Tipping Point Evaluation of a Network Intrusion Detection System (NIDS) for Peer-to-Peer Networks

Adam Magana, Benjamin Christen, and Graciela Perera
Department Computer Science and Information Systems
Youngstown State University,
Youngstown, OH. 44555, USA
Emails: anmagana@student.ysu.edu, bochristen@student.ysu.edu, gcperera@ysu.edu

ABSTRACT

Unstructured Peer-to-Peer (P2P) networks for content distribution are decentralized and robust. Their growing popularity has an impact on security because they can be use to deliver malicious code and potential remote control. Additionally, P2P networks create a hole in a firewall that can be used to obtain private and confidential information. P2P security in many organizations focuses on blocking the default port used in P2P communication. We proposed configuring and experimentally evaluating an off the shelf Network Intrusion Detection Systems (NIDS) to minimize and manage the threads posed by P2P networks in the Youngstown State University network. The NIDS selected from TippingPoint promises to be simple to use and provide concrete information when an intruder tries to penetrate the network. Furthermore, there is a default recommended setting to block malicious traffic.

Our evaluation of the NIDS selected seeks to test the methods provided by TippingPoint that deal with P2P traffic. Also, we will investigate procedures that can analyze the scenario in which a P2P network is configured to listen on the TCP port 80 (HTTP). Most organizations allow the traffic from port 80 to go through the firewall and P2P threats may disguise themselves as HTTP traffic.

1. Introduction

Unstructured Peer-to-Peer (P2P) networks for content distribution are decentralized and robust. Their immense popularity has a severe impact on security because they can be use to deliver malicious code and potential remote control. Additionally, P2P networks create a hole in a firewall that can be used to obtain private and confidential information [4,6,8,10]. Intrusion Detection Systems (NIDS) can be used to minimize and manage the threads posed by P2P networks.

Many NIDS solutions are defined over a theoretical framework or deployed using a prototype solution in a controlled environment [3,6,10]. Furthermore, the evaluation of many existing off the shelf products depend on commercial reports that lack the analysis, unbiased study and judgment of entity such as academia [8].

This investigation proposes configuring and experimentally evaluating an off the shelf Network NIDS to minimize and manage the threads posed by P2P networks in the Youngstown State University (YSU) network. The NIDS selected from TippingPoint promises to be simple to use and provide concrete information when an intruder tries to penetrate the network [4,8]. *The questions that we seek to answer are: 1) how does TippingPoint prevent P2P threats? 2) is TippingPoint simple to use and configure?*

Our evaluation will study and test the methods provided by TippingPoint that deal with P2P traffic. Also, we will investigate procedures that can analyze the scenario in which a P2P threat may disguise itself as HTTP traffic. Most organizations allow the traffic from port 80 to go through the firewall.

The experimental evaluation of TippingPoint for P2P threats will not only provide real world insight and an assessment of how to configure and use an existing NIDS but will provide a real world research experience for two undergraduate students in Computer Science and Computer Information Systems at YSU . Allowing students to build their technical skills, participate and apply the concepts learned outside of the classroom.

We hope projects like these can provide a bridge between academia and companies so students can accumulate the experience and training that will allow them to quickly be assimilated by companies. Also motivate students to expand their knowledge by pursuing graduate school.

2. Peer-to-Peer Networks

The functionality of Peer-to-Peer networks is structured in two phases. The first phase allows a host to find other P2P hosts and connect to the network. The second phase enables a connected host to search for files by broadcasting queries and allows a file to be downloaded.

2.1 Connecting to the network

In the first phase, a host joins Gnutella by obtaining a list of hosts (bootstrap list) from a

This material is based on work supported by the Ohio Latino Affairs Commission 2008 Grassroots Initiative Mini-Grant

bootstrapping host cache (bootstrapping host). The bootstrap list contains the IP address and port number of the hosts that have participated in the network (e.g., connected to the network).

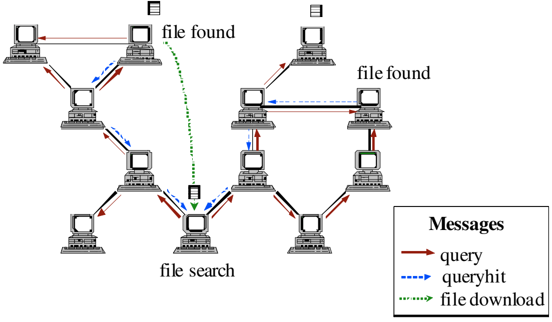

Figure 1. P2P Network

A host joins the network by directly connecting to a set of random hosts from the bootstrap list. The host connects to the hosts via a permanent TCP/IP connection (one connection per host) [7]. Each host that accepts a connection is called a neighbor, and the set of all hosts connected is denominated neighborhood. When a host loses a neighbor, that neighbor is replaced by a host not belonging to the neighborhood selected from the bootstrap list [2,7].

Discovering new hosts in the network is done by requesting a host's address and port number using a ping message. Ping messages are broadcasted, thus pings are forwarded to all neighbors. If a host receiving a ping can accept additional neighbors it will answer the request with a pong message containing its address and port number [7]. A pong message is routed back by the host that forwarded the ping. All hosts maintain a routing table that registers the unique identifier of a ping as well as the identifier of the host from which the ping was forwarded from. If a host receives copies of the same ping from different neighbors, it will only register the first copy received in the routing table. All other copies are not registered and will not be forwarded. This assures that the pong is routed back to the host that initiated the ping through a unique path.

2.2 Searching and downloading files

The second phase is summarized in figure 1 and is initiated once the host has connected to the network. The host shares a collection of files that can be downloaded by other hosts. These files are stored in the shared file directory from which files are downloaded. A user at a host can search for a file by broadcasting a query containing file name keywords [7]. The host receiving the query matches the keywords contained in the query received with the keyword file names stored in the index of shared files. The index of files shared is the data structure that associates with each file a list of file name keywords used to answer the queries [1,2,7]. All P2P hosts have the capability to search by broadcasting queries [7].

Downloading a file occurs only after one or more queryhits are received for a query broadcast. The host that wants to download the file connects directly to the host storing the file via an HTTP GET. To download the file, the host needs the IP address, port number, and file identifier obtained from the queryhit [1,2,7].

3. Experimental evaluation of TippingPoint

Over the last few years many off the shelf products for Network Intrusion Detection (NID) have been developed. In particular, when the first NID systems were introduced in the market P2P networks did not have the popularity they have today [4,6,8]. Thus, many NID systems did not considered P2P a severe threat.

Recently, P2P is considered such a important threat, that it is very difficult to find a NIDS without some kind of P2P network protection incorporated. The Gartner Magic Quadrant Report in 2006 mentions TippingPoint as one of the best network-based device that can be deployed in the front or behind a corporate firewall [8]. TippingPoint claims to protect the network from worms, viruses, Trojans, DoS attacks, spyware, and more relevant to this study P2P threats. The switch like speed in which the TippingPoint systems analyzes traffic (i.e., layers 2-7 of the OSI model) is due to their patented ASIC Threat Suppression Engine (TSE) [4,8]. This attributes makes TippingPoint the ideal candidate for our evaluation since the Youngstown State University (YSU) Network has many potential P2P users (i.e., students) [5].

The evaluation of TippingPoint focuses around two questions. The first question focuses on investigating the preconfigured methods that prevent P2P threats? The second question seeks to determine is TippingPoint simple to use and configure?

3.1 How does TippingPoint prevent P2P threats?

The answer to *how does TippingPoint prevent P2P threats?* is defined by the results obtained from two experiments. The first experiment seeks to test the defaults settings that block P2P traffic. These settings block a user attempting to establish a connection to the P2P network. That is, TippingPoint will block the connection when a host from inside the YSU network attempts to establish a connection or search for a file from a P2P network. The second experiment will test whether TippingPoint monitors file download request from a P2P network and either blocks or limits the rate in which a file is downloaded.

The P2P network client used for both experiments described before will be Shareaza. Given that the P2P network Shareaza can simultaneously connect to bandwidth demanding popular music and video file sharing networks. Lastly, because Shareaza is an open source program that can be modified. We will modified and configured Shareaza to listen on the TCP port 80 (HTTP). This will determine if a potential P2P threat may disguise itself as HTTP traffic and pass through the firewall.

3.2 Is TippingPoint simple to use and configure?

The answer to the last question *is TippingPoint simple to use and configure*? Will be determine by quantifying the amount of time that takes two undergraduate students Adam Magana (computer science major) and Benjamin Christen(computer information systems major) to configure and use TippingPoint. Additionlly, the students will keep a notebook containing the date, time, description of the activities they perform, and the problems faced during the use and configuration of the TippingPoint system.

4. Conclusions

Peer-to-Peer networks can be use as an additional vector of delivery of threats such as worms, viruses, Trojans, spyware, DoS attacks, DDoS attacks and more recent Botnets [10]. The likelihood of the success of against these threats depends on devices such as NIDS. The NIDS can inspect packets at certain important locations in the network (i.e., behind or in front a firewall), seek malicious and anomalous behavior, or drop packets that contain a certain string identified as malicious [4,6,10]. Many studies have analyzed and proposed prototype solutions that are very hard to experimentally evaluate in a real world environment [3,8,10]. We propose the evaluation of a real NIDS over an existing University network with realistic threats. The unique contribution of our investigation is twofold. The first is the experimental evaluation of an off the shelf solution NIDS (i.e., TippingPoint) with respect to P2P threats. The second is testing the simplicity of use and configuration of the NIDS.

Currently, we have purchased the TippingPoint 50 systems with a throughput of 50 megabits per second and latency of less than one millisecond. We estimate to have the system configured and evaluated by the first quarter of the year 2009.

References

1. T. Karagiannis, K. Claffy, and M. Faloutsos, File-Sharing in the Internet: A Characterization of P2P Traffic in the Backbone, *Technical Report, UC Riverside*, November 2003.
2. T. Karagiannis, A. Broido, N. Brownlee, kc claffy and M. Faloutsos, Is P2P dying or just hiding?, *Globecom*, December 2004.
3. V. Paxson, Bro: A System for Detecting Network Intruders in Real-Time, *Computer Networks*, Vol. 31, December 1999, pp. 2435-2463.
4. TippingPoint Intrusion Prevention Systems 50, URL:http://www.tippingpoint.com/products_ips.html.
5. F. Sole, A Short History of the Computer Network at YSU, http://network-services.ysu.edu/faqs.htm.
6. L. Ellis, BitTorrent's Swarms have Deadly Bite on Broadband Nets, *Multichannel News*, May 2006. URL:http://www.multichannel.com/article/CA6332098.html.
7. G. Perera and K. Christensen, Broadcast Updates with Local Look-up Search (BULLS): A New Peer-to-Peer Protocol, *ACM Southeastern Conference*, pp. 124-129, March 2006.
8. G. Young and J. Pescatore, Gartner Magic Quadrant Report on Network Intrusion Prevention Systems, 2006. URL:http://www.tippingpoint.com/pdf/analyst/tippingpoint2138.pdf
9. Peer-to-Peer network Shareaza Systems, 2008. URL:http://www.shareaza.com/
10. M. Engle and J. I. Khan, Vulnerabilities of P2P Systems and a Critical Look at their Solutions Technical Report, 2006.