Chapter 12

USING PEER-TO-PEER TECHNOLOGY FOR NETWORK FORENSICS

Scott Redding

Abstract Networked computer systems are under a barrage by combatants attempting to obtain unauthorized access to their resources. Methods must be developed to identify attacks on the systems and provide a forensically accurate description of the chain of events related to the unauthorized activity. This paper proposes a peer-to-peer (P2P) framework for network monitoring and forensics. Host-based security tools can be used to identify malicious events. The events can be communicated to other peers over a P2P network, where analysis, forensic preservation, and reporting of related information can be performed using spare CPU cycles.

Keywords: Network forensics, peer-to-peer networks

1. Introduction

Networked computer systems are under a barrage by combatants attempting to obtain unauthorized access to their computing resources. Methods must be developed to identify attacks on the systems and provide forensic evidence of the chain of events related to the unauthorized activity. While the ideal solution may be to save a record of all bits that traverse a computer network, that is practically infeasible due to the massive amount of traffic on high speed networks. Therefore, a more reasonable solution is to use end systems on the network to identify the interesting data and to inform other systems of their findings in order to: analyze events, identify attacks, share findings, protect the network, and to preserve evidence. Important data relating to unauthorized access of the systems must be preserved in a forensically acceptable manner while striving to minimize the data storage resources.

2. Peer-to-Peer Framework

The P2P network framework approach to network monitoring and forensics is a solution that utilizes end systems in a peer-to-peer manner to collect and analyze host-based security events. A security event is any incident identified by a host-based protection mechanism. The P2P network forensics architecture is reliant on cooperation from as many peers as possible within a community of interest. A neighborhood or peer group is a set of peers sharing a common attribute. This attribute could be operating system, network interface type, hardware similarity, software application, platform use, server type, network subnet, location, organizational work group, or even an end user characteristic. Each of the systems can be member of many neighborhoods in the P2P network and can effectively contribute data pertaining to the status of their neighborhoods. A community is comprised of the set of all neighborhoods. The P2P framework is based on the Java programming language, so inclusion of a variety of computing devices is easily accomplished.

All peers should be ready to contribute data to the P2P network whenever they are running. Data acquisition is performed through interaction with the existing host-based protection applications running on the peer. Data is shared with other neighborhood systems via the P2P network. On-line peers which are currently inactive, or active less than some threshold, perform analysis of the neighborhood network data that is received. The P2P network forensic system is designed to be able to deal with the transient nature of peers.

Utilization of peers in the collection and analysis of network data is a resourceful and economical use of existing systems. Since a community's infrastructure is comprised of these systems which are already in place, it is a logical step to use them to perform network data analysis. These systems already have access to the network traffic that is of the most interest which is the traffic to the systems themselves. As Denning proposed, the systems also are best able to determine what is legitimate traffic and what is illegitimate or anomalous [4]. Use of the systems themselves as network sensors is worthwhile as long as that task doesn't interfere with the official mission of the system. In addition to the ability of peer systems to capture relevant traffic, the amount of idle CPU cycles available on the peer systems during normal processing and especially during times when no active processing is going on which can be used to analyze network information can be significant. As the success that P2P applications such as SETI@HOME [1, 11], the Folding Project [5], and the World Community Grid [16] has shown, using idle

cycles to perform background processing can be an efficient utilization of computing resources.

3. Network Forensics

This work engages the network forensics definition provided by the Digital Forensic Research Workshop. Network forensics is "the use of scientifically proven techniques to collect, fuse, identify, examine, correlate, analyze, and document digital evidence from multiple, actively processing and transmitting digital sources for the purpose of uncovering facts related to the planned intent, or measured success of unauthorized activities meant to disrupt, corrupt, and or compromise system components as well as providing information to assist in response to or recover from these activities" [8].

In order to perform network forensics using P2P technology, it must be shown that the P2P techniques used are scientifically sound and that they can uncover the facts related to unauthorized access of computing resources. The network forensics process can be broken down into five distinct phases. The first phase is event acquisition. An event is any occurrence for which a host-based protection mechanism issues an alert or notice. The P2P application on a peer is responsible for collecting these alerts from the set of host-based protection systems. The event is normalized into a standard XML document format in order to ease the parsing process used by other peers. The next phase is event transmission. This is really where P2P comes into play. The P2P network is designed so that the normalized XML version of the event is transmitted to all other neighbor peers. This is done through the P2P network without any configured or pre-existing information about the identity and location of neighbor peers. The third phase is data storage. Information regarding specific events must be archived in a manner allowing investigations to proceed without the concern that the data describing the event has been corrupted. By creating databases on each of the peers containing event information for all of their neighbors, corroboration of the data can be accomplished. The fourth phase is data analysis. As each peer receives transmission of an event from one of its neighbor peers, it tries to correlate that event with other events that have been archived in its database. After analysis, the final phase, reporting, is invoked. A peer is designed to report its findings back to all neighboring peers in order to create a complete neighborhood view for all peers.

Network security personnel can configure monitor peers that are members of all neighborhoods in order to establish a profile of the complete network status. A peer can also develop a peer status showing each of

the neighborhoods in which it belongs. The techniques used in each of these phases will be described in order to show that they are scientifically sound.

4. Data Acquisition

Data acquisition is the process where host-based protection security mechanisms collect and identify events that can contribute to the network forensic process. This acquisition can be separated into two levels. The first level involves the protection mechanisms that are normally running on the peer. The events from the first level are considered important enough to be transmitted to all other peers belonging to the same neighborhood. The second level is enacted after an event or a series of events has been deemed important enough to warrant an increased amount of effort in data acquisition and an increased volume of resultant data. The first level mechanisms include host-based firewalls, anti-virus systems, and intrusion detection systems. Intrusion detections systems can be further broken down into programs which perform file system monitoring, log file analysis, and network connection monitoring. The second level mechanisms are packet capturing programs.

As an example of a first level data acquisition process, a firewall system will be described. Consider the iptables [9] firewalling product for linux. The iptables program can be configured to log using the syslog daemon any network packet that meets some criteria. These criteria are based on the security policy for the host. Packets can be logged independently of the action of the firewall to accept, reject or drop them. Like many of the host-based protection mechanisms, iptables, has its own log format. The iptables log includes information that could be very useful in the analysis process: IP header fields, TCP header fields, UDP header fields, IP options, TCP options, and TCP sequence number. Each of the log fields is formatted with the field name, an equals character, and the field value except in the case where the field is a boolean flag, and in that case, if the log entry includes the field name alone, it indicates that the flag is set. A sample log entry is:

```
Nov 22 09:55:17 myhost kernel:iptables REJECT IN= OUT=eth0
SRC=192.168.118.221 DST=192.168.1.19 LEN=60 TOS=0x00 PREC=0x00
TTL=64 ID=12320 DF PROTO=TCP SPT=36718 DPT=767 SEQ=890512859
ACK=0 WINDOW=5840 RES=0x00 SYN URGP=0
```

As long as the format of the host-based protection mechanism log entry is known, the entry can be reformatted into an XML document to ease the interpretation by other peers. Converting the information into a

XML document allows each of the peers in the neighborhood to be able to use the XML parsing mechanism built into its P2P application to process the event without having to have a priori knowledge about the specific log format of the host peer's protection program.

The second level of data acquisition occurs after a host-based protection event is determined to be important. This second level is an attempt to gather more information about the event. The analytical process 12.7is used to determine when an event meets the criteria of being important. This second level acquisition process is an attempt at collecting more complete network information. Packet capturing systems such as snort [10, 13], tcpdump [14, 15], or snoop [12] are then utilized to collect and archive relevant data. As opposed to the host-based protection mechanisms, the second level acquisition processes are designed to gather all relevant information. While this involves substantially more data collection, it is still less than the amount of data that would be collected if a capture all method were employed.

5. Communication Architecture

P2P refers to the concept that in a network of systems, any system can communicate or share resources with another without necessarily needing central coordination. Each of the systems (peers) in the network are essentially treated as equals. The common non-P2P method which is prevalent in today's network environment is client-server where the server is a centralized system providing information and coordination to a number of subordinate client systems. P2P networks on the other hand are designed such that all systems are producers as well as consumers. A pure P2P network does not rely on a centralized management system in order to control communication. It is designed to allow dynamic discovery of peer systems and permit direct communication between any of the peers without the need for intervening control. In opposition to client/server networks, P2P networks increase utility with the addition of more nodes. In order to create the distributed network forensics environment, a peer-to-peer network is established. This P2P network allows peer/node/end devices to communicate with each other to share interesting network information.

A P2P system for performing network forensics has the following requirements:

- Must permit peers to be both producers and consumers of data.
- Must allow the dynamic addition and subtraction of peers to the system.
- Must communicate with a variety of platforms.
- Must minimize network traffic overhead.
- Must facilitate the exchange of data in a common format.

- Must ensure confidentiality.
- Must ensure integrity.
- Must ensure availability.

The P2P framework that was found to best satisfy the requirements is JXTA (pronounced juxta) [6].

5.1 JXTA

5.1.1 Overview. The P2P framework is based on the JXTA open source platform from Sun Microsystems. JXTA technology is a Java based network programming and computing platform that is designed to solve a number of problems in modern distributed computing. Project JXTA's objectives are to enable interoperability between platform independent systems. The goal is JXTA ubiquity, implementable on every device with a digital heartbeat, including sensors, consumer electronics, PDAs, appliances, network routers, desktop computers, data-center servers, and storage systems.

5.1.2 Architecture. The JXTA software architecture is referred to by the Protocol Specification (v2.0) as a "three layered cake." The three layers are:

- Platform. This layer encapsulates minimal and essential primitives that are common to P2P networking, including peers, peer groups, discovery, communication, monitoring, and associated security primitives. This layer is ideally shared by all P2P devices so that interoperability becomes possible.
- Services. This layer includes network services that may not be absolutely necessary for a P2P network to operate but are common or desirable for P2P environments. Examples of network services include search and indexing, directory, storage systems, file sharing, distributed file systems, resource aggregation and renting, protocol translation, authentication and PKI services.
- Applications. This layer includes P2P instant messaging, entertainment content management and delivery, P2P e-mail systems, distributed auction systems, and many others. Obviously, the boundary between services and applications is not rigid. An application to one customer can be viewed as a service to another customer.

This system fits into the JXTA applications layer. On top of the underlying JXTA platform a P2P network is designed. Peer groups are a

central concept in the JXTA platform which provides a segmentation of the P2P network space into distinct sets. The P2P framework relies on these JXTA peer groups for implementation of the peer neighborhoods. In this P2P architecture, each peer by default is a member of the Net Peer Group. This essentially is a network where each peer knows about and can set up communication with all other peers. Peer groups are created in order to partition the network into logical neighborhoods so communication can be restricted to only those other peers who have an interest in similar data.

5.1.3 P2P Communication. The P2P framework communication consists of an end system establishing itself in the P2P network and then performing transfer of messages describing interesting network traffic to its peer neighbors. These messages could be periodic messages of average statistics, and then individual messages consisting of interesting data. The messages are of a standard XML based format that any P2P host can parse. Peers also transfer messages related to the ongoing analysis of its data. Through the communication of interesting events and individual analysis of a peer's environment/history a more thorough understanding of an event can be obtained.

5.1.4 Security. As with any computing system or environment, security of the system must be addressed. Using a completely distributed manner for performing the network forensics eliminates a big problem with a centralized server which is that it is a single point of failure. The P2P network forensics analysis technique also provides a great deal of redundancy to the process since any interesting data and any significant analytical results on a peer are likely to be duplicated on other peers in its neighborhood.

The P2P network forensics model presented here is for an administratively closed environment. That is, all systems in the community are contained within a single administrative domain. Given this architecture, a Public Key Infrastructure (PKI) approach to security, specifically trust, can be implemented. Security in an information system can be considered as three legs, confidentiality, integrity, and availability and each of those legs must be addressed.

Confidentiality is the "concealment of information or resources" [2]. Since this is an administratively closed environment model, sharing of the information within the P2P network is allowed. A larger issue is preventing outsiders, that is, those not in the community, from accessing the information. This is an community wide security issue but sill important to the design of the P2P network. If the network boundary is

not sufficient to ensure confidentiality, P2P message traffic within peer neighborhoods can be encrypted using the PKI.

Integrity refers to the "trustworthiness of data or resources, and is usually phrased in terms of preventing improper or unauthorized change" [2]. Integrity is the very important in the P2P architecture. Each peer needs to be confident that the information that it receives from other peers is accurate. This can be assured through the implementation of a PKI. The PKI is used with public key cryptography and digital signatures to verify the integrity of messages between peers. Within the administratively closed environment, a Certificate Authority (CA) is established. The CA's verification of a peer's public key is contained in a digital certificate. Each peer presents its digital certificate to other peers as a means to securely distribute its public key. To prove that a P2P message is authentic, a peer digitally signs the message with its private key and transfers the signed message. The receiving peer can authenticate the message using the public key included in the digital certificate. Integrity is then assured.

Availability refers to the "ability to use the information or resource desired" [2]. The concern here is twofold. First, it is a matter of community wide management. Peers need to be able to get interesting information from the other peers in its neighborhood. This means that other peers must be contributing in the P2P forensic network. Free-loading on P2P networks is a common concern [7], which in this environment can be dealt with administratively. Secondly, the P2P network must have controls in place so that peers are not overwhelmed with P2P messages which could create a denial of service. This must be handled in the P2P application. The application must recognize when it is sending out too many messages, and throttle back. The fact that too many messages are being generated must be taken into account in the self-analysis of the host and treated appropriately. The number of messages sent through the P2P network must be one of the items that is statistically monitored and when a significant deviation from this value is encountered an anomaly event should be triggered.

6. Archival

Two aspects of archival portion must be addressed: archival for audit purposes and archival for analytical purposes. Audit records are required to perform after the fact recreations of the unauthorized activity and are more rigorous in their requirements. These records need to be authentic (trusted). Analytical records on the other hand, are copies of the event that a peer receives from a neighbor and are used in the analytical

process of determining if malicious activity is underway. These records diminish in importance as time progresses and can be summarized or deleted with the presumption that the loss of information will not have significant impact.

Since the event information that is produced from a peer is based on the host-based protection mechanisms from that peer, that application is responsible for archiving the information for audit purposes. The P2P events can be recreated later from the archived information if needed. The P2P process needs to be responsible for archival of the received events from other peers in order to do effective analysis. A means to archive/database the event information needs to be developed. Some of the goals of this portion of the project are minimal data storage requirements, ease of searching the archive for relevant information, and ability to deploy the archive on any platform that the P2P application runs on. The archival system should automatically age the events and take appropriate steps to remove outdated information.

7. Analysis

A critical element of the network forensic process is to examine, correlate, and analyze the digital evidence that is collected and shared using the P2P process. Each peer system in the network is expected to collect the information that its neighbor peers identify as security events and perform correlation of that data with data that the peer has already archived in order to determine if the event is worthy of more detailed scrutiny. Analysis is the process by which an event is determined to be worthy of being elevated to the important category. The first step of analysis actually occurs in the data acquisition phase of the process. This is the normalization of the data. Normalization means to make the data regular and consistent.

The analysis process is triggered each time that an event arrives over the P2P network. Reception of the event causes the analytical process to compare the event with other events that have been archived on the peer. There are many different analytical approaches that can be employed to determine if an event should be elevated to the important level. Two of these approaches are statistical analysis and rule based analysis.

Statistical analysis is a process where the occurrence of an event is compared to what has been statistically determined to be normal use. Denning [4] hypothesized that exploitation of system vulnerabilities involves abnormal use of the system; thus, security violations can be detected by analyzing abnormal patterns of system usage. In particular,

Denning identifies five models by which abnormal behavior may be identified:

Operational Model: Abnormality here is defined as a comparison of an observation against a fixed value. The fixed value is not based on a statistical evaluation of some n previous events, but upon a predetermined threshold which can be based on security policy. An example is an event counter of login failures due to bad password within some time constraint.

Mean and Standard Deviation Model: Abnormality is based on comparison of an event to previous events. A mean and standard deviation of the previous events are calculated and an event is abnormal if it doesn't fall within a confidence interval based on a number of standard deviations from the mean. Chebyshev's inequality tells us that the probability of a value falling outside of that interval is at most the inverse of the number of standard deviations squared.

Multivariate Model: Similar to the mean and standard deviation model except that it is based on the correlation of multiple variables. A study by Ye, *et al.* [17] discusses this model and compares two techniques, Hotelling's T^2 test and the chi-squared distance test, X^2. They find that the more scalable X^2 multivariate analysis technique detecting mean shifts only is sufficient for intrusion detection.

Markov Process Model: Using each individual event as a state, and a state transition matrix to characterize transition frequencies between states, an event can be categorized as abnormal if the probability of transition between the previous state and the event state is low as determined by the state transition matrix.

Time Series Model: Similar to the mean and standard deviation model, but includes the timestamp of the event in its determination of abnormality. In this case an event is considered abnormal if the probability of the event occurring at that time is too low. This method has the advantage of measuring trends over time and detecting behavioral changes.

The rule based analysis approach, which is similar to the Markov Process model, is presented by Chuvakin [3]. According to Chuvakin, rule-based correlation uses some preexisting knowledge of an attack (a rule), which is essentially a scenario that an attack must follow to be detected. Such a scenario might be encoded in the form of "if *this*, then *that*, therefore *some action* is needed."

Rule based analysis deals with: states, conditions, timeouts, and actions. A state is a situation that an event from a peer describes, it is a position that a correlation rule is in. A state is made up of various conditions that describe the state like source or destination IP addresses and port numbers. The timeout describes how long the rule sequence should be in a particular state and a transition is the movement from one state to the next. Rules may describe many different state transitions from a particular state. An action is the steps taken when a rule has been matched.

Unlike many of the statistical analysis methods, rule based analysis requires a understanding of the specific operation of the system, or known attack sequences that occur. The first, specific operation, tracks state changes through normal operation of the system and deviation from the expected rules indicate abnormal activity which could be an indicator of malicious intent. The second, known attacks, tracks state changes though known malicious activities and a complete rule match will be an indicator that the malicious activity has occurred.

An analysis of an event that results in a determination that the event is significant triggers the analyzing peer to send a message to its neighbors that an important event has occurred. As stated previously, when an event is deemed to be important, neighbors are alerted in order for them to collect more detailed information regarding similar events. The message that is sent contains critical parameters based on the event type which will allow the neighbors to determine what to further scrutinize. The message will also be an important factor in alerting the system user and administrator to the fact that a possible unauthorized access of information resources was identified.

8. Reporting

Reporting is the documentation part of network forensics. The analytical results that each peer produces need to be provided to the user/administrator of the peer system. Since the analytical process is designed to run in the background using spare CPU cycles, the best way to display results would be in the form of a screensaver. While this provides information to the user of the peer, overall community results are also necessary. The designed method for this type of operation is to dedicate an analytical/reporting peer or set of peers that operate exclusively for the purpose of performing monitoring. These peers operate in the same manner as normal peers, but will be configured to join any neighborhoods that they are interested in monitoring.

9. Conclusions

A P2P framework used to share security events among information systems in order to perform network forensics has been proposed. The security events that are shared are normalized versions of alerts gathered from existing host-based protection mechanisms. A prototype P2P application using JXTA technology has been developed and shows promise in effectively transferring the information throughout a community comprised of a number of peer neighborhoods.

Future work will focus on employing scientific techniques to collect, fuse, identify, examine, correlate, analyze, and document digital evidence from multiple, actively processing and transmitting digital sources. The ultimate goal is to uncover facts related to the planned intent, or measured success of unauthorized activities meant to disrupt, corrupt, and/or compromise system components as well as providing information to assist in the response and/or recovery from these activities.

References

[1] D. Anderson, J. Cobb, E. Korpela, M. Lebofsky and D. Werthimer, SETI@home, *Communications of the ACM*, vol. 45(11), pp. 56-62, 2002.

[2] M. Bishop, *Computer Security: Art and Science*, Addison-Wesley, Reading, Massachusetts, 2003.

[3] A. Chuvakin, Security event analysis through correlation, *Information Systems Security*, vol. 2(13), pp. 13-18, 2004.

[4] D. Denning, An intrusion-detection model, *IEEE Transactions on Software Engineering*, vol. 13(2), pp. 222-231, 1987.

[5] FOLDING@home (folding.stanford.edu).

[6] JXTA (www.jxta.org).

[7] A. Oram (Ed.), *Peer-To-Peer Harnessing the Power of Disruptive Technologies*, O'Reilly, Sebastopol, California, 2001.

[8] G. Palmer, A road map for digital forensic research, *Proceedings of the Digital Forensic Research Workshop*, 2001.

[9] netfilter/iptables (www.netfilter.org).

[10] M. Roesch, SNORT - Lightweight intrusion detection for networks, *Proceedings of the Thirteenth Systems Administration Conference*, 1999.

[11] SETI@home (setiathome.ssl.berkeley.edu).

[12] snoop (docs.sun.com).

[13] snort (www.snort.org).

[14] R. Stevens, *TCP/IP Illustrated, Volume 1*, Addison-Wesley, Reading, Massachusetts, 1994.

[15] tcpdump (www.tcpdump.org).

[16] World Community Grid (www.worldcommunitygrid.org).

[17] N. Ye, S. Emran, Q. Chen and S. Vilbert, Multivariate statistical analysis of audit trails for host-based intrusion detection, *IEEE Transactions on Computers*, vol. 51(7), pp. 810-820, 2002.