12-2008

A Model to Use Denied Internet Traffic to Indirectly Discover Internal Network Security Problems

Chet Langin
Southern Illinois University Carbondale

Hongbo Zhou
Southern Illinois University Carbondale

Shahram Rahimi
Southern Illinois University Carbondale, rahimi@cs.siu.edu

Published in Langin, C., Zhou, H., & Rahimi, S. (2008). A model to use denied Internet traffic to indirectly discover internal network security problems. IEEE International Performance, Computing and Communications Conference, 2008. IPCCC 2008, 486-490. doi: 10.1109/PCCC.2008.4745091

Recommended Citation

Langin, Chet, Zhou, Hongbo and Rahimi, Shahram. "A Model to Use Denied Internet Traffic to Indirectly Discover Internal Network Security Problems." (Dec 2008).

A Model to Use Denied Internet Traffic to Indirectly Discover Internal Network Security Problems

Chet Langin
Information Technology
Southern Illinois University
Carbondale, Illinois USA

Hongbo Zhou and Shahram Rahimi
Dept. of Computer Science
Southern Illinois University
Carbondale, Illinois USA

Abstract

We propose a model for using firewall log entries of denied inbound Internet traffic for indirect discovery of local IP addresses that have security problems. This method is used successfully to discover two computers on the network of Southern Illinois University which were infected with malicious feral software, as well as two more IP addresses on the university network with other security problems.

1 Introduction

Malicious software began to become dangerous to computing environments soon after the first computer virus was created on November 3, 1983[2]. Intrusion detection and prevention became paramount after the Internet (Morris) Worm of 1988 disrupted computer operations on a wide scale. However, efforts at information security have not been able to completely prevent malicious Internet activity. Other notable infections include Code Red in 2001, and the SQL Slammer in 2003. In 2005, a sniffer on the TJ Maxx network was used to steal over 45 million credit card numbers costing that corporation over $100 million in settlement claims.

In 2007, an army of infected computers was used in coordinated Denial of Service (DoS) attacks in cyber warfare to disable the network of Estonia, a small country[3, 8]. These armies are called *botnets* with each infected computer called a *bot* (short for *robot*.) Bots take their orders from bot masters via command and control centers (C&C) using various protocols, such as HTML, Internet Relay Chat (IRC), and Peer-to-Peer (P2P). Estimates of the numbers and sizes of botnets vary, but one study, for example[10], discovered 3,290 unique IRC botnets with 700,700 distinct IP addresses. P2P botnets are more problematic because they encrypt their traffic, and their distributed system makes it difficult to trace and find the command and control centers. Botnets are particularly insidious because they can accomplish whatever code their malicious master is capable and imaginative enough to deliver to them. *It is clear that bonets have become the most serious security threat on the Internet*[7].

However, traditional packet analysis intrusion detection is stymied by the encrypted P2P botnet command and control network traffic. Our model assists in overcoming this barrier by using Internet traffic recorded in firewall log entries to indirectly indicate locally infected computers. We have successfully used this model to locate previously unknown infected computers.

The P2P botnet model and the general version of the model are explained in Section 2. Then in Section 3 we relate the methodology on how the model worked to advantage. Section 4 provides references for related work, and Section 5 is the conclusion.

2 Model

We developed the model while studying P2P bots and then we generalized the model. In this section, first the model specific to P2P bots is described. Then, we describe the general version of the model using Fig. 2.

2.1 P2P Botnet Model

When a computer gets newly infected with a P2P bot, it attempts to notify the bot master's C&C of its infection so that the C&C knows that the newly infected computer is ready for further instructions. This scenario describes the specific P2P botnet model as illustrated in Fig. 1.

Step 1 in Fig. 1.a illustrates the initial contact, which can be accomplished in a virtually unlimited manner of mundane ways imitating normal Internet traffic to one or more IP addresses and/or domain names encoded in the malicious software which infected the computer.

Since a border firewall is typically configured to allow outgoing traffic to the Internet which was initiated by a

978-1-4244-3367-4/08/$25.00 ©2008 IEEE

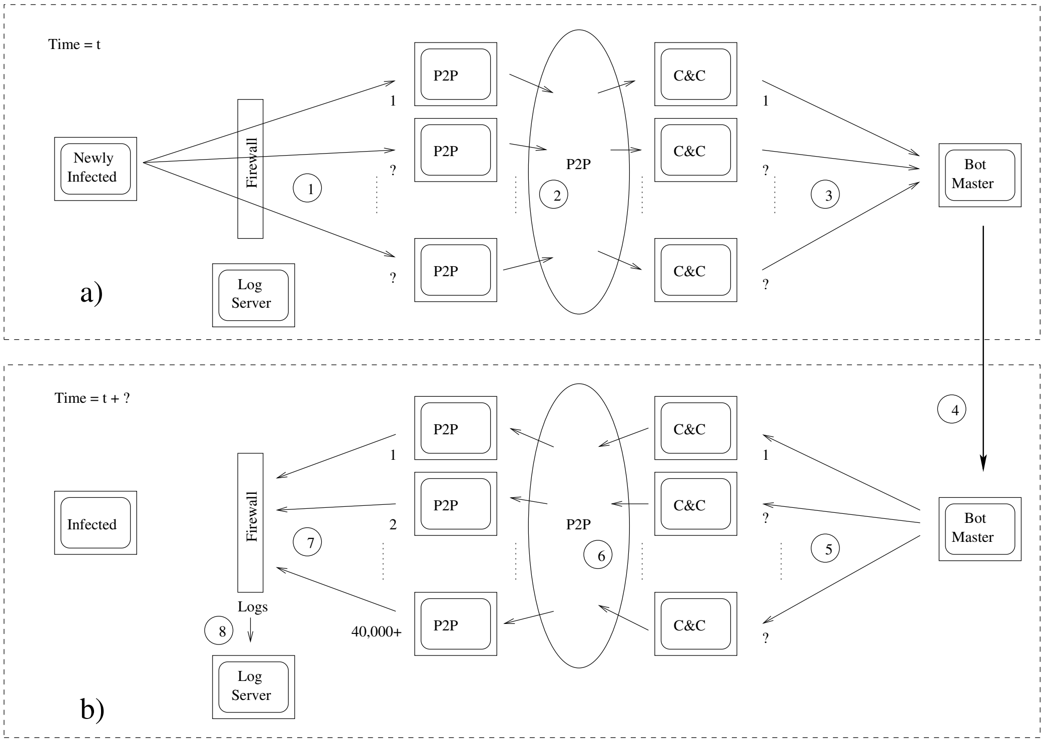

Figure 1. This figure illustrates the P2P Botnet Model. See the text for a detailed description.

computer on the internal network, this initial contact from the infected computer, which appears to be normal network traffic, goes out unchallenged and likely unlogged. If any logs of this traffic do go to a log server, these logs are indistinguishable from normal network traffic in the model.

The initital contact goes to one or more computers in or associated with the P2P botnet. Step 2 in Fig. 1.a symbolizes how the contact from the newly infected computer transmits via this P2P network through the Internet cloud obscuring its route to the C&C of the bot master, making it very difficult to locate and identify the bot master.

Step 3 in Fig. 1.a symbolizes that the C&C has one or more contacts with the bot master, who may not even get a notice of each newly infected computer, but who at least gets periodic summary information of the extent of his or her botnet.

Step 4 represents a transition from Fig. 1.a to Fig. 1.b. At some point in time, which could be days later, the process begins for the bot master to send commands to his or her bots.

We do not actually see what happens when the actions in Fig. 1.a occur in the model. The newly infected computer in the model is not a honeypot; it is a computer being used by someone who responds to spam e-mail, clicks on a malicious website, or takes other action which gets the computer infected. (While we can monitor known e-mail spam and known malicious web sites, we cannot know in every case when a user takes an action to infect his or her computer, at the time the infection occurs.)

The bot master initiates action with one or more of his or her C&C servers as illustrated in Step 5 of Fig. 1.b. Step 6 illustrates that the action is propogated through the Internet via the P2P botnet, which hides the route back to the bot master. We have seen the logs from apparently over 40,000 unique IP addresses in a P2P botnet attempting to make contact back to an infected computer.

In the model, the incoming botnet traffic is blocked by the firewall because these botnet sessions were initiated outside of the network, which is consistent with a properly configured default deny firewall. Step 7 in Fig. 1.b illustrates this network traffic being blocked. Step 8 symbolizes logs of this blocked network traffic being sent to the log server for analysis. The IP addresses being blocked by the firewall in Step 7 do not have to include the same IP addresses the newly infected computer contacted in Step 1 of Fig. 1.a, even though they look similar. Also, in Step 1, only one contact has to get out, whereas in Step 7, many thousands of attempted contacts come back.

The model was used to practical advantage when the firewall logs from Step 8 were analyzed to discover two previ-

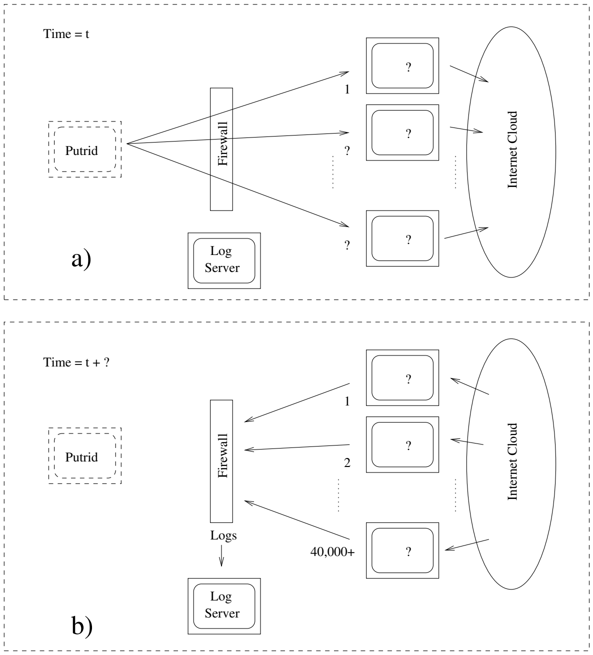

Figure 2. This figure illustrates a generic form of the P2P Botnet Model. See the text for more detailed information.

ously unknown feral bots on the network.

2.2 General Model

The P2P Botnet Model can be generalized for other types of malignant network behavior as illustrated in Fig. 2. In Fig. 2.a, a malignant IP address produces a putrid network *scent* which is picked up by other computers on the Internet. We cannot distiguishable this scent from normal network traffic in the model. The putrid network scent is spread via P2P or other means on the Internet. In Fig. 2.b, attempted responding contacts are made to the malignant IP address, where they are denied by the firewall and logged.

3 Methodology

We tested the P2P botnet model on border firewall log entries for the university's Class B network. We had become aware of two P2P bots on the network which we labeled Bot1 and Bot2.

We analyzed log entries for 24-hour periods which included inbound traffic for Bot1 on February 6, 2008, and for Bot2 on February 12, 2008. These logs also contained data for potentially all of the other IP addresses on the network. We had over 20 million border firewall entries in the test data, the bulk of which were logs of incoming traffic which was denied by the border firewall.

We parsed the log entries and loaded the data into a MySQL database which summarized the data, creating an eight-dimensional vector for each IP address of the following elements: tot_cnt (total count) is the total number of log entries; src_cnt (source count) is the number of unique IP addresses from the outside; port_cnt (port count) is the number of unique destination ports; lo_port (low port) is the lowest destination port; hi_port (high port) is the highest destination port; and, icmp_cnt (ICMP count), tcp_cnt (TCP count), and udp_cnt (UDP count) are the counts of each of those protocols.

As an example, the vector for the local Bot1 IP address approximated these values: 400,000; 40,000; 80; 80; 60,000; 0; 100; 400,000—meaning that there were (approximately) 400,000 firewall log entries from 40,000 outside IP addresses attempting connections on 80 destination ports from Port 80 to Port 60,000, with no ICMP entries, 100 TCP entries, and 400,000 UDP entries. (The same traffic on another network would likely have different values because these values depend on the specific firewall settings that produce the log entries.) Approximately 60,000 of the IP addresses had entries, giving an input matrix size of 8 by approximately 60,000.

We used a Self-Organizing Map (SOM), a type of Artificial Neural Network (ANN), to cluster the vectors in the matrix. See [6] for more information about SOM, a technical description of which is beyond the scope of this paper. We clustered the data using 1,000 nodes, with each node representing a cluster, providing an average of 60 local IP addresses per cluster. Node 996 was the Best Matching Unit (BMU) for both Bot1 and Bot2, and no other IP addresses had Node 996 as the BMU, meaning that SOM successfully isolated the bot IP addresses from the other IP addresses.

We then also used SOM for classification of new firewall log data to look for additional bots on the network. We processed new feral firewall log entries on a daily basis looking for local IP addresses which had Node 996 as the BMU. The distributed nature of the university prevents general access to suspect computers, but we have, nonetheless, been able to confirm some results. The SOM produced less than one Node 996 match per day on the average. We labeled the matches starting with Suspect01, and the confirmed results are as follows.

The SOM discovered Suspect01 on March 29, 2008, and again on April 1, 2008. Another appliance later detected Suspect01 making probes of IP addresses in a manner indicating an infection. Suspect01 was found to have both P2P software and multiple infections, some dating back to the SOM discovery. However, the exact P2P software and infections were not made available to us.

The SOM discovered both Suspect02 and Suspect03 on

April 1, 2008, and the users stated that they were using non-malicious P2P software.

Suspect04 was discovered on April 4, 2008. We could not find any evidence that this IP address was assigned or active on the network. One possible explanation of this could be that the traffic was backlash from a spoofed IP address.

Suspect10 was discovered on July 24, 2008, and was an appliance which was using outdated firmware and which had a web interface which was vulnerable to a cross-site scripting attack.

The SOM discovered Suspect13 on August 12, 2008. A couple of hours later, an alert from another detection method indicated that Suspect13 was infected with a P2P bot. Suspect13 was disabled from network traffic. However, Suspect13 was discovered again by the SOM on August 13, 2008, for the network traffic on the previous day before it was disabled. A couple of hours later, again, the other detection method provided an alert stating that Suspect13 was positive for a P2P bot.

Suspect17 was a DHCP IP address used at a station to assist incoming students in finding and cleaning malware off of their computers and in setting up Windows firewalls, automated patching, and anti-virus software. No computer was logged in as using this IP address when the pertinent log entries were recorded by the firewall. We believe a putrid network scent on this IP address was caused by one or more of several possible problems with student computers, including vulnerabilities, and possible malware.

No other P2P bots of the same type were detected on the network by other means on days in which we ran SOM analyses. Some of the above suspects do not fall clearly into true positive and false positive catagories, but it is apparent by observation that both the P2P bot model and the general version of the model have validity.

4 Related Work

Ilgun[4] categorizes three types of intrusion detection: rule-based penetration identification, anomaly detection, and model-based intrusion detection.

Intrusion Detection System (IDS) appliances are primarily rule-based: They look for strings, flags, ports, and other information in packets and traffic flow data that indicate known intrusion patterns. Two disadavantages for rule-based systems in looking for P2P bots are that 1) you have to know in advance what you are looking for, and 2) encrypted packets obfuscate needed information for rule-matching.

Anomaly detection uses statistical or other methods to determine normal network traffic, and then reports what is not normal. Disadvantages of anomaly detection of P2P botnets are that the initial outgoing contacts of the newly infected computers are indistinguishable from normal traffic, and the incoming traffic from the botnet never reaches the local network through a properly configured firewall.

Many papers have been written on rule-based and anomaly intrusion detection, but only a few have been written on intrusion detection models.

Kemmerer [5] proposed a network intrusion detection model in 1997 which he called Network State Transition Analysis Tool (NSTAT) and which was based on earlier host-based models. He noted that IDS did not take into consideration two or more users working together to execute a penetration!

Cho [1] proposed a Hidden Markov Model (HMM) model in 2003 to improve intrusion detection performance by only considering the privilege transition flows based on the domain knowledge of attacks.

More recently, in 2007, Zhou [9] noted the *massive number of simple alerts of low-level security-related events* for signature-based (rule-based) IDS, and proposed a formal model utilizing the concept of *capability* to implement an alert correlator for complex multistage intrusions, expanding an earlier *requires/provides* model.

None of these previously published network models are appropriate for P2P botnets because there are no observable state transitions, privilege transitions, or alert correlations to consider.

5 Conclusion

We proposed a P2P botnet model and a general version of the model to describe Internet responses to bot infected computers and malignant IP addresses on the university network. We showed how this model can be used to indirectly discover infected computers and other network security problems on the network by analyzing firewall log entries of denied Internet traffic.

Advantages of the model are that it suggests techniques in addition to rule-based intrusion detection, malware detection, and intrusion prevention in the effort to eradicate malicious software and other security problems. A disadvantage of the model is that the specific techniques for utilizing it will vary for different locations because of different network environments and different firewall configurations. Even so, we found some security issues faster using the model than we did by using other methods. We also found a security issue involving an appliance which likely would never have been discovered by other means.

References

[1] S.-B. Cho and H.-J. Park. Efficient anomaly detection by modeling privilege flows with hidden markov model. *Computers and Security*, 22(1):45–55, 2003.

[2] F. Cohen. Computer viruses. In *7th DoD/NBS Computer Security Conference*, pages 240–263, 1984.

[3] J. Davis. Hackers take down the most wired country in europe. *Wired*, 15(9), September 2007.

[4] K. Ilgun, R. A. Kemmerer, and P. A. Porras. State transition analysis: A rule-based intrusion detection approach. *IEEE Transactions on Software Engineering*, 21(3), 1995.

[5] R. A. Kemmerer. Nstat: A model-based real-time network intrusion detection system. Technical report, University of California-Santa Barbara, November 1997.

[6] T. Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer-Verlag, Berlin Heidelberg New York, third edition, 2001.

[7] W. Lee, C. Wang, and D. Dagon. *Botnet Detection: Countering the Largest Security Threat*. Advances in Information Security. Springer, 2008.

[8] J. Robb. When bots attack. *Wired*, 15(9), September, 2007 2007.

[9] J. Zhou, M. Heckman, B. Reynolds, A. Carlson, and M. Bishop. Modeling network intrusion detection alerts for correlation. *ACM Transactions on Information and System Security (TISSEC)*, 10(1), 2007.

[10] J. Zhuge, T. Holz, S. Y. Han, J. Guo, and W. Zou. Characterizing the irc-based botnet phenomenon, 2007.