Implementing Distributed Internet Security using a Firewall Collaboration Framework

J. Lane Thames
Georgia Institute of Technology
lane.thames@gatech.edu

Randal Abler
Georgia Institute of Technology
randal.abler@gatech.edu

Abstract

Society has grown to rely on Internet services, and the number of Internet users increases every day. As more and more users become connected to the network, the window of opportunity for malicious users to do their damage becomes very great and lucrative. The computer industry is combating the rising threat of malicious activity with new hardware and software products such as Intrusion Detection Systems, Intrusion Prevention Systems, and Firewalls. However, malicious users are constantly looking for ways to by-pass the security features of these products, and many times they will succeed. In order to enhance the capabilities of computer network security devices such as firewalls, the Distributed Network Applications Laboratory at the Georgia Institute of Technology is investigating the use of a Distributed Internet Security architecture known as the General Network Security Collaboration Framework (GNSCF). This paper describes a particular instance of the GNSCF using firewalls as the basic network element of the framework, and this system is referred to as a Firewall Collaboration Framework (FCF).

1. Introduction

Over the past few years, there has been accelerating growth in the use of network based services. Because of the ubiquity of computer systems and the Internet, this growth will continue. Corresponding to this continued growth in network usage, there will be increases in malicious user activity. The threat of malicious users is becoming more serious every day. Terms such as phishing, pharming, spam and identity theft make their way into news headlines all the time. The Internet security community is seeing malicious activity enter a new level of sophistication. The tools and concepts of malicious users are entering the underground crime scenes where hackers are being paid for their services [9]. Corporate information espionage, extortion, and identity theft are very lucrative commodities within the Internet underground. Therefore, there exists an engineering need to develop reliable and robust network security systems that can thwart the actions of malicious users as their tools and techniques continue to evolve.

This paper discusses the concept of Distributed Internet Security and how it can be implemented when using a Firewall Collaboration Framework (FCF). The firewall collaboration framework is a particular instance of the General Network Security Collaboration Framework (GNSCF). The paper will give an overview of computer network security fundamentals and firewall technology. Related work in the area of Distributed Firewalls will be discussed, and the concept of the Firewall Collaboration Framework will be introduced. An experimental evaluation of the framework will be described, and the paper will end with a discussion of future work and conclusions.

2. Overview of Computer Network Security

The field of computer network security provides the technologies to prevent users with malicious intent from doing damage. The term "damage" is used loosely to refer to issues ranging from unauthorized access of data to unauthorized access of computer systems to actual physical damage of computing resources. Computer network security can be thought of encompassing five main services: *confidentiality*, *authenticity*, *integrity*, *availability,* and *access control*. Confidentiality concerns the concealment of information such that only those that are authorized to view data can view the data (information privacy). Authenticity is the means by which to identify and assure the origin of information. Integrity is charged with verifying the trustworthiness of information (non-altered information), and availability provides the mechanisms needed to ensure that information and resources are able to be used in the manner for which they are designed (permanence and non-erasure). Access control is designed to prevent misuse of resources.

Malicious users have developed many techniques for attacking computers and network systems. The following is a list of the most common attack techniques, also known as attack vectors:

- Buffer overflow exploits
- Denial of service attacks
- Password attacks
- Exponential attacks
- Trojan horses
- TCP/IP protocol exploitation

To implement a buffer overflow attack, the malicious user (hacker) discovers a software package that contains an unprotected memory region such as a buffer used for user input to a program. Once the unprotected memory is discovered, the hacker can construct a special sequence of data that can redirect the vulnerable program to code that the hacker wants to execute. If done correctly, the hacker can gain complete control of a computing system. Denial of service attacks are designed to deny legitimate users access to needed services. Classically, this is done by either flooding a system's network in order to consume its bandwidth or by flooding the actual target in such a way to consume its resources such as internal software queues. Password attacks are techniques used to gain unauthorized access to systems. This is commonly done by using password guessing software (by using password dictionary databases), by using software to decrypt encrypted passwords, or by using social engineering and information leakage to determine user passwords. Computer worms and viruses are classified as exponential attacks. They are referred to as exponential attacks because their propagation profiles can be modeled with differential equations that have exponential curves as solutions. Worms and viruses can spread at alarming rates and have the potential to do much damage to the Internet infrastructure. Trojan horses are software modules installed in a piece of software that would otherwise be considered trustworthy. These modules can contain code that can be used as "back-doors" into computer systems. TCP/IP protocol exploits are techniques that make use of weaknesses in the TCP/IP protocol suite. For example, IP spoofing, which is the ability for a host to send a packet onto a network that has the IP address of another host, can be used to cause denial of service on a specified host such as with a "Smurf" attack. With the Smurf attack, a hacker sends a broadcast message to a network (or group of networks), but the hacker spoofs his IP address with that of the victim. In turn, the hosts that receive the broadcast packet will reply to the victim, thereby causing denial of service on the victim. There are many other attack vectors available, but describing them is beyond the scope of this paper.

3. Overview of Firewall Technology

Firewalls have become a very common technology to use for securing computers and networks. A firewall can be classified as any device that limits network access. It is a collection of components inserted between two networks that filter traffic between the networks according to a local security policy [4]. The device can be software running on a personal computer or server. Or, it could be a dedicated hardware device within a computer network. There are three common types of firewall devices: packet filtering devices, application filtering devices, and stateful packet filtering devices. A packet filtering device analyzes network traffic at the network and transport layers of the TCP/IP model. These are the layers that control network routing with IP addresses and software routing with port numbers. Packet filters rely on access control lists based on packet types and local security policies, i.e. allowing users access to the world wide web. Application filtering devices work at the application layer of the TCP/IP model. These devices examine the communication packets in greater detail based on the particular type of application being used. For example, special filters can be created for local email systems. Stateful packet filters are the most complex devices of the three. These filters maintain state information for each network flow. This state information includes variables such as source and destination IP addresses, source and destination port numbers, sequences numbers, and various other protocol flags. Firewalls are extremely effective security mechanisms and should be consider as part of any security implementation.

4. Distributed Firewalls and Related Work

One of the first discussions of distributed firewalls was given by Bellovin [3], which described a distributed system of firewalls that enforce a security policy, but the security policy is centrally managed. This system can be viewed as a star topology where the management unit is at the center and the other units connect to the center. Indeed, this conceptual model gave rise to the end-host software firewall applications that are common today. But, these common end-host applications do not usually rely on a central policy manager. The policies are determined by the end-host's user. The system described in [3] is based on central management of the distributed firewalls. In contrast, the FCF does not use central management. Instead, the individual elements work independently. As they observe new malicious activity, they distribute the newly discovered information to the collaborating elements. In [10], Smith describes a distributed firewall system that uses a cascade model. This model can be viewed as a tree, where end-hosts are at the bottom of the tree and firewalls are cascaded downward along the tree. The idea is to increase the level of security as the network flow traverses the tree. Zou [13] presents a similar model that uses a "defense in depth" strategy where networks are divided into domains protected by there own firewalls. This strategy is to place firewalls at key locations within a given organization. In [8], the authors propose a distributed internet security system called the Intruder Detection and Isolation Protocol (IDIP) whose functionality is similar to the framework described in this paper. However, their system focuses on Intrusion Detection Systems (IDS) whereas this paper focuses on a firewall implementation of a general network security collaboration framework where classification, i.e. IDS, is a component of the framework.

5. Firewall Collaboration Framework Concept

The philosophy of this work is to dynamically limit the impact of malware by attempting to stop the malicious behavior as close to the source as possible thus preserving network resources for intended applications. The high-level design goals of the Firewall Collaboration Framework are as follows: The system is built with a federation of firewalls that collaborate with each other and share a "global" pool of information. The global pool of information contains data such as attach signatures, firewall rules, access control lists, and related security policies. Once new information regarding malicious activity is obtained either in real time with an information classification system or by an administrator that learns of new rules that need to be added, the new information is distributed to the firewall units that belong to the federation. The collaborating firewalls can then update their rule bases and policies to incorporate this new information. Metcalfe's law states that the value of a network is proportional to the square of the number of nodes in the system. The FCF follows this model as well. As more nodes join the federation, the usefulness or utility of the system increases. Essentially, using this architecture, the firewall members of the federation act like distributed network sensors. As new information is obtained, it is distributed to the other members. Hence, as more elements join the federation, the overall view of the network increases and the probability of obtaining and distributing data describing malicious behavior increases. The implications of this system architecture are that worms, viruses, spam email, and other malware could, in some cases, be stopped at the source (assuming the source is behind a collaborating firewall). Figure 1 will be used to further illustrate this concept.

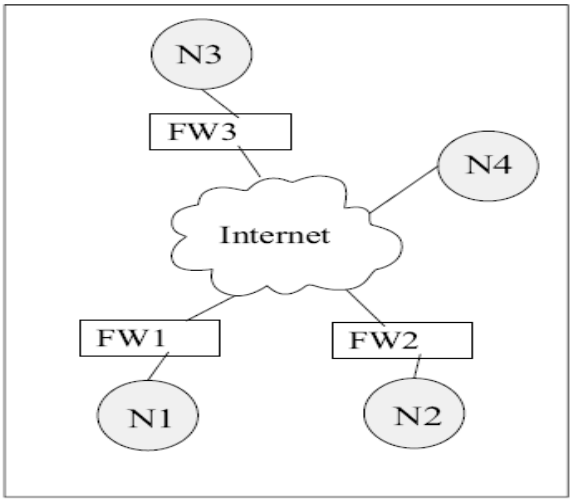

Figure 1. Illustration of the FCF concept.

From Figure 1, N1, N2, N3, and N4 are end user networking nodes, and FW1, FW2, and FW3 are firewall elements that belong to the FCF federation. Consider the following scenario: N3 is a malicious source such as a host that contains a worm that is trying to propagate. N3 issues an attack on N1. At some time Δt after the attack begins, FW1 classifies the attack vector, updates the local firewall policy to deny N3 access to the network that FW1 protects, which includes N1, and distributes the new information to the federation. In this scenario, one can see two of the major strengths of this system: First, when the new information is obtained by FW2 and its rules are updated, its protected network, which includes N2, will never be impacted by the malicious activity of N3 (assuming the classification, distribution, and updating happens before N3 issues an attack onto N2 and its network). Second, the malicious node (N3) is behind a collaborating firewall, FW3. Since FW3 will obtain the distributed data from FW1, it will know to deny N3 networking access beyond the firewall perimeter. Hence, in this case, the origin of malicious activity is stopped at the source. The major weakness of the FCF is denial of service. Consider the following scenario: N4 issues an attack on N1, but N4 spoofs its IP address with N3's IP address. FW1 classifies the attack and distributes the policy information to the federation. The policy will be to deny N3 access, as it has been classified as performing malicious activity. However, N3 is not the true source since N4 has spoofed N3's address. Hence, N4 has issued denial of service on N3 once the policies take effect at FW3. These are very simple and intuitive cases that illustrate the usefulness of the system, but also describe its major weakness.

Figure 2 shows the functional components of the FCF.

Federation Management
Trust Relationship Management
Policy Management
Network Traffic Classification
Information Management
Resource Management

Figure 2. Functional components of the collaboration framework.

These components are in the initial design stage. Solution spaces for each component will not be presented in this paper. However, the various components and their purposes will be presented. As shown in Figure 2, there are 6 functional components (in this stage of the design) needed to implement a reliable firewall collaboration framework.

Federation Management: The purpose of federation management is to control membership of new firewall elements to the federation. This layer is responsible for establishing an initial trust between the firewall and the federation. If membership is not carefully controlled, denial of service attacks would be easy to orchestrate. For example, a rogue firewall could join the federation and inject falsified policies into the system. This could be used to deny service to members of the federation and their respective networks. Further, it could be used to allow unauthorized access to federation networks.

Trust Relationship Management: Once firewall elements have been allowed to join the federation by establishing the initial trust between firewall element and the federation, the Trust Relationship Management (TRM) component will be used to maintain the member relationships. TRM will address issues such as information authentication and credential management. Information authentication is needed to prevent denial of service. Credential management is needed because of the possibility that members could become rogue after being allowed to join the federation. The TRM and federation management components will need to be tightly coupled with each other.

Policy Management: In most organizations that follow security best-practices, security policies are designed based on business and end user needs. These policies will differ from organization to organization. For this system to operate effectively there is a need to differentiate local policy from the global policy. The local policy refers to rules and specifications that are site specific. For example, allowing organizational users to browse the web. Global policies will be those that are distributed to the collaborating elements in the federation. For example, if a firewall is running an intrusion detection system and it classifies a new attack vector, a new policy (or rule) will be created and distributed to the federation. In most cases, local policy will be strictly partitioned from global policy, as most organizations will not want to distribute private policies to the outside world. However, there could be cases where various local policies will need to be distributed to the federation. For example, if a large organization has sites at various locations, they might want to distribute various policies that would normally be strictly local policies.

Network Traffic Classification: Network traffic classification is one of the most important components of this system. As stated earlier, network administrators can update the rules and policies of a collaborating firewall. But, the time delay between the start of an attack and the human detection (using network analysis tools or security bulletins) of the attack is usually too large. However, automated systems for detecting network attacks do exist, and these systems are the logical choice for this type of proactive, pre-emptive based system. There is much active research in the area of network traffic classification. Some of these classification systems use statistical based anomaly detection and rule based detection [11]. With statistical based anomaly detection, data that models the behavior of "normal" users is collected for various period of time. Then, statistical analysis is performed on observed behavior to measure deviations from the norm. Rule based detection relies on rules that are established that can be used to determine normal versus abnormal network usage. Other work has been performed for classifying network traffic by using artificial intelligence and machine learning algorithms [1], [2], [5], [7],[12]. These types of systems collect data from computers and networks with known behavioral profiles and use the data to train the systems. Once the systems are trained, they can then be used in real time to classify observed computer and network data. However, these systems are only reliable if the observed behavioral profiles are within domains comparable to that of the training data. Hence, as new attack vectors are created, the systems have to be re-trained so they can reliably observe and detect the new behavior. The most common way for systems to classify network traffic anomalies is with the use of attack signatures [6]. These methods rely on engineers to construct signatures based on the observed behavior of new anomalies. Once the signatures for a new attack vector are created, they need to be uploaded into the classification system's signature database. The problem with this is two-fold. First, the signatures are discovered by humans, so there will be a noticeable time delay between detection and signature creation. Second, attack vectors change at a very fast rate, which leads to large signature databases. As the databases continue to grow, the processing time for these detection systems to classify network flows will increase. Network traffic classification will be thoroughly analyzed with future work as it is a key component of the framework.

Information Management: The information management component will address an array of issues. It will govern the way in which information is transported throughout the federation. This mechanism will address issues such as centralized versus peer-to-peer information distribution (this is not to be confused with central "management"). The distribution mechanism could also be a combination of centralized and peer-to-peer mechanisms. With the combined approach, the federation could be partitioned into domains where the central unit coordinates information into and out of its respective domain, and information is distributed within the domain from a peer-to-peer perspective. The information management component will also address the issue of data caching and staleness. In certain cases, the policies that are distributed will be long-term policies, and staleness will not be of concern. However, because of the dynamic nature of networks and attacks, there will be other policies that will not be needed for long periods of time. This will be the case if IP addresses are used as part of the attack signature. For example, a particular host might be classified as malicious at some point in time. Later, a system administrator could

quarantine the host and clean the system. After the system has been cleansed, it should be able to participate in network activities again. Therefore, the original attack signature would now be deemed unnecessary and will need to be removed from the collaborating firewalls. Lastly, this component will also address the issue of information confidentiality and integrity. This will be accomplished with encryption mechanisms, and this component will govern the types of encryption techniques that are used throughout the federation.

Resource Management: In order to provide scalability, resources within the federation will need to be governed. The resource management component will administer the mechanisms needed for robustness and scalability. For example, it will manage issues such as which elements will be central units in the partitioned domains.

6. Experimental Evaluation

The firewall collaboration framework concept is still in the initial design stage. The solution spaces for each of the layers are currently being evaluated. However, to ascertain some of the perceived properties of the framework, a simple experimental evaluation was performed. A system similar to Figure 1 was constructed. Three firewall units were built using laboratory PCs running the Fedora Core 4 [18] operating system. The firewall functionality was implemented with the iptables software package [14]. The classification mechanism used for this experiment was the PortSentry software package [15]. PortSentry is a software package that can detect network scanning. Once a scan is detected, it can be configured to automatically update the firewall rules in iptables. The nmap software package [16] was used to scan the hosts in the experimental network. For the purposes of this experiment, the network scan was used to "simulate" real network attacks. The netcat software package [17] was used as the transport mechanism.

In order to test the system, nmap was used to scan the nodes in the experimental network. As the nodes were scanned, PortSentry would detect the scan and modify the iptables of the scanned host. A Perl script was created and was called by a cron job every 60 seconds. The script would examine the current state of the iptable and compare it to the previous version, which was stored in a file. If the iptable rules had changed, this indicated that PortSentry had detected a scan. If the iptable rules had changed, the data was written to a file to be used as a comparison when the script was called at the next 60 second interval. This new information was then distributed to the other nodes of the federation by using a client-server system built with the netcat software. The goal of the experiment was to evaluate the properties of the framework. The following questions were answered with the experimental evaluation:

1. Can the FCF stop malicious activity if the source has to pass through a collaborating firewall? **Yes**
2. Can the FCF enable networks to be preemptively protected because of the propagation of classification information (i.e. if one of the collaborating firewalls detects an attack and distributes the information to the federation before the attacker targets the other networks)? **Yes**
3. Is denial of service possible in the FCF because of IP spoofing? **Yes**
4. Is denial of service possible if a rogue firewall joins the federation? **Yes**
5. Can unauthorized access be granted to a network protected by a collaborating firewall if a rogue firewall joins the federation and injects fraudulent information into the system? **Yes**

There are many more questions that could be conceived. Even though most of these ideas are intuitive to those that are familiar with networking and security fundamentals, experimental proof is needed before moving forward with a concept such as this. The experimental results show that the concept is valid and has great potential for implementing a reliable, distributed internet security model if the six functional components are implemented in a way to prohibit denial of service attacks.

7. Future Work

The firewall collaboration framework is a specific instance of the general network security collaboration framework. Future work will focus on evaluating the solution spaces of the six functional components of the general framework. The work performed in [12] for network traffic classification will also continue as network traffic classification is a fundamental function of the framework.

8. Conclusion

Computer network attacks evolve on a daily basis. Malicious users are continuously developing new tools to wreak havoc on computer and networking systems. In response to this, security professionals must continue to research and develop better ways of defending their systems. This paper discussed a new idea of implementing distributed Internet security by using a firewall collaboration framework. The distributed nature of the system will allow proactive and preemptive measures to take place in such a way to protect some systems before the attack makes it to their perimeter, and, in some cases, the source of attack can be completely isolated and banned from network usage. Denial of service is a major weakness of this system. But, the security benefits provided by this mechanism are great. The idea that malicious sources such as worms, viruses, spam, and malware can be stopped at the source provides a strong reason to move forward with this concept.

12. References

[1] N. Abouzakhar, A. Gani, G. Manson, M. Abuitbel, and D. King, "Bayesian Learning Networks Approach to Cybercrime Detection", *Proceedings of the 2003 PostGraduate Networking Conference* (PGNET 2003), Liverpool, United Kingdom, 2003.

[2] N. Amor, S. Benferhat, and Z. Elouedi, "Naïve Bayes vs Decision Trees in Intrusion Detection Systems", *Proceedings of the 19th Annual ACM Symposium on Applied Computing,* pp. 420-424, 2004.

[3] S. M. Bellovin, "Distributed Firewalls", *;login*, pp. 37-39, 1999.

[4] W. R. Cheswick, S.M. Bellovin, and A.D Rubin, *Firewalls and Internet Security, 2nd Edition*, Addison-Wesly, Boston, MA, 2003.

[5] S. Lee and D. Heinbuch, "Training a neural network based intrusion detector to recognize novel attacks", *Information Assurance and Security*, pp.40-46, 2000.

[6] B. Laing, "Intrusion Detection Systems", *Internet Security Sytems*, http://www.iis.com , 2000.

[7] P. Lichodzijewski, A. Zincir-Heywood, and M. Heywood, "Host based intrusion detection using self-organizing maps", *Proceedings of the 2002 IEEE World Congress on Computation Intelligence*, 2002

[8] D. Schnackenberg, K. Djahandari, and D. Sterne, "Infrastructure for Intrusion Detection and Response", *Proceedings of the 2000 DARPA InformationSurvivability Conference and Exposition* (DISEX'00), Hilton Head, SC, 2000

[9] E. Skoudis, *Counter Hack: A step by step guide to computer attacks and effective defenses*, Prentice Hall, Upper Saddle River, NJ, 2002.

[10] R. Smith, Y. Chen, and S. Bhattacharya, "Cascade of Distributed and Cooperating Firewalls in a Secure Data Network", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 15, NO. 5, pp. 1307-1315, 2003.

[11] W. Stallings, *Network Security Essentials,* Prentice Hall, Upper Saddle River, NJ, 2003.

[12] J. L. Thames, R. Abler, and A. Saad, "Hybrid Intelligent Systems for Network Security", *Proceedings of the 2006 ACM Southeast Conference* (ACMSE06), Melbourne, FLA, 2006.

[13] C. Zou, D. Towsley, and G. Weibo, "A Firewall Network System for Worm Defense in Enterprise Networks", *Technical Report: TR-04-CSE-01*, University of Massachusetts, Amherst, 2004

[14] http://www.netfilter.org/projects/iptables/index.html

[15] http://sourceforge.net/projects/sentrytools/

[16] http://www.insecure.org

[17] http://netcat.sourceforge.net/

[18] http://www.redhat.com