SECURITY METRIC BASED NETWORK RISK ASSESSMENT

A Thesis
Presented to
The Academic Faculty

by

Moazzam Khan

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
May 2013

SECURITY METRIC BASED NETWORK RISK ASSESSMENT

Approved by:

Dr. John A. Copeland, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. David C. Keezer
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Raheem A. Beyah
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Mustaque Ahmed
College of Computing
Georgia Institute of Technology

Dr. Henry L. Owen III
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: Dec 12, 2012

To my parents, my teachers and my friends.

ACKNOWLEDGEMENTS

I wish to thank office of information technology (OIT) for their support through the project. I want to thank all my friends who guided me with brainstorming sessions and technical and moral support when I needed them. Special thanks to Muhammad Omer, Aleem Mushtaq, Bilal Anwar, Azhar Hasan, Meer Adeel, Farasat Munir, Selcuk Uluagac, Sungin Park, Shiin Chang. My professor and my co advisor, Dr. John A. Copeland and Dr. Raheem A. Beyah were instrumental in this project, without their supervision this work would not have been possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Objective of this research is to identify the probability of attack on a communication infrastructure. A communication infrastructure becomes prone to attack if certain elements exist, such as vulnerabilities in the comprising elements of the system, existence of an attacker and motivation for an attacker in the system. We initially focus our study on vulnerability assessment and break down our analysis based on certain security metrics such as user behavior, operating systems, user applications, updates etc. In order to achieve a quantitative value of risk we observe a finite set of machines with the matrices that we have identified for this study. We collect the historical data of machines that have these matrices and have gotten compromised and apply statistical analysis on this data to predict a quantitative value of risk for systems analyzed later. We present analytical results by comparing the collected metric statistics visually via graph. For a thorough mathematical analysis we apply machine learning and statistical approaches such as Bayesian Regression, Multivariate Regression, Neural Networks, Decision Theory, Clustering etc. on the collected matrices.

CHAPTER 1

INTRODUCTION

Modern day computer networks have become very complex and attackers have benefited due to this complexity and have found vulnerabilities and loopholes in the network architecture. In order to identify the attacks from an attacker all aspects of network architecture needs to be carefully examined such as packet headers, network scans, versions of applications, network scans, network anomalies etc. and after the examination attributes playing a significant impact on the security posture of the organization needs to be highlighted so that resources and efforts are directed towards those attributes. In this work we extensively look at network traffic at dormitory network of a large campus and try to identify the attributes that play a significant role in the infection of a machine. Our scheme is to collect as much attributes from the network traffic applying the heuristic of network infection and then devise a scheme called decision centric rank ordering of security metric that gives the priority to the security metrics so that network administrators can channel their efforts in the right direction.

Another aspect of this research is to identify the probability of an attack on a communication infrastructure. A communication infrastructure becomes prone to attack if certain elements exist in it, such as vulnerabilities in the comprising elements of the system, existence of an attacker and motivation for him to attack. Focus of this study is on vulnerability assessment and security metrics such as user behavior, operating systems, user applications, and software updates. To achieve a quantified value of risk, a

set of machines is carefully observed for the security metrics. Statistical analysis is applied on the data collected from compromised machines and the quantified value of risk is achieved.

Besides the use of the historical data, techniques from other domains are borrowed to predict risk in a communication system. The following schemes are investigated to quantify the network security.

- Machine learning schemes to identify the significant metrics.
- Game theory approaches to model the analysis as a game between an attacker and a defender, where an attacker only attacks when he has high probability of launching a successful attack.
- Financial market knowledge of risk calculation.
- Differential stochastic equation to model risk over a period of time.

The motivations for this work are diverse as emphasized in [1]. As the saying goes, "you can't control what you can't measure", a manager in higher position need to know quantified information either in the form of impact or associated cost. By identifying the probability of an attack a manager can take defensive measures to prevent the attack from being successful. For example, after applying the proposed technique you get a score of nine and after taking some preventive measures, such as installing a firewall, your score becomes three. This lowering of score shows that it is cost effective to take preventive measures to avoid considerable losses. Also, this technique would enable organizations to compare their security level with other organizations and sell their products more effectively, because customers would be able to trust their security. For example, if a

bank gets a rating of three compared to another bank with a rating of seven, then you would feel safer doing business with the first bank rather than the second bank. Another very big motivation for quantifying security is for the insurance companies that want to get a better picture about the security of a system, but unlike auto or life insurance they have no way of quantifying the security level of a communication network.

In summary, this study would enable the communication-network community in general and security community in particular to predict the probability of attacks with a certain confidence level, thus enabling them to take effective measures to prevent those attacks.

Chapter two discusses work that has been done towards identification of security metrics and quantification of network security. We highlight the areas where our research differs from the state of art, and the contributions made. In chapter three we streamline the identification process for the security metrics. In this chapter we highlight the standards used for collections of metrics, commonly used metrics in the industry, and the set of metrics that we identified from the network traffic. Chapter four presents an in depth analysis of network traffic and presents statistics on the security metrics identified in chapter three. In chapter five we present an analytical approach to gauge the effectiveness of the identified security metrics by in depth analysis of infected and non-infected profiles. For this purpose we look at user browsing behavior and user update behavior and examine all the identified metrics that fall under these classifications. Our analysis compares these metrics by comparing the metrics of infected machines identified by BotHunter, which is dialogue based malicious traffic identification engine, with the metrics of not infected machines. Chapter six presents a scheme called decision centric rank ordering (DCRO) that identifies the most significant metrics and prioritize them for

the network administrators to efficiently allocated resources. Chapter seven explores schemes that can utilize the identified metrics in order to predict the future risk. We compare the results of regression, decision tree, classification and clustering schemes and identify schemes with the highest prediction value. Chapter eight concludes by identifying the areas that require further investigation and suggest improvements that can be done in the future work.

CHAPTER 2

IMPORTANT CONCEPTS AND PREVIOUS WORK

In this chapter we explain some basic concepts that will be used throughout the document and highlights the previous work in the area of network security metrics, user behavior and network traffic characterization, threat intelligence and risk analysis.

1. Important Concepts

1.1 Asset

An asset is any resource that is critical to an organization and the organization wants to protect it. It could be a computing resource, hardware, software or people. Loss of an asset could impact the C.I.A (Confidentiality, Integrity and Availability) of the organization, and it could also result in monetary loss.

1.2 Threat

Threat is any event that could is harmful for the confidentiality, integrity and availability of an organization. A threat could be manmade like a disgruntled employee, external attacker, rival organization, or it could be a natural threat like an earthquake, hurricane, etc.

1.3 Vulnerability

Vulnerability is a weakness in the assets or safeguards of a system that could make a threat highly likely. Vulnerabilities could be due to poorly written code, lack of test and validation, careless administration, neglect on updates etc.

1.4 Risk

In simple words, risk is the impact on the asset if a vulnerability results in materializing a threat. The risk is usually measured in the dollar amount of the loss that could result if an attack is successful to an organization.

1.5 Safeguard

A safeguard is the control or countermeasure employed to reduce the risk associated with a specific threat, or group of threats.

1.6 Security Metric

A security metric is a result of measurement process that can give us a good idea about the security posture of an entity. The entity could be an organization or it could be a piece of software. Using a security metric we quantize the security level of that entity.

1.7 User Behavior Characterization

This process characterizes the behavior of network users in terms of sites they visit, frequency of their visits, and application and operating system updates. The reason for characterizing user behavior is that it could act as a very good security metric. For example, a user who is careless in his browsing and visits sites infested with malware is prone to get compromised, so the type of sites visited is a good security metric.

1.8 Network Traffic Characterization

Similar to user behavior characterization, we can characterize network traffic to identify good security metrics. For example can observe the normal network traffic at certain times of the day and if we observe a deviation from normal network traffic this could

indicate an attack going on the network, thus attributes identified via network traffic characterization can serve very good security metrics.

Efforts toward the quantification of security had started as early as 1980 when people tried to borrow ideas from computer science, economics, statistics, reliability theory and other disciplines. Some of the early research in quantification area is [2], [3], [4], [5], [6], [7]. The work done ranges from defining metrics [19], [20], [21]; defining security with approaches such as attack graph [8], [11], [12]; applying artificial intelligence [9], [10], [16]; statistical techniques to determine security of the network [13]; determining the weakest link in the network; figuring out the aggregate security [14], [15]; and quantifying network threat level [17]. Some researchers have even tried to question the merits of quantifying the network security [22]. The objective of all this research is to determine a predictable way to assess the security level of a network. This chapter tries to highlight some of the prominent schemes of security quantification and their drawbacks.

2. Background of User Behavior and Network Traffic Characterization

There have been many studies in the past that try to characterize network traffic but they are more network centric than end-user based. Almost all those studies have tried to find out statistics such as the mobility patterns of users on a campus network, application usage, and network usage over a particular period of time. Their focus is not to develop a profile of each user that can help determine the security posture of the network. The most closely related work that tries to characterize user behavior in terms of security was found in [62]. In this paper they try to see if users are properly securing their machines when they connect to a wireless network. They checked what percentage of machines is using properly configured firewalls, what percentage has ports open, and what percentage have

a serious vulnerability associated with any open ports. Another study [63] analyzes data from an extensive set of machines and end users. Their basic objective is to find the trends in the wireless network usage over the years. They determined that in the beginning Web traffic dominated the wireless networks, and the new trends show increases in the peer-to-peer traffic, streaming multimedia, and VoIP traffic. Authors in [64] studied the traces from a public LAN with the objective of characterizing user behavior such as connection session length, mobility of user, protocol distribution and access point usage. The objective is to better understand the issues in wireless networks and optimize the network. There are studies that take more targeted approach such as [65] which looks at the Web sites that network users visit and try to classify them into malicious Web sites, and claims to get 95 – 96% accuracy of prediction. Studies like [66, 69, 70, 71, 72] take a higher-level view of the user behavior and target things users do in a Web 2.0 such as social networking, photo sharing sites, and online games. They tend to characterize traffic based on usage pattern, popularity of use, classes of users etc. Studies in [67, 68] looked at an ISP data and characterized behavior of two types of users, residential and small office/home office (SOHO) users. They modeled the distributions for session arrival, session duration, number of bytes transferred and user requests for each type of users. Authors in [73] characterize user behavior of three European countries for a period of over a year. Their investigated parameters are application types and file download services. For this purpose they looked into statistics of five classes of applications: File Hosting (FH), Streaming Services (SS), P2P services (P2P), Social Networking (SN) and other HTTP traffic (Web). They considered file sharing

applications BitTorrent, Emule, Megaupload and Rapidshare. Authors in [74] predicted demographic data such as age and gender from the browsing behavior.

This study not only characterizes network traffic of a campus dormitory, but also identifies attributes from user behavior that reflects the security state of a network and thus can be used to effectively measure the security.

3. Background of Security Metrics

3.1 Identification of Network Security Metrics

Early work in the network security quantification involved defining the security metrics. The study focused on objectives and properties of an ideal metric. Authors in [21] and [22] try to identify the objectives of an ideal metrics as follows:

- Focus scarce resource on pressing problems.
- Make a business case for needed change.
- Help spot problems or successes early.
- Address outside concerns or criticisms fairly and objectively.

Authors also define that an ideal metric should have the following characteristics:

- Motivate good/correct behavior (not promote evasive tactics just to make the numbers look good).
- Prompt additional questions ("Why? How?") to understand what is influencing the numbers.
- Answer basic questions of goodness (e.g., "Are we doing better or worse?").
- Be objective and measurable, even if correlation may not equal causality.

3.2 Measuring Network Security Using Attack Graphs

Figure 2.1 shows different directions that researchers have taken using attack graphs for measuring network security.

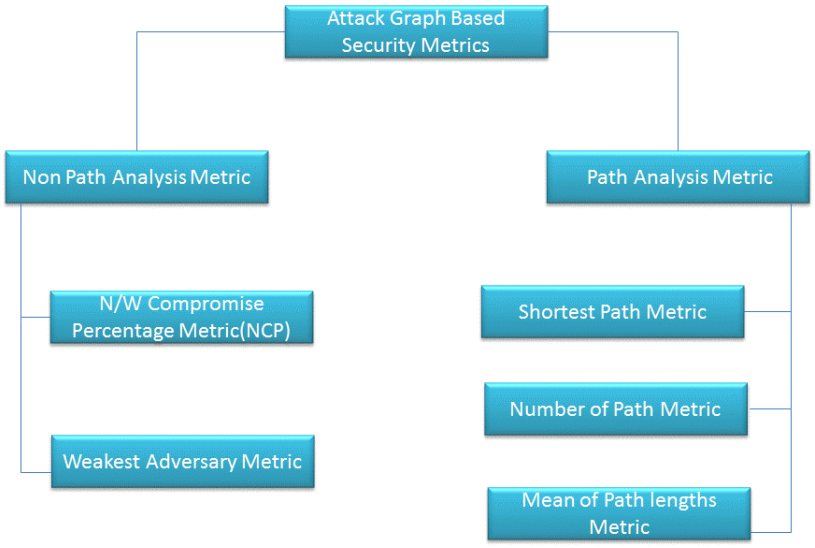


Figure 2.1: Attack Graph Schemes for Measuring Security.

Much effort has been made toward measuring the network security using attack graphs. An attack graph is a representation of all the possible paths that an attacker can take to enter a network. In other words, it is a graphical representation of all the vulnerabilities that exist in a system. Theoretically, an attack graph is defined in [23] as:

Definition 1: *Supposing V is the set of vulnerabilities in a network, an attacker breaks into the network through a chain of exploiting vulnerabilities, CP={v1 v2 ... vn| vi ε V, i=1,2, ...,n}, where each exploit in the chain helps to execute subsequent exploits. Such a chain, CP, is called an attack path.*

Definition 2: *Attack graphs are used to describe network security. To a network, the set of all possible attack paths from an attack graph, that is AG= {CPi |CPi is an attack path, i=1,2,....,m}*

3.2.1 Architecture of an Attack Graph System

Some papers such as [23] illustrate how an attack graph can be applied to measure network security. Their framework is presented in Figure 2.2.

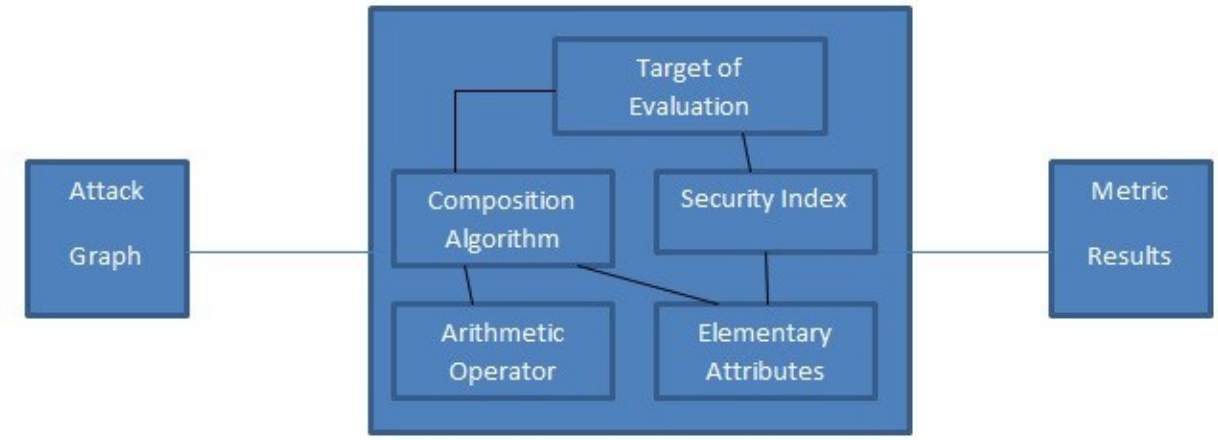


Figure 2.2 : Attack graph framework.

Different elements of the above framework are defined below:

3.2.1.1 Security Index (SI)
An SI is a set of measurable security attributes that indicate the network security level. Categories of security indices are integral security, probability of success, attack cost, and loss.

3.2.1.2 Target of Evaluation (ToE)
An object that would be evaluated by the framework and it is the entity to which the security index should be applied.

3.2.1.3 Elementary Attribute (EA)

The basic measurable security attribute, which is associated with atomic attacks and vulnerabilities, such as probability, time and, state.

3.2.1.4 Composition Algorithm (CA)

A set of well-defined rules to determine the resultant security metric of a specific ToE in a finite number of steps. These rules are based on attack graphs.

3.2.1.5 Arithmetic Operators (AO)

Arithmetic operators are the symbols or functions applied to the CA. The operands of AO are the values of the security attributes.

This paper also gives an example of how the framework can be applied to real-world networks and explains all the above mentioned elements. They consider an example of a news provider via webpages. The provider is concerned about the security of its webservers and want to know the likelihood of the webpages getting tampered with by a malicious attacker.

In this scenario, the ToA is the web system that hosts the webpages containing news so all the states related to the security of the web system in the attack graph are key states. The security index is the probability of gaining the write right to the webpages, thus the elementary attribute is the probability of executing an atomic attack successfully.

3.3 Attack Graphs with Metrics

In [11] the authors use the attack graph approach to define two metrics, namely a probabilistic security metric and an attack resistance metric to determine the security levels of different network configurations. These security metrics are defined in 2.3.1 and 2.3.2.

3.3.1 Probabilisict Security Metric (PSM)

A PSM quantifies the probability of successfully executing an exploit and measures the likelihood of compromising a network in terms of the number of attackers reaching the goal. In short, this metric can be used to measure the degree of security strength of a network configuration.

3.3.2 Attack Resistance Metric (ARM)

ARM defines the effort an attacker requires until he succeeds. Hence, the higher the resistance of an exploit, the more secure is the system.

All the approaches discussed earlier only consider a fixed attack graph, but in a real-time network, certain temporal elements also play a part such as availability of exploit code or the patch for some vulnerability that existed before. For such cases [24] presents a dynamic Bayesian approach that continuously measures the network security in a dynamic environment.

3.3.3 Limitations

There are tools that try to find graphs covering all the network vulnerabilities in a system but due to temporal and spatial limitations it is extremely difficult to have a comprehensive graph of system vulnerabilities and their dependence on each other.

3.4 Common Vulnerability Scoring System (CVSS)

CVSS [25] and its version 2 [26] is an effort toward the standardization of quantization of network security. According to their official documentation it offers the following benefits:

- Standardized Vulnerability Scores

- Open Framework
- Prioritized Risk

The scores from the CVSS system are taken and used to devise schemes that can measure the network security of a system as explained in [28], [29], [30]. CVSS consists of three metric groups that are defined below:

Base Metric: Represents the intrinsic and fundamental characteristics of a vulnerability that are constant over time and user environments.

Temporal Metric: Represents the characteristics of a vulnerability that change over time but not among user environments.

Environmental Metric: Represents the characteristics of a vulnerability that are relevant and unique to a particular user's environment.

A detailed representation of three groups is given in Figure 2.3.

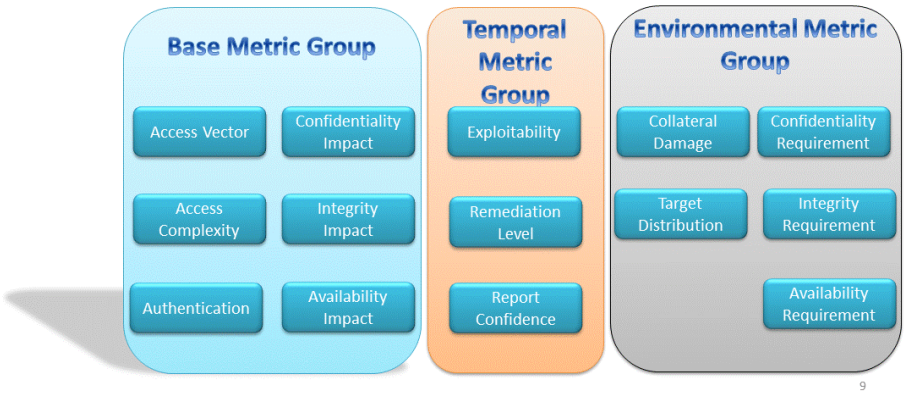


Figure 2.3: CVSS architecture.

3.4.1 How Does CVSS work

Figure 2.4 shows how the base metric is assigned a value and a vector is created. The score range from zero to ten. The vector communicates the score for each vulnerability, and for this reason vector is always displayed with the vulnerability. Temporal and environmental metrics are optional but can be used to refine the base score.

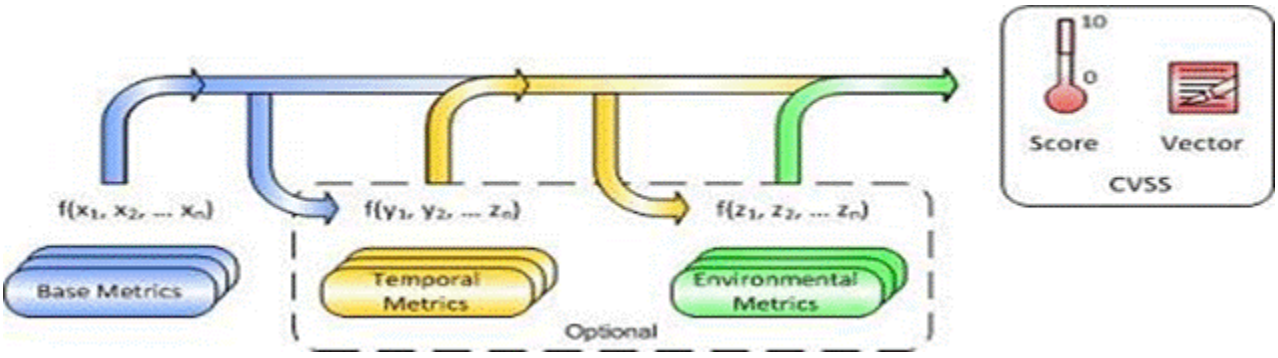


Figure 2.4: CVSS working.

3.4.2 Other Scoring Systems

Along with CVSS there are other standards that exist in the market.

Table 2.1: Other standards for vulnerability scoring.

1	Computer emergency response team /Coordination center (CERT/CC)	CERT/CC produces a numeric score ranging from zero to one eighty but considers factors such as whether the Internet infrastructure is at risk and what sort of preconditions are required to exploit the vulnerability.
2	System Administration, Networking, and Security Institute (SANS)	SANS vulnerability analysis scale considers whether the weakness is found in default configurations or client or server systems.
3	Microsoft	Microsoft's proprietary scoring system tries to reflect the difficulty of exploitation and the overall impact of the vulnerability.
4	US Department of	A threat rating system such as those used by the US Department of

Table 2.1 Continued

	Homeland Security	Homeland Security, and the SANS Internet Storm Center. These services provide an advisory warning system for threats to critical US and global IT networks.
5	National Vulnerability Database	A vulnerability database such as the National Vulnerability Database (NVD), Open Source Vulnerability Database (OSVDB) or Bugtraq. These databases provide a rich catalogue of known vulnerabilities and vulnerability details.
6	Common Vulnerabilities and Exposures (CVE)	A vulnerability identification system such as the industry-standard Common Vulnerabilities and Exposures (CVE) or a weakness dictionary such as the Common Weakness Enumeration (CWE). These frameworks are meant to uniquely identify and classify vulnerabilities according to the causes as they are manifested in code, design, or architecture.

3.5 Vulnerability and Impact Analysis

Another attempt at the quantification of network security and the prediction of the likelihood of an attack is [31]. In their efforts the authors take the following two approaches:

First, they measure the vulnerabilities in the services of the system. In this analysis they not only consider the present vulnerabilities but also the dormant risk because of past vulnerabilities.

Second, they consider the policy aspect of the security risk, which gives an idea of the impact of a successful attack on the network and the cost associated with such an attack.

According to the authors, a combination of vulnerability analysis and policy evaluation provides a framework that can help when comparing security policies with each other to determine the most secure policy. System architecture of the scheme presented is shown in Figure 2.5.

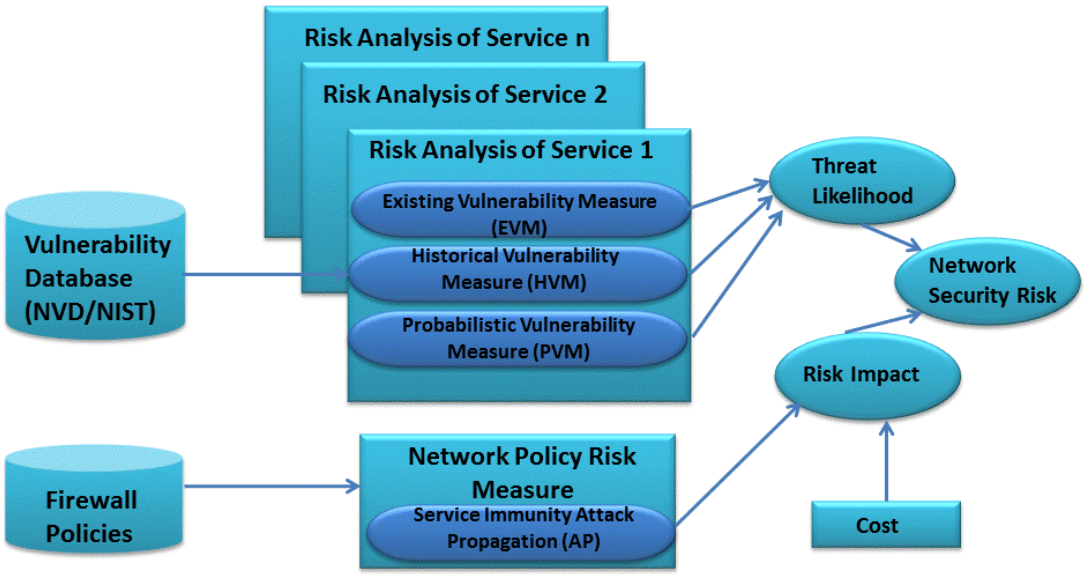


Figure 2.5: System Architecture for vulnerability and impact analysis.

3.6 Configuration Analysis for Risk Management

Another study [32] takes into consideration the flaws in a network configuration that renders a network vulnerable to attack. They present a metric called the VEA-bility security metric, which can help compare different network configurations and select the most secure one. The motivation given by the writers is that it is the objective of any network administrator to have the smallest number of vulnerabilities in his network, and to have every software and hardware system securely configured. The metric proposed by the authors enables different configurations to be compared with each other, thus helping in the selection of the most secure configuration.

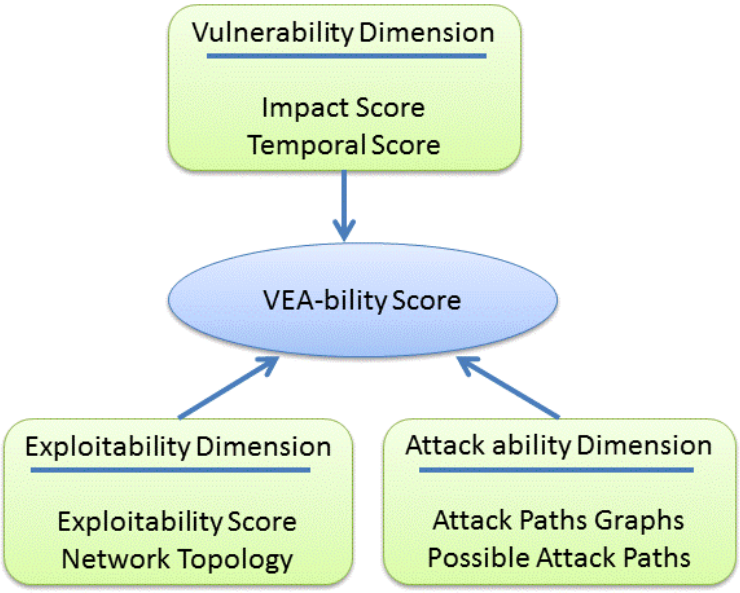


Figure 2.6: VEA-bility metric

The VEA-bility metric is actually the security scores of vulnerability, exploitability and attackability. The scores for these three dimensions are obtained from the network topology, attack graphs and the Common Vulnerability Scoring System (CVSS).

These VEA scores are for each host in the system. The combined scores are the exponential averages and summations of the individual host scores. For example, the vulnerability score of the network is the exponential average of the host vulnerability scores. And the network exploitability and the attackability scores are the summations of the exploitability and attackability scores of each host.

$$V_N = \min(10, \ln \sum e^{V(host)})$$

$$E_N = \sum E(host)$$

$$A_N = \sum A(host)$$

$$\textbf{VEA-bility}_N = 10 - ((V+E+A)_N / 3)$$

4. Background of Risk Analysis

4.1 Decision Tree Model to Quantify the Intentional Attacks.

Many papers discuss the risk associated with non-anomalous failures and the costs associated with such failures, but very little work has been done on determining the risk factor associated with intentional failures and attacks. Authors in [33] present a decision tree model to quantify the intentional attacks. They argue that in using their quantitative approach network managers can ascertain the risk level as a percentage that can be tested, improved, compared, and budgeted compared to the qualitative schemes.

Their security model consists of following elements:

Vulnerability: A weakness in any information system, which can be a coding bug or a design flaw.

Threat: Any circumstance or event that has the potential to adversely impact an information system through unauthorized access, destruction, disclosure, or modification of data or denial of service.

Countermeasure (CM): An action, device, procedure, technique, or other measure that reduces risk to an information system.

Residual Risk: The risk remaining in the system after the CM is applied. It is zero if a perfect CM exist.

All the elements of their frame work are explained in Figure 2.7.

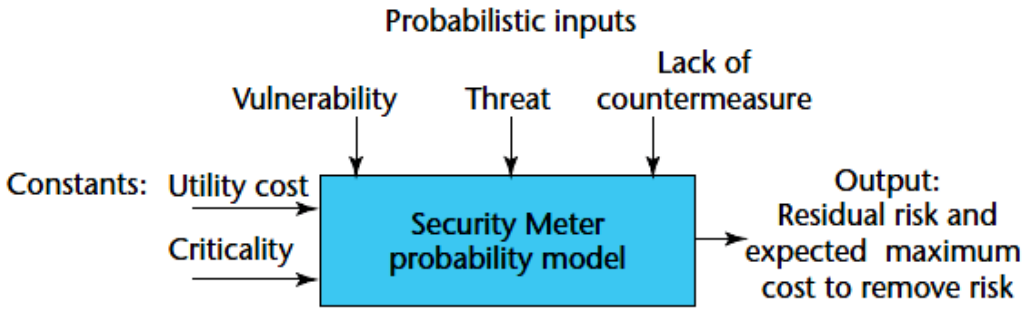


Figure 2.7: Security meter framework

Using the above model they calculate the residual risk as follows:

Residual Risk = Vulnerability*Threat*LCM

Where LCM is the lack of countermeasures.

Using the criticality factor, final risk can be calculated as:

Final Risk = Residual Risk*Criticality

And the expected cost associated with the risk is

ECL = Final Risk*Capital Cost

An example chain for risk calculation is explained in Figure 2.8.

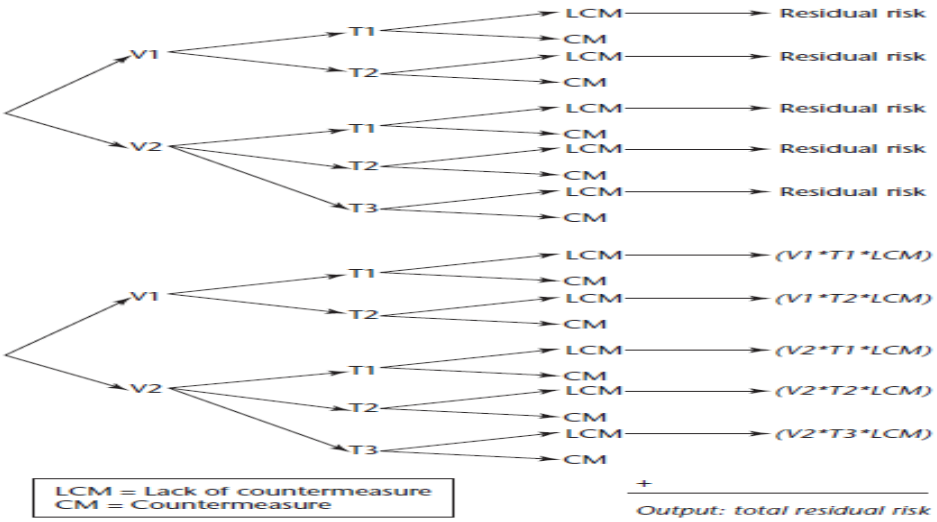


Figure 2.8: Risk calculation using security meter

4.2 Measuring Security in Next Generation Networks (NGN)

In [34] the authors consider the security issues in the next generation networks (NGN). International telecommunication union (ITU) defines seven dimensions to describe the security of NGN. Because of network layers and the attributes on each layer, which help measure the security of a given network, they describe a three dimensional model which is described in Figure 2.9.

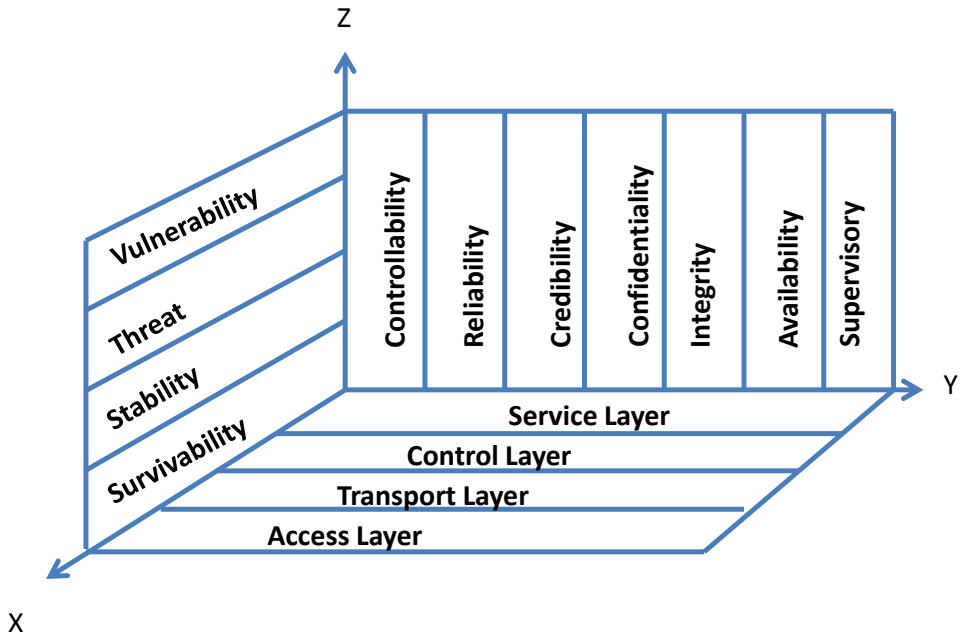


Figure 2.9: Security metric for next generation networks

The x-axis is the NGN network level; the y-axis is NGN security dimension; and the z-axis is NGN security metric.

In Figure 2.9 it can be seen that NGN is divided into four layers and the security of each layer can be measured for seven dimensions shown on the y-axis. Since these security attributes can be mapped to network attributes, the paper uses four attributes on x-axis to represent the security level of NGN.

4.3 Application of Game Theory to Determine Future Risk

Some research has borrowed ideas from artificial intelligence, such as machine learning and game theory. One such paper [35] applies a Markov game theory approach to determine the effects of future risk on the present risk calculated. The farther away the risk is in the future, the more minimal the impact it has on the present risk. They also present an automated scheme to generate a remedial scheme for the network administrators. The architecture of their scheme is explained in Figure 2.10.

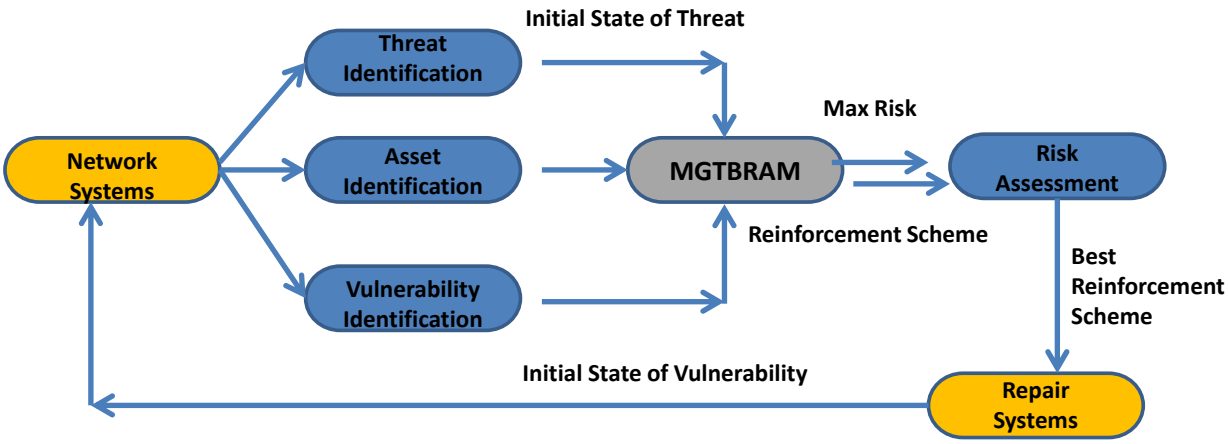


Figure 2.10: Markov model for risk assessment.

The proposed model consists of five main modules.

Threat Identification Module: This module detects the threats in the network systems and logs the data.

Vulnerability Identification Module: This module identifies the vulnerabilities in the system and stores it in a certain format.

Asset Identification Module: This module identifies and assesses the value of the assets in the system.

MGTBRAM Module: This module takes data collected by the previous three modules and after applying algorithms outputs data to the risk assessment module.

Risk Assessment Module: This module calculates risk for each vulnerability and records it as R(i) where i=0,1,2….,n.

After taking corrective measures, the value of risk for each vulnerability becomes RR(i), i=0,1,2,….,n

Thus, the eliminated risk is given as:

ER(i) = R(i) –RR(i) i=0,1,….,n

5. Conclusion

After carefully reviewing the previous work in the area of risk quantification following observations were made:

- Many papers make unrealistic assumptions that will fail in the real security environments.
- Ideas that are applicable in other domains may not be applicable in computer network security.
- Many papers simulated their proposed models and there is no knowledge about the practical application of their proposed models.

CHAPTER 3

IDENTIFICATION OF SECURITY METRICS

The debate towards an effective security metric is a big one. Every solution provider and security vendor claims that the set of metrics chosen by themselves is the best, but no quantitative study or analytical approach has been presented to identify a good set that can effectively monitor the level of security. Securitymetrics.org, an organization dedicated to the effort of defining metrics has published a list commonly used, but it may not suit every organization's needs. The best approach for an organization is to identify a larger set of security metrics that works well in their environment, and then focus on metrics that play a significant role towards improving their security. Therefore, in this chapter we first outline the process of identifying a set of security metrics by passive analysis of network traffic. Later in chapter six we present an analytical scheme that identifies the metrics that play the most significant role towards enhancing security, so that resources can be focused on them. The detection and ordering of significance in these metrics is accomplished using ideas from decision theory and information science.

1. Identification of the Network Security Metrics

To find a mechanism for quantifying the security level of a network, a survey on the existing work was done as summarized in previous section. The next process was to identify a set of network attributes that could correctly define the security of a network. These attributes will be called security metrics throughout this document. The objective was to identify what a metric is and what should be selected as a metric that gives a

correct picture of the security. A SANS document [36] was used as a guide for defining the security metrics. SANS [36] defines a metric by comparing it with measurements, and the comparison can be seen in Table 3.1.

Table 3.1: Comparison of measurements and metrics

Measurements	**Metrics**
Measurements provide single-point-in-time views of specific, discrete factors.	Metrics are derived by comparing to a predetermined baseline two or more measurements taken over time.
Measurements are generated by counting.	Metrics are generated from analysis.
Measurements are objective raw data.	Metrics are either objective or subjective human interpretations of those data.

The basic idea is to identify the metrics and try to focus on them one by one. Thirty metrics were defined. Metrics were selected primarily by looking at the attributes that were seen in historical attacks, by looking at system vulnerabilities, and domains that calculate risk. One such example of the latter are the insurance companies that use attributes to set their premium for life and auto insurance. A metric should have the following properties:

1) It should be easily collected.
2) It should be consistent across different environments.

SANS [36] defines a metric to be SMART, i.e. specific, measurable, attainable, repeatable, and time dependent.

The main purpose of a metric is to answer the following questions:

• Are we more secure today than we were before?

• How do we compare to others in this regard?

• Are we secure enough?

For defining a metric following standards were followed:

1. Define the metrics program goals and objectives.

2. Decide which metrics to generate.

3. Develop strategies for generating the metrics.

4. Establish benchmarks and targets.

5. Determine how the metrics will be reported.

6. Create an action plan and act on it.

7. Establish a formal program review or refinement cycle.

2. Development of Tools

Having identified the metrics, the next important task was the development of tools that could collect and process the metrics. To achieve this goal existing tools to collect metrics were used. Tools were also developed for metrics that could not be collected using existing tools. Main tools that were found useful and learned are listed in Table 3.2.

Table 3.2 : Tools utilized

Tool	Purpose
Snort	For detection of past intrusions and attacks
Nessus/Qualysguard	Active vulnerability scanner
RNA, PVS	Passive vulnerability detection
P0f, PRADS	OS finger printing
Wireshark, tsharq, colasoft, Caci Pilot	Network traffic analysis
Python, perl , matlab, C	Parsing, filtering, analysis

Httpry, python, tsharq	Http analysis
Bothunter	For detection of compromised machines

For collection, parsing, filtering, processing, storage and retrieval of data we have developed a python based tool that is described in Figure 3.1.

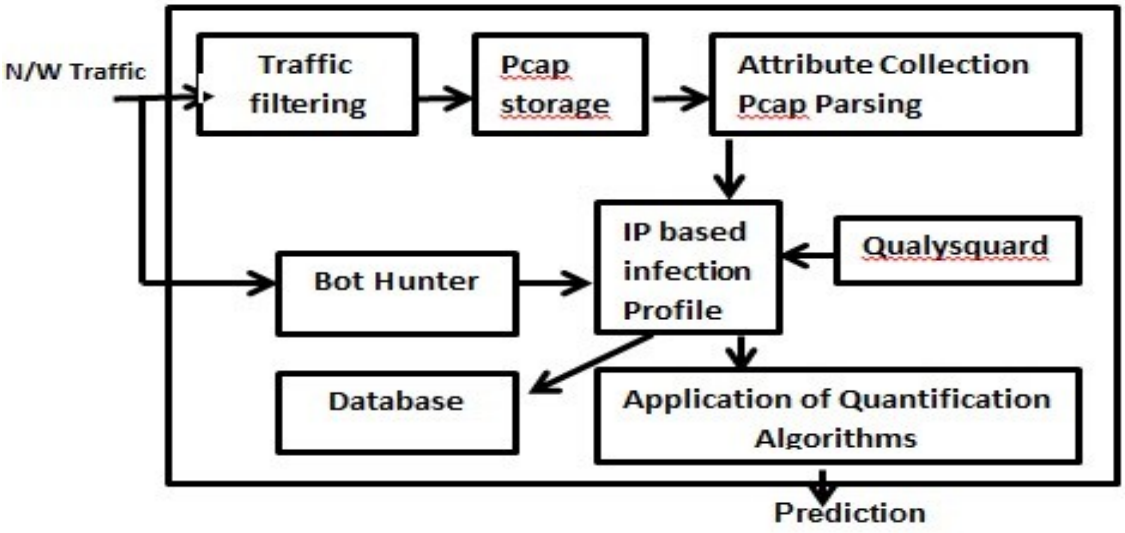


Figure 3.1: System architecture.

3. Experimental Setup and Data Collection

The most important aspect of this research was network data. After assuring the IT department that no one's privacy would be violated, access was given to network traffic from one of the dorms on campus. The subnet provided access to approximately one thousand unique IP addresses, but number of active users was far less. Another challenge was to monitor the data passively, which limits sending active probes to the scanned machines and also limits installation of any software on the machines. With the limitations of passive analysis things that could be detected very easily through scanning

tools had to be inferred. Attributes such as port scans, vulnerabilities, versions of OS and applications, OS updates, firewalls installed are very easy to capture with scanning tools. But these attributes had to be inferred from the network traces because active scanning wasn't allowed. For example, http headers had to be looked into to infer browser types, port status was inferred from the network activity, vulnerability information was deduced from the OS and applications detected, and patch status was detected from the update links. A diagram of the network setup and collection point is shown in Figure 3.2.

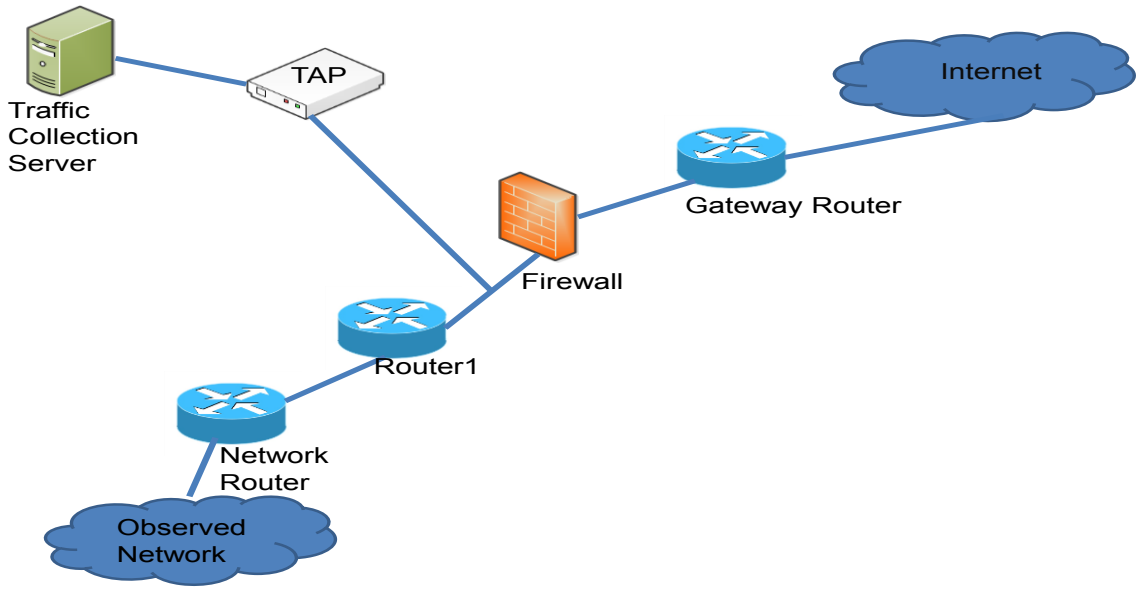


Figure 3.2: Data collection and processing from network tap.

4. Metrics Refined from Network Data

The metrics collected from the traffic traces along with their significance are given below:

1) Operating System: Known vulnerabilities in the detected versions of operating system.
2) Patch Frequency: How often a user patches its OS, browser, antivirus software.
3) Browser : The version and the known vulnerabilities for the detected versions of the browsers.
4) Plugins: Third party vulnerable plugins detected for the browsers such as activeX, Flash, Adobe.
5) Websites visited : Categorization of websites into classes of activity, threat level , frequency of visit, duration of visit, referrer information, hosting country, etc.
6) Web-searches: Categorization of search into classes.
7) Activity on a social network: Clicks on a third party links, apps, photos, time spent, frequency.
8) Links from chat: Encrypted chat usage, frequency of receiving links in the chat, frequency of replying to links.
9) Firewall and antivirus installation: Inference from the updates.
10) Ports open: Inference from the backscatter of the OIT port scans.
11) Targeted ports: Communications to the port other than the standard ports.
12) Vulnerable protocols: Usage of FTP, Telnet.
13) Cookies sent: Prone to XSS, CRFS types of attacks.
14) System uptime: Higher system time means more vulnerable to attack. Similar to the auto and life insurance companies that determine higher premium based on cars or a person's age.

15) Online streaming: Youtube, Megaupload, Netflix, sporting event sites, free movies, tv shows. These sites are the main reasons for the machines getting infected.

16) File type: Types of the file downloaded through http such as, movies, mp3, pdf, jpeg, exe.

17) P2P activity: Types of the file download, activity duration.

18) Network scans targeting a machine: The assumption is that next step after a scan is an attempt by the attacker to find the vulnerabilities and compromise them.

19) Malware: The malware binaries detected that were destined to the targeted machine.

20) Packets with incorrect fields: These packets are likely attacks generated by tools to exploit weaknesses in the protocols.

21) Network game usage: The gamers need to install third party applications that are vulnerable to exploits.

22) Communicating countries: Communication to countries where attacks/scans have originated and detected in the network.

23) Daily traffic trends: Based on a normal behavior something abnormal can be detected.

24) Usage patterns: Sites a user visits, time he spends there, frequency of visiting a particular site in a day. If a user changes his pattern drastically than that's an indication of something wrong and a user that is online less often is less likely to be effected than a user who is online more frequently.

25) Popup links: After opening a link a lot of popups indicate careless browsing behavior.

26) Links from email: There is no referrer information in the links and no way to analyze emails as the data is encrypted.

27) Spam: The number of spam messages a user gets, how much downloaded data is due to emails, how frequently he replies. These statistics aren't possible without decrypting the email payload.

28) Authenticity of certificates: Detection of a user clicking the certificate option.

29) Compromise history: Similar to an auto insurance if a user had accidents in the past he is more likely to be in an accident in the future.

30) Discovery of vulnerability and patch availability: The data about the date of discovery of an exploit on an O.S or a browser and its patch availability.

31) DNS attributes: Time to live (TTL), number of IPs, autonomous systems in the DNS packets could be a very good indicator of a phishing based sites that a user is visiting.

CHAPTER 4

NETWORK TRAFFIC CHARACTERIZATION FOR COLLECTING SECURITY METRICS

As the use of web is increasing, so are the complexities of applications available on it, and these complexities result in loopholes and susceptibilities that unscrupulous hackers try to exploit. An increasing number of exploits are making use of vulnerabilities in browser plugins, operating systems and other software. Statistics show that such types of attacks, where attackers make use of vulnerabilities on the systems, are on the rise. User activities, such as a lack of efforts to secure the machine by doing updates and reckless browsing behavior, tend to make the picture even worse. In this chapter we first look at the network traffic to see trends such as network bandwidth, protocols in use, active hosts, operating systems and applications etc. Then we passively capture the user browsing behaviors, i.e. what sites he visits, frequency of visits, searches he makes. We look at similar statistics for update behavior of a user passively. We also look at the different attacks going on over the network. For this study we did passive traffic analysis on the dormitory network of a college campus. The main purpose of this characterization is to identify a set of attributes that can help us effectively measure the security of a network, and that act as a set of good security metrics.

1. Introduction

A network intruder always benefits from slackness on a user's part. It could be because of a user visiting questionable and malicious Web sites that enabled an attacker to download malicious code on a user's machine, or it could be the user's carelessness by not updating his application software, operating system, or not installing safeguard tools such as

firewalls, antiviruses, or IDS/IPS. Studies presented in [38, 39, 40] show how attackers are utilizing weakness in the user habits, especially through social engineering attacks. They also discuss how such attacks can be prevented with minimal efforts. The objective of our study is to survey our target network and find out which user behaviors might result in making the network vulnerable to unscrupulous attackers. We also characterize the network traffic that may have an indirect influence on the security of the network. The primary objective is to look at the profile of the users, and determine how their behavior and habits on their machines affects overall network security. This work is sort of like developing a passive 'nmap' from the administrator's point of view: i.e. to determine what applications a user is running, which OS types and versions he uses, if he updates them regularly, what type and version of browsers he uses, and whether they are vulnerable or not. We also look at the attacks and intrusion events as detected by SNORT, an open source intrusion detection system, to determine a correlation between user habits and attacks seen on the network. This study not only profiles typical user behavior on a campus network and presents statistics observed on the network, but it also carefully looks into these dimensions of network and comes up with a set of attributes that can effectively measure the security of a network. As discussed in [45] the best strategy for defining security metric is to investigate your own environment for attributes that effectively measure the security, rather than choosing widely adopted metrics as described in [46] and expecting it to meet specific security needs. So this study exhaustively looks for attributes that can be best metrics for the investigated network. From our study we have the following observations.

- Users are active at a certain time, communicate to an average number of servers, and communicate a certain number of bytes. If these parameters deviate from normal they can indicate a suspicious activity and thus can be used as a security measure.
- Social networking sites are the most popular. The average flow duration is very short in length.
- Users indicate a certain pattern in sites they browse, contents they download, and time they spend on a site. These parameters can be used to measure their security level. Web search classification also indicates behavior characteristics similar to Web browsing.
- Very few users on the network show a consistent update behavior. Most of the updates that were made were configured automatically.
- A majority of the attacks events happen at a particular time, fall into the bad traffic category, and involve contacting known malicious servers.

Our study not only characterizes network traffic of a campus dormitory, but also identifies attributes from user behavior that reflects the security state of a network and thus can be used to effectively measure the security level. In section 3 we outline our data collection setup and an efficient scheme that we utilize to collect traffic of interest that optimizes storage space. Section 4 characterizes network traffic and discusses how we passively identify attributes such as applications and operating systems on each user's machine, protocols in use, bandwidth utilization, defense mechanisms installed etc. Section 5 discusses the update behavior of a user. We look at trends of which top applications are updated, frequency of updates etc. Section 6 looks into browsing behavior of each user and discusses statistics on sites visited, popular sites, frequency of

visits, searches made, geo location of Web sites. Section 7 characterizes trends on attacks seen on the network.

2. Experimental Setup and Efficient Filtering of Network Traffic

Our data collection setup is shown in Figure 4.1.

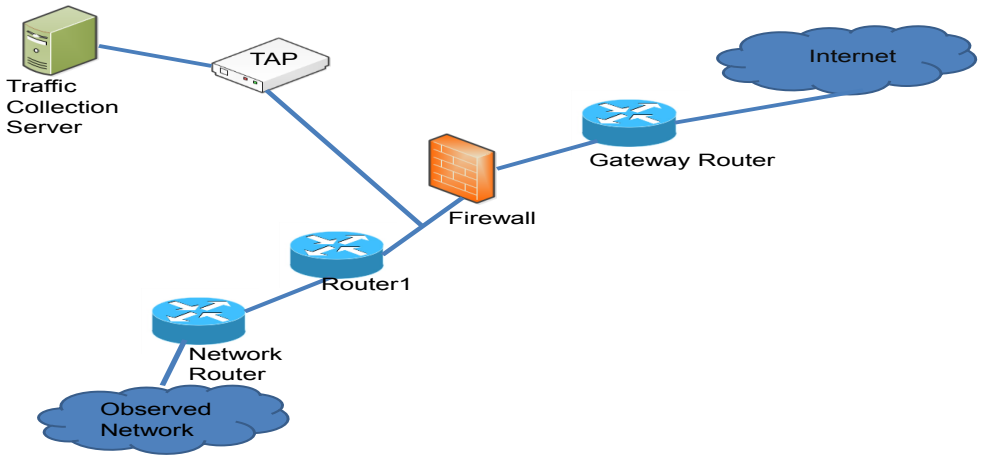


Figure 4.1: Data Collection Setup.

For our study we monitored a college dormitory network that has an active user base of about six hundred students. Data considered for this study spans a period of two months, from March 2012 to April 2012. We captured packets and stored them as tcpdump files for later "passive" analysis.

Ideally packets should be analyzed in real time rather than doing a full tcpdump type (pcap) capture for later analysis. However, having a pcap file is useful for later debugging and analyzing anomalies. In our case, many of the hosts were downloading large files, so a full pcap capture could not be supported by the size of the hard drives available on our monitoring host.

Since we wanted the first 2000 bytes of data complete, setting the tcpdump "snaplen" parameter to capture less than full packet size would have left gaps in the application-

header data that we wanted. The solution was to use our real-time analysis (RTA) program to write a reduced pcap file that would include only the packets containing first 2000 data bytes of each connection, as well as the TCP packets that had the SYN, FIN, or RST flags set. The RTA program was keeping track of each connection and its data bytes in real time, so it knew how many packets to output in pcap format. If a packet had more than enough data bytes to reach the 2000-byte limit, it was shortened, and the "capture-length" field in the pcap record header was adjusted accordingly.

Alternate methods of mitigating excessive traffic have been studied. The study in [37] used bulk traffic mitigation strategies based on curtailing per IP traffic to limit the excessive resource consumption by applications such as P2P.

The following table shows the tools that were used for this study.

TABLE 4.1. Tools utilized for the study.

Tool	*Purpose*
C	Real time analysis (RTA)
Python (dpkt)	Filtering, log parsing and analysis
Tcpdump/Tshark	Traffic capture and analysis
Tstat	Flow analysis
Matlab	Visualization and analysis
Snort	Intrusion event detection
Bothunter	Detecting botnet events

3. Network Traffic and User Characterization

This section tries to profile our network by identifying the number of machines that were observed, amount of traffic that was seen, operating system and applications that were in use, and the presence of safeguard mechanisms such as a firewall.

3.1 Number of Local Machines

We saw on average five hundred and sixteen local machines during the period of our study. This number is consistent with the population and number of machines on the network. In the third week of March there was a dip in Figure 4.2 due to students leaving dorms during the spring break.

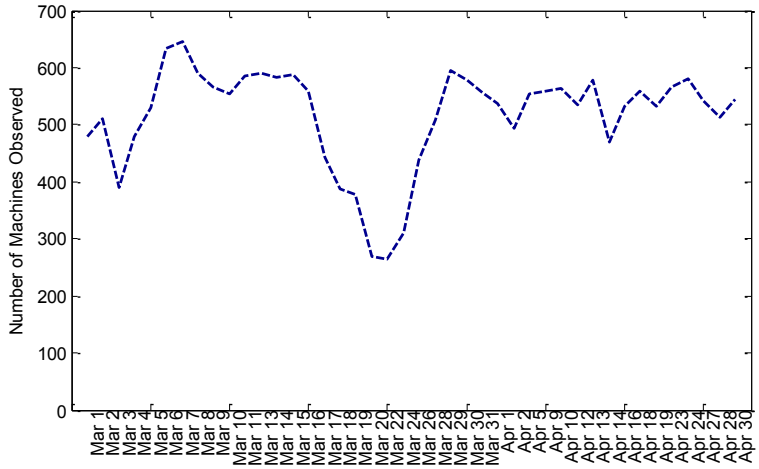


Figure 4.2: Statistics on the number of local machines.

Per day statistics are shown in TABLE 4.2.

TABLE 4.2. Per day statistics.

Parameter	*Average*
Local IPs	516
Foreign IPs	366223
Traffic Volume	~1.26 Tera Bytes

3.2 Network Traffic Volume

Figure 3 shows the traffic on the network, it can be seen that there is very limited activity during the school holidays. Also in the month of April traffic is relatively lower compared to March. This trend could be because of students are staying away from heavy downloads such as movies and online streaming, because of finals in April. Hourly traffic volume and stats are shown in Figure 4.3 and TABLE 4.3.

Another interesting thing to note is that during the month of March data that is sent is higher than the data received. Since we are observing a stub network and this is not a typical behavior of a stub, but this stub belongs to a college dormitory and applications employed by students in a dorm explain this behavior. For example figure 4.5 shows that Bittorrent traffic is the second highest traffic on the network, this explains that students are letting peers download files from their machines and thus heavy upload traffic. Streaming applications such BBC iplay also use p2p protocols to store videos on a user machine and let other users download it from a user machine.

In April the behavior represents a typical stub network, which could be as a result of a policy against p2p traffic put in place by the administrators that either blocks the p2p traffic altogether or traffic shaping that results in rate limiting or connection limiting the

peer to peer traffic which not only explain the normal trend but considerable reduction in the volume of traffic.

Figure 4.3: Daily traffic trend.

.

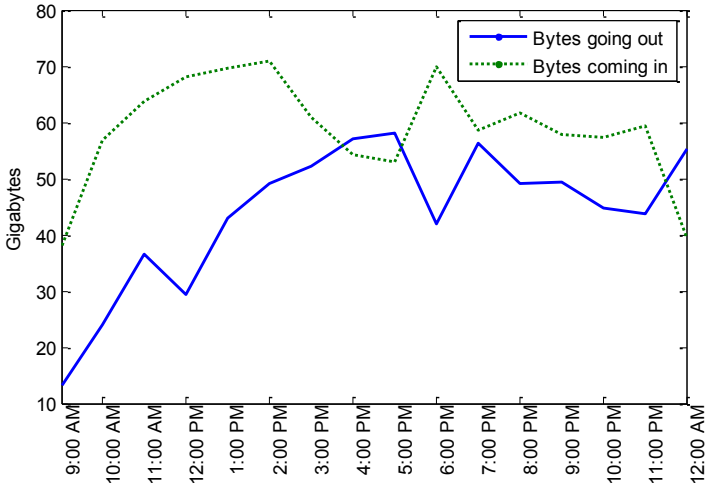


Figure 4.4: Hourly traffic trend.

TABLE 4.3. Average traffic per hour

Traffic	*Average (GB)*
Coming In	~58.7
Going out	~43.9

3.3 Protocols Observed

HTTP was the dominant protocol, closely followed by BitTorrent and SSL/TLS traffic as can be seen from Figure 4.5. There is consistent email traffic on the network as seen by the presence of POP3, IMAP4, and SMTP protocols. Different versions of protocols have known vulnerabilities associated with them, so a machine's security can be defined in terms of vulnerable protocols found running on it. For example machines running Telnet and FTP send password in clear text and can be susceptible to man-in-the-middle attack (MIMT).

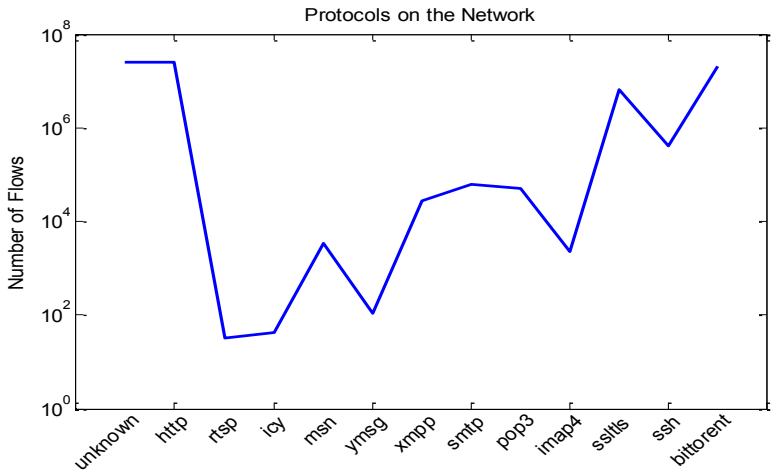


Figure 4.5: Protocols observed on the network.

3.4 User Agents

Figure 4.6 shows the statistics on most common user agents seen on the college network. Mozilla Firefox and Google Chrome are the leading browsers on the network followed by Internet Explorer. This usage of a specific user agent can be indicative of security state of a user when combined with the vulnerability status of a particular version and thus act as a good metric.

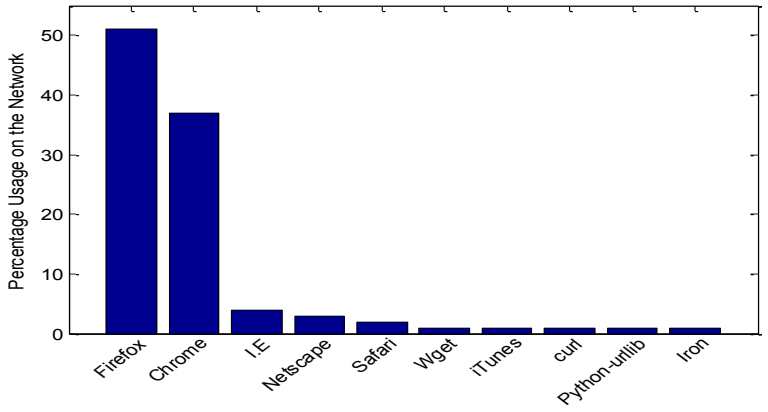


Figure 4.6: User agent statistics of the network.

3.5 Operating System

Like the user agents, vulnerable versions of the operating system can give us information on the security state of a machine. The statistics we collected about the type of operating systems in use on the college network are shown in Figure 4.7. There were instances of operating system as old as Windows 95, which has not had any updates in decades. This shows how effective this metric could be when determining vulnerable machines on the network.

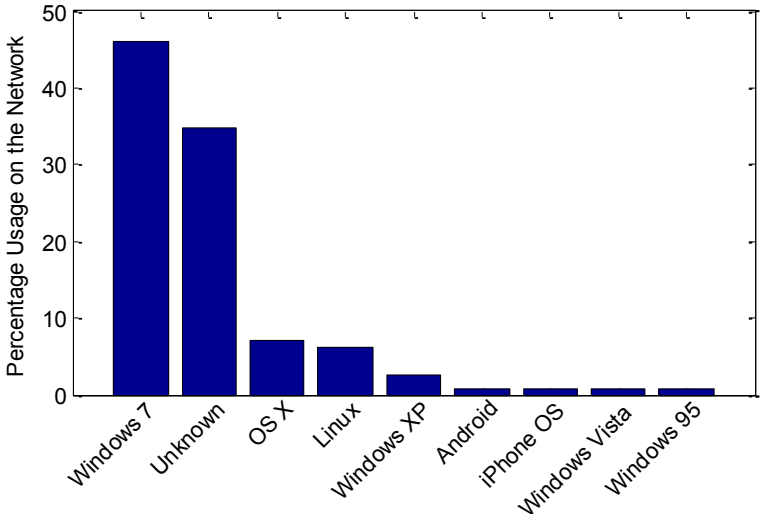


Figure 4.7: Operating system statistics on the network.

3.6 Defense Mechanism and P2P

Having a defense mechanism such as a firewall, antivirus, or an IDS/IPS system is very crucial for the security of the system. Since we are doing a passive analysis of the network traffic we have to infer the presence of these mechanisms on a host from network traffic. If there is a firewall installed on a host machine it will either do a "silent drop" in response to TCP SYN packets or will reply with a RST. Similarly an antivirus program is detected when an antivirus client updates itself for the latest definitions of threats.

Recently p2p is a favorable technique for bot command and control (C&C) and distribution. So p2p can be used an effective metric to measure malicious activity on the network. Statistics collected about defense mechanisms used on the network, such as firewall and antiviruses, and p2p usage is shown in Figure 4.8. Figure shows the percentage of machines for which these parameters were detected and not detected. It is

clearly seen that a little over half of the users have some sort of defense mechanism installed.

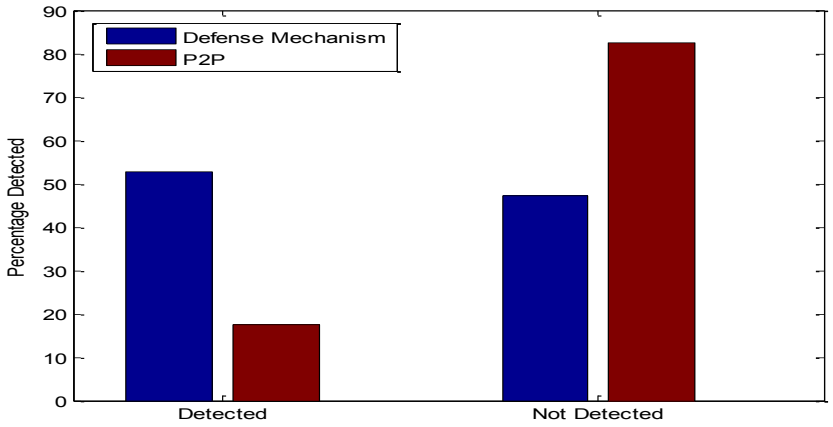


Figure 4.8: Defense mechanism and p2p activity on network.

4. Update Behavior Characterization

Updating applications and operating systems is very important for keeping the machine secure. In this section we look into update trends we observed on the network.

4.1 Update Detection

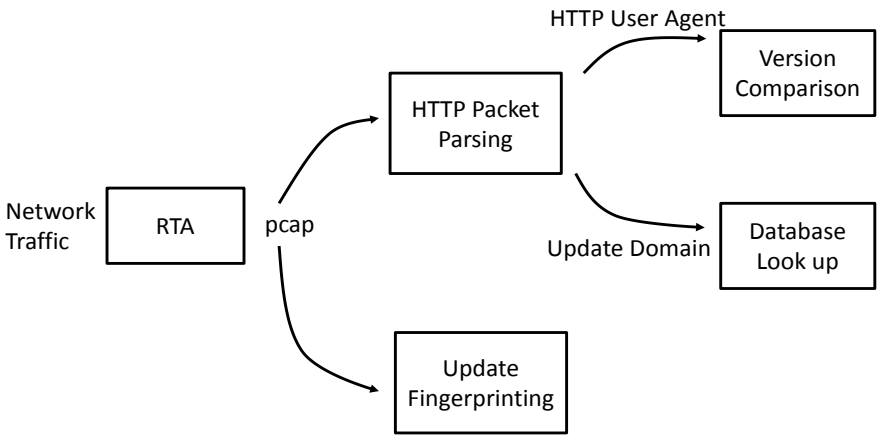


Figure 4.9: Network based passive update detection.

Detecting if a machine is updating itself passively from network traffic is a very difficult process. Owners of a machine or a computer administrator know when an update is triggered, because they personally configured the machines to update at a certain time. But in our case, we did not have any access to the individual machines that were active during the period of study. We are not the administrator of the network either so we do not know when and how machines trigger updates.

To detect updates we used three ways. First, we collected a list of update domains and matched every request from hosts inside the network. Second, we looked into the http header's user agent field. This field gave us the version of browser and operating system used on a machine. We keep track of versions for each machine and if this version changes, we identify that browser or operating system as updated. Third, we created an update fingerprint by analyzing the network traffic of an update process. If any communication from a client matches that finger print, we declared that machine as updated. The entire detection process using this three dimensional approach is shown in Figure 4.9. Using this three dimensional detection process, we were able to capture update trends that we will discuss in the following sub sections.

4.2 Overall Monthly Update Trend

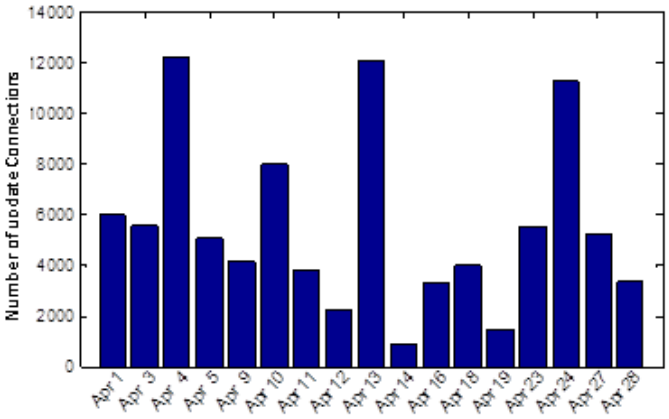


Figure 4.10: Update traffic for the month of April.

Figure 4.10 shows the number of update connections that machines on the network made each day in April. There are peaks on 4th, 12th and 24th of April indicating heavy update processes. We further investigated these peaks and realized that on these days vendors introduced some very critical updates. For example on April 12 Microsoft released a cumulative security update for Internet Explorer. This security update resolves five privately reported vulnerabilities in Internet Explorer. The most severe vulnerabilities could allow remote code execution if a user views a specially crafted Web page using Internet Explorer. An attacker who successfully exploited any of these vulnerabilities could gain the same user rights as the current user [41]. Similarly there are some heavy updates from Symantec on these three days.

4.3 Application Based Monthly Trend

Windows-based software and Symantec antivirus are the most updated software on the network as shown by the connections made to these update sites in Figure 4.11.

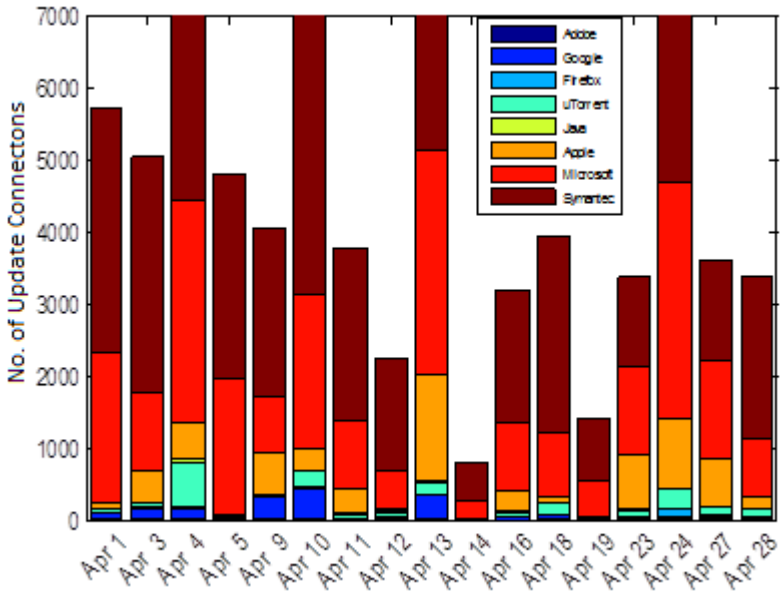


Figure 4.11: Update trends per application.

This trend is quite understandable as Microsoft regularly releases updates and most of the machines are configured to update automatically. Similary, antivirus software has to do definition updates. Other applications that need user involment are far behind in terms of the number of updates, which is an indication of a poor update behavior by a user.

4.4 Frequency of Updates

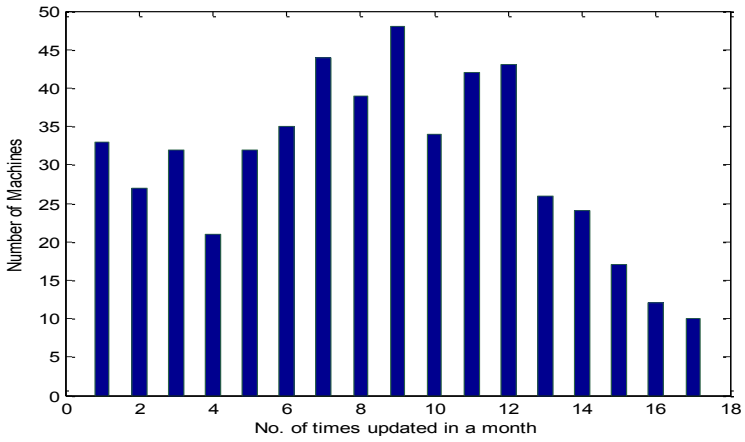


Figure 4.12: Frequency of updates by users per month.

From the security point of view, it is very important to know how often a user updates its machine. Figure 4.12 gives us a trend on the number of particular machines that updated something in a month. It can be seen that only 40 to 50 machines out of average 516 were updating more than 10 times a month.

4.5 Number of Applications Updated

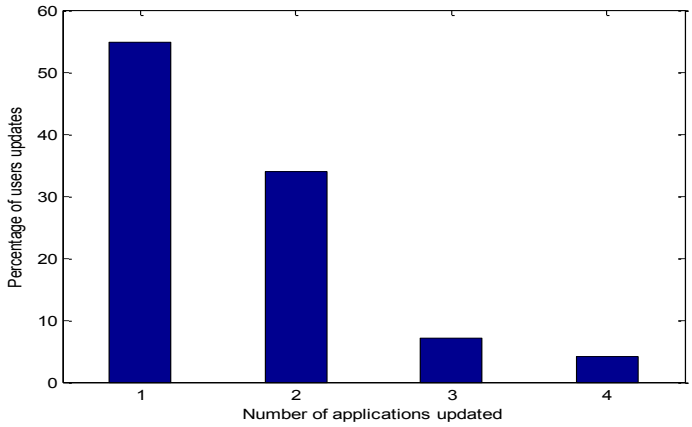


Figure 4.13: Number of applications updated.

Data in Figure 4.13 shows the percentage of users that do a certain number of updates in a month. It is seen that more than half of the users just do one application update. This is an indication of poor update behavior and shows the vulnerability level of a user. Most applications could be configured to automatically do updates, so practially half of the users on the network are slacking on this front. Only 2% of the users updated four or more applications which is an indicator that very few users on the network take this important security measure.

4.6 Percentage of Application Updates

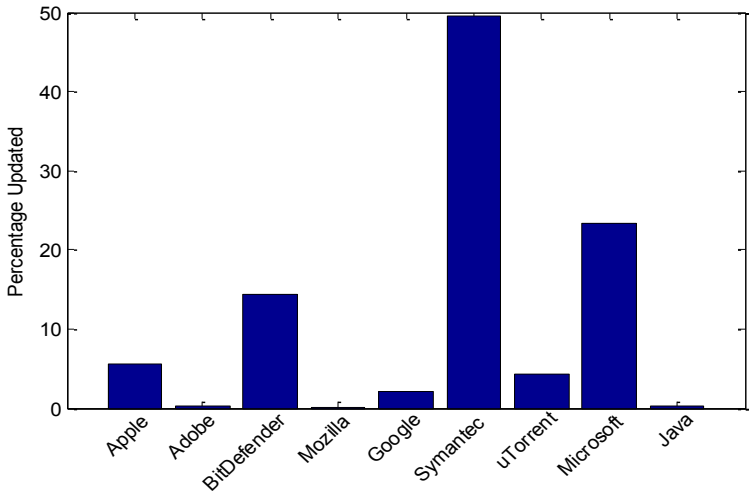


Figure 4.14: Most commonly updated applications.

Figure 4.14 gives us the statistics about the most commonly updated applications on the network. As explained in the previous sections, applications which do updates automatically are leading. Applications which require manual updates are laging far behind in terms of the number of updates. This supports our observation about the poor update behavior of users.

4.7 Correlation with Critical Updates by Vendors

To get a glimpse of user update behavior we looked at the most critical updates announced by vendors during our study period. We looked at the updates shown in following table with their dates and criticality.

TABLE 4.4. Major updates by Microsoft in March and April.

Date	*Number*	*Title*	*Rating*
3/13	MS12-020	Vulnerabilities in Remote Desktop Could Allow Remote Code Execution	Critical
4/12	MS12-023	Cumulative Security Update for Internet Explorer	Critical

We correlated these updates on our network and saw a correspondence between the release of an update and the number of users updating their machines on that date as discussed in section 5.2. This shows a good security reaction by Microsoft.

5. Browsing Behavior Characterization

As discussed earlier browsing trends can give us insight into the security state of a machine, so in this section we carefully look into user browsing trends such as popular sites, frequency of visits, time spent etc.

5.1 Popular sites visited by users

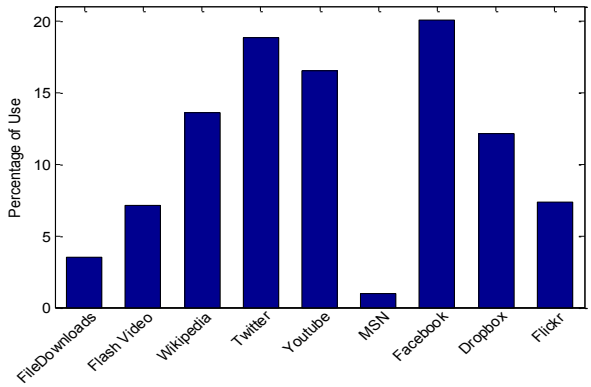


Figure 4.15: Popular sites visited by users.

Figure 4.15 gives us the popular sites during the study period. Social networking sites such as Facebook and Twitter and online video streaming are the dominant sites visited by users. These statistics were collected from the flows gathered by tstat [47].

5.2 Frequency of visits

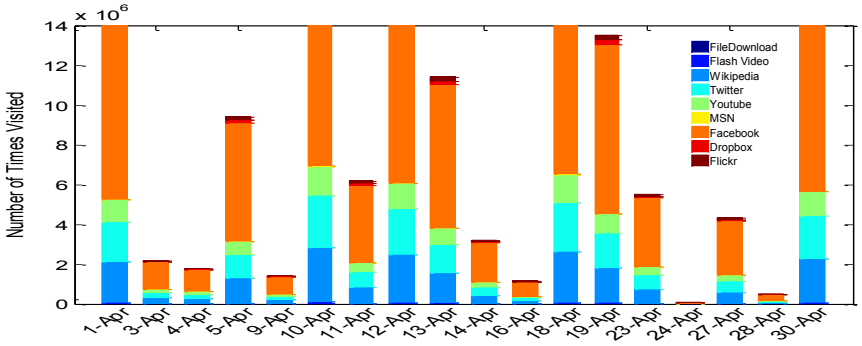


Figure 4.16: Daily sites visited by users.

Figure 4.16 shows the daily Web usage in terms of popular sites for the month of April. Similar to above section it can be seen that social network sites are the most popular sites visited on daily basis.

5.3 Average Flow Length

CDF for the average flow length is shown in Figure 4.17, which shows that on average an flow duration is very short. Almost 90% of the connection durations are under two minutes. This is an indication of a fleeting browsing behavior where users click from one site to another quickly. This is a common theme in modern Web browsing, for example in jumping from one Youtube video to another, from one Wikipedia page to another, or clicking through the profiles of friends of friends in a social networking site.

This also validates existing studies on Elephant vs Mice flows, which say that only less than 4 % of the flows are elephant flows [76]. For example a lot of traffic flows on the network are DNS queries which take only seconds to complete. Another important characteristic that we wanted to observe was how much time a user spends on a particular site and that can't be done without having instrumentations on the end machines such as monitoring software that records user activities.

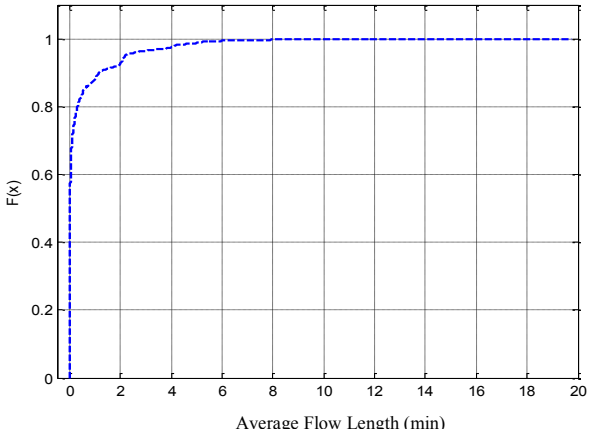


Figure 4.17: CDF for the average flow length.

5.4 Number of sites visited by users

Cumulative distributive function (CDF) for the number of sites visited by users in a month is shown in Figure 4.18, it shows that almost 40% of the users access more than 1000 sites each month, and 10% more than 20,000. This is an indication of heavy Web usage on the dormitory network that makes sense considering the observation made in 6.3 about fleeting browsing behavior.

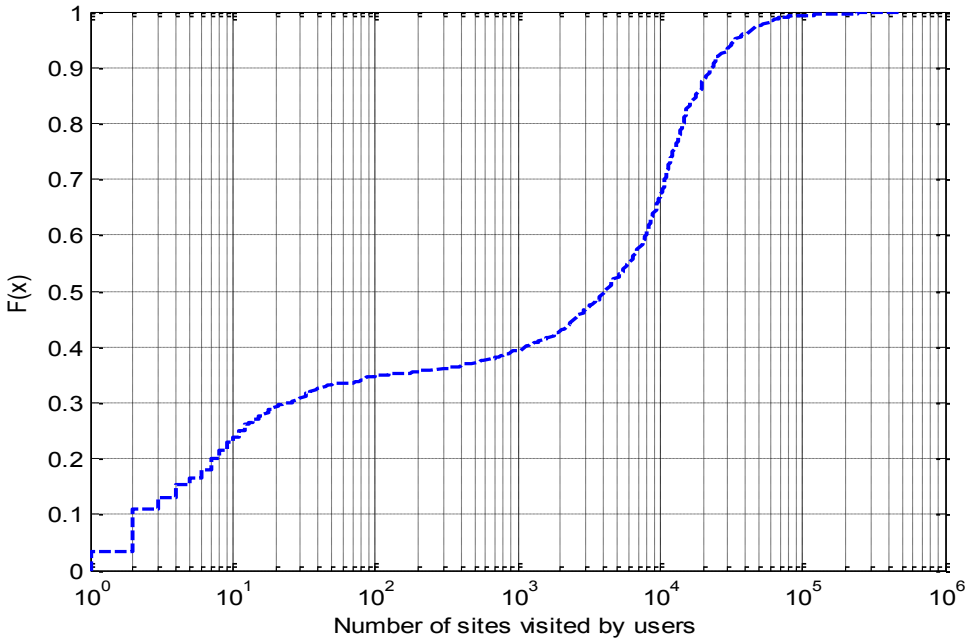


Figure 4.18: CDF for no. of sites visited per month.

5.5 Website Classification

To get a better idea of what users are doing we wanted to identify the sites they visit into classes such as news media, sports, religion etc. Because most of the URLs are just text strings without any semantics attached, we used the database provided by [42]. Figure 4.19 gives us the percentage of sites that were placed in a specific category.

This gives us a clear picture about the maliciousness of the website and more information about the activities of the users, which was not possible from popular sites analysis in 6.1. It also shows that most of the users are visiting sites such as sites that host malware, adult sites that are known to be infested with spywares and viruses, phishing sites. Therefore a good or bad browsing behavior gives a good picture about the security state of the machine and can be used as an effective metric.

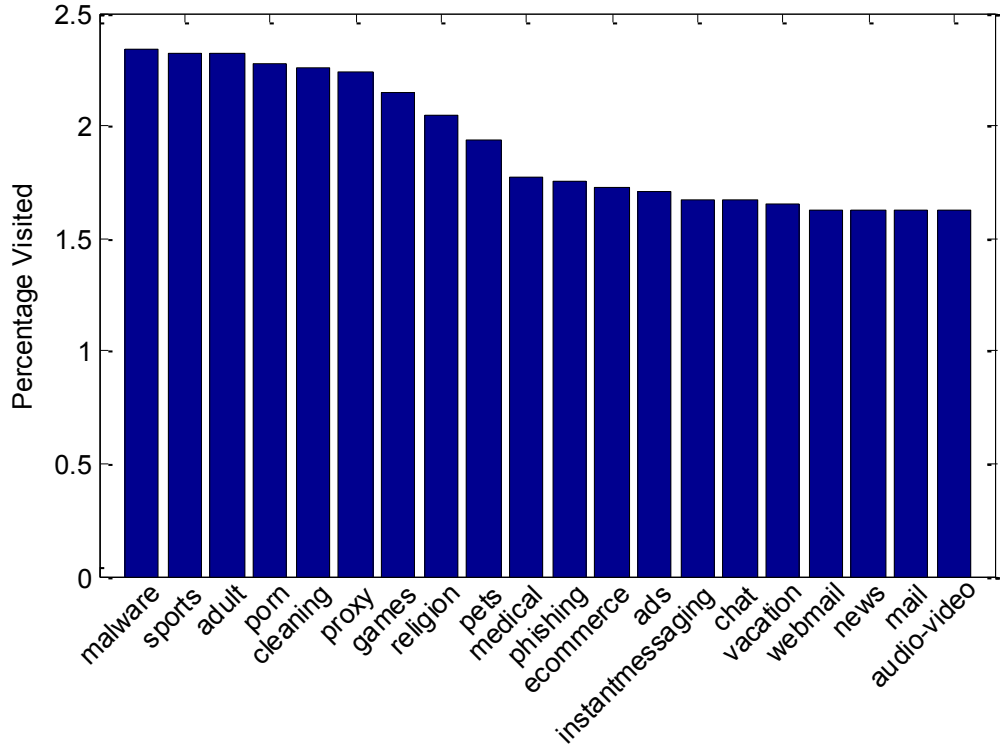


Figure 4.19: Website classification.

5.6 Web Content Analysis

The type of Web content downloaded is a very important candidate for a security metric, because these files have a direct correlation with infection rate. An exploit has to be downloaded on the machine in order to infect it, and certain types of files have long been known to be the carriers of malware. For example javascript, flash and images files can be used to obfuscate attacks and infect a user machine. Figure 4.20 shows the top contents that were downloaded.

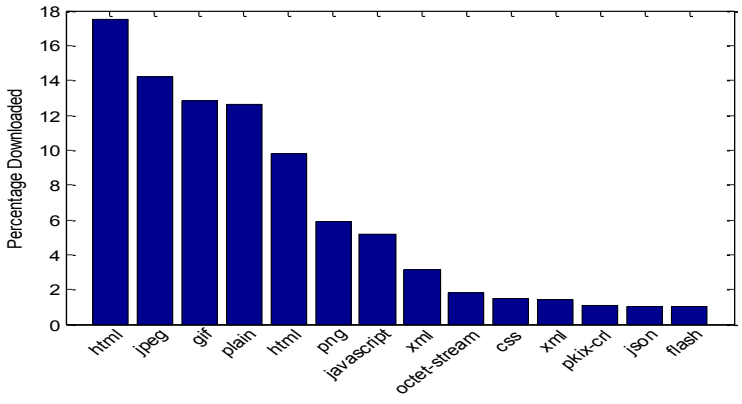


Figure 4.20: Web content analysis.

5.7 Server Response Code Analysis

Server codes tell us a lot about the browsing behavior of a user such as if the requests he is making are legitimate and if the user has permission to the file (code 403), if the file requested was not found (404), or the request was a success (200).

Figure 4.21 shows us that most of the web requests were successful: code 200. Second and third most found codes were 3xx and 4xx which deals with redirections and errors made by users respectively.

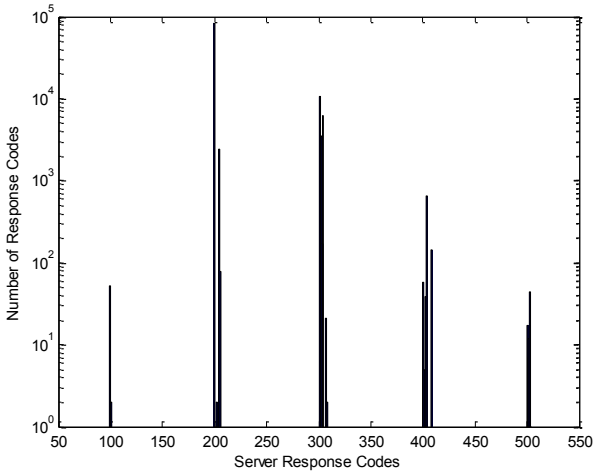


Figure 4.21: Server response codes.

5.8 Web Searches Classification

Web searches can tell us a lot about the browsing behavior of a user. We collected the user searches and classified them based on a machine learning API provided by [43]. It can be seen from Figure 4.22 that most of the searches made by users fall into categories of recreation, science, and society; sites that relate to the activities of an undergraduate living in a dorm. This data correlates with the Web site classification data in 6.5.

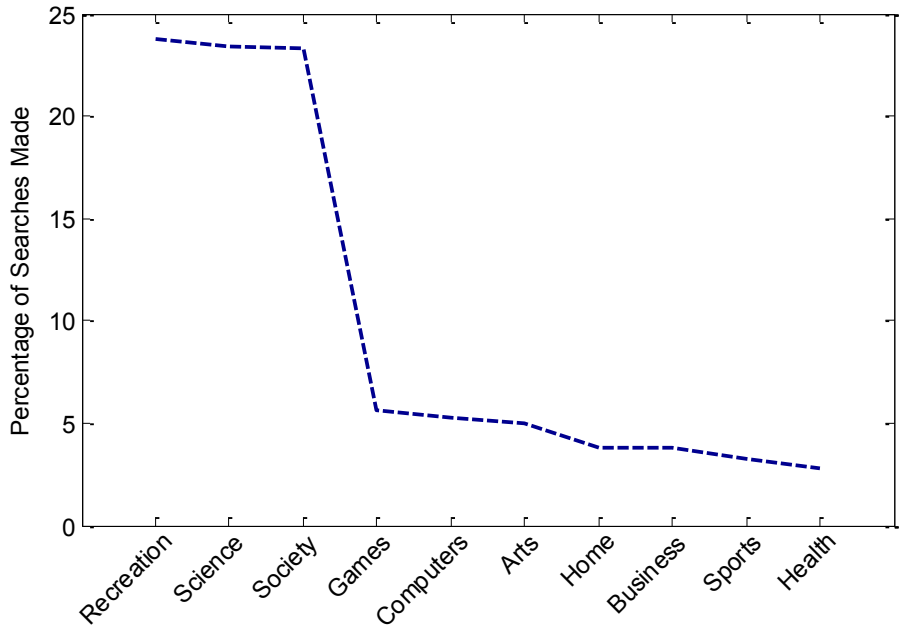


Figure 4.22: Web search classification.

5.9 Top Geo location of Servers

Figure 4.23 gives us statistics on the geo location of the server IP. This is an important attribute because studies [44] have shown a correlation between the origins of malicious sites and certain countries and therefore it can be used as an effective metric.

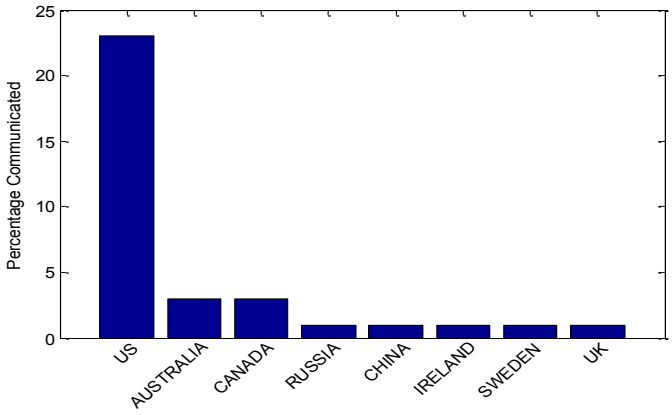


Figure 4.23: Geolocation of External IP.

6. Network Attack Characterization

In order to correlate the trends we observed from the network traffic, browsing and update behavior with the attacks observed on the network, we looked at the events detected by Snort, a freely available intrusion detection system (IDS). The observations are presented in this section.

6.1 Severity Level of Attacks Detected

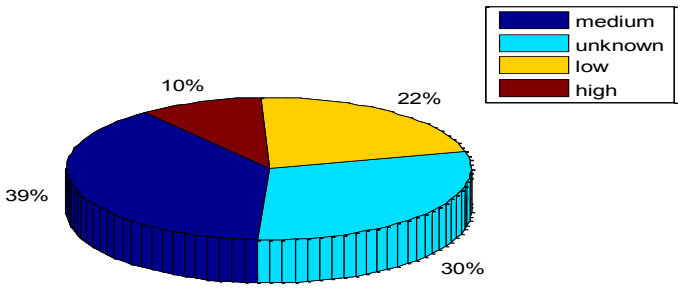


Figure 4.24: Severity of attacks.

Figure 4.24 represents the severity level of the attacks detected. Since Snort is a rule based system and organizations have rules specific to their own needs, users can define the events severity value themselves. For example an organization which holds Availability (A in confidentiality, integrity, and availability (as in CIA)) most important would assign a high value to denial of service (DoS) attacks. For our study we used the severity levels that are defined by default in Snort.

6.2 Per Hour Attack Distribution

From Figure 4.25 it cans be seen that most of the attacks are at 2000 and 2200 hours. This is an interesting observation because this time is considered to be the peak time where students are back to their rooms from classes, which means most of the attacks originate when users are active. This could be an indication of coordinated attackers that only launch attacks such as scanning attacks when machines are up.

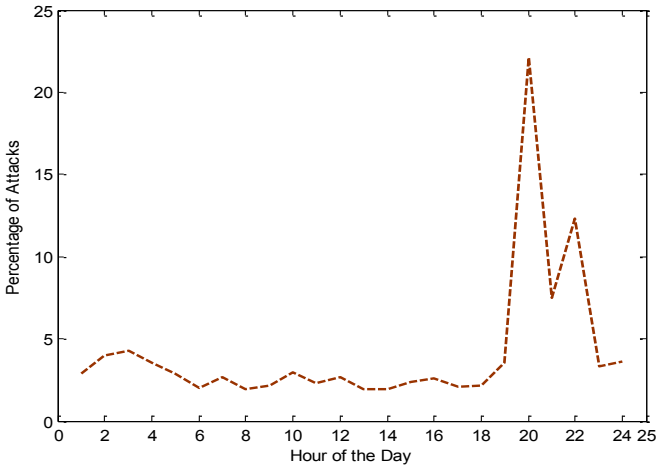


Figure 4.25: Attacks per hour.

6.3 Attack Distribution per Port

In the Figure 4.26 we can see that port 0 is the most attacked port. Heavy attacks on port 0 in our network indicate network-mapping activity by attackers. In TCP/IP port 0 is a reserved port and cannot be used by any protocol so the attackers use this port because most systems do not block this port. The second most attacked port is the Web server port 80.

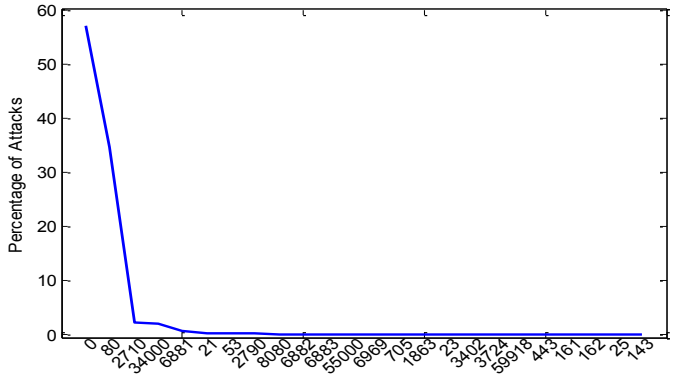


Figure 4.26: Attack distribution per port.

6.4 Classification of Attacks

The following TABLE 4.5 gives a distribution of attacks observed on the network. Bad traffic was highest percentage of attacks detected. This could be packets crafted with incorrect fields to exploits TCP/IP stack.

There were a lot of "http_inspect" BARE BYTE UNICODE ENCODING attacks. Bare byte encoding is an IIS trick that uses non-ASCII chars as valid values in decoding UTF-8 values. This is not in the HTTP standard, as all non-ASCII values have to be encoded with a %. Bare byte encoding allows the user to emulate an IIS server and interpret non-

standard encodings correctly [48]. This is a very common alert generated by Snort and does not mean that machines on the network are infected and launching these attacks. If the machines are clean and patched than Snort can be configured to filter this noise out.

TABLE 4.5. Classification of Attacks

Percentage	Classification	Severity
35.67	Potentially Bad Traffic	medium
27.17	http_inspect	unknown
21.6	Misc activity	low
8.39	Potential Corporate Privacy Violation	high
1.91	Attempted Information Leak	medium
1.36	Web Application Attack	high
1.17	TCP	unknown
1.01	access to a potentially vulnerable web application	medium
0.11	Generic Protocol Command Decode	low
0.04	Attempted Administrator Privilege Gain	high
0.04	Detection of a non-standard protocol or event	medium
0.03	Information Leak	medium
0.01	Misc Attack	medium
0.01	Attempted Denial of Service	medium
0.01	Attempted User Privilege Gain	high
0.01	A suspicious string was detected	low

6.5 Bot Activity and Attacks Detected by Bothunter

Bothunter [49] is a bot detection tool that monitors the traffic between host in our network and the Internet. It then tries to correlate this communication with the communication pattern of a typical malware. If a communication from inside of the network matches with the steps identified as part of a malware life cycle, it is declared infected. Each identified machine is assigned a score from 0.8 to 3.8 depending on how closely the communication matches with a typical malware communication.

The results of detection by Bothunter are shown in the Figure 4.27. E8 [rb] is the most commonly detected event, which indicates that an internal host connects to a known malware site.

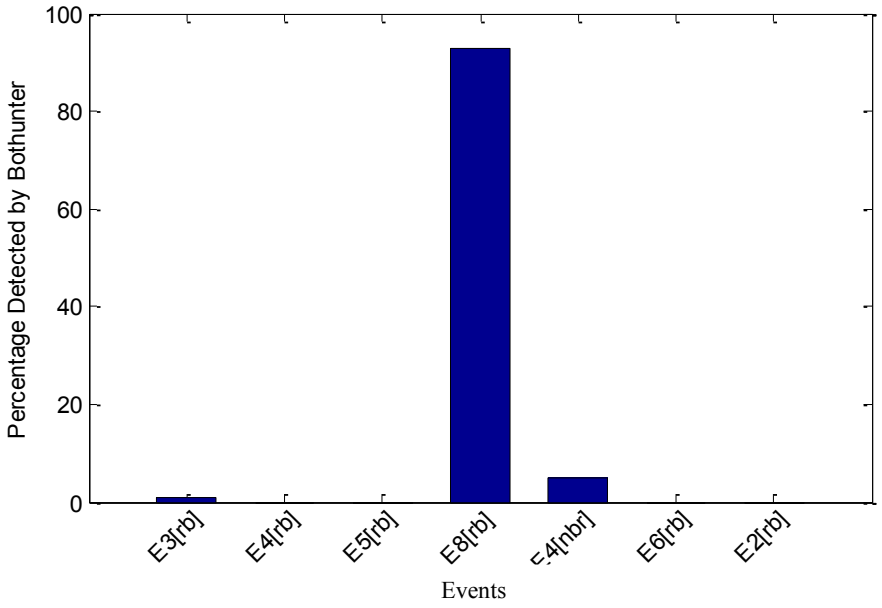


Figure 4.27: Events detected by Bothunter on the network.

7. Contribution of this Study

This study contributes in two ways. First it characterizes network traffic, user browsing and update behavior, and network attacks with a passive analysis. And then it points out security value that each of these discussed attributes bring and can act as a security metric. This study was geared more towards showing statistics of these attributes without analytically investigating their effectiveness as a metric. For future studies we plan to analytically determine the effectiveness of each of these attributes as a security metric.

8. Observations from Network Statistics

In this study we investigated network traffic extensively from the user behavior and the security point of view. Our objective is to study the network parameters to evaluate their effectiveness as an overall network security metric. For this purpose we first looked into network traffic itself to profile network usage, trends in bandwidth, protocols in use, operating systems and applications running, and firewalls and other defense mechanisms installed. Then we looked at the users' browsing behavior and looked at parameters such a classification of sites a user visits, how often the user visits a particular site, searches a user makes, countries the servers are located in, type of contents download etc. We also look at the update behavior of a user and parameters such as frequency of updates, most commonly update applications etc. Finally we look at the attacks and security events detected on the network. We presented them based on a per-day and a per-hour basis. We looked at things like network scans, malicious code downloads, contacts to malware servers (bothunter), etc. From the parameters observed on the network, we highlighted the security value of these network attributes. For example, we identified that average traffic volume on the network could be used as a metric, something that significantly deviates from this value could indicate an anomalous event. The versions of operating systems and application software can indicate how vulnerable the system is to known exploits, and can be a very effective security metric. Update frequency can also be used to effectively measure the security of the system, as not updating means leaving system more vulnerable to attacks We also observed that careless browsing behavior expose a user's machine to many types of attacks, so it can also act as an effective metric. Things like number of scans, and successful intrusions can also be effectively used to measure

the security level of the network. Using a similar investigation process, network managers can assess attributes and parameters of their own network and identify metrics that best suit and effectively represent their security.

CHAPTER 5

ANALYTICAL EVALUATION OF USER BROWSING BEHAVIOR AND UPDATES ON MACHINE INFECTION

User habits and browsing behavior play a critical role in the security of a system. There are certain actions on a non-malicious user's part that can bring the security of a system to a standstill, without his realization. In this paper we look at the interplay of those actions and the way a machine gets infected. We look at the profiles of infected machines generated by a bot detection engine, and investigate any correlation between the user behavior and infection of the machine. Apart from looking at the different network traffic parameters in TCP/IP stack, we also look at user behavior to provide context for his actions, while sitting at the network gateway. We look at several important attributes that comprise a user behavior such web searches, sites visited, frequency of visit, content download etc. We compare these characteristics of user behavior of machines that were infected with that of non-infected ones.

1. Introduction

Hackers are always looking for vulnerable machines on the Internet to compromise and to fulfill their nefarious designs. Motives behind their efforts are many fold, it could be to launch denial of service attacks, send spam emails, store illegal contents, or simply use it to compromise more machines. Vulnerabilities manifest themselves in a system due to poor designs, lack of testing, debugging and urgency by vendors to bring the product into the market as quickly as possible. But the successful exploitation is only made possible by lack of caution from a user side. Accessing malware infested sites on the internet is one of the big insecure user behaviors. Such sites

fool users into clicking on links that can exploit vulnerabilities or just do a drive by download of malware and infect it. In this study we want to take a holistic approach towards characterizing a malicious and non-malicious browsing behavior. Our study will look into several dimensions of Web browsing starting from sites a user visit, searches he makes, content he downloads, response codes that he gets from the destination, download content size, and geolocation of the server IP. Our investigation will focus on answering the question: "If there any correlation between these parameters of a user who gets infected and the one that does not?" And we will evaluate browsing behavior of user as an effective security measures in terms of attributes identified.

We have following important observation about user behavior trends.

1. Social networks are the most popular sites on the network.
2. Most users are searching terms related to recreation, science and technology which explains the behavior on a typical dormitory network.

After comparing user behavior of infected and non-infected machines we have following observations.

1. Infected machines were communicating with servers located in countries know to be originating most attacks.
2. Machines that were regularly updating were less infected than the machines that weren't frequently updated.
3. Users that were visiting questionable and malicious sites were highly likely to get infected than users not visiting those sites.
4. Users on file and photo sharing sites were more likely to get infected.

2. Existing research on Effectiveness of Metrics

In this study we investigate the browsing behavior of users on a college dormitory network. We look extensively at attributes that define a browsing habits of a user such as sites visited, frequency of sites visited, time spent on each site (from session information), searches made, contents download etc. and then try to correlate these attributes for infected and non-infected machines. To the best of our knowledge, user browsing behavior has never been studied so extensively before and never been correlated with the infected status of hosts. Section 3 discusses our data collection setup and an efficient filtering scheme to only look for traffic of interest and optimize storage. Section 4 discusses important web browsing trends we observed on the studied network. Section 5 and 6 outline the infected machine identification process and correlate infected machine browsing behavior with non-infected machine behavior.

3. Efficient Data Collection and Filtering

For our study, we looked at two subnets, a /20 and a /22, with an active user base of 600 users. We look at the network packets and extracted http header information from them. We look for URLs and searches that are made and log times corresponding to the IP address. To ensure that an IP represents a unique user, IP assignment in the DHCP has be tied to MAC addresses.

For our study, we monitored a college dormitory network that has an active user base of about a thousand students. Data considered for this study spanned over a period of two months from March 2012 to April 2012. We capture packets and store them as tcp dump files for later “passive” analysis. There are certain challenges for doing a passive

analysis vs. active analysis, some of them we discussed in section D. Our data collection setup is shown in Figure 4.1 in previous chapter.

4. Browsing Behavior trends and analysis

As discussed earlier in chapter 4 browsing trends can give us insight into the security state of a machine. In this section we carefully look into some additional user browsing trends such as that were not discussed in previous chapter and do analytical evaluation by comparing these attributes for infected and not infected machines in the next section.

4.1 Web Content length

Following figure shows that most of the contents downloaded were short in nature, in the later section we will see if length of the content has any correlation with the infection.

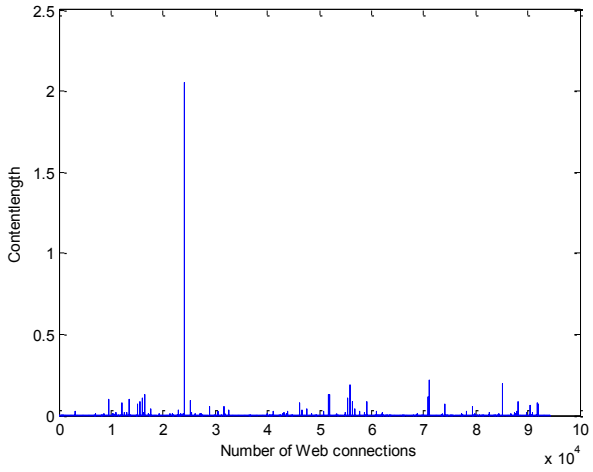


Figure 5.1: Web content length downloaded.

4.2 TLDs of Sites

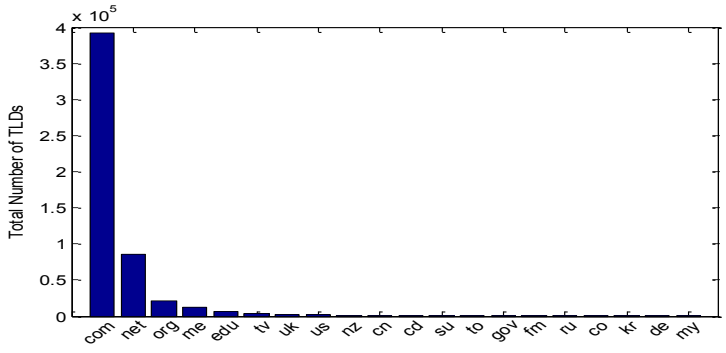


Figure 5.2: Top level domains.

It is not surprising to see that com is the dominating TLD, followed by net and org in Figure 5.2. It will be interesting to note the relationship between the TLDs of the infected and non-infected machines.

4.3 DNS attributes of sites

Figure 5.3 shows the statistics about the DNS attributes on the network. Only attributes that play a significant role in the infection are shown such as A records, NS records and TTL values. Later section will compare these attributes of infected and non-infected machines.

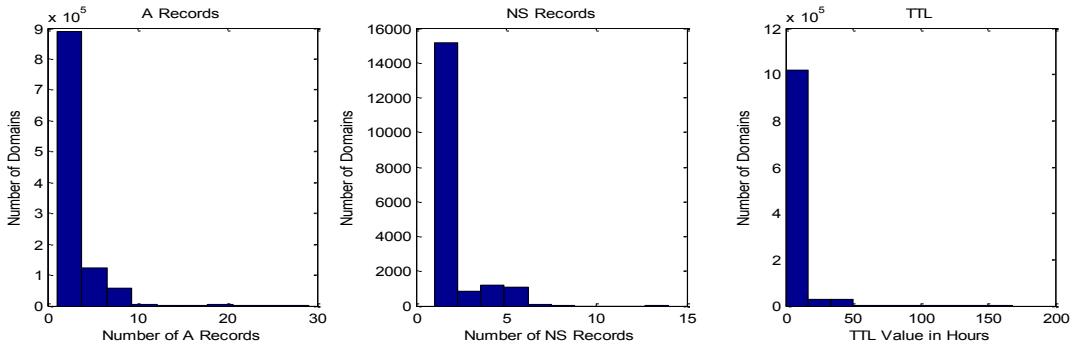


Figure 5.3: DNS attributes.

5. Bothunter for Infected Machine Detection

For the analysis we compare trends discussed in the previous section for infected and non-infected machines. We want to determine if any correlation exists between the profiles of infected and non-infected machines. Determining if a machine is infected or not is very difficult to do passively from the data captured at the tap as shown in Figure 4.1. If we had access to the user machines we could go and run malware detection software and determine the infection. Since we have to do it just by looking at the traces we are collecting at the tap we rely on a bot detection tool called Bothunter. Detection mechanism and attack trends observed through bothunter are discussed in the following sections.

5.1 Bothunter Detection Mechanism

Bothunter monitors the traffic between host in your network and the Internet. It then tries to correlate this communication with the communication pattern of a typical malware. If a communication from inside of your network matches with the steps identified as part of a malware life cycle, it is declared infected. Each identified machine is assigned a score from 0.8 to 3.8 depending on how closely the communication matches with a typical malware communication. Figure 5.4 explains the life cycle of a typical malware against which all network flows are matched.

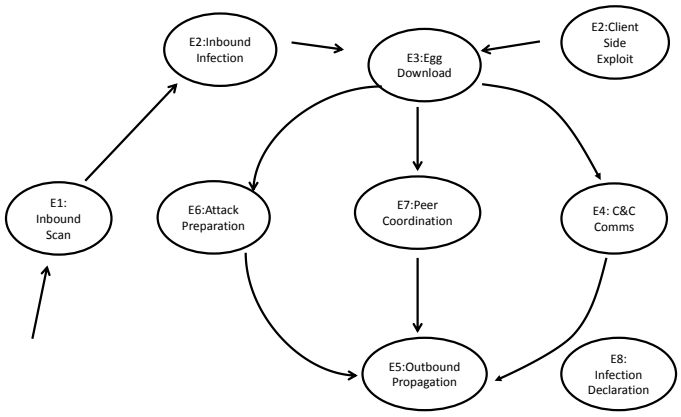


Figure 5.4: Bothunter infection life cycle.

5.2 Bothunter Attacks Trends

We tried to get a glimpse on the malware life cycle on our network and the events detected by Bothunter are shown in the Figure 5.5. E8 [rb] is the most commonly detected event, which indicates that internal host connects to a known malware site.

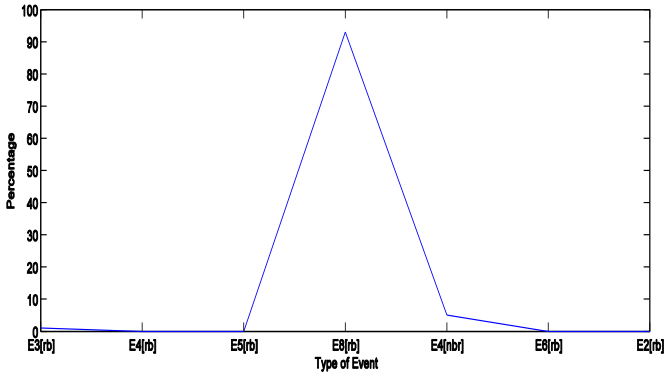


Figure 5.5: Events detected by Bothunter on the network.

6. Analysis of Infected and Non Infected Behavior

6.1 Popular Sites for Infected and Non Infected Machines

Apart from categorizing the web traffic into categories we also looked into most popular sites and tried to find out if there is any correlation between infections and visiting any of those sites.

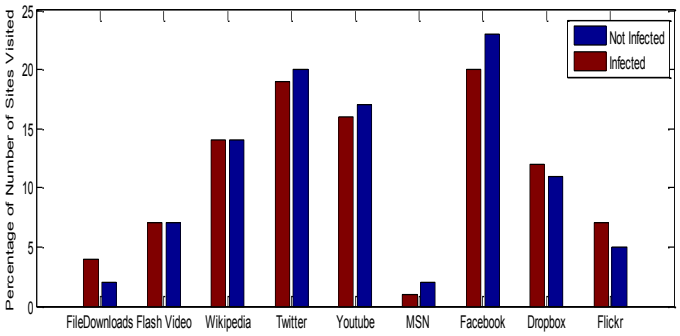


Figure 5.6: Popular sites visited.

In the above figure we can see that for file download sites that number of infection is higher than non infection, Flicker and Dropbox also show higher number of infections vs non infection. This behavior is an indication that a lot of attacks utilizing this vector. Social media sites such as Twitter, Facebook and Youtube show that visiting these sites machines are less prone to get infected.

6.2 Web Content Analysis of Infected and Non Infected

This was one of the most interesting parameter that we studied because files have direct correlation with infection as a exploit has to be downloaded on the machine in order to infect it.

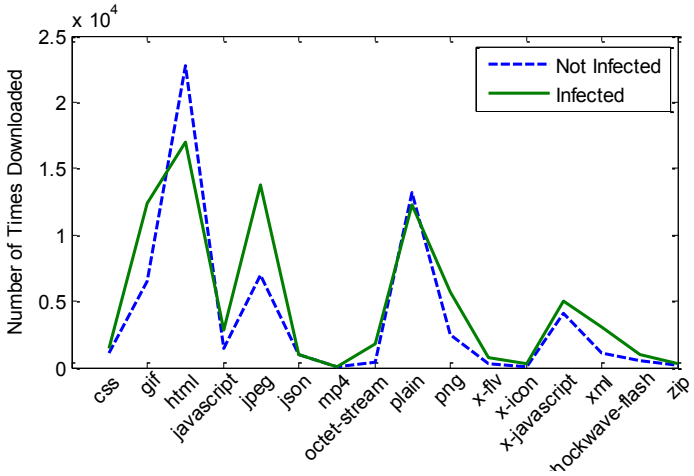


Figure 5.7: Web content downloaded.

6.3 Server Response Code Analysis

Server codes tell us a lot about the browsing behavior of a user. If the request he is making are legitimate or not. You can see that there is a spike in infected machines at 3xx which is the code used for redirection of sites and further action needs to be taken by the user agent to fulfill the request.

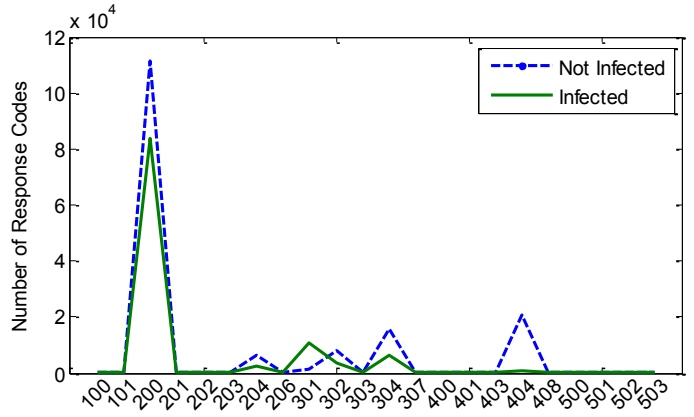


Figure 5.8: Server response codes.

6.4 Web Content Length Analysis for Two Profile

Here we analyzed the total length of payload that users are downloading. Our premise is that size of payload might be indicative of a good or a bad browsing behavior. For example video streaming, movie downloads are bigger in sizes, bot communications have a payload size that can be statistically identified from content length.

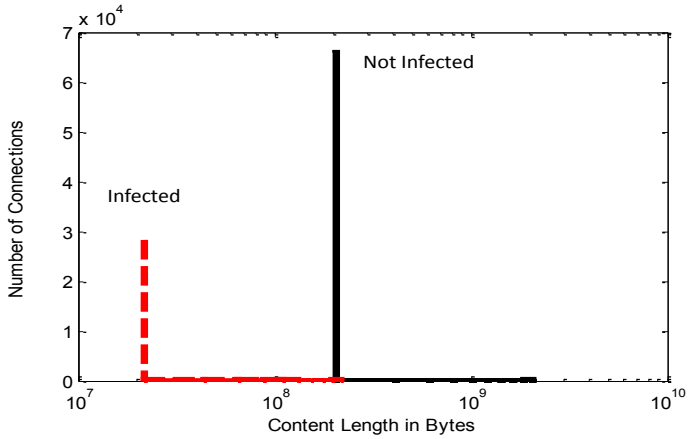


Figure 5.9: Web content length analysis.

6.5 Web Searches Classification for Infected and Non Infected

Web searches also give us an indication of a user browsing behavior of infected and non infected machines. Figure 5.10 shows that infected machines were visited sites associated with recreation, science and society.

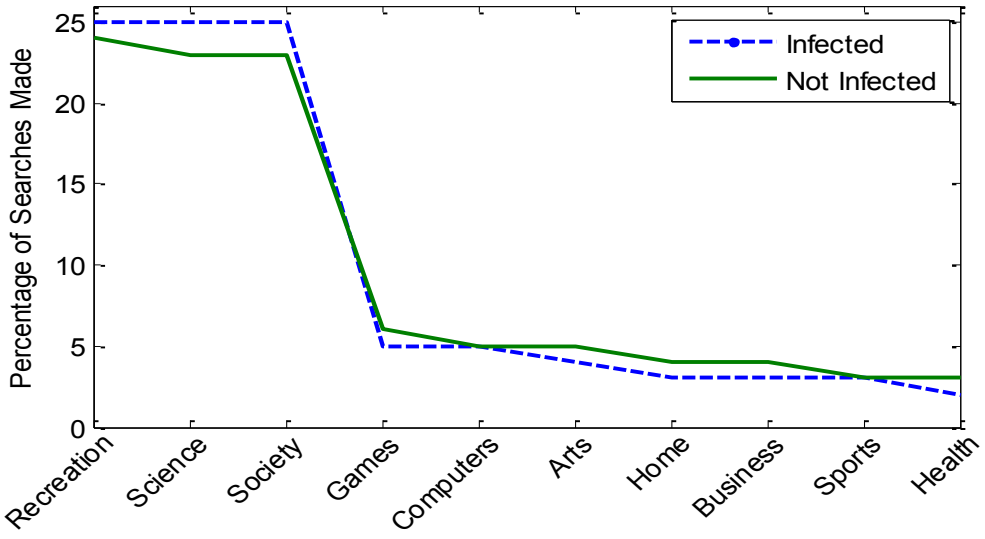


Figure 5.10: Web searches classification.

6.6 Geo-location of Servers for Infected and Non Infected

Figure 5.11 shows a clear relationship between the countries and the infections. Russia and China are known to be the originators of most of the attacks which is attested by the statistics in the following figure.

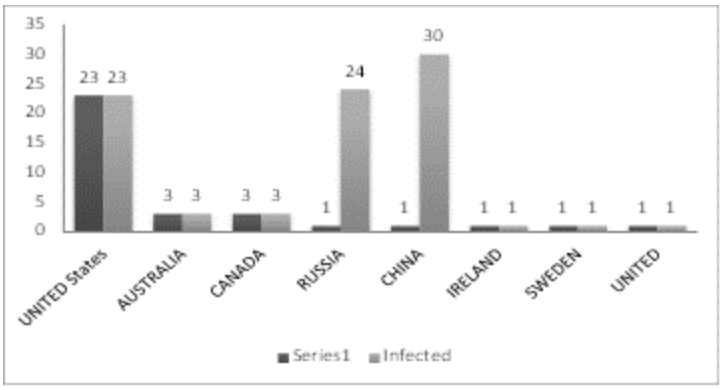


Figure 5.11: Geo-location of IP talked to.

6.7 Updates for Infected and Non-Infected Machines

Figure5.12 shows the number of updates on the vertical axis for the infected and non-infected machines. It is clear that machines that were not infected had the highest number of updates.

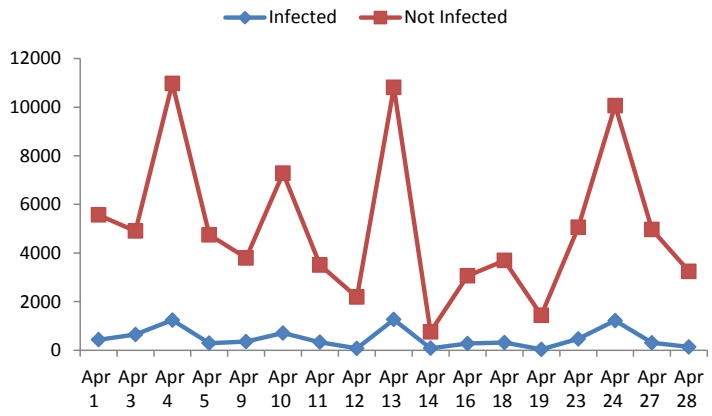


Figure 5.12: Updates for infected and non-infected machines for April.

Figure 5.12 clearly shows that machines that were infected were doing far fewer updates than the machines that were not infected. This shows a correlation between the update process and infection of a machine.

6.8 Frequency of updates for Infected and Not Infected

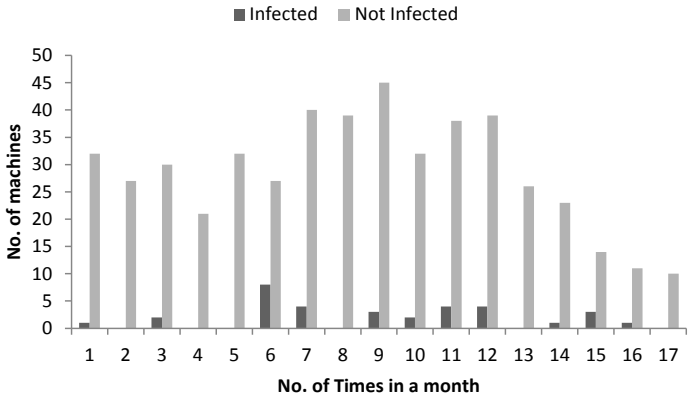


Figure 5.13: Frequency of updates for infected and non-infected.

How often a machine is updated determines how well it is protected from latest vulnerabilities. This fact can be seen from Figure 5.13 showing one instance of infected machines where only less than ten users updated their machines six times while the other users were updating little to none.

6.9 Number of Applications Updated

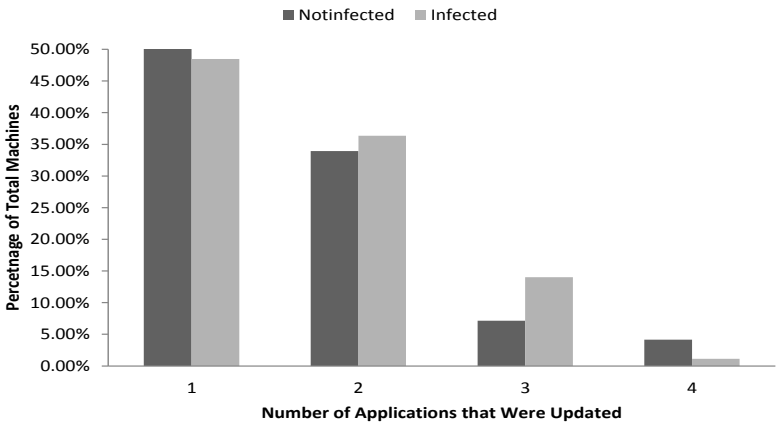


Figure 5.14: Number of applications update for infected and non-infected.

It can be seen from Figure 5.14 that users who updated only one application were the most infected ones. Also for users who updated most of their applications, the number of compromised machines was relatively lower. This is especially seen in the case of users who updated four or more applications.

6.10 Types of Updates

Figure 5.15 is the strangest graph that we observed during our study because even though a number of hosts were updating Microsoft and Symantec applications, they still got compromised. This could be because the update process was not able to patch the vulnerabilities that compromised on the machine. In Symantec's case, it could be that the bot was able to obfuscate itself to hide detection.

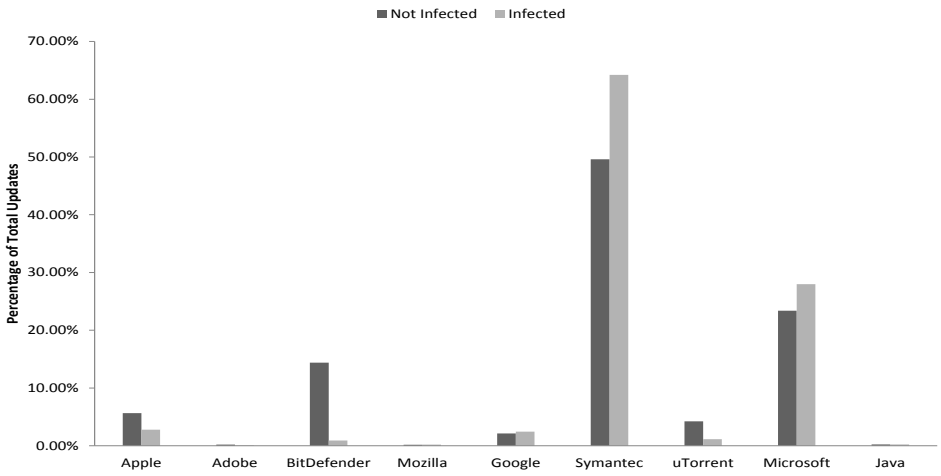


Figure 5.15: Types of updates for infected and non-infected.

7. Conclusion

In this chapter we have characterized the user browsing behavior from security point of view and highlighted trends that we observed in an infected and not infected machine.

Using these trends network administrators can focus on areas that result in most security incidents like if a user is going to sites or downloading contents that are increasing the threat level of the network they can devise a policy and implement safeguards that can improve the security posture of the organization.

CHAPTER 6

DECISION CENTRIC RANK ORDERING FOR IDENTIFYING SIGNIFICANT SECURITY METRICS

A network has several attributes which play a role in determining the security of the network and its robustness to external threats. These attributes are related to security through a complex nonlinear function which is generally unknown to the network management personnel. Since the network security is an explicit function of these attributes, the managers do not realize when the value of a certain attribute has become critical to the point of threatening network security. In this chapter we take a theoretically rigorous approach to quantifying the effect, each of the individual attributes has on network. By using correlation based analysis techniques for identifying principal components in the data, we can order the significant factors determining network security. This is similar to dimensionality reduction which is commonly employed for model based identification methods. Based on our analysis we can come up with critical coefficients which must be maintained by the administrator if the network is to remain secure to external threats. We show how the mathematical framework leads to the prediction of network's security health, and how real world data can be used to substantiate these claims. Chapter concludes by validating our results on a list of compromised machine and determining if our identified metrics played the significant role in the infection.

1. Introduction

A good size modern network observes millions of security events every day and these events pertain to hundreds of network parameters. Identification of events that has an impact on the organization's assets is an uphill task. The process becomes even more difficult due to today's complex and dynamic networks. A mechanism is needed that can help administrators channel their energies and resources in the right direction. Most of the attacks in recent years are successful because heavy effort is made on insignificant factors while leaving the most significant ones open for exploits. In 2012's data breach investigations report Verizon reports that 97 % of the attacks could be prevented with little effort because they were launched by mostly script kiddies without much skills and resources [59, 61]. This exploitation is going on despite organizations spending millions of dollars on defense strategies whereas hackers can compromise their security with simple exploits available online. The lesson learned is that having many controls does not provide effective security but having the right ones does. You can have firewall, antivirus, intrusion detection and intrusion prevention systems (IDS/IPS) on every machine on the network but if the users are not trained on security policy they could bring an infected USB, or a wireless device infected on an unsecure network, or download malware a network machine after falling prey to social engineering tricks.. Security metrics tries to answer the crucial question about where an administrator should focus his efforts, where should the control mechanism be placed to optimize security while being efficient on resources.

The debate towards an effective security metric is big, every solution provider and security vendor claims that their metric is the best but no empirical study or analytical approach has been presented to identify a good set that can effectively monitor the

security. Securitymetrics.org, an organization dedicated to the effort of defining metric has published a list commonly used [56] but it may not suit every organizations need. Best approach for an organization is to identify a larger set of security metrics which works well in their environment and then try to focus on metrics that play a significant role towards the security of the network. Therefore, in this study we first outline the process of identifying a set of security metrics by passive analysis of network traffic and then present an analytical scheme that help identify metrics that play most significant role towards security so that resources can be focused on them. For our analysis we use the infection profile generated by Bothunter, which is a dialogue base bot detection tool.

2. State of Art in Security Metric Research

Majority of the work in the security metrics domain is towards identifying a good security metric. Kun Sun et al. [50] identify a system for security metrics collections, management and visualization. They quantified the present vulnerabilities, historical trends in vulnerabilities, and predicted the future vulnerabilities in a service. They also corroborated it with network policy and measured how much a policy allows an attack to propagate through the network. Another study [51] takes into consideration the flaws in a network configuration that renders a network vulnerable to attack. They present a metric called VEA-bility security metric, which can help compare different network configurations and select the most secure one. The motivation given by the writers is that, it is the objective of any network administrator to have the least amount of vulnerabilities in his network and to achieve that every software and hardware system should be securely configured. The metric proposed by the authors enables different configurations to be compared with each other, thus helping in the selection of the most secure

configuration. Authors in [52] use metrics based on attack graphs. They measure the likelihood of successfully exploiting a network in terms of number of attackers reaching their goal. Another metric they define is the resistance an attacker faces in order to launch an attack. Status of temporal elements such as available exploits and patches for a service keep changing and a static attack graph cannot incorporate such a change. Dynamic nature of operational security and temporal elements of attack graphs are considered in [53]. In our observation different approaches have been taken to define a security metric and it varies across different organization so the question arises which metric provides best bang for the buck. To answer this important question we take an analytically extensive approach towards identifying a good set of metrics that can optimize the security posture of a network.

3. Network Setup and Collection of Metrics

For our study we monitored a college dormitory network that has an active user base of about thousand students. Data considered for this study spans a period of two months from March 2012 to April 2012.

4. Identifying Network Security Metrics

Efforts toward identifying security metrics have been going on for quite some time but security practitioners have never reacheched a consensu on what a good security metric should be. In his book [57] Andrew Jaquith defines a good metric should be; consistently measured, cheap to gather, expressed as a cardinal number or percentage, expressed using at least one unit of measure, and contextually specific.Using the criteria stated by him we identify eight metrics to test our scheme, this scheme can be applied to any number of

metrics to get an optimal set. Along with identification of metrics we also characterize them based on their statistics seen on the network.

4.1 User Agent Version

We use the user agent version as a security measure for the machine. The premise here is that number of vulnerabilities and exploits are associated with the particular version of a product [54]. A version with higher number of exploits and vulnerabilities makes a user machine more vulnerable to attacks. Figure 6.1 shows the statistics on most common user agents seen on the college network. This figure just gives the class of a user agent, for our analysis we have used specific versions that were detected for each class.

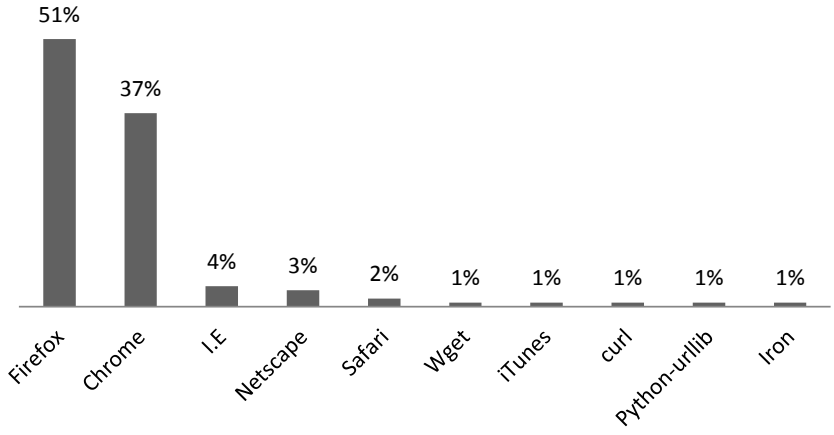


Figure 6.1: User agent statistics of the network.

4.2 Operating System Version

Similar to user agent we try to find vulnerable versions of the operating system to get the security state of a machine. The statistics we collected about the type of operating systems in use on the college network are shown in Figure 6.2.

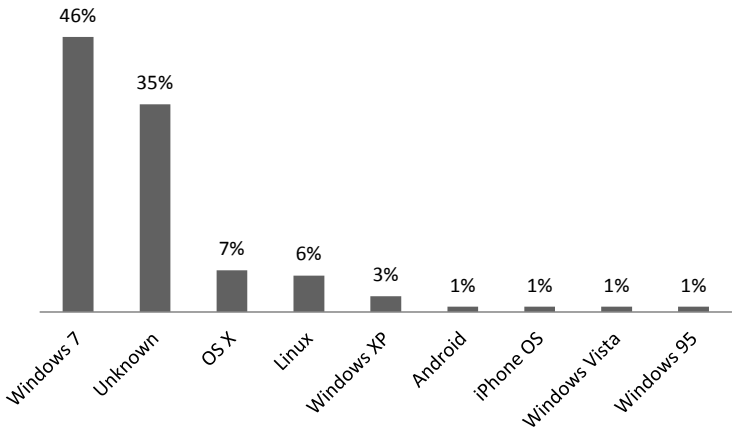


Figure 6.2: Operating system statistics on the network.

4.3 Software Updates

Software updates is closely related to the metrics defined above and is a very crucial measure for the security of a machine. A regular update regimen ensures that vulnerabilities on the system are kept to a minimum, thus reducing the attack surface and making the system more secure. We have used frequency of updates as a metric because a frequently updated machine leaves a very little room for the attacker to attack. Most updated software on the campus are shown in Figure 6.3.

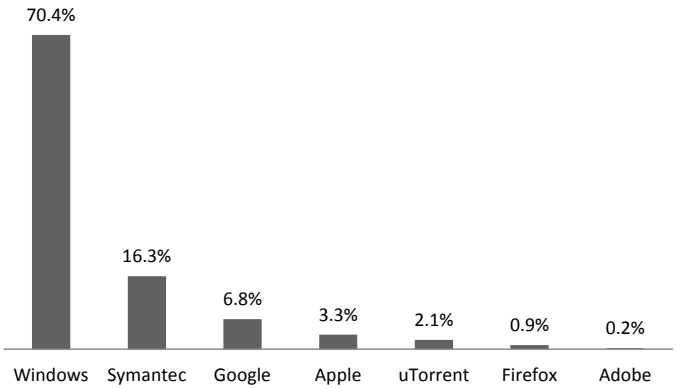


Figure 6.3: Update trends on the network.

4.4 Defense Mechanism and P2P

Having a defense mechanism such as a firewall, antivirus or an IDS/IPS system is very crucial for the security of the system. Since we are doing a passive analysis of the network traffic we have to infer the presence of these mechanisms on a host from network traffic. If there is a firewall installed on a host machine it will either do a "silent drop" in response to TCP SYN packets or will reply with a RST. Similarly an antivirus is detected when an antivirus client on a user machine updates itself for the latest definitions of threats.

Recently p2p is a favorable technique for bot command and control (C&C) and distribution. So p2p can be used an effective metric to measure malicious activity on the network. Statistics collected about defense mechanisms used on the network such as firewall and antiviruses and p2p usage is shown in Figure 6.4. Figure shows the percentage of machines for which these parameters were detected and not detected. It is clearly seen that a little over half of the users have some sort of defense mechanism installed.

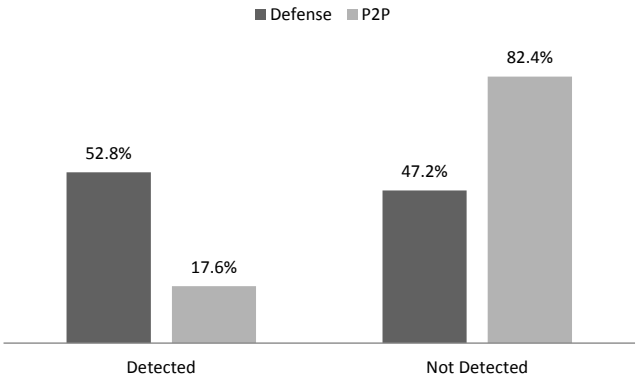


Figure 6.4: Defense mechanism and p2p activity on network.

4.5 Web Browsing

To determine the impact of a single site on the security of a host is difficult so we classify websites into different categories using the database provided by [55]. There are sites that host malware contents, have cross site scripting (XSS) vulnerabilities or simply can result in drive by download. Host security can be measured based on which category of sites a user is visiting. Web browsing statistics observed based on site classification are shown in Figure 6.5.

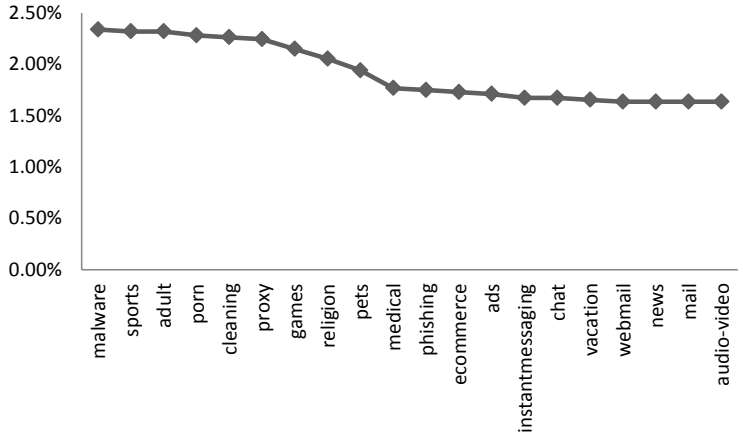


Figure 6.5: Site classification.

4.6 Content Download

Contents downloaded on a user machine can be a good indicator of an infection as binary download is an important part of the life cycle of a botnet. Following figure gives the distribution of contents that were downloaded by users during the study period.

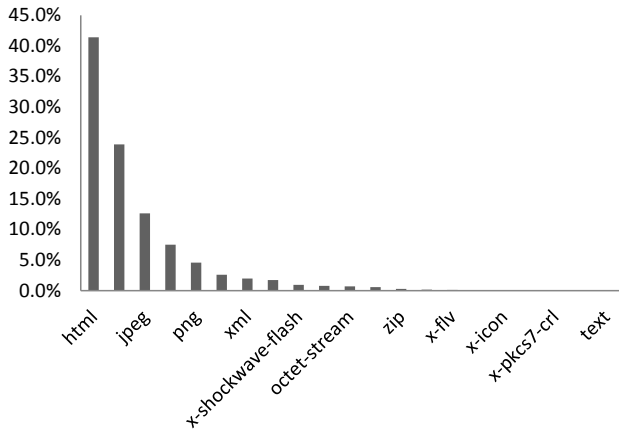


Figure 6.6: Percentage content download on campus network.

4.7 Geolocation of Destination

Recent studies have shown a correlation between the countries and the servers hosting a malware. Majority of the attack have originated from certain part of the world. Therefore we have used geolocation of the destination machine as a valuable metric. Figure 6.7 shows a distribution of most commonly talked to countries during our study.

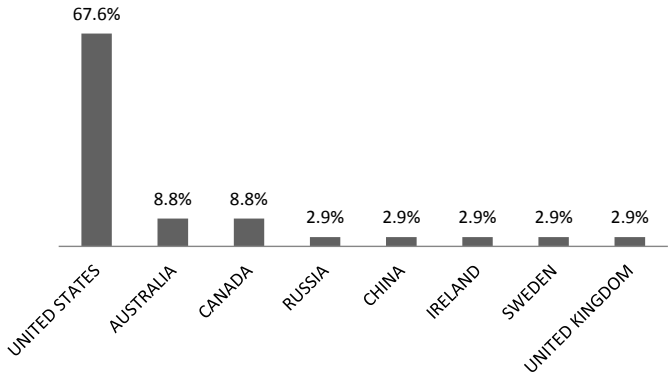


Figure 6.7: Geolocation of the servers talked to.

4.8 Scaling the Metrics for Analysis

We scaled the attributes mentioned in the previous section to help us with the mathematical analysis. Basic idea is to represent these attributes on a quantifiable scale

from zero to ten. Ten being the riskiest and zero being the safest. We use the age of the user agent and operating system to determine the score. Older version are assigned higher score because in [54] we observed that as a software gets older more vulnerabilities and exploits are found thus increasing the attack surface of the product. For scaling the update metric we use frequency of update, a more frequently updated machine is assigned a lower risk level. We scale web sites based on the category they belong. Sites such as; malware, warez, spam, and phishing which are known to have attacked clients are assigned high score, whereas harmless sites such as sports, news etc. are assigned lower scores. For content download type of content which is known to be used as an attack vector such as flash, javascript, exe etc. are assigned higher scores. Countries that orginated higher number of attacks are assigned higher scores. We used [58] to determine countries with top attack origins and to assign scores.

5. Infection profile from Bothunter

5.1 Bothunter Detection Mechanism

Bothunter [60] monitors the traffic between host in our network and the internet. It then tries to correlate this communication with the communication of a typical malware. If a communication from inside of our network matches with the steps identified as part of a malware life cycle it is declared infected. Each identified machine is assigned a score from 0.8 to 3.8 depending on how closely the communication matches with a typical malware communication. The data collected from Bothunter is used to classify metrics as belonging to an infected or a non-infected machine and this data is used by our decision engine as shown in Figure 6.8.

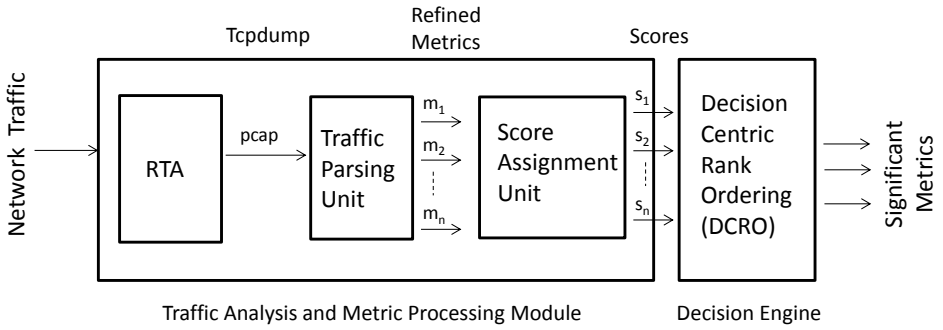


Figure 6.8: Architecture of decision centric identification and ranking system.

Once we have scaled our metrics, it is fed into the decision centric rank ordering system (DCRO) as shown in Figure 10. A detailed explanation of DCRO is given in the next section.

6. Decision Centric rank ordering (DCRO)

In this section we look at the problem of ranking security metrics in the order of their impact in determining network security. The ranking of security metrics will allow us to formulate effective solutions for managing network security as pointed out in the beginning of the paper. The problem of ranking security metrics is a difficult one. Several different approaches can be taken, and they have their pros and cons. One of the most obvious and theoretically viable approach is to perform correlative analysis on the network security data. The correlation between a certain metric and the network security score can give us an idea how strongly the network metrics impact network security. This can act as potentially useful information, but gives us a single number that defines the mutual interactions of security and security metrics. This kind of ranking is also prone to bias because of the paucity of data. A large data must be collected in order to establish definite correlations, and be able to rank them sufficiently apart from each other.

In order to understand the relationship between network security and security metrics we look for inspiration towards decisions theory and probabilistic distributions. Our hope is that my using this mathematical machinery, we can provide a much deeper, fuller, and clearer insight into how different security metrics impact security. We begin by assuming that x represent a certain metric, and C represents a certain class of machines that have been impacted through their presence in the network. C has a generic form given by C_k where k can take a value of either 0, or 1 which represents the infected and non-infected machine. The idea is to observe a certain security metric and determine whether the machine has been infected or not. In other words, we are interested in the probability that given an observation of a certain security metric, what is the probability that the particular machine belong to a particular class C_k. Using baye's theorem these probabilities can be expressed in the form,

$$p(C_k \mid x) = \frac{p(x \mid C_k)p(C_k)}{p(x)} \qquad (1)$$

$p(C_k)$ is the probability that a machine belong to the infected or not-infected class and is called the prior probability. $p(C_k \mid x)$ represents the probability that a machines belongs to a particular class C_k, once the security metric x has been observed for the machines. $p(C_k \mid x)$ is therefore referred to as the posteriori probability. If we want to minimize the number of decisions where we assign x to the wrong class C_k, we must intuitively choose the class having a higher $p(C_k \mid x)$. In other words, we must compare the posterior probabilities to arrive at the correct classification of whether a machine is infected or not.

The intuitive reasoning of comparing posteriori probabilities can be put on further mathematical basis. Suppose we want to minimize the probability of assigning a security

metric observation to the wrong class of machines. We need a rule that assigns each observation x of a security metric, to one of the two classes of infected and non-infected machines. Such a rule will divide the range of input values of the metric into two regions R_1 and R_2, such that all points in R_1 are assigned to the class C_1 and all points in the region R_2 are assigned to the class C_2. When we make a classification mistake, it means the machines belonging to R_1 are classified as belonging to class C_2 and vice versa. The probability of making this mistake can be represented as

$$p(\textbf{inference error}) = p(x \in R_1, C_2) + p(x \in R_2, C_1) \quad (2)$$

and since the probabilities are continuous functions of dynamic range of x we can represent the two terms of the sum as an integral of the joint pdf function as follows,

$$p(\textbf{inference error}) = \int_{R_1} p(x, C_2) + \int_{R_2} p(x, C_1) \quad (3)$$

It is clear that in order to reduce the inference error we must select the decision regions such that x is assigned to whichever class has the lower value of the integrand in (3). Since we know from the product rule of probability that

$$p(x, C_k) = \frac{p(C_k \mid x)}{p(x)} \quad (4)$$

we can say that $p(x, C_k)$ is equivalent to observing $p(C_k \mid x)$ since $p(x)$ is the common divisor for all classes k. Thus in order to decide which class to assign to any x, we must decide by observing which of the probabilities $p(C_1 \mid x)$ or $p(C_2 \mid x)$ is higher. We therefore determine a threshold for the decision based on when the two pdfs intersect each other. This is shown in the Figure below,

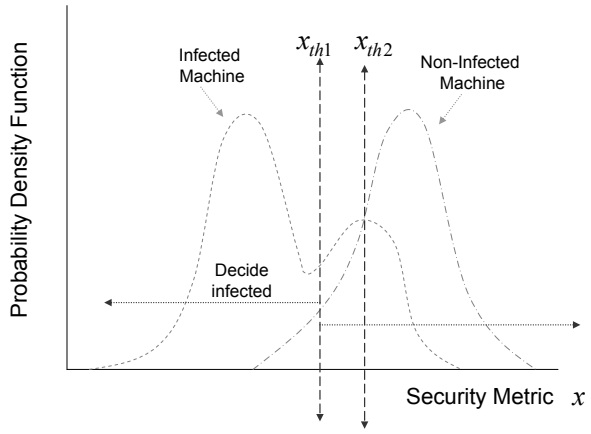


Figure 6.9: Pdfs of infected and non-infected machines.

Two different pdfs each representing the condition density function for the two classes is shown. A non-optimal threshold for the security metric is shown as x_{th1} and an optimal decision threshold which minimizes the probability of error is shown as x_{th2}. One can see from the figure that selecting the threshold as x_{th2} will yield the lowest probability of mis-classification.

From the above discussion we understand that the decision theory gives us an optimum decision threshold which leads to a minimum, but non-zero probability of error. When would this probability of error be zero? If the two pdfs shown in Figure are widely separated from each other on the parametric axis, the probability of mis-classification will gradually tend towards zero. We therefore define the difference between the two pdfs as pointing to the fact that the distance between the two pdfs of the security metric is a powerful indicator of the ability of the metric to impact the network security score. This is a very important idea, and to the best of our knowledge has never been exposed in the network community.

The separation between two probability density functions, each representing the conditional probability of infection, given the value of a certain security metric, can act as

a powerful descriptor of the most relevant metrics impacting network security. The question then naturally arises as to how can we define the difference between the two probability density functions. One of the methods can be the difference of the parameters describing the probability distribution. For example a Gaussian distribution has the parametric form given by

$$p(x) = \frac{1}{(2\pi\sigma^2)^{0.5}} \exp\left\{-\frac{(x-\mu)^2}{\sigma^2}\right\} \qquad (5)$$

where the distribution is characterized by two parameters μ and σ. We can define a parametric vector which defines a certain Gaussian distribution as being $a_p = \left[\mu_p \quad \sigma_p\right]$ while a similar parametric vector for another distribution $q(x)$ would be given by $a_q = \left[\mu_q \quad \sigma_q\right]$. We can define the difference between the two distributions as a difference of the two parametric vector $a_p - a_q$ and it can be used as measure of their separation, by taking the magnitude of the difference vector. The separation between the two pdfs can then be defined in an approximate fashion by writing,

$$\| p(x), q(x) \| = \sqrt{\left[a_p - a_q\right]^2} \qquad (6)$$

where the two parameters of the parameter vector μ and σ can be approximated from the data in a maximum likelihood fashion by writing,

$$\mu_{ML} = \frac{1}{n}\sum_{n=1}^{N} x_n \qquad (7)$$

and

$$\sigma_{ML} = \frac{1}{n}\sum_{n=1}^{N} (x_n - \mu_{ML})^2 \qquad (8)$$

where μ_{ML} and σ_{ML} are the sample mean and sample variance respectively and they tend towards the true mean and true variance of the distribution as N tends to infinity. It can be shown that the true parameters and the sample parameters are related to each other by

$$E[\mu_{ML}] = \mu \qquad (9)$$

and,

$$E[\sigma_{ML}] = \frac{N-1}{N}\sigma \qquad (10)$$

We observe that sampled based maximum likelihood estimates of the pdf tend towards the true values of parameters as N tends to infinity. So as the number of sample points grow, our estimate is closer to the ideal, and we can use (6) to calculate the distance between two probability distributions to come up with a measure of separability of observed patterns. If we need to form an unbiased estimate of the sample variance without resorting to a large sample size we can do so by writing,

$$\hat{o} = \frac{}{N-1}\sigma_{ML} = \frac{1}{N-1}\sum_{n=1}^{N}(x_n - \mu_{ML})^2 \qquad (11)$$

Although (6) combined with (7) and (8) along with the knowledge of (9) and (10) can give us expressions for the calculation and determination of separation between two density functions, what happens in the case of an arbitrary pdf. In general, the real world pdfs do not adhere to the Gaussian pdf, and we must have a way of determining the separation of an arbitrary pdf. We look to information theory for help and we find that there, we often need to estimate a certain density function, and the difference between the estimated and actual function is the additional amount of information which must be transmitted. This is referred to as kullback-leiber divergence or KL divergence, as is defined as

$$KL(p \| q) = -\int p(x)\ln q(x)dx - \left(-\int p(x)\ln p(x)dx\right) \qquad (12)$$

which can be written as

$$KL(p \| q) = -\int p(x)\ln\left\{\frac{p(x)}{q(x)}\right\}dx \qquad (13)$$

where $p(x)$ and $q(x)$ are the two probability distributions whose difference needs to be measured. KL divergence is an information theoretic criterion, as it is obtained by subtracting the fundamental entropy of the two distributions. This measure is not symmetric i-e in general

$$KL(p \| q) \neq KL(q \| p) \qquad (14)$$

and hence its not a norm in the conventional sense of norm, but it can still be used to measure the difference of two pdfs and performs reasonably well in differentiating the components in all situations.

Along with the means of estimating distance between pdfs, we also have to come up with a form of the pdf themselves. This form can be based on the parametric or non-parametric description. We can use histogram method by partitioning x into a number of bins, each of width Δ_i and then count the number of observations n_i lying within each bin. To turn this count into a probability density function we simply divide by the product of total number of observations, and the width of each bin. We can express by writing,

$$p_i = \frac{n_i}{N.\Delta_i} \qquad (15)$$

where p(x) is constant over each bin, and generally the bins are chosen to have equal width. Once the histogram density is calculated, we can use the density expressions for different metrics to estimate the difference of two probability distributions, and their potential for security evaluation.

7. Analysis to Determine Significant Metrics

In the light of theoretical development done in Section IV we now calculate the probability density functions for different security metrics. To reiterate, the underlying

theme of ranking these metrics is intuitively based on the idea of how much decision power they provide to the analyzer. If the metric can easily distinguish between infected and non-infected machines, it will be given a higher relevance score for network security. Below we will provide a qualitative analysis of different metrics collected from the network. The x-axis in the following figures will represent the scores and y-axis will represent the pdf. Higher the score more vulnerable to get infected.

7.1 User Agent

The graph for user agent data is shown in Figure 6.10. The figure consists of two probability density functions. The blue trace shows the probability density function for machines with a high infection score, whereas the red curve is for machines having a low infection score. In other words, the two curves are the conditional probability density functions for infected and non-infected machines. We observe that the two probability density functions are separated from each other, but also have a significant overlap. The decision of infection based on this metric alone, will have many false alarms, and missed detections. This is therefore a medium relevance metric, which has a clear boundary at the single point of intersection of two pdfs, but the integrals in (3) hold significant values beyond this boundary.

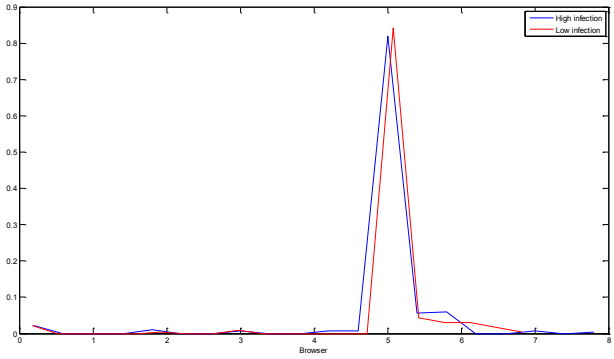


Figure 6.10: User agent.

7.2 Operating System

The next metric analysis is for operating systems. The

Operating system plays a critical role in determining security of a particular machine. The operating system metric can also provide important insights into the security trends for networks. The pdf data of the metric is plotted in Figure 6.11. The pdf shows that in the middle of the operating system range, the pdfs overlap each other almost entirely, rendering the metric un-usable for infection detection. However, towards the higher end of the operating system security score, the pdfs again depart from each other and establish a difference amongst themselves, and a decision boundary becomes available. This is a very important data, as it is telling us that the operating system score, which is based on their year of introduction, becomes irrelevant to security, once a certain number of years have elapsed. This could possibly be due to the fact that number of vulnerabilities are patched year after year and at a certain point in time, hackers and the developers are evenly matched, which leads to a blurring of the decision power of this metric. However, as we move towards high risk OS, this metric becomes progressively more important, as the decision can now be made based on the scale of OS score. We therefore categorize is as a medium low relevance to network security.

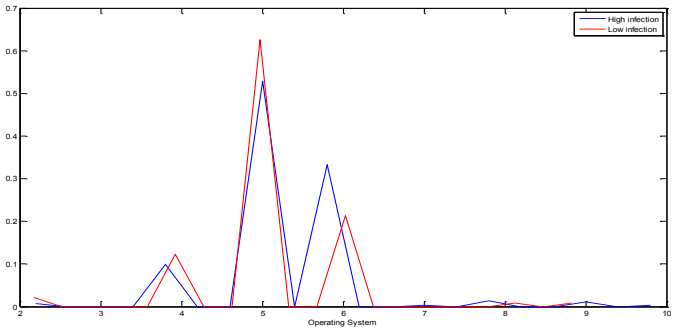


Figure 6.11. Operating System.

7.3 Software Updates

Updates on software can play ha role in keeping the

System immune to attacks. The software update metric shown in Figure 6.12 clearly demonstrates that, where the infected and non-infected profile are completely separated from each other. We therefore assign a high security relevance to this metric and it can be used to determine the infection possibility with high accuracy.

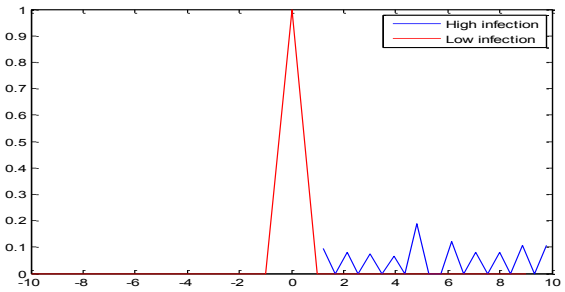


Figure 6.12: Software Updates.

7.4 Peer to Peer

Peer to peer metric graph shown in Figure 6.13 shows

That the conditional pdf profiles completely overlap each other. Therefore from this data, one can derive the conclusion that peer to peer is un-important in predicting the security score. We therefore assign it the lowest relevance to security.

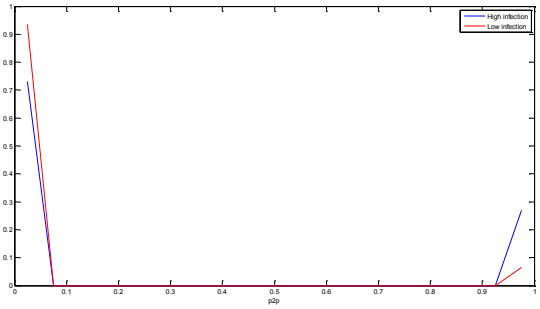


Figure 6.13: p2p.

7.5 Browsing Habits

Browsing habits can be an indicator of a machine's

Security threat as well. We see the pdf results of browsing habits in Figure 6.14. The distribution for the non-infection machines is confined to the left. This gives a clear demarcation for the maximum likelihood decisions but also gives a significant probability of false alarm because of the appreciable overlap of the infected distribution with the non-infected one. We will therefore categorize this metric as of medium high relevance to the security profile.

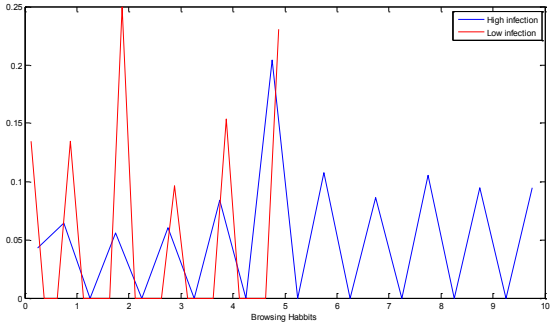


Figure 6.14: Browsing Habits.

7.6 Defense Mechanism

The graph for defense mechanism distribution is

Shown in Figure 6.15, and demonstrates that a clean decision boundary can be drawn between two distributions, as the infected is skewed completely to the left and the conditional distribution of non-infected machines is skewed completely to the right. This metric gets a high rating for its relevance to security.

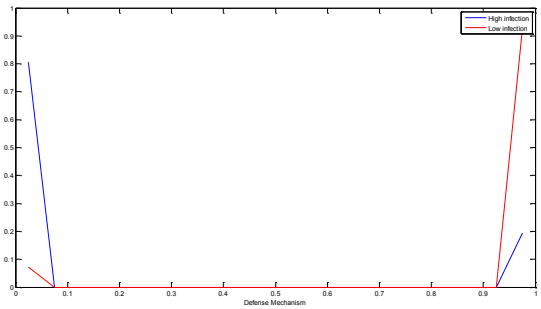


Figure 6.15: Defense Mechanism.

7.7 Contents Downloaded

Figure 6.16 shows that there is not a very clear boundary for contents downloaded between the infected and non-infected machines. Therefore this attribute can't be selected as a good metric.

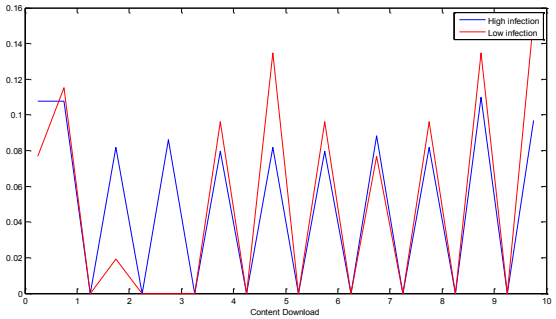


Figure 6.16: Contents downloaded.

7.8 Geolocation

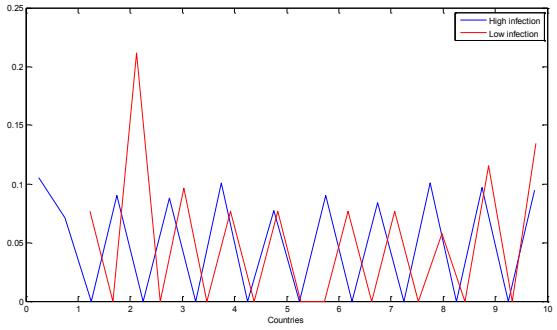


Figure 6.17: Geolocation of server.

For geo location there is a distinct boundary in some regions where as other regions are not as much distinguishable.

8. Conclusion

This chapter was geared towards providing a framework for the identification of significant security metrics that directly impact the security profile of machines contained within the network. We first presented a scheme to passively identify a set of security metrics from network traffic. We then proposed a mathematical engine that can help us identify the most significant metrics from the collected data. This mathematical engine is intuitively based on the reasoning that the most significant metrics are those, which can play a part in determining network security profile in the most un-ambiguous manner. We make use of probabilistic analysis, and decision theory methods along with information theoretic measures to come up with a ranking for the collected metrics. Once these metric rankings are determined, administrative efforts can be focused to optimize security. We also characterized the network traffic and present trends that we observed on the network during our study.

CHAPTER 7

COMPARISON OF SCHEMES FOR PREDICTION OF COMPROMISE

1. Introduction

Careful identification of attributes that give us insight into the security of the network have also enabled us to determine the future security state and the risk associate to the network infrastructure. Since we have attributes and corresponding vector that tells us if a machine is infected or not we can use this profile to observe if any machine in the future is found to have the same attribute set than it is very likely that it will also get infected. Visual representation of the infection profile created is shown in the following equation.

		Attribute1	Attribute2	.....	Attribute N
X =	Computer1	x1	x2		xn
	Computer2	x1	x2		xn
	.				
	.				
	ComputerN	x1	x2		xn

		Infection
Y =	Computer1	yes
	Computer2	no
	.	.
	.	.
	.	.
	ComputerN	yes

2. Neural Network Learning Methodology

For learning the patterns for this problem we have employed a multi-layered perceptron methodology. The multilayered perceptron commonly referred to as MLP can be used to separate patterns using a nonlinear manifold. What this basically means is that the data is approximated using nonlinear basis functions. The structure of MLP is multi-layered, which gives it a universal approximation capability. Inputs are used at each layer to create outputs using linear and nonlinear operators. A network also had what is referred to as input, output and hidden layers. The input layer of course preprocesses the inputs for passing through the whole network. The output layer generates the output and the hidden layers are used to process the inputs to create better approximations to the underlying pattern of the data. The process of output formation is now detailed in the next paragraphs.

Let x represent different inputs to the network which needs to identify patterns, classes, or develop prediction functions for an input-output mapping. Let the number of such inputs be d and the weights of the network be given by w. Assuming that the network only has one hidden unit, the inputs to such a unit are computed by using

$$a_j = \sum_{i=1}^{d} w_{ji}^{(1)} x_i + w_{j0}^{(1)} \qquad (1)$$

It can be seen from (1) that outputs are formed using a weighted combination of the inputs along with a constant term $w_{j0}^{(1)}$ which is referred to as the bias of the network. The weights $w_{ji}^{(1)}$ denote the weights in the first layer of the network. The outputs of the hidden layer, can then be obtained using a nonlinear activation function. This nonlinear activation function is the one that gives universal approximation ability to such networks. The inputs z of the hidden layer can then be formed using,

$$z_j = g(a_j) \qquad (2)$$

Where $g(.)$ is a nonlinear operator on the inputs a_j. The $g(.)$ can be selected to be a $\tanh(.)$ or a sigmoidal function. Any other different nonlinear function is also possible depending upon the problem definition. We chose a sigmoidal nonlinearity for this experiment. The ouptuts of the network are obtained using a weighted combination of the nonlinearly transformed weighted inputs. We can thus represent the mathematical expression of the output layer as

$$a_k = \sum_{j=1}^{M} w_{kj}^{(2)} x_j + w_{k0}^{(2)} \qquad (3)$$

Where $w_{kj}^{(2)}$ represent the weights of the second layer, and z_j are the inputs to this layer. We can form the final output using another nonlinear mapping function on the inputs a_k by writing

$$y = \tilde{\xi} \qquad (4)$$

If we combine (1), (2) and (3), the output of the multiplayered perceptron can be written using as expression of the form,

$$y_k = \tilde{\xi} \left(\sum_{j=1}^{M} g\left(\sum_{i=1}^{d} w_{ji}^{(1)} x_i \right) \right) \qquad (5)$$

We see that y_k represents a nonlinear functions of the inputs x_i and can be used to find an arbitrarily complex nonlinear mapping using more hidden layers and corresponding weights. Note that in formulating the final expression for y_k, the bias terms of the

network has been absorbed into the linear expression by setting the input $x_0 = 1$ permanently.

The NLP can be trained using a variety of gradient descent approaches, the most popular of which is the back-propagation method. Out experiment uses a back propagation method to train the network. The results of training the network using only two hidden layers, and sigmoidal nonlinearities will be shown later.

In the Figure 7.1 the confusion matrix of the analysis is displayed. Confusion matrix summarizes the performance of the algorithm in terms of number of false positives and true negatives, and displays them in a single format. In this table we see the results of target class, which is the actual number of infected computers versus output class, which represents the predicted number of injected computers. The second columns contains the critical data set. This column stands for all the machines that were actually infected. The rows of this column tell the classifier performance. We see that 51.6 percent of the machines which were infected were correctly identified as infected by the algorithm. However it also made a number of mistakes, but identifying 48.6 percent of the infected machines as benign and healthy (by assigning them a zero). The total performance of the algorithm was therefore marginal.

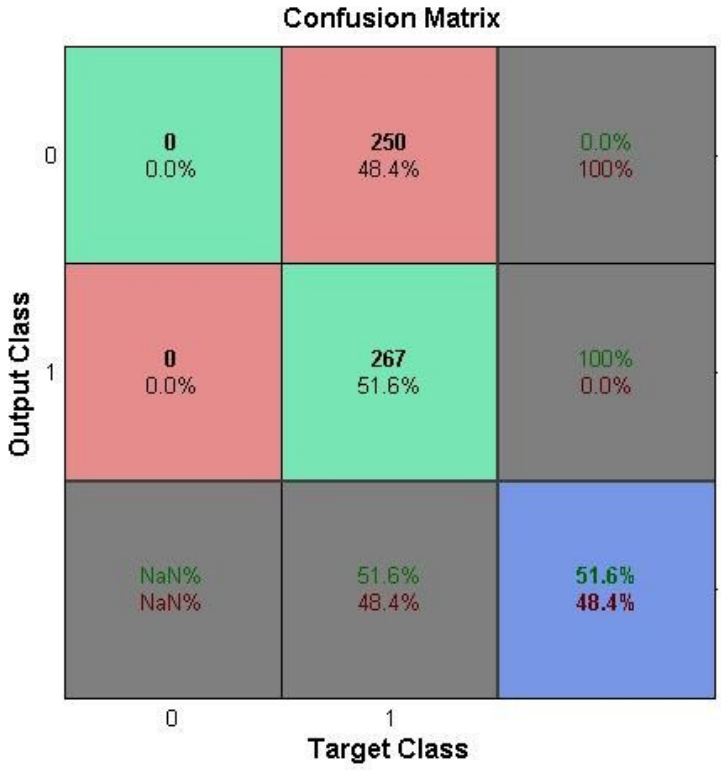


Figure 7.1: Confusion Matrix.

Figure 7.2 represents the distribution of errors from the classifier. This graph enables us to see the decay of errors as we move away from the classification boundary. This is useful in indentifying the behavior of algorithm away from the critical points, and tells us that the classification boundary is well-defined. If the errors decay monotonically away from the classification boundary, we can rest assured that we are operating at the optimal boundary point and not much can be done to increase the performance of the system. If the error graph oscillates across the boundary point, we know that a fundamental revision of the classification criterion is needed in order to refine the statistical estimate that determines the decision boundary.

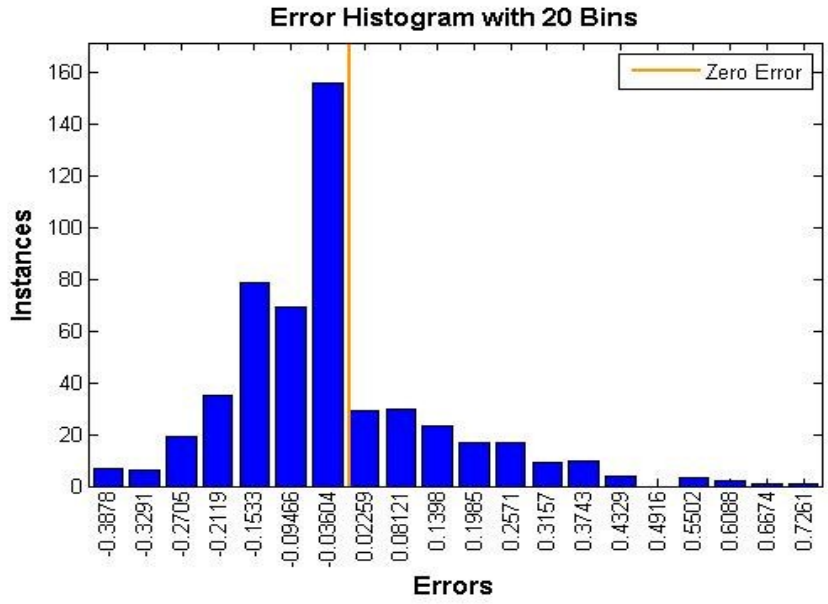


Figure 7.2: Distribution of errors from the classifier.

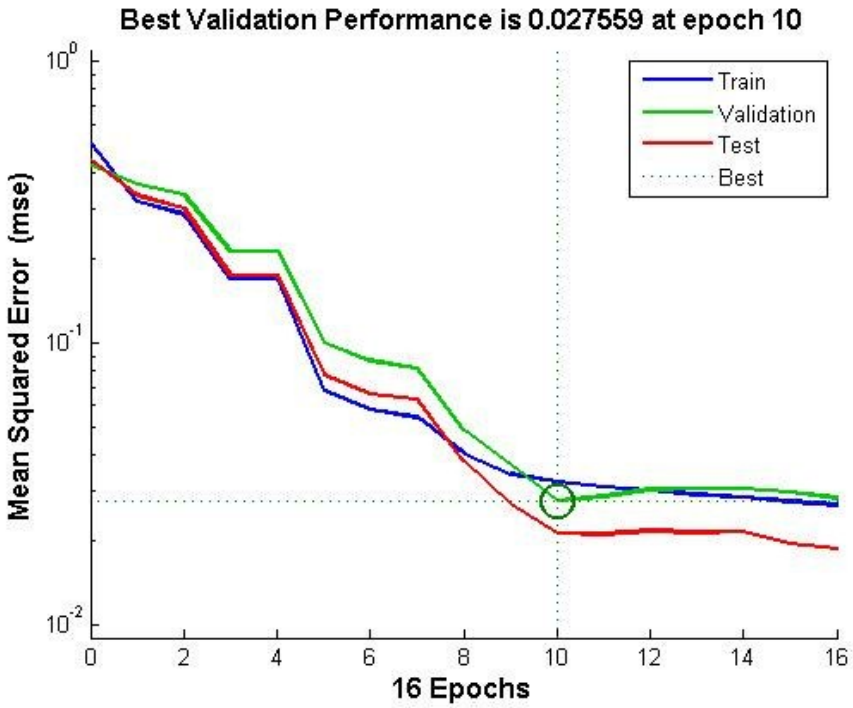


Figure 7.3: Mean square error vs time of the algorithm.

The Figure 7.3 shows the mean square error of the algorithm vs time. This is a useful stability test to make sure that the algorithm is not diverging with time. As time passes, and more training is performed on the algorithm, the mean square error continues to go down. The mean square error represents the fundamental statistical criterion on which many of the training methods like steepest descent, least mean square and recursive least squares are based. We see from the graph that error converges to its steady state value after 10 epochs. Moreover, the validation curve shows nearly the same behavior as the training curve, and reaches the minimum mean square error point below the training curve (at the optimal epoch time). This shows that the training method and the algorithm is stable, and no divergence shall be encountered if more training data is added into the system.

3. Multi-variable Regression

We used the regression modeling available in Weka machine learning software. Similar to mathematical model presented in the Baysian regression case it uses least square estimation to evaluate a model that can predict an outcome given new parameters. Following figure represents the out of Weka when data from five hundred and seventeen hosts was presented to it with the attributes that have been marked in the figure. The regression engine gave us the following formula as shown in Figure 7.4.

Infectionscore = 0.0344 * p2p - 0.2257 * defense + 0.532 * updates + 0.0541 * browsing + 0.0044 * countries - 0.256

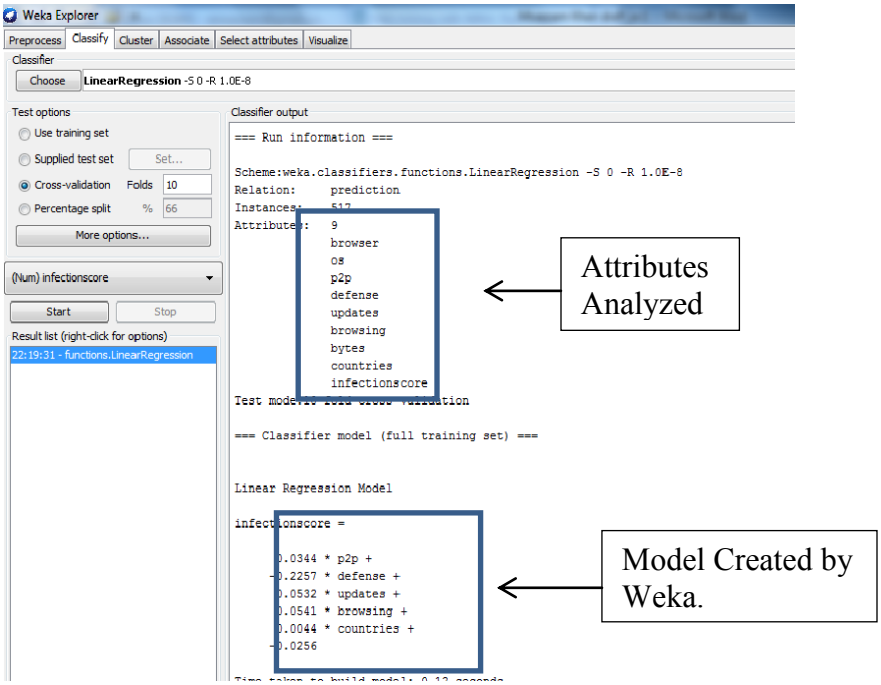


Figure 7.4 : Regression results from Weka.

3.1 Interpretation of Regression Model

Regression model obtained gives us some valuable information according to the model:

- OS, browser and bytes transferred and received by a user aren't playing any role in the infection.
- Defense installed on the machine is playing a negative role, meaning higher the defense score lower will be the infection score which is logical.
- Compared to other attributes countries talked to play some but not a big role towards infection.
- Similarly if user is not doing updates and browsing questionable sites that he/she will have higher infection score.

4. Bayesian Regression

4.1 Introduction

Following figure represents Bayesian regression model for the analysis of our selected attributes.

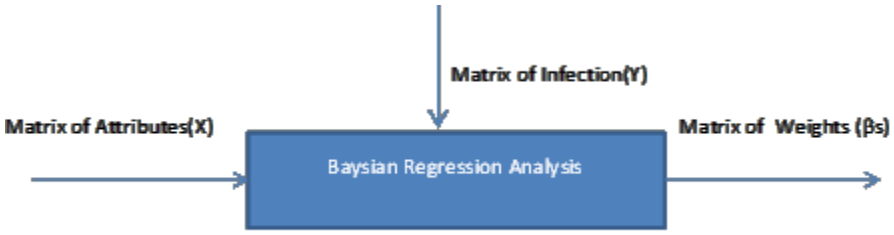


Figure 7.5: Bayesian model.

Once we have the matrix of weights (β) we can use it for the following purposes.

1) Reduce the number of attributes for analysis.
2) Find the values of attributes for final risk calculation.
3) Determine the most significant attributes to focus our security measures.

4.2 Mathematical Explanation of the Model

For the prediction purposes we use linear regression model, which states that:

A linear regression model is a particular type of smoothly changing model for p(y|x)

that specifies that the conditional expectation E[Y |x] has a form that is linear

in a set of parameters:

$$\int yp(y|x)dy = E[Y|x] = \beta 1x1 + \ldots + \beta p\ xp = \beta^{T} x \qquad (6)$$

Which is the conditional probability of a machine being affected given a set of attributes.

Where 'x' are the values of our attributes specified in preliminary research section.

β's are the weights assigned to each attribute by our model. How we arrive to the values of β will be explained in the following section.

The normal linear regression model specifies that, in addition to E[Y |x] being linear, the sampling variability around the mean is i.i.d. from a normal distribution:

$\epsilon_{1,\ldots,}\epsilon_n \sim$ i.i.d normal $(0,\sigma^2)$

$Y_i = \beta^T x_i + \epsilon_I$ (7)

This model provides a complete specification of the joint probability density of observed data y1, . . . , yn conditional upon x1, . . . , xn and values of β and σ^2:

p(y1, . . ., yn | x1, . . .xn, β and σ^2)

$=(2\pi\ \sigma^2)^{-n/2}$ exp{ -1/2 $\sigma^2 \sum_{i=1}^{n} (yi - \beta\ xi)^2$} (8)

Another way to write this joint probability density is in terms of the multivariate normal distribution: Let y be the n-dimensional column vector (y1, . . . , yn)T , and let X be the n × p matrix whose ith row is xi. Then the normal regression model is that

{y|X, β, σ^2} ~ multivariate normal (X β, σ^2), where I is the p × p identity matrix and

$$\mathbf{X}\boldsymbol{\beta} = \begin{pmatrix} \boldsymbol{x}_1 \rightarrow \\ \boldsymbol{x}_2 \rightarrow \\ \vdots \\ \boldsymbol{x}_n \rightarrow \end{pmatrix} \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_p \end{pmatrix} = \begin{pmatrix} \beta_1 x_{1,1} + \cdots + \beta_p x_{1,p} \\ \vdots \\ \beta_1 x_{n,1} + \cdots + \beta_p x_{n,p} \end{pmatrix} = \begin{pmatrix} \mathrm{E}[Y_1|\boldsymbol{\beta}, \boldsymbol{x}_1] \\ \vdots \\ \mathrm{E}[Y_n|\boldsymbol{\beta}, \boldsymbol{x}_n] \end{pmatrix}$$

The density depends on B through the residuals (yi – β^T xi). Given the observed data, the term in the exponent is maximized when the sum of squared residuals,

$$\mathrm{SSR}(\boldsymbol{\beta}) = \sum_{i=1}^{n}(y_i - \boldsymbol{\beta}^T \boldsymbol{x}_i)^2 \qquad (9)$$

is minimized. To find the value of β at which this minimum occurs it is helpful to rewrite SSR(β) in matrix notation:

$$\mathrm{SSR}(\beta) = \sum_{i=1}^{n}(yi - \beta\, xi)^2\} = (y - X\beta)^T(y - X\beta)$$

$$= y^T y - 2\,\beta^T X^T y + \beta^T X^T X\beta$$

Solving for the minimum value of SSR(B)

$$\frac{d}{d\beta}\mathrm{SSR}(\beta) = \frac{d}{d\beta}(y^T y - 2\,\beta^T X^T y + \beta^T X^T X\beta)$$

$$= -2\,X^T y + 2\,X^T X\beta \qquad \text{, therefore}$$

$$\frac{d}{d\beta}\mathrm{SSR}(\beta) = 0 \iff -2\,X^T y + 2\,X^T X\beta = 0$$

$$\iff 2\,X^T y = 2\,X^T X\beta$$

$$\iff \beta = (X^T X)^{-1} X^T y \qquad (10)$$

The value $\beta ols = (X^T X)^{-1} X^T y$ is called the "ordinary least squares" (OLS) estimate of β , as it provides the value of β that minimizes the sum of squared residuals. This value is unique as long as the inverse$(X^T X)^{-1}$ exists.

4.3 Application of Baysean Regression Model on Subset of Data

We have selected the following attributes to test Baysean Regression.

- Firewall
- OS Updates
- Cookies
- P2P
- Weak Protocols
- Malicious Sites

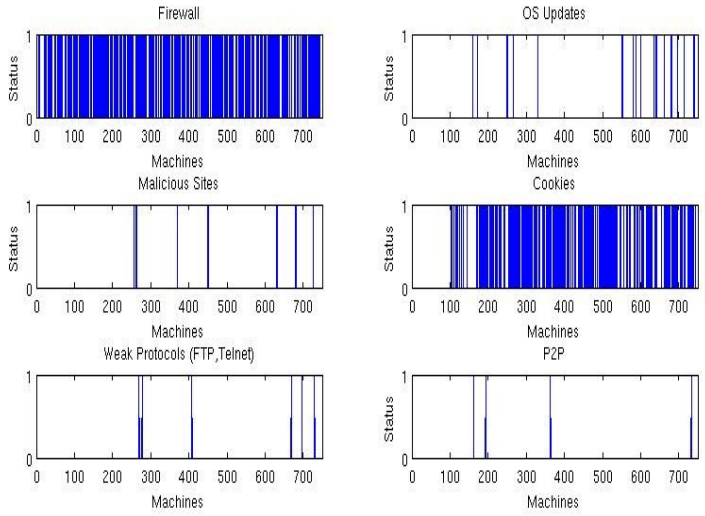
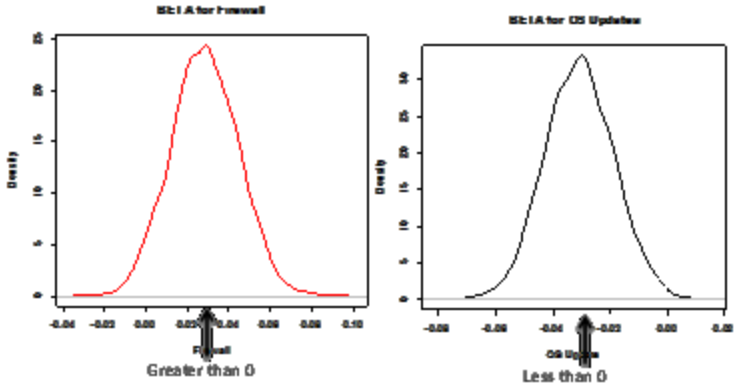


Figure 7.6: Statistics of attributes on the network.



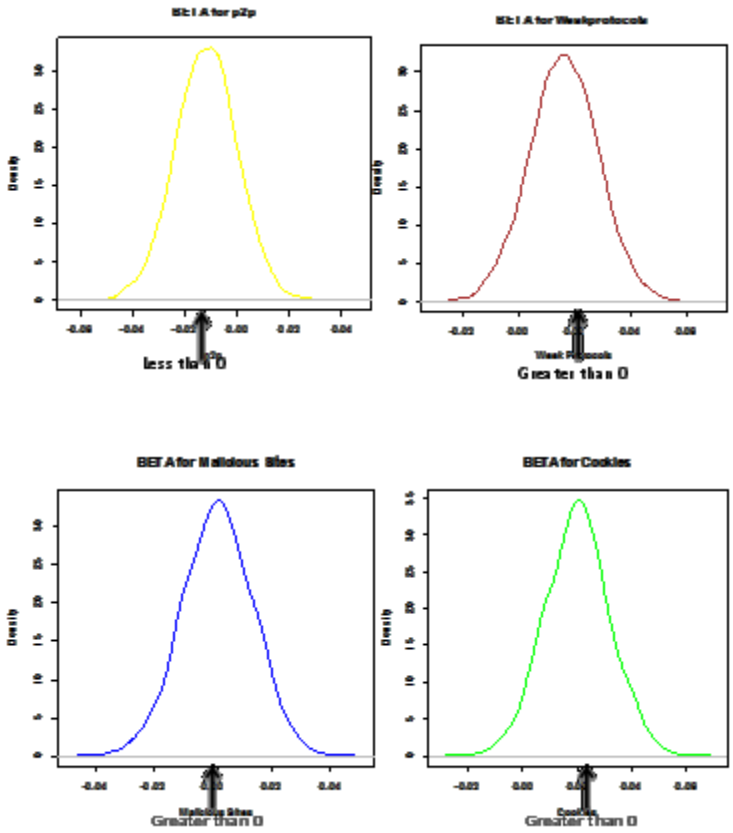


Figure 7.7: Distribution of Betas of the metrics.

4.4 Observations

Cookies, weaker protocols and absence of firewall are the factors that play an important role in infecting a machines.

5. Attribute Clustering

Using Weka's clustering engine we can divide the samples into clusters with unique attributes. This gives us an indication about the way certain set of attributes behave and we can group them accordingly. We have to carefully choose the number of cluster to divide. If too many samples are falling into same clusters than we have too few clusters

for the design. Similarly if we have some clusters with very few or no elements than we have selected more number of cluster than needed. For example following figure shows that choosing five clusters puts a lot of elements in cluster # 2 and 3. Which isn't a very good indicator of the behavior of the users. So ran another model with ten clusters and the results are shown in the in Figure 7.8.

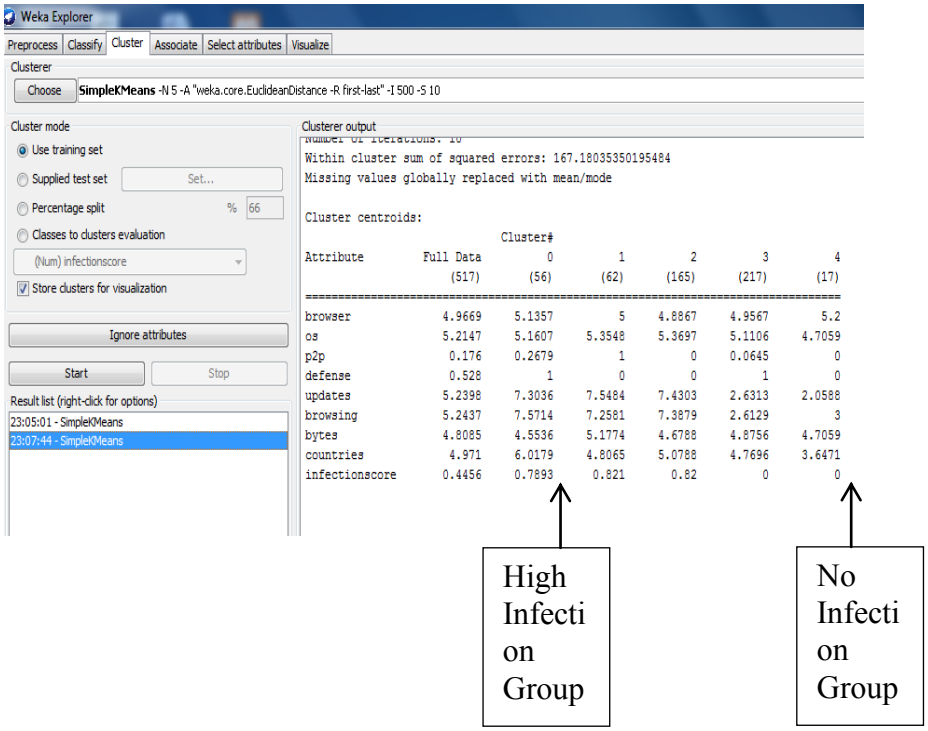


Figure 7.8: Results with five clusters.

Looking at a particular cluster we can deduce some characteristic from each. For example in the above figure we can classify cluster 0 as a group of user with high infection and we can see that those with high infection had high scores on malicious browsing and low score on defenses. Whereas the group with no or low infection were not using p2p applications and had a lower score on malicious browsing.

5.1 Increasing Number of Clusters

Following figure 7.9 is the result with ten clusters.

Number of iterations: 13
Within cluster sum of squared errors: 105.89390302037256
Missing values globally replaced with mean/mode

Cluster centroids:

		Cluster#								
Attribute	Full Data	0	1	2	3	4	5	6	7	8	9
	(517)	(41)	(62)	(80)	(63)	(17)	(52)	(56)	(85)	(29)	(32)
browser	4.9669	5.122	5	4.88	5.1048	5.2	5.0231	4.8911	4.8929	5.0448	4.6938
os	5.2147	5.122	5.3548	5.35	5.2698	4.7059	4.9231	4.9821	5.3882	5.3448	5.1875
p2p	0.176	0	1	0	0	0	0	0	0	1	0
defense	0.528	1	0	0	1	0	1	1	0	1	1
updates	5.2398	7.1951	7.5484	7.5	2.5079	2.0588	2.8654	2.875	7.3647	5.2414	2.0313
browsing	5.2437	7.5854	7.2581	7.325	2.7302	3	2.2692	2.8393	7.4471	5.4483	2.2813
bytes	4.8085	4.4878	5.1774	7.525	7.4444	4.7059	2.0769	2.5893	2	5.0345	8.1563
countries	4.971	5.8049	4.8065	5.3875	3.0794	3.6471	1.6731	7.3393	4.7882	5.8966	8.4688
infectionscore	0.4456	0.8024	0.821	0.81	0	0	0	0	0.8294	0.3897	0

Figure 7.9: Increasing the number of Clusters.

6. Conclusion

We have used methods like neural networks, simple and Bayesian regression and clustering. Each method has its pros and cons. The simplest and the most elegant of the techniques is the least squares regression, (so called simple regression). The least square is a powerful method for training classifiers and can lead to good classification. Basically the method works by forming an optimization criterion based on the square of the errors from different samples. This method suffers from the outlier effect, whereby the data points falling way outside the cluster of training points can lead to large error terms. Since squared error is used as an optimization criterion this can lead to the buildup of large offsets from the optimal solution. It also suffers from the curse of dimensionality problem, where the number of unknown parameters explode exponentially with the dimensionality of the problem.

The Bayesian regression circumvents some of the problems of least squares estimator by assuming an apriori distribution on the set of unknown parameters of the problem. The exact relationship of the distribution needs not be known, and the Bayesian classifier will learn it on the fly using the training data sets. Bayesian classifier can help with the reduction of offsets associated with the least squares regression, and can lead to optimal solutions in many instances. The form of Bayesian distribution is always a hypothesis justified by experimental data and has been hotly contested as a means of parametric estimation by different camps of statisticians. The Bayesian classifier can somewhat circumvent the explosion of dimensionality by using clever statistical criterions which measure the complexity versus performance tradeoff.

The neural network classifier does a clever trick of using the data itself to form an inference of the exact basis functions to use for parametric estimation. In other words we are no longer restricted to using an apriori model to perform parametric estimation, but rather we learn the structure of the model on the fly. By using this approach the method is able to closely adapt itself to the particular dataset and can have the remarkable property to become very accurate as the training data increases. Unfortunately the accuracy of the method may beat other methods in cases of very large training data and very long network pruning, but will not be effective on short data bursts or where the statistics of the data maybe changing on the fly. For a constant data structure this may turn out to be an ideal technique, but for instances where the intrinsic structure of the problem is changing all the time, this technique may not be very suitable.

We observe that in the particular case of determining network health and security, the nature of the problem is extremely variable. The structure of the network and the creation

of different security problems is a very dynamic process and can be very faster than the learning time constants of complex networks. It is therefore speculated that adaptive, online least square methods like recursive least squares, recursive oja's rule, can prove to be invaluable in giving instantaneous indications of network health. The time constants of these adaptive methods can be scaled for the particular problems at hand, and can be a topic of future research.

Comparing the results from the schemes above we can see that each scheme helps us in prediction in a different way. For example by looking at the clustering we can group users based on infection or not and draw conclusions by observing the parts that attributes are playing in that cluster or group. Regression gives us a model considering the significant attributes and their weights so that values for a new user can be plugged into the model and infection score can be calculated. Similarly, Bayesian Regression gives us the most significant attributes and their corresponding weights.

CHAPTER 8

CONCLUSION

This work quantizes network security by identifying security metrics from network traffic and user behavior that can effectively envisage the security posture of an organization. To prove the effectiveness of the selected metrics we analytically showed that there is a strong correlation between them and the infection.

Main observations of characterization study are following:

- Users are active at a certain time, communicate to an average number of servers, and communicate certain number of bytes. If these parameters deviate from normal they can indicate a suspicious activity and thus can be used as a security measure.
- Social networking sites are the most popular. The average connection duration is very short in length. Site classification indicates that majority of sites accessed are questionable in nature and could result in infections with high probability.
- Users indicate a certain pattern in sites they browse, contents they download, time they spend on a site and these parameters can be used to measure their security level. Web search classification also indicates such a behavior similar to Web browsing.
- Very few users on the network show a consistent update behavior. Most of the updates that were made were configured automatically.
- A majority of the attacks events happen at a particular time and fall into the bad traffic category and involve contacting known malicious servers.

We not only characterized the network attributes for the identification of security metrics but we also compared those attributes across infected and non-infected profiles. Main

observations from analysis of metrics for infected and non-infected profiles are following.

- Machines that were regularly updating were less infected than the machines that weren't frequently updated.
- Users that were visiting questionable and malicious sites were highly likely to get infected than users not visiting those sites.
- Users on file and photo sharing sites were more likely to get infected.
- Users with communications to certain countries were more likely to get infected.

Some of our observations may be very obvious but this kind of analysis gives us a clear idea about the attributes that are playing a significant role towards infection in our studied network. And as reiterated many times before this helps us prioritize our resources and focus only on those attributes.

We also presented an analytical approach for identifying the significant metrics from the set of attributes identified from the characterization process. We tested a subset of attributes we collected from the campus network and applied a decision centric based scheme to identify the most significant attributes. The benefits of this scheme are given below:

- It presented an analytical approach towards defining security metrics compared to previous approaches this approach can help in standardization of metrics across organizations.
- It identified the most significant metrics so that all the administrative efforts and resources can be channeled towards them.
- It minimized the cost by taking care of areas that really matter the most.

- It identified a set of attributes that can be used in a model to predict future infections.

The next part of this work focuses on using the identified metrics for risk analysis and prediction of infection. Towards this goal we applied several schemes for the prediction of future infection. Although, none of the schemes can predict with high degree of certainty, each presents a different perspective and more information from the other. For example,

- Using multi variable regression we can identify the significant attributes and a model that can be used to predict future infections.
- Clustering can help us group representing a certain characteristics. This can help use with the prediction if a user falls into a particular cluster.
- Bayesian Regression can use a prior to give us model and significant attributes similar to simple regression even when we don't have a historical data.
- Neural networks can train themselves based on the pattern in the dataset.

Although we tried to identify as many metrics that could be collected from network traffic our list is not comprehensive. We had initially identified thirty two metrics for investigation but were not able to collect them from the traffic for reasons such as encryption, no access to user machines, not being allowed for active probing, probabilistic nature of detection tools etc. But with further investigation more metrics can be identified, tools can be improved for accuracy of detection. So our suggestions for future work are following.

- Setting up a controlled test bed so that every parameter can be deterministically determined and with enough number of machines to account for good data set for machine learning schemes and statistical analysis.
- Exhaustive collection of metrics and then using the DCRO scheme that we presented to identify the metrics that are most effective.
- We relied on Bothunter for detection of compromised machines. Bothunter predicts about the compromise based on anomalous network traffic so the detection is non-deterministic and the fact that a machine is infected or not is a deterministic attribute which can be obtained. We recommend doing analysis with the infection knowledge obtained deterministically. This could be through the testbed option suggested above or with the collaboration of network administrator.

REFERENCES

[1] John Hauser and Gerald Katz. Metrics: You are what you measure! European Management Journal, 16(5):517–528, October 1998.

[2] Jim Alves-Foss and Salvador Barbosa. Assessing computer security vulnerability. SIGOPS Oper. Syst. Rev., 29(3):3–13, July 1995.

[3] Bob Blakley, Ellen Mcdermott, and Dan Geer. Information security is information risk management.In NSPW ’01: Proceedings of the 2001 workshop on New security paradigms, pages 97–104, NewYork, NY, USA, 2001. ACM.

[4] C. Wang and W. Wulf. Towards a framework for security measurement. In NISSC, 1997.

[5] Carlos Villarrubia, Eduardo F. Medina, and Mario Piattini. Towards a classification of security metrics. In WOSIS, pages 342– 350, 2004.

[6] Rodolphe Ortalo, Yves Deswarte, and Mohamed Kaâniche. Experimenting with quantitative evaluation tools for monitoring operational security. IEEE Trans. Softw. Eng., 25(5):633–650, September 1999.

[7] James W. Meritt. A method for quantitative risk analysis. In Proceedings of the 22nd National Information Systems Security Conference, 1999.

[8] Nirnay Ghosh , Saurav Nanda , S. K. Ghosh, An ACO based approach for detection of an optimal attack path in a dynamic environment, Proceedings of the 11th international conference on Distributed computing and networking, January 03-06, 2010, Kolkata, India.

[9] Elie Bursztein , John C. Mitchell, Using strategy objectives for network security analysis,Proceedings of the 5th international conference on Information security and cryptology, December 12-15, 2009, Beijing, China.

[10] David Basin, Felix Klaedtke, and Samuel. Monitoring security policies with metric first-order temporal logic. In *Proceeding of the 15th ACM symposium on Access control models and technologies* (SACMAT '10). ACM, New York, NY, USA, 23-34

[11] Haihui Ge; Lize Gu; Yixian Yang; Kewei Liu; , "An attack graph based network security evaluation model for hierarchical network," Information Theory and Information Security (ICITIS), 2010 IEEE International Conference on , vol., no., pp.208-211, 17-19 Dec. 2010

[12] Anming Xie; Zhuhua Cai; Cong Tang; Jianbin Hu; Zhong Chen; , "Evaluating Network Security With Two-Layer Attack Graphs," Computer Security Applications Conference, 2009. ACSAC '09. Annual , vol., no., pp.127-136, 7-11 Dec. 2009

[13] Yinqian Zhang; Xun Fan; Zhi Xue; Hao Xu; , "Two Stochastic Models for Security Evaluation Based on Attack Graph," Young Computer Scientists, 2008. ICYCS 2008. The 9th International Conference for , vol., no., pp.2198-2203, 18-21 Nov. 2008

[14] Holm, H.; Ekstedt, M.; Andersson, D.;, "Empirical Analysis of System-Level Vulnerability Metrics through Actual Attacks," Dependable and Secure Computing, IEEE Transactions on , vol.9, no.6, pp.825-837, Nov.-Dec. 2012

[15] Hui Wang; Roy, S.; Das, A.; Paul, S.; , "A framework for security quantification of networked machines," Communication Systems and Networks (COMSNETS), 2010 Second International Conference on , vol., no., pp.1-8, 5-9 Jan. 2010

[16] Rass, S.; Schartner, P.; , "Game-Theoretic Security Analysis of Quantum Networks," Quantum, Nano and Micro Technologies, 2009. ICQNM '09. Third International Conference on , vol., no., pp.20-25, 1-7 Feb. 2009

[17] Wang Chun-zi; Huang Guang-qiu; , "A new method for network threat quantification analysis," dvanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on , vol.1, no., pp.V1-601-V1-605, 20-22 Aug. 2010

[18] Andrea Atzeni and Antonio Lioy. Why to adopt a security metric? A brief survey. In Quality of Protection, 2005

[19] Victor-Valeriu Patriciu, Iustin Priescu, and Sebastian Nicolaescu. Security metrics for enterprise information systems. Journal of Applied Quantitative Methods, pages 151–159, 2006

[20] Marianne Swanson, Nadya Bartol, John Sabato, Joan Hash, and Laurie Graffo. Security metrics guide for information technology systems. Technical report, NIST, 2003.

[21] S. C. Payne. A guide to security metrics. Technical report, SANS Institute, 2006.

[22] Vilhelm Verendel. 2009. Quantified security is a weak hypothesis: a critical survey of results and assumptions. In *Proceedings of the 2009 workshop on New security paradigms workshop* (NSPW '09). ACM, New York, NY, USA, 37-50

[23] Anming Xie; Weiping Wen; Li Zhang; Jianbin Hu; Zhong Chen; , "Applying Attack Graphs to Network Security Metric," Multimedia Information Networking and Security, 2009. MINES '09. International Conference on , vol.1, no., pp.427-431, 18-20 Nov. 2009

[24] Frigault, M.; Lingyu Wang; , "Measuring Network Security Using Bayesian Network-Based Attack Graphs," Computer Software and Applications, 2008. COMPSAC '08. 32nd Annual IEEE International , vol., no., pp.698-703, July 28 2008-Aug. 1 2008

[25] Peter Mell; Karen Scarfone; Sasha Romanosky; , "Common Vulnerability Scoring System," Security & Privacy, IEEE , vol.4, no.6, pp.85-89, Nov.-Dec. 2006

[26] Mell, P.; Scarfone, K.; , "Improving the Common Vulnerability Scoring System," Information Security, IET , vol.1, no.3, pp.119-127, Sept. 2007

[27] Scarfone, K.; Mell, P.; , "An analysis of CVSS version 2 vulnerability scoring," Empirical Software Engineering and Measurement, 2009. ESEM 2009. 3rd International Symposium on , vol., no., pp.516-525, 15-16 Oct. 2009

[28] Gallon, L.; , "Vulnerability Discrimination Using CVSS Framework," New Technologies, Mobility and Security (NTMS), 2011 4th IFIP International Conference on , vol., no., pp.1-6, 7-10 Feb. 2011

[29] Houmb, S.H.; Franqueira, V.N.L.; , "Estimating ToE Risk Level Using CVSS," Availability, Reliability and Security, 2009. ARES '09. International Conference on , vol., no., pp.718-725, 16-19 March 2009

[30] Harada, T.; Kanaoka, A.; Okamoto, E.; Kato, M.; , "Identifying Potentially-Impacted Area by Vulnerabilities in Networked Systems Using CVSS,"

Applications and the Internet (SAINT), 2010 10th IEEE/IPSJ International Symposium on , vol., no., pp.367-370, 19-23 July 2010

[31] Ahmed, M.S.; Al-Shaer, E.; Khan, L.; , "A Novel Quantitative Approach For Measuring Network Security," INFOCOM 2008. The 27th Conference on Computer Communications. IEEE , vol., no., pp.1957-1965, 13-18 April 2008

[32] Tupper, M.; Zincir-Heywood, A.N.; , "VEA-bility Security Metric: A Network Security Analysis Tool," Availability, Reliability and Security, 2008. ARES 08. Third International Conference on , vol., no., pp.950-957, 4-7 March 2008.

[33] Sahinoglu, M.; , "Security meter: a practical decision-tree model to quantify risk," Security & Privacy, IEEE , vol.3, no.3, pp. 18- 24, May-June 2005

[34] Rui Huang; Danfeng Yan; Fangchun Yang; , "Research of security metric architecture for Next Generation Network," Network Infrastructure and Digital Content, 2009. IC-NIDC 2009. IEEE International Conference on , vol., no., pp.207-212, 6-8 Nov. 2009

[35] Cui Xiaolin; Tan Xiaobin; Zhang Yong; Xi Hongsheng; , "A Markov Game Theory-Based Risk Assessment Model for Network Information System," Computer Science and Software Engineering, 2008 International Conference on , vol.3, no., pp.1057-1061, 12-14 Dec. 2008

[36] http://www.sans.org/reading_room/whitepapers/auditing/guide-security-metrics_55

[37] Zhang, Qianli, Jilong Wang, and Xing Li. “Evaluation of Bulk Traffic Mitigation Practices in Campus Network.” In Proceedings of the Sixth Asian Internet Engineering Conference, 9–15. AINTEC ’10. New York, NY, USA: ACM, 2010.

[38] DBIR Myopia: If You Think the Big Problem is Hacktivism, You're Wrong, http://blog.redsealnetworks.com/category/security-metrics/

[39] Verizon 2012 Data Breaches Investigation Report , http://www.verizonbusiness.com/resources/reports/rp_data-breach-investigations-report-2012_en_xg.pdf

[40] IBM X-Force 2011 Trend and Risk Report, http://www.fbiic.gov/public/2012/mar/2011IBMReport-March2012.pdf

[41] Cumulative Security Update for Internet Explorer , http://technet.microsoft.com/en-us/security/bulletin/ms12-023

[42] Urlblacklist: Database of blacklisted sites with classification, http://urlblacklist.com/

[43] Uclassify: Free Text Classification API, http://uclassify.com/

[44] Top Cyber Security Risks - Origin and Destination Analysis, http://www.sans.org/top-cyber-security-risks/origin.php

[45] Moazzam Khan, Muhammad Omer, John A. Copeland, " Decision Centric Identification and Rank Ordering of Security Metrics," Submitted to to 37th Annual IEEE Conference on Local Computer Networks (LCN).

[46] Metric Definition by Securitymetric.org, http://www.securitymetrics.org/content/Wiki.jsp?page=MetricsDefinitions

[47] Tstat-TCP STatistic and Analysis Tool, http://tstat.tlc.polito.it/index.shtml

[48] Snort Documentation, http://www.snort.org/docs

[49] Gu, Guofei, Phillip Porras, Vinod Yegneswaran, Martin Fong, and Wenke Lee. “BotHunter: Detecting Malware Infection Through IDS-driven Dialog Correlation.” In Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium, 12:1–12:16. SS’07. Berkeley, CA, USA: USENIX Association, 2007.

[50] Kun Sun; Jajodia, S.; Li, J.; Yi Cheng; Wei Tang; Singhal, A.; , "Automatic security analysis using security metrics," MILITARY COMMUNICATIONS CONFERENCE, 2011 - MILCOM 2011 , vol., no., pp.1207-1212, 7-10 Nov. 2011.

[51] Tupper, M.; Zincir-Heywood, A.N.; , "VEA-bility Security Metric: A Network Security Analysis Tool,"Availability, Reliability and Security, 2008. ARES 08. Third International Conference on , vol., no.,pp.950-957, 4-7 March 2008.

[52] Ghosh, N.; Ghosh, S.K.; , "An Approach for Security Assessment of Network Configurations UsingAttack Graph," Networks and Communications, 2009. NETCOM '09. First International Conferenceon , vol., no., pp.283-288, 27-29 Dec. 2009

[53] Frigault, M.; Lingyu Wang; , "Measuring Network Security Using Bayesian Network-Based Attack Graphs," Computer Software and Applications, 2008. COMPSAC '08. 32nd Annual IEEE International , vol., no., pp.698-703, July 28 2008-Aug. 1 2008.

[54] Khan, M.; Omer,M,; John, C.; , ” What is more secure, open or closed source?An investigation from vulnerability and exploit’s perspective,”Submitted to Globecom 2012 - Communication and Information System Security Symposium.

[55] Urlblacklist Website ,” http://urlblacklist.com/”.

[56] Securitymetrics,”http://www.securitymetrics.org/content/Wiki.jsp?page=Metrics Definitions.”

[57] Jacquit, A.; ,”Security Metrics: Replacing Fear, Uncertainty, and Doubt”.

[58] Top Cyber Security Risks - Application vs. Operating System Patching, “http://www.sans.org/top-cyber-security-risks/patching.php”

[59] DBIR Myopia: If You Think the Big Problem is Hacktivism, You’re Wrong,”http://blog.redsealnetworks.com/category/security-metrics/”

[60] Bothunter Website,”http://www.bothunter.net/about.html”

[61] Verizon 2012 Data Breaches Investigation Report ,http://www.verizonbusiness.com/resources/reports/rp_data-breach-investigations-report-2012_en_xg.pdf.

[62] Tim Chenoweth; Robert Minch; Sharon Tabor; , "User Security Behavior on Wireless Networks: An Empirical Study," System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on , vol., no., pp.145b, Jan. 2007 J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[63] Henderson, Tristan, David Kotz, and Ilya Abyzov. “The Changing Usage of a Mature Campus-wide Wireless Network.” In Proceedings of the 10th Annual International Conference on Mobile Computing and Networking, 187–201. MobiCom ’04. New York, NY, USA: ACM, 2004.

[64] Anand Balachandran, Geoffrey M. Voelker, Paramvir Bahl, and P. Venkat Rangan. 2002. Characterizing user behavior and network performance in a public wireless LAN. In Proceedings of the 2002 ACM SIGMETRICS international conference on Measurement and modeling of computer systems (SIGMETRICS '02). ACM, New York, NY, USA, 195-205.

[65] Ma, Justin, Lawrence K. Saul, Stefan Savage, and Geoffrey M. Voelker. "Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs." In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1245–1254. KDD '09. New York, NY, USA: ACM, 2009.

[66] Gill, Phillipa, Martin Arlitt, Zongpeng Li, and Anirban Mahanti. "Youtube Traffic Characterization: a View from the Edge." In Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement, 15–28. IMC '07. New York, NY, USA: ACM, 2007.

[67] Zhang, Qianli, Jilong Wang, and Xing Li. "Evaluation of Bulk Traffic Mitigation Practices in Campus Network." In Proceedings of the Sixth Asian Internet Engineering Conference, 9–15. AINTEC '10. New York, NY, USA: ACM, 2010.

[68] Humberto T. Marques Neto, Jussara M. Almeida, Leonardo C. D. Rocha, Wagner Meira, Pedro H. C. Guerra, and Virgilio A. F. Almeida. 2004. A characterization of broadband user behavior and their e-business activities. SIGMETRICS Perform. Eval. Rev. 32, 3 (December 2004), 3-13.

[69] Humberto T. Marques, Nt., Leonardo C. D. Rocha, Pedro H. C. Guerra, Jussara M. Almeida, Wagner Meira, Jr., and Virgilio A. F. Almeida. 2004. Characterizing

broadband user behavior. In Proceedings of the 2004 ACM workshop on Next-generation residential broadband challenges (NRBC '04). ACM, New York, NY, USA, 11-18.

[70] Marcelo Maia, Jussara Almeida, and Virg; lio Almeida. 2008. Identifying user behavior in online social networks. In Proceedings of the 1st Workshop on Social Network Systems (SocialNets '08). ACM, New York, NY, USA, 1-6.

[71] Rao, Ashwin, Arnaud Legout, Yeon-sup Lim, Don Towsley, Chadi Barakat, and Walid Dabbous. "Network Characteristics of Video Streaming Traffic." In Proceedings of the Seventh COnference on Emerging Networking EXperiments and Technologies, 25:1–25:12. CoNEXT '11. New York, NY, USA: ACM, 2011.

[72] Roelof van Zwol. 2007. Flickr: Who is Looking?. In Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI '07). IEEE Computer Society, Washington, DC, USA, 184-190.

[73] Michael Kwok and Gary Yeung. 2005. Characterization of user behavior in a multi-player online game. In Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology (ACE '05). ACM, New York, NY, USA, 69-74.

[74] Garcia-Dorado, J.; Finamore, A.; Mellia, M.; Meo, M.; Munafo, M.; , "Characterization of ISP Traffic: Trends, User Habits, and Access Technology Impact," Network and Service Management, IEEE Transactions on , vol.PP, no.99, pp.1-14.

[75] Hu, Jian, Hua-Jun Zeng, Hua Li, Cheng Niu, and Zheng Chen. "Demographic Prediction Based on User's Browsing Behavior." In Proceedings of the 16th

International Conference on World Wide Web, 151–160. WWW '07. New York, NY, USA: ACM, 2007.

[76] Mori, T.; Kawahara, R.; Naito, S.; Goto, S. (2004). "On the characteristics of Internet traffic variability: spikes and elephants". Applications and the Internet Proceedings. 2004 International Symposium on Applications and the Internet: 99–106.

VITA

Moazzam Khan

Moazzam Khan was born in Karachi, Pakistan. He attended NED University of Engineering and Technology for his B.E in Computer Engineering. He completed his M.S in Electrical and Computer Engineering from Georgia Tech in 2007. He started his PhD with Communications Systems Center (CSC) under the supervision of Dr. John A. Copeland. Along with research Moazzam likes to teach and has been a TA and instructor for graduate and undergraduate course at Tech. He has also been an adjunct instructor for Southern Polytechnic State University (SPSU) and Kennesaw State University (KSU). Moazzam Khan is an aficionado of Cricket and Tennis. When not in the lab, he can be found on the cricket field or at a tennis court.