Towards Practical Framework for Collecting and Analyzing Network-Centric Attacks

Napoleon Paxton
ncpaxton@uncc.edu

Gail-Joon Ahn
gahn@uncc.edu

Bill Chu
billchu@uncc.edu

College of Computing and Informatics
University of North Carolina at Charlotte

Abstract

Since nearly the beginning of the Internet, malware has been a significant deterrent to productivity for end-users, both personal and business related. A particular malware, known as a bot, can create networks of compromised machines called botnets, which are some of the most threatening adversaries over the Internet due in large part to the difficulty of identifying botnet traffic patterns. We have witnessed that existing signature-based detection and protection methods are ineffective, when used alone, in dealing with new unknown bots.

In this paper, we introduce a risk-aware network-centric management framework to detect and prevent targeted botnet attacks as well as propagation attempts within the network. As the first step in that direction we focus on learning more information about the bots by identifying malicious characteristics through the network traffic. Once we have their characteristics we then decide whether or not those characteristics present a significant risk to the network that is being protected by our architecture. Using risk as a factor in the decision process helps identify the bots more systematically. We present two scenarios that describe the risk-aware process and show that our framework shows great promise.

1. Introduction

Botnets are one of the largest problems that computers on the Internet face. Botnet commanders have many purposes for their army of compromised machines ranging from spam for hire to distributed denial of service (DDoS) and phishing attacks. The most destructive issue involving botnets is their capability to perform DDoS attacks [1]. Enterprises that offer web services have a difficult time in distinguishing a DDoS attack from a spike in legitimate customers accessing their service sites [2]. Malware includes a broad range of techniques that snoop on a user's activity, deploy Trojan horses, exploit with key and mouse logging software, and finally allow an adversary to control compromised machines in use [3-5]. Botnets are successful in part because of the difficulty in detecting new bots. This is the case mainly because static methods such as firewalls and anti-viruses can be bypassed by slightly changing the code on existing bots. To remedy this problem we propose identifying bots by their network traffic patterns. Since most bots are descendants of other bots that have been previously discovered many of the patterns are the same. In this paper, we describe a systematic framework to detect botnet traffic based on patterns and protect a network against bots using risks. Our framework includes a characteristic discovery system which sits on the subnet with the local area network and discovers malicious characteristics that are targeting the specific subnet, and also a risk-aware protection mechanism that sits in-line with the network that allows and blocks traffic based on the risk presented to the network by the characteristics in the packets.

The rest of the paper is organized as follows. Section 2 discusses background technologies and related work. The risk-aware framework is presented in Section 3. In Section 4, we discuss our preliminary results. Section 5 concludes the paper.

2. Background Technologies and Related Work

Honeynets have been used to learn as much about bots and the attacker sending bots as possible [6]. Even though this approach allows us to gather attackers' footprints, a systematic data analysis method is still needed. In the botnet, the command and control is where the attacker sends commands to the botnet. Currently most malicious bots use IRC to communicate with the command and control. IRC's built-in multicast capabilities make it easy for the commander to send orders to all the bots in the botnet without much effort [7]. A more destructive form of communication for bots is with the P2P protocol. These bots contain P2P clients and can communicate with one another without the use of a central command center. With this type of command and control the attacker can initiate commands by posing as a peer anywhere in the network.

1-4244-1500-4/07/$25.00 ©2007 IEEE

Other forms of command and control are also being used to a lesser degree, such as instant messaging and cellular phones. As researchers continue to find ways to protect against IRC based command and control structures, the number of botnets controlled by other protocols will continue to increase. As mentioned earlier, DDoS attacks are extremely difficult to detect. Most existing mechanisms have limitations to properly distinguish botnet traffic from legitimate traffic, generating a high false positive rate [8]. A high false positive rate may be its own denial of service, since legitimate traffic is blocked from accessing the network. Botnets continue to be a growing threat until a trustworthy mechanism is presented that effectively detects and blocks botnet attacks while allowing a very low false positive rate [9, 10].

Defending networks against botnet attacks is an emerging issue in network security and cyber crime research communities. To our knowledge, there are a few works using risk as a deciding factor such as a newly released McAfee's Advanced Botnet Protection in Intrusion Prevention System [8]. This tool takes a similar approach of our framework in that it uses a proxy to accept or block traffic that appears to be botnet related, but it doesn't use the risk value rigorously but mainly relies on a signature based approach. In [11], taxonomy of botnets was introduced to provide a response to botnets in degrading or disrupting them. This method involved discovery and proactive attack to the botnet. There is some similarity in building a taxonomy of bot characteristics used to identify botnets. Our work focuses on a bot taxonomy, as opposed to a botnet taxonomy to build up properties for our risk-aware mechanism. Some earlier works addressed issues on tracking botnets [12]. Such works adopted sensors and honeypots to investigate a pathway to and from botnets. Our approach uses a virtual space much like honeypots to capture bots and track botnets. In addition, we attempt to move one step forward by providing a way to categorize the bots and to record scanning activities targeted for vulnerable services. This allows us to grasp more details of the intent of the adversary and gives us a way to keep track of what services are being attacked the most. Our architecture is very similar to the approach as noted by Rajab, Zarfoss, Monrose, and Terzis[13]. Some key differences are that instead of creating "drones" to connect to a command and control, we "install" the actual bot on a honeypot to connect to its command and control. Our correlation system component is also a major difference in that we are keeping track of similarities in the bots and the sources that download the bots. Their approach is also more geared towards discovering the level of activity of botnets on the Internet without discovering characteristics for identifying similar unknown variants of each bot and corresponding botnet traffic.

3. Overview of our framework

This section gives detail of our approach that is based on three critical requirements as follows:

- *Systematically collect and analyze bot traffic over the Internet;*
- *Comprehensively discover characteristics and unique behaviors of bots; and*
- *Dynamically determine associated risks and generate corresponding detection rules.*

Our risk-aware framework is consisted of several modules with three core components to support these requirements: *Bot detection*, *Bot characteristics*, and *Bot risks* as shown in Figure 1.

3.1 Bot detection

Bot detection involves identifying both known and unknown bots that are trying to enter the protected network. We accomplish this by installing a malware collection system component on the network. The type of bots collected on each network is different because attackers target certain subnets with different bots.

The **malware collection system component** uses emulated vulnerabilities to entice attackers and to trick them into believing they are interacting with the actual vulnerable services. The attackers send their bots to this system and it is captured and not run. To realize this architecture we built upon a tool call Nepenthes [14]. Nepenthes has to restart each time a new module is created for a vulnerability. This is not feasible for us, since we want to start capturing bots as soon as a new vulnerability is found. To correct this discrepancy, we use a Ruby program to provide script space where modules are added on the fly and the system does not have to restart to go into effect. As new vulnerabilities are found in software, modules to capture bots that target these vulnerabilities will be added to the collection system.

In addition to the vulnerability modules to capture bots, is a scanning module to capture the scanning activity produced before the bot is downloaded to the collection system. This enables us to know the full story of each compromise, which helps us to develop concise characteristics for blocking the bots in the future.

3.2 Bot characteristics

To comprehensively discover characteristics and unique behavior of bots, the system is required to identify known malware, discover new malware, discover traffic patterns of individual malware, and discover a correlation between more than one instances of malware. **The network monitoring component** is employed to identify known malware signatures. In this system, both an anti-

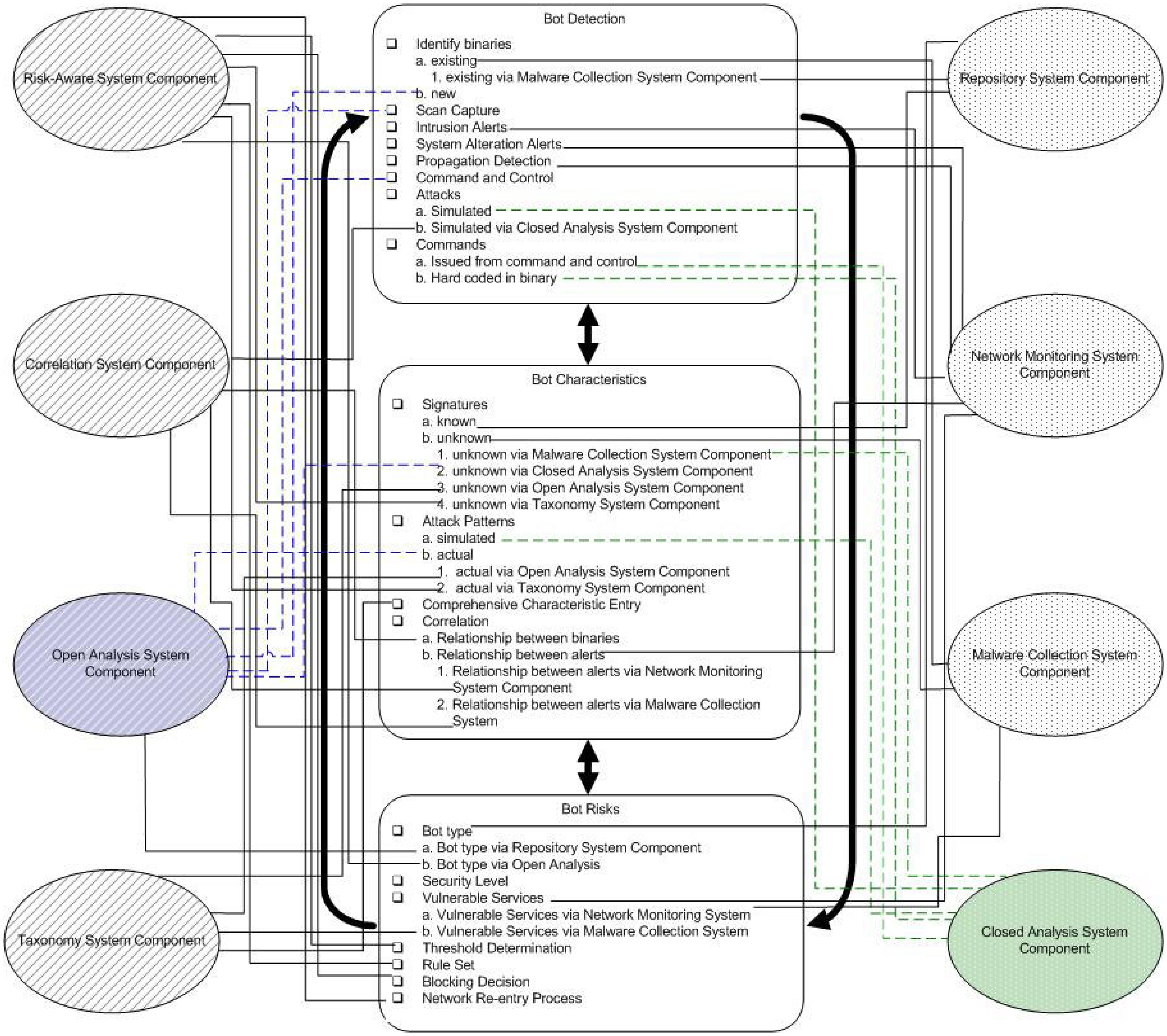

Risk-Aware System Component
Correlation System Component
Open Analysis System Component
Taxonomy System Component
Repository System Component
Network Monitoring System Component
Malware Collection System Component
Closed Analysis System Component
Bot Detection
□ Identify binaries
a. existing
1. existing via Malware Collection System Component
b. new
□ Scan Capture
□ Intrusion Alerts
□ System Alteration Alerts
□ Propagation Detection
□ Command and Control
□ Attacks
a. Simulated
b. Simulated via Closed Analysis System Component
□ Commands
a. Issued from command and control
b. Hard coded in binary
Bot Characteristics
□ Signatures
a. known
b. unknown
1. unknown via Malware Collection System Component
2. unknown via Closed Analysis System Component
3. unknown via Open Analysis System Component
4. unknown via Taxonomy System Component
□ Attack Patterns
a. simulated
b. actual
1. actual via Open Analysis System Component
2. actual via Taxonomy System Component
□ Comprehensive Characteristic Entry
□ Correlation
a. Relationship between binaries
b. Relationship between alerts
1. Relationship between alerts via Network Monitoring System Component
2. Relationship between alerts via Malware Collection System
Bot Risks
□ Bot type
a. Bot type via Repository System Component
b. Bot type via Open Analysis
□ Security Level
□ Vulnerable Services
a. Vulnerable Services via Network Monitoring System
b. Vulnerable Services via Malware Collection System
□ Threshold Determination
□ Rule Set
□ Blocking Decision
□ Network Re-entry Process

Figure 1: Risk-Aware Network-centric Attack Detection and Prevention Framework

virus and a firewall analyze the traffic to discover if anything matches their current rule set. The anti-virus is located in the **repository system component** where all data is stored.

The discovery of new malware is done by detecting the variations in the traffic using the **capture system component**. Once the traffic has been determined to have malicious packets, it will update the vulnerability list in the component and create a new module to capture the malware. To discover the traffic patterns, the bot is then automatically sent to a **closed analysis system** where it is ran in a virtual network and analyzed to return characteristics based on the behaviors it showed in the analysis. The initial characteristics obtained using the closed analysis will include the strings and pcap packets from simulated attacks.

The **open analysis system component** is then used to run the malware on the Internet. All transactions to and from the network are monitored and blocked if the malware we have installed is a significant contributor to an attack. The characteristics discovered here are the actual characteristics of the communication between the malware and the attacker.

In addition, we propose an IRC Sandman analysis tool that works with the open and closed analysis systems and it is written in Perl for the purposes of monitoring known hostile IRC channels, often used in conjunction with botnets. IRC Sandman's major purpose is to download secondary injections and store them under the heading of the original bot for later use. Figure 2 shows the architecture of IRC Sandman analysis tool.

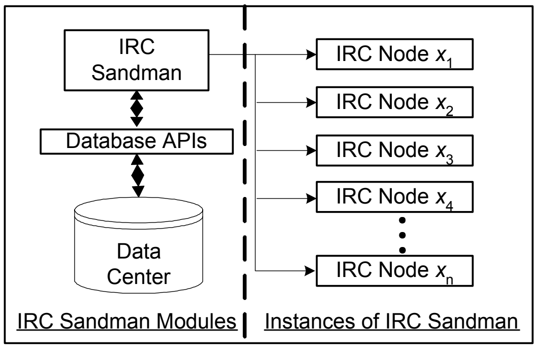


Figure 2: IRC Sandman Architecture

The malware characteristics are then correlated and displayed using the **correlation system component**. This system provides the ability to display the intelligence discovered from the characteristics. All alerts from the **network monitoring system component** are displayed in the intelligence report. This report is generated by querying the **open and closed analysis system components** and connecting the characteristics found by their md5 value, which is discovered in the **malware collection system component**

The correlation system component uses keywords found using the strings command from the closed analysis system to search the internet for possible characteristics, such as motives or frequency of metadata of the attacker. This can include irc commands, hard-coded dns entries, usernames, and so on. These characteristics are then added to the data from the open and closed analysis systems in the correlator. The **correlation system component** also finds links between different malware to discover patterns that may exist between different types of malware.

The **taxonomy system component** keeps a record of each bot's characteristics. All characteristics in the taxonomy are able to be correlated. When a relation is found that connects a bot to another bot or another type of malware, the **correlation system component** will send an update to the **taxonomy system component** which makes a reference to the malware that has been correlated.

3.3 Bot risks

An examination of the system is taken to discover what the vulnerabilities are. This examination takes into account the applications installed, the operating system and the importance of their vulnerabilities to the mission of the network. The more important an application or service is to a network, the more weight the vulnerability carries as it pertains to the risk-value. As we identify evidence of targeting these vulnerabilities in increasing the risk-value of that particular IP traffic, subsequent packets are recorded in a suspicious traffic storage component and not allowed to access the network until a certain level of legitimate traffic is recorded in the proxy. This is dynamic since every window of traffic would be different and increase & decrease the risk-level on the fly.

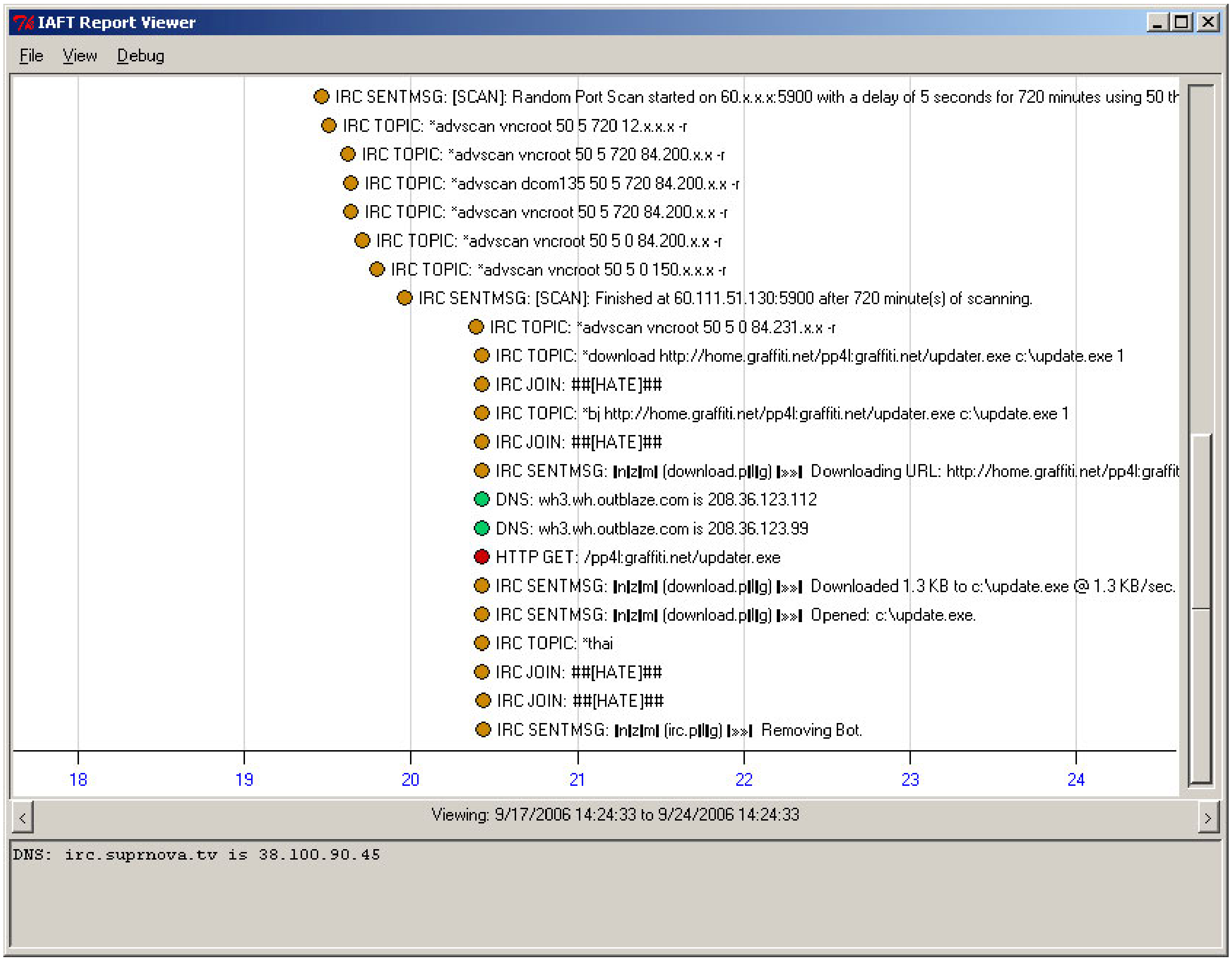
The **risk-aware system component** is the focal point of the system components. All the other components are working to discover as much information about the malware as possible to be able to discover whether or not they present a risk or not to the network. The risk-aware engine is unique for each particular system it is protecting because risk is relative to the individual, or company that is being protected. Each entity has its own risk factors and different weights for each factor. The first step in this engine is to discover the personalized risks. These risks include the vulnerabilities of the operating systems being used on the protected network, the applications, the known vulnerable services on the network, and the importance of the services. After the personalized risks are detected, we receive input from the **Taxonomy System Component** to give the malicious characteristics and then use the combination of the personalized risks and the characteristics to give a risk value that is used to aid in the determination of whether traffic is blocked or not. Due to the page limit, we omit the details of this part.

4. Preliminary Results

Our results demonstrate how our framework can help us identify different classes of bots. Table 1 illustrates a list of bots that we examined and analysis results of each bot. Each bot has an identification based on MD5 value. Also, we identified whether the collected bots are known or unknown bots. The targeted vulnerabilities were captured and further interactions with our system were also monitored. Those interactions include system changes, DNS queries and IRC communication, network service, and IRC communications with the intent. Once the analysis is completed, each analysis item would be stored in our bot taxonomy to categorize the bot for articulating patterns of bot-centric attacks. Using the open and closed analysis results, we also developed a correlation report so that we can identify all relevant system and network activities performed by a particular bot. As shown in Figure 3, more fine-grained intelligence report for the bot can be generated. Also, it allows us to further examine network and system activities at the specific time frame during the course of investigation actions.

Table 1: Comparison of Bots Ran through the Analysis Process

Bot	*Unknown /Unknown*	Trojan.Mybot-7706/ W32.virut	Unknown/Unknown	Trojan.Mybot-7669/ W32.IRCBot
Identification: MD5	3d35…	0c28…	5525…	C36d…
Vulnerabilities Targeted	ms04-011,ms03-039	ms04-011	ms04-011	ms04-011
System Interaction	19 files incl: lssas.exe	msnserve.exe	x.exe downloaded via ftp	lssas.exe
DNS Queries		bacho.hassouna.us	DNS asechka.ru	Info.prison-server.net
IRC Communications	Yes: Checks for paypal account	Yes: Checks for paypal account	Yes: Reptile Welcomes You	Yes: Reptile Welcomes You



```
IRC SENTMSG: [SCAN]: Random Port Scan started on 60.x.x.x:5900 with a delay of 5 seconds for 720 minutes using 50 th
IRC TOPIC: *advscan vncroot 50 5 720 12.x.x.x -r
IRC TOPIC: *advscan vncroot 50 5 720 84.200.x.x -r
IRC TOPIC: *advscan dcom135 50 5 720 84.200.x.x -r
IRC TOPIC: *advscan vncroot 50 5 720 84.200.x.x -r
IRC TOPIC: *advscan vncroot 50 5 0 84.200.x.x -r
IRC TOPIC: *advscan vncroot 50 5 0 150.x.x.x -r
IRC SENTMSG: [SCAN]: Finished at 60.111.51.130:5900 after 720 minute(s) of scanning.
IRC TOPIC: *advscan vncroot 50 5 0 84.231.x.x -r
IRC TOPIC: *download http://home.graffiti.net/pp4l:graffiti.net/updater.exe c:\update.exe 1
IRC JOIN: ##[HATE]##
IRC TOPIC: *bj http://home.graffiti.net/pp4l:graffiti.net/updater.exe c:\update.exe 1
IRC JOIN: ##[HATE]##
IRC SENTMSG: |n|z|m| (download.p||g) |»»| Downloading URL: http://home.graffiti.net/pp4l:graffit
DNS: wh3.wh.outblaze.com is 208.36.123.112
DNS: wh3.wh.outblaze.com is 208.36.123.99
HTTP GET: /pp4l:graffiti.net/updater.exe
IRC SENTMSG: |n|z|m| (download.p||g) |»»| Downloaded 1.3 KB to c:\update.exe @ 1.3 KB/sec.
IRC SENTMSG: |n|z|m| (download.p||g) |»»| Opened: c:\update.exe.
IRC TOPIC: *thai
IRC JOIN: ##[HATE]##
IRC JOIN: ##[HATE]##
IRC SENTMSG: |n|z|m| (irc.p||g) |»»| Removing Bot.
```

Viewing: 9/17/2006 14:24:33 to 9/24/2006 14:24:33

```
DNS: irc.suprnova.tv is 38.100.90.45
```

Figure 3: Correlation Intelligence Report

5. Conclusion

In this paper, we have introduced a risk-aware network-centric attack detection and prevention framework. Also, we described functionalities and features of each component in the framework. In addition, we elaborated our preliminary results based on the honeynet-based testbed which demonstrated the feasibility of our framework.

For the future work, we would attempt to integrate risk-aware system component with our current testbed architecture to seek more systematic way for identifying and calculating risk values involved with bot traffics. Also, our correlation system would be enhanced to generate more meaningful intelligence report.

6. Acknowledgements

This work was supported, in part, by funds provided by e-Business Technology Institute at UNC Charlotte.

References

[1] Phishing Reaches an All-time High in March. Available at www.it-observer.com

[2] S. Kandula, D. Katabi, M. Jacob, and A. Berger, "Botz-4-Sale: Surviving Organized DDoS Attacks That Mimic Flash Crowds," In Proceedings of NSDI, 2005

[3] R. Dhamija and J. D. Tygar. "The battle against phishing: Dynamic security skins". *Symp*. On Usable Privacy and Security, 2005.

[4] S. Saroiu, S. D. Gribble, and H. M. Levy. "Measurement and Analysis of Spyware in a University Environment". *Proc. NSDI,* 2004.

[5] E. Skoudis and L. Zeltser. "*Malware: Fighting Malicious Code*". Prentice Hall, 2004.

[6] The Honeynet Project & Research Alliance. "Know Your Enemy: GenII Honeynets". Available at www.honeynet.org/papers/

[7] J. Li, T. Ehrenkranz, and G. Kuenning, "Simulation and Analysis on the Resiliency and Efficiency of Malnets," *In the Proceedings of the 19th Workshop on Principles of Advanced and Distributed Simulation*, 2005

[8] J. Dunn. McAfee launches first bot-killing system. Available at www.techworld.com

[9] The Honeynet Project & Research Alliance, "Know Your Enemy: Tracking Botnets," Available at www.honeynet.org/papers/

[10] T. Holz, "A Short Visit to the Bot Zoo," *Security & Privacy Magazine*, IEEE 2005

[11] D. Dagon, G. Gu, C. Zou, J. Grizzard, S. Dwivedi, W. Lee, and R. Lipton,"A Taxonomy of Botnets," Available at www.math.tulane.edu/~tcsem/botnets/

[12] F. Freiling, T. Holz, and G. Wicherski, "Botnet Tracking: Exploring a Root-Cause Methodology to Prevent Distributed Denial-of-Service Attacks," *In the Proceedings of the 10th European Symposium on Research in Computer Security*, 2005

[13] Moheed Abu Rajab, Jay Zarfoss, Fabian Monrose, and Andreas Terzis, "A Multifaceted Approach to Understanding the Botnet Phenomenon," *In the Proceedings of ACM IMC*, 06

[14] Nepenthes-Finest Collection, Available at nepenthes.mwcollect.org/

[15] Truman – The Reusable Unknown Malware Analysis Net, Available at www.lurhq.com/truman/

[16] The Honeynet Project, Know Your Enemy: Sebek-- A kernel based data capture tool, November 2003, Available at www.honeynet.org