabytes of data and are collecting terabytes of data with no available capability for analyzing said data.
- **Sensor positioning** – Fundamentally, administrators will position a sensor at the network boundary in order to detect inbound attacks from external entities. The difficulty lies with insider threats. With insider threats internal sensors are required. However, where should such sensors be positioned and to what extent, as the volume of data multiple internal sensors collect will quickly overwhelm analysis capabilities. Additionally, should these techniques forward the data to a central repository or force the analyst to perform analysis on many remote systems. The collation of data on a central repository has the advantage of providing for correlation of activity which can rapidly identify attacks otherwise not detectable.
- **Changing normal activity** – Most data mining techniques are dependent on what is considered normal activity. The first difficulty is identifying what is normal. The second difficulty is the fact that normal often changes. This change may be from one time of the day to another, one time of the year to another, random changes throughout time, or gradual changes in the nature of usage by an organization. Normal can also change more drastically during a crisis. These changes can easily cause the detection techniques to fail. More critically, attackers can slowly inject elements into the network in order to

change the nature of normality over time and ensure that their attack when finally initiated will go undetected.

2.4 Overriding problems of intrusion detection systems

Much of the problems inherent in these techniques derive from a lack of complete understanding of the underlying data. For instance, simple changes in a data attack can bypass detection even if the change is meaningless as far as the network protocol or attack strategy is concerned. Similarly, attacks can be hidden within the morass of normal activity.

Thus, intrusion detection systems currently suffer from the fact that too much data is being collected of which much isn't relevant to intrusion or attack detection. Consequently, sophisticated attackers can use this excess data to hide *their* attack. Ultimately, rather than collecting all data we must be more selective in which data is collected. This requires a better understanding of both the underlying data as well as attacks themselves. The understanding of attacks must be such that new attacks, i.e. zero day attacks, can not bypass our reduced data collection functionality.

3 Expanding Data Collection

Theorem 2: Current selective data collection strategies are inadequate and more complete collection strategies are necessary. This requires more fundamental research into identification of what data elements need collection in order to reduce the amount of data needing to be transferred and collated on a central repository.

3.1 Current data sources

Intrusion detection research has examined a wide range of data sources in the attempt to identify anomalous or malicious activity. These data sources are generally considered independently with focus on identifying specific types of anomalies. Given the complexity of many of today's sophisticated attacks, potential ramifications of insider threat, concerns with cyber terrorism, and fear of the use of cyber warfare as a form of asymmetric warfare, there is a definite need to integrate more of these data sources into a single intrusion detection strategy. This will provide more avenues for identification of the attacks. However, the additional data greatly increase the complexity of the analysis. Current data sources include [10]:

- **Network traffic data** – Network traffic data includes typical packet information and/or network flows. It is network traffic data that creates the bulk of the volume of data needing analysis. Generating multi-gigabyte data sets is trivial and terabyte datasets is not unusual.
- **System log files** – With system log files we are referring to typical text readable files generated by systems. Such files can include: snort alert logs, message logs, security logs, email logs, router logs, web logs, database logs, etc. The usefulness of these logs varies. However, it is quite clear that most insider threats can not be detected without the aid of some form of system based data. Identification of contrasts between system log files and network traffic data is one mechanism of rapidly identifying intrusions or otherwise anomalous activity.
- **Firewall logs** – Firewall logs could be considered specialized system logs. These logs include details related to blocked and accepted activity. Firewall logs are important as initial indicators of attacks. The logs also aid in fine tuning of the firewall itself by identifying what and what is *not* blocked and allowed through.
- **System statistics** – System statistics provide information such as system load, disk usage, etc. Such information can rapidly identify anomalous usage of a system.
- **Router logs and transmittal statistics** – Routers collect information as to the connectivity they are seeing as well as statistics on how much data is being transmitted across each of its ports/links. This data can be useful for identification of network health, network misuse, worm spread, etc.
- **Process accounting logs** – Process accounting logs identify the processes or commands a user has executed. This is often a first line of defense in the identification of insider threats, especially when masquerading is incorporated.
- **Library function accounting logs** – As it is a fairly simple matter to cloak the actual processes being executed by changing the name of the process and adding in non-functional code, monitoring the functions that a process executes aids identification of the real purpose or functionality of a process [7].
- **Biometric data** – Biometric data as it relates to intrusion detection refers to data collected as to users' behavioral patterns in their fundamental use of the computer hardware; the keyboard and mouse for instance [1] [2] [13]. This biometric data in essence attempts to ensure the validity of the systems physical security.
- **IDS alerts** – Given the wide range of capabilities of IDS tools, we must assume that these alerts will provide value in terms of identifying anomalous activity that could be of assistance to the analyst in focusing their initial efforts.

Generally, administrators only collect a fraction of the amount of data available on their network and often of only limited types. The data typically collected is exemplified in the first row of table 1. Thus, in this format the majority of the data collected is simply network traffic data from the border router and limited system log data from critical servers. This data collection metaphor is exemplified by figure 1. While several of the servers lie in the demilitarized zone, the data must still be collated on a central repository for integrity, correlation, and analysis. This is the typical paradigm for intrusion data collection [16].

This limited data collection greatly limits the ability of administrators and analysts to detect attacks and intrusions. When anomalous activity *is* detected analysts and administrators will often begin collecting or monitoring additional data sources in order to aid identification of the source and nature of the anomaly.

In terms of current research, most research focuses on the analysis of a single source of data. Currently, each data

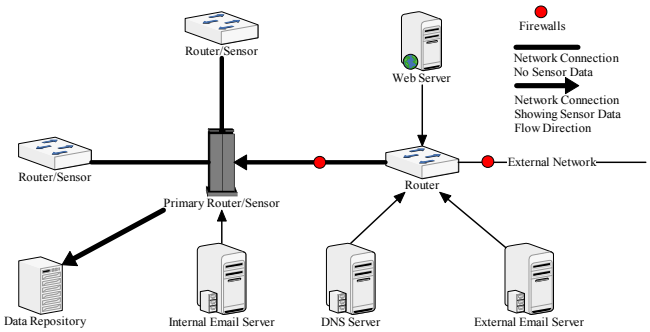

Figure 1: Basic data collection scheme. Typically, organizations will collect network traffic data from a border router and syslog data from servers. No other data is generally collected.

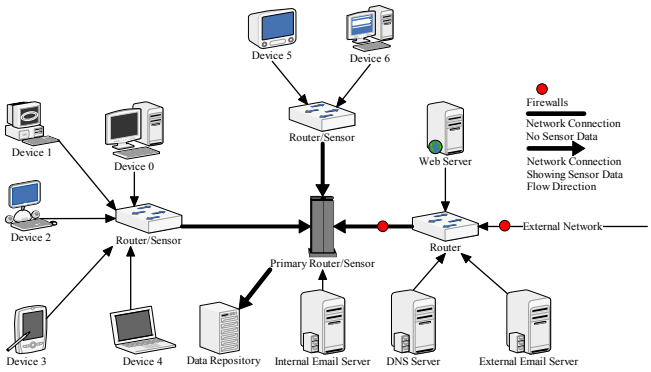

Figure 2: Proposed expansion in which data from all sources are collected and collated on a central repository. In addition to network traffic data and system log files from critical servers, process and library accounting logs are collected.

source is sufficiently challenging to develop needed techniques for that while researchers have examined hybrid data source approaches such research has been limited. However, the potential advantages of correlating wide ranging data sources to greatly improve detection capabilities imply it is a route that must be examined.

In order to improve the detection ability of analysts and administrators it is necessary to collect far more data than currently is being done. In particular, in the second row of table 1 we show a fuller range of potential data source to be collected and the extent to which these additional data sources will aid in the detection of additional potential attacks, intrusions and compromises. The full range of data sources matches the list of potential data sources described previously. The goal with such a complete range of data sources is to limit the range of vulnerabilities, such as those identified in [12], and enable identification of insider threats, novel attacks, and more insidious intrusions than is currently possible.

This second metaphor is exemplified by the network configuration in figure 2. Ultimately, the interconnections within this setup are identical to that in figure 1. However, in this setup far more data is being collected, both from the already monitored servers and also from additional system and routers currently unmonitored. For example, currently desktop systems are unmonitored and network traffic data is not collected from the internal network. By collecting these additional data sources, the administrator or analyst can greatly improve their detection ability. In essence, the goal is to provide some of the functionality urged in [14]. The data can be acquired directly, indirectly, or through a combination of methods. Acquiring the data directly implies monitors integrated into the application [3]. The network traffic data similarly can either be acquired by the routers or

	Firewalls	**Servers**	**Routers**	**Managed Switches**	**Desktop Systems**	**Enterprise Wide Detection Ability**
Basic Data Collection Setup	Firewall Logs	System Log Files	Router Logs	Network Flows (Packet headers) Border Network Traffic Data (Packet headers and payloads, Mirrored Switch)	-	External network attacks Externally controlled Systems Network misuse (peer to peer) Basic compromised servers
Complete Data Collection Setup	Firewall Logs	System Log Files *Process Accounting Logs* *Library Function Accounting Logs* *System Statistics*	Router Logs	Network Flows (Packet headers) *Data Transmittal Statistics* *Network Traffic Data (Packet headers and payloads, Mirrored Switch)*	*Process Accounting Logs* *System Log Files* *Library Function Accounting Logs* *System Statistics*	External network attacks Externally controlled Systems Network misuse (peer to peer) *Sophisticated compromised servers* *Sophisticated compromised desktops* *Computer misuse* *Insider threats* *Attacker migration* *Internal DOS attacks* *Internal network health and needs* *Identification of vulnerabilities* *Masquerades* *Spoofed/duplicate IPs*

Table 1: Basic (typical) data collection paradigms compared with the proposed data collection paradigms and the advantages of the proposed technique. Detection is considered at the enterprise level and does not consider detection within individual subnets.

from individual hosts. We do argue that the information from these multiple hosts must be correlated in order to identify many sophisticated attacks.

Often this type of arrangement will be performed through mobile agents controllable by the administrator. These mobile agents will be responsible for identifying what information needs to be collected based upon rules specified by the administrator and transmitting the appropriate elements to the data storage system for correlation with other available data [4]. These agents would need a level of cooperation to avoid data duplication. In our model, these agents are primarily responsible for collecting and preprocessing data. In other models these agents can be responsible for full data analysis [5] [6]. This prevents the need for data transmittal but can not provide the correlation necessary to identify the most sophisticated attacks. It is because of this need for correlation as well as the need for integrity that techniques such as proposed in [15] are insufficient; such Internet scale techniques require identification of attacks at the local level before being passed on to a higher level. For instance, as soon as a system becomes compromised its associated agents become unreliable. Much of the data that agents are supposed to avoid transmitting by performing the analysis themselves will again be needed to detect that said agent has been compromised.

3.2 Scalability issues

The desired data collection scheme discussed above creates enormous scalability issues. While collecting network traffic data from the border router creates more than enough data to make analysis an extreme challenge, collecting internal network traffic data will significantly explode the amount of data needing analysis. While log files, process accounting logs, systems statistics, etc. will not increase the sheer volume of data on a comparable level it will acquire significant amounts of new data which will likely go unanalyzed without significantly more effective tools.

This volume of data needing to be collated onto a central repository will create a misuse of the internal network; we will in essence be creating a denial of service attack against our own network. While the value of this additional data collection was shown, the negative impact of this data collection paradigm requires that new capabilities be developed to support this data collection stream. Polla et al. [11] have discussed needs with respect to architectures for such an implementation and have proposed one such implementation. However, even with an effective architecture there will be a great need for new techniques to handle (reduce) the volume of data being injected into the architecture.

If we are to be able to effectively collect and collate the full range of data sources available and perform the full range of detection and analysis capabilities current research is attempting to achieve, then we must examine techniques for getting this data to the central data repository. Additionally, we require new techniques to perform the basic analysis. As it is we have five areas which make such data collection and thus effective intrusion and attack detection completely unfeasible:

- **CPU usage and data preprocessing** – CPU usage is likely the least concern. Here we are referring to the CPU usage solely associated with the data collection process on the remote hosts. Acquiring, processing, storing, and transmitting system log files will take effort from each remote system, i.e. desktop systems; however, this CPU usage will likely not be substantial but may be of concern to individuals, especially those with existing high compute needs.
- **Disk storage space requirements** – Given that the network traffic data collected by many organizations easily achieves the terabyte range, there can be enormous costs associated with the collection and storage of sufficient data sets. Even our small data sets easily reach the 2-50 GB size depending upon current network load.
- **Network communication requirements** – The data collected on individual hosts and routers must be transmitted to a central repository for correlation, unless of course a more effective means of processing that does not require full transmittal can be developed. This transmittal can cause an enormous bottleneck on the network, especially when network traffic data collected on internal routers must be correlated for insider threat detection and analysis.
- **Data analysis requirements** – The resulting volume of data places great demands on analysts attempting to examine it. Currently it is unfeasible to expect all the data we desire to be collected to be fully analyzed.

As it is, many organizations will collect enormous volumes of information, either as a one time process, a repeated process, or a cyclical process. However, at this stage the extent to which this data is used is essentially to announce that they have acquired *N* terabytes of data but be unable to analyze it.

4 Data Reduction Needs

Theorem 3: The volume of data currently being collected and which we propose to be collected must be greatly reduced. Rather than collecting all available data from a particular source we need a more fundamental understanding of the characteristics and classifications of the underlying data such that the data can be selectively reduced before collection and transmittal to an analyst's workstation. While taxonomies have been created, as in [10], more detailed analysis is needed.

4.1 Scalability issues of data collection

In the previous section we saw the scale of data needing to be collected in order to perform a complete monitoring as well as attack, insider threat, and intrusion detection of the entire network. Dealing with this volume of data requires new paradigms in data collection processes. Typically, processes are fairly rudimentary, as exhibited in figure 3.

In this traditional paradigm, data collection is based on an administrator's hard gained comprehension of network vul-

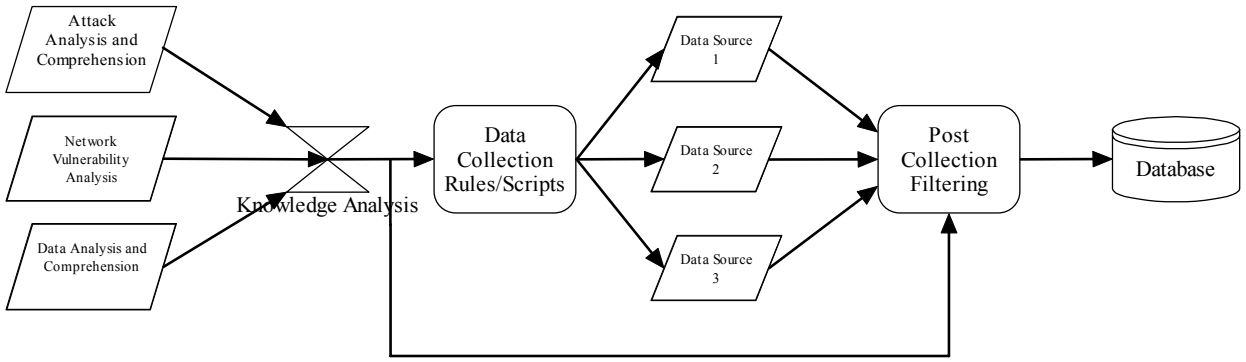

Figure 4: Our proposed set of processes for more effective data collection. Of critical importance is the development of the knowledge basis one left most portion of the process. With respect to vulnerability analysis we aren't creating the knowledge base directly but creating the facilities for administrators to more effectively perform and collate vulnerabilities within the network as well as identify how to *control* these vulnerabilities.

nerabilities. In many cases this knowledge is based solely on the configuration of border firewalls and border intrusion detection system. This implementation lacks knowledge of actual internal system configuration and completely avoids issues of insider threat. This configuration is a typical configuration; there are always exceptions to the rules. For instance, department of defense organizations provide a much more complete process and much more controlled environments. This configuration is severely limited in terms of the effectiveness of the environment. As we are interested in collecting far more data we need to develop more complete and effective data collection processes, figure 4.

The more advanced process proposed in figure 4 is designed to provide the capabilities needed to collect the volume of data we are proposing. Of note is the need for additional capabilities to handle this volume of data, for instance:

- **Data classifications and categorizations** – A more detailed classifications of available data sources is needed such that better judgments can be made as to which elements require collection.
- **Attack Comprehension and Select Parameter Acquisition** – Identification of attack characteristics and parameters to aid identification of which data elements require collection but which parameters of those elements are needed.
- **Vulnerability analysis capabilities** – Tools are needed to better identify system vulnerabilities throughout a network, based on known system type such that administrators can better control these vulnerabilities.
- **Data collection recommendations** – Given the range of available data sources we are proposing, recommendations are needed as to the extent of data collection needed for a given risk mitigation level or legal remediation capability.
- **Performance vs. risk mitigation recommendations** – Similarly we must provide recommendations as to how to tradeoff performance vs. risk mitigation.
- **Scalable Data Analysis Techniques**

Through the analysis of attacks and fundamental data the need is to better identify what data elements must be collected. This is in contrast with today's typical paradigm in which either nothing or everything is collected. For instance, through this research we expect the data collection tools to be able to make determinations as to what network packets validly and intrinsically belong to accepted network connections and thus do not need to be collected for attack or insider threat detection. Similarly, we expect much more in the way of system log files to be collected with the collection tools able to determine which entries can be rejected without collection. With respect to post filtering, the goal is to identify the specific parameters that need to be collected and eliminate those that aren't needed. Thus, when examining network packets, rather than collecting entire packets we must be more selective with respect to what portions (parameters) of the packets we collect. For instance, in most scenarios the payload of a packet itself is useless; for instance if the packet is encrypted. However, the fact that the

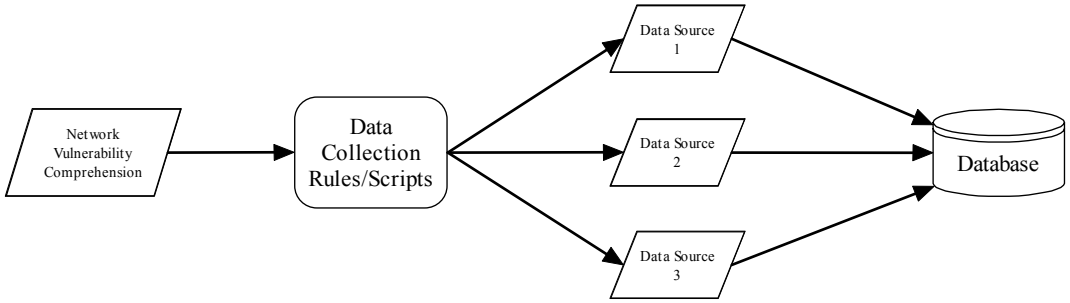

Figure 3: Simple data collection process typically implemented by system administrators. Data collection scripts are based on an administrator's knowledge of perimeter security and limited knowledge of internal security and vulnerabilities.

packet was encrypted would be of valuable. Similarly, in the case of buffer overflow attacks the contents of the payload would be of enormous value and such payloads should not be filtered. In this fashion the goal is to identify precisely what needs to be collected and what does not. This data collection paradigm will have enormous advantages:

- Allow collection of all available data
- Provide for maximum identification of attacks, intrusions, and insider threats
- Reduce data storage on each machine within the network
- Reduce network traffic in transmitting data to a central repository
- Improve analysis efficiency due to the minimization of insignificant data

Even though there are significant advantages there are disadvantages of this proposition. The primary disadvantage is the extra CPU cycles required on each system to perform the data collection scheme. However, in most environments the power of today's systems are rarely used and thus the majority of individuals would not notice any impact. Given that the bulk of the data, i.e. network traffic data, is only collected on routers or mirrored switches, the small amount of data collected on desktop systems should not have a significant impact on such systems.

4.2 Data Classification and Categorization

Analysis is needed of the underlying data itself, particularly in relation to the original networking RFCs. If we can classify and categorize legal network traffic data at a fundamental level we will have more of a basis for identifying data elements that need to be collected and those that do not. This is not saying we will be able to precisely eliminate *all* data related to valid network traffic. We will still be collecting data deemed anomalous, not necessarily an attack. We can essentially identify what combinations and sequences of flags and options might be valid which will greatly aid identification of anomalous event streams that do not match expected sequences at a fundamental level. However, there are many instances in which innocuous activity does not match valid network traffic sequences. For instance, many operating systems will implement protocols differently even if the slight deviation in implementation may not precisely implement the network protocol as specified in the RFCs.

For example, consider the typical connection/disconnection process associated with TCP. Generally, the connection and disconnection process consists of the following:

Connection:	**Disconnection:**
Syn→	Fin →
← Syn + Ack	← Ack
Ack →	← Fin
	Ack →

We have observed scenarios in which the termination with certain operating systems will exhibit an additional Ack. Technically, this is incorrect and this extraneous Ack would appear to be unassociated with any connection. This example in fact provides an example of a source of false alarms as far as port scan identification goes.

This is where an extensive understanding of the fundamental data and an ability to characterize the data will greatly aid identification and analysis of the data very early in the process in order to determine if the data is truly part of an attack and needs to be collected or not. Again, the goal here is to provide a better comprehension of network traffic data at a fundamental level in order to identify what data must be collected such that the volume of data being collected can be greatly reduced.

4.3 Attack Comprehension and Select Parameter Acquisition

In direct association with the analysis of underlying data at a fundamental level, we must examine attacks at a fundamental level. Here we are particularly concerned with sophisticated attacks, as opposed to naïve attacks; though both are of value. Understanding attacks at a fundamental level is both a separate task from data comprehension and directly complimentary. They will ultimately be used in a tightly integrated fashion but we must have this level of understanding. By understanding attacks we will have a better comprehension as to:

- How attacks in general are exhibited in the data [8]. This will allow us to identify the event streams related to attacks and which are not.
- How sophisticated attacks are exhibited differently from naïve attacks. This will allow control over which data elements and what level of attacks we focus our analysis on. Since naïve attacks are often used to cover sophisticated attacks we may be more interested in the basic fact that it exists with the details not being relevant.
- What the potential ramifications of a specific attacks are. This will allow for identification of the potential severity of the attack and the priority the attack should be given.
- How the steps in a given attack will follow on. This will allow for immediate deployment of counter measures and protections in order to prevent the specified attack from having significant negative impacts.
- What parameters are important and which are not.

This will greatly aid our ability to control what data elements and more importantly data parameters are collected. Even the ability to identify which portions of a network packet need to be collected and which do not can have potentially enormous impact on the volume of data collected. This becomes particularly important if we can determine when the payload needs to be or does not need to be collected.

In addition to the identification of overall collection or removal of data elements and data parameters we will also need to examine summarizations of data components. This can also have significant ramifications on the volume of data needing collecting. For example, merely indicating that the payload is encrypted or identifying the fact that the payload contains a known buffer overflow code sequence is sufficient. This allows identification of the critical characteris-

tics of the data elements without the need for storing the entire data stream which is what leads to our scalability issues. In essence we can consider this to be performing a portion of the analysis as a pre-storage step. This of course does bring up the potential ramifications of such preprocessing in terms of the computational costs. However, the potential benefits in terms of data reduction could be enormous.

4.4 Vulnerability analysis capabilities

As mentioned previously, currently system administrators rarely know the extent or characteristics of vulnerabilities within their own network; aside from vulnerabilities at the perimeter. This lack of knowledge greatly inhibits their ability to refine their data collection and analysis methodologies. This essentially requires knowledge of which OS is running in conjunction with which network ports are open, i.e., which network services are running. While there are tools to identify certain such activities, such as through nmap or ISS, more manageable tools that work in conjunction with the data collection strategy are necessary. For instance, if you can imagine administrators attempting to manage potential vulnerabilities on several thousand machines simultaneously, the task quickly becomes unfeasible.

4.5 Data collection recommendations

Given a set of potential vulnerabilities within the network as well as potential limits on performance impact, risk mitigation, and need for legal remedies, there is a need for sets of recommendations as to what extent data needs to be collected. Given these recommendation sets, i.e. graphs of tradeoffs between performance impact, risk mitigation, attack detection probability, and legal liability, administrators should have the ability to make accurate decisions as to how much data and of what type to collect. On the administrators end determinations will essentially be made based on these characteristics as well as their expected cost of implementing the recommended data collection paradigm. This will reduce the cost of the data collection paradigm by reducing the volume of data that will ultimately be collected.

4.6 Scalable Data Analysis Techniques

Even with effective pre-filtering of the data we will not be able to completely forego the need for scalable data analysis techniques. As networking bandwidth increases, the size of internal networks increase, and new devices are added to the network, the volume of data will only continue to increase. Similarly, the number of attacks on a given network continues to increase at a phenomenal rate. While this provides further evidence for the development of such fundamental capabilities, as we have discussed here, it also makes it clear that we need to continue to explore scalable analysis techniques. The goal at one level will be for *effective* data reduction techniques to meet the data transmittal needs while the scalable data analysis techniques meet the processing needs. Scalable techniques have focused on all avenues in an attempt to deal with the volume of data, such as:

1) High Performance Computing

The goal of high performance computing is essentially to throw massive amounts of CPU cycles at the problem. High performance computing can either refer to specialized multiprocessor systems or cluster systems. The issue here is whether it is feasible to expect general administrators and analysts to have the CPU cycles available in a typical environment to deal with their volume of data.

2) Data Filtering

In this context we are referring to the more arbitrary data filtering applied by administrators as a preprocessing step. Administrators will arbitrarily identify streams or characteristics not to be analyzed in order to reduce the volume of data. Often administrators will only collect selective data, i.e. when unusual activity is already identified. These forms of data filtering aid in data reduction but often eliminate critical data associated with attacks as well as innocuous data.

3) Algorithm Enhancement

Essentially, algorithm enhancement is the process of identifying improvements in existing techniques, both in terms of effectiveness as well as efficiency.

4) Algorithm Development

Rather than single steps in the improvement of techniques, the development of completely new algorithms attempt to make leaps forward with respect to the analysis of data. Such algorithm development is extremely challenging and the development of new successful algorithms occurs infrequently.

5) Visualization

Visualization techniques take an alternative approach to the analysis of data. Rather than performing an extensive computational analysis of the data, the visualization algorithms will perform transformations on the data to convert it from a raw form to a graphical form. This graphical form relies on the human visual systems ability to perceive anomalies and trends, especially in conjunction with an analyst's domain knowledge.

5 Conclusions

We have proposed collecting far more data sources than can currently be analyzed in a realistic fashion. To make this data collection paradigm feasible we have identified the need for additional capabilities for the reduction of the amount of data from each source that actually needs to be collected. Additionally, by improving our comprehension of the underlying data and attacks we will be able to create far more powerful and effective analysis tools. For instance, we discussed some of the weaknesses of typical intrusion detection techniques in which attackers can make trivial changes to known attack sequences in order to fool most intrusion detection sequences. Improvement of our comprehension of how attacks are exhibited at a more fundamental level will prevent this from occurring. Additionally, it will allow for the development of far more effective techniques.

From a managerial perspective we must have the ability to present analysis relating data collection strategies with

performance impact and associated risk. It is this analysis of risk that alone will greatly aid organizations in identifying a data collection scheme. Thus, we will be able to associate data collection schemes either with desired levels of performance, desired levels of risk, to ensure identification of specific types of attacks, or any combination of these needs.

While we have not examined current research; existing research in the areas of high performance computing, distributed storage, networking, stimulus response, and data compression can aid in reducing but not solving the identified problems. These are broad and abstract directions of research which have direct applicability to the identifies challenges. Clearly, there are numerous other research activities which will provide benefit to this proposed direction of research.

6 References

1. A.A.E. Ahmed and I. Traore, "Detecting Computer Intrusions Using Behavioral Biometrics," *Proceedings of the Third Annual Conference on Privacy, Security and Trust,* October 2005.
2. A.A.E. Ahmed and I. Traore, "Anomaly Intrusion Detection based on Biometrics", *Proceedings of the 6th IEEE Information Assurance Workshop*, pp. 452- 453, June 2005.
3. M. Almgren and U. Lindqvist. "Application-Integrated Data Collection for Security Monitoring," *Proceedings of Recent Advances in Intrusion Detection* (RAID), pp. 22-36, October 2001.
4. Joseph Barrus and Neil C. Rowe. A distributed autonomous-agent network-intrusion detection and response system. *Proceedings of Command and Control Research and Technology Symposium*, pp. 577–586, 1998.
5. M. Eid, H. Artail, A. Kayssi, and A. Chehab, "AN ADAPTIVE INTRUSION DETECTION AND DEFENSE SYSTEM BASED ON MOBILE AGENTS," *Proceedings of the 2004 International Research Conference on Innovations in Information Technology*, pp. 108-116, 2004.
6. Rajeev Gopalakrishna and Eugene Spafford, "A Framework for Distributed Intrusion Detection using Interest-Driven Cooperating Agents, *Proceedings of the Fourth International Symposium on Recent Advances in Intrusion Detection* (RAID), October 2001.
7. S. A. Hofmeyr, S. Forrest, and A. Somayaji, "Intrusion detection using sequences of system calls," *Journal of Computer Security* (JCS), Vol. 6, pp. 151–180, 1998.
8. M. Y. Huang, R. J. Jasper, and T. M. Wicks, "A large scale distributed intrusion detection framework based on attack strategy analysis," *Proceedings of the Intl. Symp. on Recent Advances in Intrusion Detection* (RAID), pp. , 1998.
9. A.K. Jones and R.S. Sielken, "Computer System Intrusion Detection: A Survey," University of Virginia, Computer Science, *Technical Report*, 2000.
10. A. Lazarevic, J. Srivastava, and V. Kumar, "A Survey of Intrusion Detection techniques", chapter 2 of *Managing Cyber Threats: Issues, Approaches and Challenges*, Kluwer, 2004.
11. D. Polla, J. McConnell, T. Johnson, J. Marconi, D. Tobin, and D. Frincke, "A FrameWork for Cooperative Intrusion Detection," *21st National Information Systems Security Conference*, pp. 361-373, October 1998.
12. Thomas H. Ptacek and Timothy N. Newsham, "Insertion, evasion, and denial of service: eluding network intrusion detection," *Technical report*. Secure Networks Inc., January 1998.
13. Maja Pusara, Carla E. Brodley, "User re-authentication via mouse movements Full text," *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, pp 1-8, 2004.
14. Eugene Spafford and Diego Zamboni, "Data collection mechanisms for intrusion detection systems," *CERIAS Technical Report 2000-08*, CERIAS, Purdue University, 1315 Recitation Building, West Lafayette, IN, June 2000.
15. Lawrence Teo, Yuliang Zheng, and Gail-Joon Ahn, "Intrusion Detection Force: An Infrastructure for Internet-Scale Intrusion Detection," *Proceedings of IWIA*, pp. 73-88, 2003.
16. Michael Treaster, " A Survey of Distributed Intrusion Detection Approaches ," *ACM Computing Research Repository (CoRR) Technical Report cs.CR/0501001*, January 2005.
17. "IEEE Workshops on Visualization for Computer Security," *vizsec*, p. iv, IEEE Workshops on Visualization for Computer Security (VizSec'05, VizSec'06, VizSec'07), 2005-2007.
18. http://www.snort.org/