

# The Use of Honeynets to Detect Exploited Systems Across Large Enterprise Networks

John Levine\*, Richard LaBella\*\*\*, Henry Owen\*, Didier Contis\*, Brian Culver\*\*

\*School of Electrical and Computer Engineering

\*\*Office of Information Technology

Georgia Institute of Technology

\*\*\* South Florida Honeynet Project

*Abstract – Computer Networks connected to the Internet continue to be compromised and exploited by hackers. This is in spite of the fact that many networks run some type of security mechanism at their connection to the Internet. Large Enterprise Networks, such as the network for a major university, are very inviting targets to hackers who are looking to exploit networks. Large Enterprise Networks may consist of many machines running numerous operating systems. These networks normally have enormous storage capabilities and high speed/high bandwidth connections to the Internet. Due to the requirements for Academic Freedom, system administrators are restricted in what requirements they can place on users on these networks. The high bandwidth usages on these networks make it very difficult to identify malicious traffic within the enterprise network. We propose that a HoneyNet can be used to assist the system administrator in identifying malicious traffic on the enterprise network. By its very nature, a HoneyNet has no production value and should not be generating or receiving any traffic. Thus, any traffic to or from the HoneyNet is suspicious in nature. Traffic from the enterprise network to a machine on the HoneyNet may indicate a compromised enterprise system.*

**Index terms – Computer crime, hacking, intrusion detection, Honeynets, HoneyPots**

## I. INTRODUCTION

Computer networks that are currently connected to the Internet are vulnerable to a variety of exploits that can compromise their intended operations. Systems can be subject to Denial of Service Attacks that prevents other computers from connecting to them for their provided service (e.g. web server) or prevent them from connecting to other computers on the Internet. They can be subject to attacks that cause them to cease operations either temporary or permanently. A hacker may be able to compromise a system and gain root access, i.e. the ability to control that system as if the hacker was the system administrator. The number of exploits targeted against various platforms, operating systems, and applications increases on a daily basis. System administrators are

usually responsible for monitoring the overall security of their networks.

System administrators use a variety of methods to protect the security of their networks. The use of firewalls at the border of their network and the Internet is one such method that is in current use today. Firewalls are used to control the flow of traffic between the local network and the Internet. Based on the characteristics of the network traffic, to include requested services, source and destination addresses, and individual users, a firewall will make a decision on whether to allow the traffic to pass through the network. A firewall can be considered as a “traffic cop” for the network [1]. Firewalls can also be utilized on individual host based systems.

Another method that may be used by system administrators is the use of an Intrusion Detection System (IDS). An IDS is used to detect and alert on possible malicious events within a network. IDS sensors may be placed at various points throughout the network, to include the interfaces between the local network and the Internet, critical points within the local network, or on individual host systems. An IDS is normally signature based, i.e., it will look for predefined signatures of bad events. These signatures normally reside in a database associated with the IDS. They may also perform statistical and anomaly analysis of network traffic to detect malicious intrusions. When malicious activity is detected they can notify the system administrator [2].

The use of IDS and firewalls provide a level of security protection to the system administrator. However, there are recognized shortfalls with the use of an IDS and firewalls to protect a network. The shortcomings associated with a firewall include the following:

1. The firewall cannot protect against attacks that bypass it, such as a dial-in or dial-out capability.
2. The firewall at the network interface does not protect against internal threats.
3. The firewall cannot protect against the transfer of virus-laden files and programs [3].

We speculate that in certain cases high volumes of network traffic may overwhelm the network monitoring capability of the firewall resulting in the possible passing of malicious traffic between networks.

The use of IDS as a network security device also leads to shortcomings. It has been speculated that in some cases an IDS fails to provide an additional level of security to a network and only increases the complexity of the security management problem. Shortcomings associated with an IDS include a high level of false positive and false negative alerts [4].

We propose that the use of a Honeynet within a network can provide an additional layer of network security. The Honeynet can serve as a compliment to the use of the firewall and IDS and help to overcome some of the shortcomings that are inherent to these systems.

#### *A. Definition of a Honeynet*

A Honeynet is a network, placed behind a reverse firewall that captures all inbound and outbound data. The reverse firewall limits the amount of malicious traffic that can leave the Honeynet. This data is contained, captured, and controlled. Any type of system can be placed within the Honeynet, to include those systems that are currently employed on the network that the Honeynet is intended to protect. Standard production systems are used on the Honeynet, in order to give the hacker the look and feel of a real system. A Honeynet is a network that is intended to be compromised, to provide the system administrator with intelligence about vulnerabilities and compromises within the network [5].

#### *B. Concept of Data Capture and Data Control*

There are two critical principles concerning the successful operation of a Honeynet. These two principles are the concept of Data Capture and Data Control. Both of these principles must be followed in order for the Honeynet to be successfully employed in protecting a network.

The principle of Data Capture concerns information gathering. All information that enters or leaves the Honeynet must be collected for analysis. This data must be collected without the knowledge of the individuals who are conducting malicious activity against the network that is to be protected. This is to prevent the hacker from bypassing the Honeynet network. The data that is collected must be stored in a location different from the Honeynet. This is done so that if the hacker compromises a Honeynet system, the data cannot be destroyed or altered. The goal is to be able to capture data on the hacker without the hacker knowing that this data is being collected.

©2003 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

The principle of Data Control concerns protecting other networks from being attacked and compromised by computers on the Honeynet. If a hacker compromises a Honeynet system, then this hacker must be prevented from using this system to attack and compromise production systems on other networks. The process of Data Control must be automated to prevent the hacker from getting suspicious. We do not want the hacker to become aware of the fact that the system he has compromised is on a Honeynet [6].

#### *C. GEN I vs. GEN II Honeynets*

There are currently two types of Honeynets that can be employed on a network. These are GEN I, or first generation, and GEN II, or second generation. The type of Honeynet that one chooses to use depends on many factors to include availability of resources, types of hackers and attacks that you are trying to detect, and overall experience with the Honeynet methodology.

GEN I Honeynets are the simpler methodology to employ. This technology was first developed in 1999 by the Honeynet Alliance. Although GEN I Honeynets are somewhat limited in their ability for Data Capture and Data Control, they are highly effective in detecting automated attacks or beginner level attacks against targets of opportunity on the network. Their limitations in Data Control make it possible for a hacker to fingerprint them as a Honeynet. They also offer little to a skilled hacker to attract them to target the Honeynet, since the machines on the Honeynet are normally just default installations of various operating systems.

GEN II Honeynets were developed in 2002 to address the shortcomings inherent with GEN I Honeynets. The primary area that was addressed by GEN II Honeynets is in the area of Data Control. GEN I Honeynets used a firewall to provide Data Control by limiting the number of outbound connections from the Honeynet. This is a very effective method of Data Control, however, it lacks flexibility and allows for the possibility of the hacker fingerprinting the Honeynet. GEN II Honeynets provide data control by examining outbound data and making a determination to block, to pass, or to modify by changing some of the packet contents so as to allow data to appear to pass but rendering it benign. GEN II Honeynets are more complex to deploy and maintain than GEN I Honeynets [7].

We have chosen to initially deploy a GEN I Honeynet on our enterprise network. We are initially concerned with detecting machines within our enterprise network that have been compromised by automated script type attacks.

#### D. Description of the Georgia Tech Campus Network

The Georgia Institute of Technology, or Georgia Tech is an engineering and research institutes in the United States [8]. There are over 15,000 undergraduate and graduate students enrolled at the university as well as approximately 5,000 staff and faculty. Undergraduate and graduate degrees are offered in the Colleges of Architecture, Engineering, Sciences, Computing, Management, and the Ivan Allen College of Liberal Arts.

The Georgia Tech Office of Information Technology has the primary mission of providing technology leadership and support to Georgia Tech students, educators, researchers, administrators, and staff. OIT consists of seven directorates including the Information Security Directorate [9].

The Information Security Directorate is responsible for numerous tasks including: educating the campus community about security related issues, assessing current policies and developing new policies, assisting in strengthening technical measures to protect campus resources, and developing mechanisms to react to incidents and events that endanger the Institute's information assets [10]. There are 69 separate departments at Georgia Tech with between 30,000-35,000 networked computers installed on campus. The campus has two OC-12's and one OC-48 connection to the Internet with an average throughput of 600Mbps.

Georgia Tech processes over four terabytes of data on a daily basis.

Because of the high data throughput as well as the requirement for Academic Freedom and the research requirements of the various departments, the Information Security Directorate does not run a firewall at the Internet connection to the campus. However, individual departments and campus agencies do run firewalls designed to meet their security requirements.

The Information Security Directorate does at present run an Intrusion Detection System (IDS) at the campus gateway in order to monitor possible exploits against campus computer systems. This monitoring is done out of band and suspicious traffic is not terminated when detected. Suspicious activity will undergo a follow-on investigation.

## II. ESTABLISHMENT OF THE HONEYNET ON THE GEORGIA TECH CAMPUS

The Georgia Tech Honeynet was initially established during the Summer of 2002. It was established using open source software and with equipment that is not currently state of the art. Initially, it was established as a single computer but has since been expanded to include several different machines running various operating systems. The following diagram (Figure 1) shows the current configuration of the Georgia Tech Honeynet.

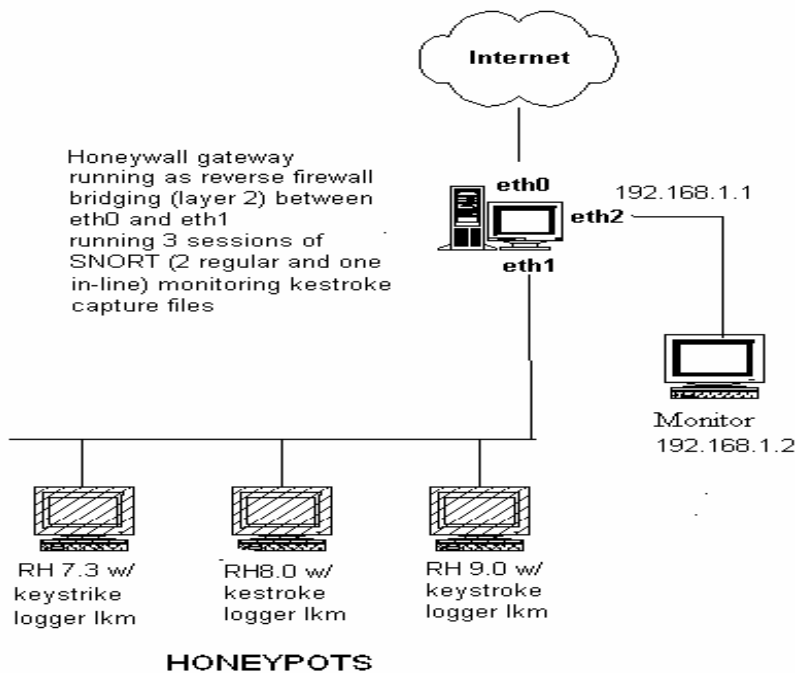


Figure 1 – Georgia Tech Honeynet

An IP address range was provided to us by the Georgia Tech OIT office to establish this Honeynet. This block of addresses is within the address range that belongs to Georgia Tech.

### A. Hardware and Software

As previously mentioned, the hardware that was used to establish the Honeynet is not current state of the art equipment. Current state of the art equipment is not necessary since the machines running on the Honeynet have no production value. The amount of traffic going to and from the Honeynet should be minimum since these systems are not running any production software. The system that runs as the Firewall does only that, it has no other applications running on it. We are using the LINUX operating system on the firewall. We are currently running Red Hat version 7.3 and we are planning an upgrade to Red Hat 8.0. Therefore, it is entirely possible to set up a Honeynet on a network using surplus equipment that may be available within the enterprise.

Although there are commercial versions of software and products available to establish a Honeynet, we have chosen to establish the Georgia Tech Honeynet using Open Source Software. We feel that open source software provides us with the greatest flexibility.

We used the rc.firewall script developed by The Honeynet Alliance to set up our firewall and establish Data Control for our Honeynet. This script is available from The Honeynet Alliance [11]. The purpose of this script is to perform Network Address Translation (NAT) for the target machines on the Honeynet and to provide data control.

The rc.firewall script was modified to control the Honeynet systems by restricting the number of outbound connections that are allowed from target systems on the Honeynet. This script should work with any version of LINUX software.

### B. Intrusion Detection System (IDS)

The IDS that we chose to use to monitor the Honeynet is SNORT. SNORT is open source software [12]. SNORT is primarily signature based but does have an anomaly detection plug-in available. Signatures are available periodically from the SNORT website and it is possible to write your own signatures. The IDS runs on a network monitoring system that is separate from the Honeynet network. The network monitoring system is currently running Red Hat 7.3 software. This system monitors the Honeynet by utilizing a network interface card (NIC) set in the promiscuous mode in a hub that connects all of the computers on the Honeynet (See Figure 1). This NIC card does not have an IP address assigned to it so that a hacker on the Honeynet will not have visibility of the network monitoring system.

We currently run two sessions of SNORT on the network monitoring system. One session uses the SNORT signature database to match signatures of potential hostile activities against traffic that is bound for the Honeynet. Alerts generated from this signature database are stored in an SQL database on the monitoring system and displayed on a web-based console. The console that we use is the Analysis Console for Intrusion Detection, or ACID, developed by the Computer Emergency Response Team (CERT). Instructions to configure SNORT, the SQL database, and ACID are available [13]. The following figure (Figure 2) shows the SNORT alert output on the web-based ACID console.

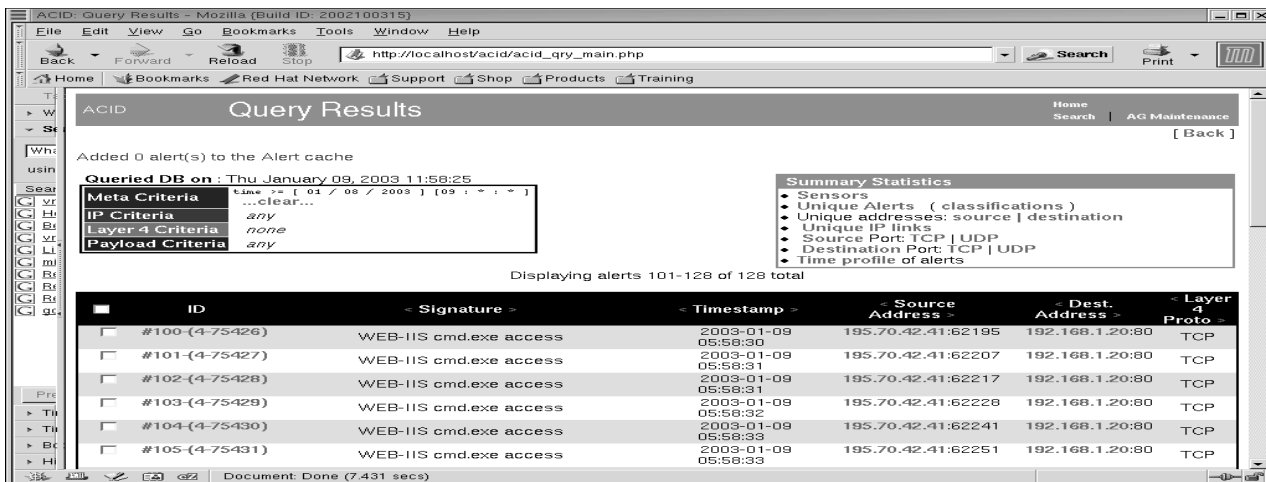


Figure 2 – ACID screen of SNORT alerts

The second session of SNORT runs in the packet capture mode. This session captures all of the data that goes to or comes from the Honeynet network. This session provides us with our Data Capture capability. The monitoring computer is isolated from the Honeynet network, protecting the integrity of this data from a hacker who may compromise the Honeynet.

### C. Logging and review of Data

The data that is collected on the Honeynet is stored in two separate locations on the monitoring system. Alerts that are triggered by the SNORT signature database are stored in an SQL database.

These alerts are retrievable for display by using the ACID console that was previously discussed (See Figure 2).

The data that is collected using the Data Capture capability of SNORT is stored in a daily log file on the monitoring system. A new directory is created each day for this data. We analyze this data using Ethereal. Ethereal is Open Source software that uses the libpcap library. Ethereal comes currently installed with Red Hat 7.3. Analyzing the data with Ethereal shows us all of the traffic that was sent to or originated from the Honeynet. Ethereal displays the source and destination addresses of this traffic, protocol used, source and destination ports and packet content. A sample of the Ethereal output is displayed in the following figure (Figure 3).

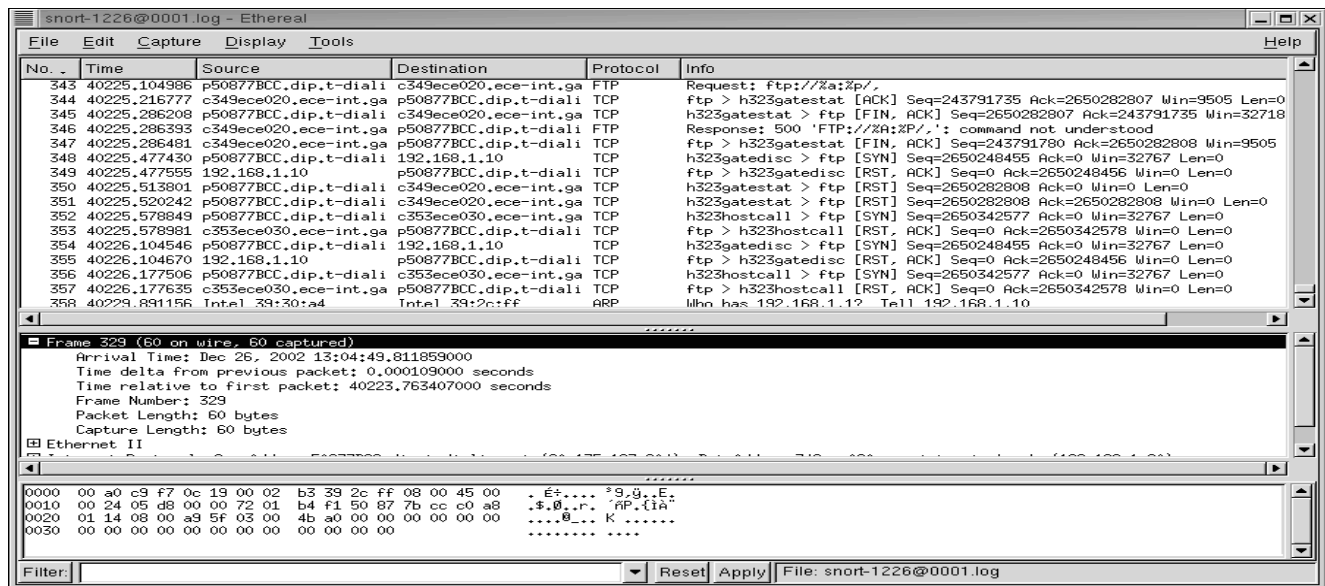


Figure 3 – Ethereal screen shot of Honeynet Data

The data provide by SNORT is analyzed on a daily basis. This analysis can be very time consuming. We spend at least one hour per day analyzing our Honeynet data for our three computer network Honeynet. When the Honeynet is attacked and compromised by a hacker we usually spend much longer analyzing the data and conducting a forensic analysis on the compromised system or systems. When we finish analyzing the previous days data, we archive the data to a read-only medium. We are currently using the X-CD-Roast package on a LINUX machine to archive our data to a CD-ROM. X-CD-Roast is Open Source Software [14]. It has the ability to create multi-session copies on the same CD. This capability is necessary for us since on average we collect less than one megabyte of data per day.

### III. EXPLOITATIONS DETECTED ON THE GEORGIA TECH NETWORK

In the six months that we have been running the Georgia Tech Honeynet we have detected 16 compromised Georgia Tech systems on networks other than our Honeynet. These compromises include automated worm type exploits as well as individual systems that have been targeted and compromised by hackers. Whenever a compromised system is detected by the Honeynet a report is made to the Georgia Tech OIT office. In some cases, the OIT personnel were already aware of some of these compromises. In other cases our report to the OIT personnel was the first report of an infected system on campus. This demonstrates the benefit of running a Honeynet on a large enterprise network in order to

augment other security measures that are currently running in order to detect compromised systems. The Gen I Honeynet that we are currently running here at Georgia Tech is very good at detecting certain automated type worm attack against specific operating systems.

#### *A. Exploitation pattern of a typical Internet worm*

The Code Red and Nimda worms are examples of two such automated worms that attack computers on the Internet. Both of these worms targeted vulnerabilities in applications running on specific operating systems. Later variants of the Code Red worm, specifically Code Red II, used localized scanning to propagate through the Internet. The principle of localized scanning is to try and infect machines in a close proximity to the currently infected machine. The infection pattern for Code Red II is as follows: 3/8 of the time it attempts to infect a machine within its own Class B address space (/16 network), 1/2 of the time it tries to infect a machine within its own class A address space (/8 network), and 1/8 of the time it would choose a random address from across the entire Internet. Localized scanning appeared to be successful for the Code Red II worm. It allowed this worm to spread quickly within parts of the Internet that had a high concentration of vulnerable hosts. This strategy allows a worm to spread very quickly within an internal network after it has already bypassed any external firewall or IDS system [15].

#### *B. Detection of worm type exploits*

Our Honeynet is well suited to detect worms that use localized scanning to infect other systems. By its very nature, any traffic to our Honeynet is suspicious and warrants investigation. Repeated scans across our Honeynet for a specific port indicate that an infected machine may be looking for a vulnerability within our enterprise network. We analyze the data collected by the SNORT session in the Data Capture mode to look at the time lapse between the various port scans that occur across our Honeynet systems. We speculate that scans that have occurred in under one second across numerous systems on the Honeynet are most likely automated worm type exploits. We can inform the OIT personnel of these systems that are suspected of being exploited. This allows the OIT personnel to take action in the quickest possible manner preventing the infection of additional systems across the Georgia Tech Enterprise network. We are also capable of providing all of the data that was transmitted to the Honeynet, allowing the OIT personnel to develop a specific signature for any new exploit that they are not already familiar with.

We also have the capability to employ a specific operating system on the Honeynet that the OIT personnel are concerned about being exploited from within the campus network. If the OIT personnel are concerned about a specific vulnerability against a specific operating system, we can install a system on the Honeynet that matches the characteristics of the system that concerns the OIT personnel. We can then monitor that system and provide a report of any suspected compromises or other suspicious traffic to that system from computer systems within the Georgia Tech campus network.

#### *C. Identification of a system with a compromised password*

Our Honeynet enabled us to identify a Georgia Tech system with a password that had been compromised by a hacker. This system was used by the hacker to connect to another system on the Honeynet that we suspect had been compromised by this same hacker. The system that the hacker connected to on the Honeynet was running Microsoft NT 4 Workstation software. Previously, a hacker had compromised this system using a Microsoft Internet Information Server (IIS) type exploit and set this system up as a WAREZ server. The hacker also set up a back door port in order to connect to this compromised computer at a later time. We knew that this system had been compromised but keep it up and running in order to track the hacker's behavior. Several days after this system was compromised the hacker connected to the back door port established on this computer using another computer from within the Georgia Tech Enterprise Network. We immediately notified the OIT personnel of this other potential compromised computer on campus. The OIT personnel took this computer off-line for analysis.

Upon conducting analysis the OIT personnel could not find any indication that this other machine had been compromised. However, this machine's owner was not the person who had connected to the WAREZ server on our Honeynet. OIT personnel speculated that the hacker used some method to get the password of this machine. The hacker could have used a brute force technique to guess this password. He could have harvested this password from a dummy web site set up to harvest usernames and passwords from unwitting users. The OIT personnel instructed the user to change his password by selecting a password that was more secure and to not use this password when establishing accounts at other web sites.

The Georgia Tech OIT personnel have stated that it would have been very difficult for them to detect that this system had been compromised using the existing security measures that they have available. Our GEN I Honeynet

allowed us to detect a system that most likely had been compromised by a hacker with some skill.

#### IV. LESSONS LEARNED

We have learned numerous lessons in the establishment and maintenance of the Georgia Tech Honeynet. Some of these lessons are:

1. Start Small – If you are going to install a Honeynet within your enterprise, start small. Begin initially with a single machine and operating system that you are familiar with installed behind the reverse firewall. This will allow you to begin to understand how to analyze the data that you will receive on the Honeynet. You will also be able to fine tune your configuration. The more machines that you have, the more data you will most likely receive going to and from the Honeynet.

2. Maintain good relations with your enterprise administrators. Inform your network administrators of the types of exploits that you are seeing. In some cases, they will already be aware of these exploits, but in other cases, you will have been the first person to notice them. The enterprise administrators should benefit from your efforts since they most likely provided you with the range of IP addresses that you are using for the Honeynet.

3. Focus on attacks and exploits originating from within your enterprise network. These are the attacks that can do the most damage to your enterprise. Inform your enterprise administrators immediately of these types of attacks since they indicate machines that have already been compromised within the enterprise.

4. Don't publish the IP address range of the Honeynet. There is no need to do this. Hackers and worms are constantly scanning across the Internet for machines to exploit. Your Honeynet will be found and attacked.

5. Don't underestimate the amount of time required to analyze the data collected from the Honeynet. This data must be analyzed every day. You will be collecting lots of information and it must be analyzed to provide any benefit. Most attacks take seconds to compromise and take over a vulnerable system. It can take weeks to analyze and document such an attack.

6. Powerful machines are not necessary to establish the Honeynet. The Georgia Tech Honeynet did not use state of the art machines and it functioned as intended.

#### V. CONCLUSIONS AND FURTHER RECOMMENDATIONS

The Georgia Tech Honeynet has served to increase the level of network security across the Georgia Tech Campus Enterprise network. We have identified numerous compromised systems on campus and provided this information to the campus network administrators. As a result, compromised systems can be rapidly investigated and the spread of these compromises across campus can be curtailed in the quickest possible manner.

A further benefit of the Honeynet is one of research in the areas of Information Assurance and Intrusion Detection. The possibility exists to detect new exploits launched against the campus network. Under the principles of data capture, all data associated with these exploits is collected for further analysis. As a result, counter measures can be taken against these new exploits and signatures targeting these exploits can be developed for the Enterprise IDS systems.

An area for further research would involve the establishment of a distributed Honeynet across the Georgia Tech Enterprise network. At present, the Honeynet only encompasses a single range of addresses within the Georgia Tech address space. In the future, we may look to develop a distributed Honeynet network throughout the entire Georgia Tech address space. This Honeynet network would be controlled by a dedicated management network that would be isolated from the rest of the Enterprise network [16]. Signatures could be developed for this network to alert on any instance of a computer from within the Georgia Tech network attempting to establish communication with one of these Honeynet machines. We believe that a distributed network would greatly enhance the security capabilities of the Honeynet. This area warrants further research.

#### VI. REFERENCES

- [1] E. Skoudis, *Counter Hack*, Upper Saddle River, NJ: Prentice Hall PTR, 2002, p. 47.
- [2] S. Northcut, L. Zeltser, S. Winters, K. Kent Fredericks, R. Ritchey, *Inside Network Perimeter Security*. Indianapolis, In: New Riders, 2003, p.5.
- [3] W. Stallings, *Network Security Essentials*, Upper Saddle River, NJ: Prentice Hall PTR, 2000, p. 322.

[4] R. Stiennon, M. Easley, *Intrusion Prevention Will Replace Intrusion Detection*, Gartner Research Notes, 30 August 2002, available at <http://www.gartner.com/reprints/intruvert/109596.html>, Dec 2002.

[5] The HoneyNet Project, *Know Your Enemy*, Indianapolis, IN: Addison-Wesley, 2002, pp. 12-17.

[6] The HoneyNet Project, *Know Your Enemy*, p. 20.

[7] L. Spitzner, *Honeypots- Tracking Hackers*, Indianapolis, IN: Addison-Wesley, 2003, pp. 242-261.

[8] <http://www.gatech.edu> SEP 2002.

[9] <http://www.oit.gatech.edu>, SEP 2002.

[10] <http://security.gatech.edu>, SEP 2002.

[11] <http://project.honeynet.org/papers/honeynet/tools/rc.firewall>, JULY 2002.

[12] <http://www.snort.org>. JULY 2002

[13] <http://www.snort.org/docs> JULY 2002

[14] <http://www.xcdroast.de>, JULY 2002.

[15] V. Paxson, S. Staniford, N. Weaver, "How to Own the Internet in Your Spare Time", *to appear in the Proceedings of the 11<sup>th</sup> USENIX Security Symposium (Security '02)*, <http://www.cs.berkeley.edu/~nweaver/cdc.web/>.

[16] Spitzner, *Honeypots- Tracking Hackers*, pp. 295-298.