Profiling Honeynet Attackers

Connie Li Taufik Parsioan

December 12, 2005

Abstract

This paper describes the value and process of using honeynets to profile attackers in the blackhat community. Data were taken from a compromised honeynet deployed by The Honeynet Project and were analyzed for significant events. The data were then used to create a profile of the type of attackers attempting to break in to systems similar to the honeynet and the exploits that may have been used. After an extensive analysis of the alerts given out by the Apache and Snort logs, we find that only inexperienced hackers attack the honeynet for the pure opportunity of it.

1 Introduction

Honeynets are networks of systems that are deployed for the purpose of luring hackers into a network that can monitor their activities. There are two different types of honeynets: production and research. The former are low interaction networks that emulate services for the purpose of protecting organizations while the latter are high interaction networks that provide real services to the attacker [Spib]. Honeynets are valuable because of their ability to collect large amounts of information about attackers and the types of attacks without doing any harm to the system or putting any valuable information at risk. Data collected on a honeynet is also easier to analyze since any activity recorded is assumed to be malicious, given that the network serves no practical purpose.

Honeynet forensics is a specific type of computer forensics in which specialized data analysis techniques are applied only to the data collected on the honeynet. The data provided by a honeynet is almost certainly easier to view than forensic data from any normal system, given that specialized software capturing all activity on a system is often previously installed on honeynets. The goal of honeynet forensics is to recreate attacks from the information collected to obtain a better understanding of what took place after the system has been compromised [RBBK04]. The final results should be able to describe the basic who, what, where, when, and why of an attack.

A subfield of honeynet forensics is profiling using honeynets. Using the results obtained from analyzing the honeynet data, profiling attempts to identify the person or group responsible for the attack and their motives. Profiling is a useful tool because past profiles might help predict what future and current cases of attackers may be like. Profiles of computer hackers also give valuable insight into the blackhat community (hackers working towards negative goals) which is often seen as a mysterious or underground subculture [KAS04].

In the next section, we will discuss in more detail the previous research that has been conducted in the area of honeynet forensics and profiling. We will then go on to discuss our attempt at profiling attackers using data taken from a compromised honeynet. First, we will explain the origin of the data that were used during the project and detail the significant events that occurred on the honeynet. Following that will be our analysis on the data that we've collected. We

Proceedings of the Class of 2006 Senior Conference, pages 19–26,
Computer Science Department, Swarthmore College

will then conclude the paper with our thoughts on how the information we gathered can be used to profile the attackers.

2 Related Work

The field of honeynet forensics, or more specifically, profiling using honeynet data, is a relatively young field of computer security. While there are numerous papers giving step by step analyses of various attacks on regular systems [Che92, Spia], many do not specifically attempt to create a profile of the attacker or identify him or her. Past research mainly attempts to describe how to perform the information gathering stages of computer forensics. There has also been a lot work in establishing stages in the process of honeynet profiling [RBBK04], creating an attacker profile, and finally, being able to classify both an attacker and the exploits used.

Two separate groups of data are left behind by a blackhat after an attack on the honeynet: network clues and system and file information [RBBK04]. The network activity information contains all traffic going to and from the honeypot while the compromised host provides the system logs and tools which were used by the intruder. System information can also be obtained by examining scripts and binaries of rootkits or other files installed or left behind by the blackhat. These two groups of information can be used to create two separate timelines of the events of an attack, which can then be merged and used to answer questions such as who was responsible for the attack [RBBK04].

Once all of the data has been collected, an attacker profile can be created from what is known. An attacker profile is made up of four things: characteristics of the event, consequences of the event, characteristics of the blackhat and characteristics of the target [KAS04]. Characteristics of the event describe why the blackhat might have carried out the attack. These characteristics, such as revenge, greed, or anger, help give a better understanding of the blackhat's motivation. Consequences of the event describe why the attacker might have chosen the particular target and timing of his or her attack. This helps to determine whether the attack was targeted or merely a random exploit of a known vulnerability. Characteristics of the blackhat discuss the person or group of people responsible for the attack with information such as the motivation, skill, experience, knowledge, nationality, and funding of the attacker. Finally, characteristics of the target give information about the system that was compromised. A set of answers to these questions will help build a profile that may assist in identifying attackers and anticipating or eliminating targets [KAS04].

There have been several attempts to try to understand and define the blackhat community. Hacker taxonomies are often constructed using one or a combination of the following factors: activities, knowledge, motivation, experience and intent. As an example, Kilger, Arkin and Stutzman [KAS04] borrow from the FBI's MICE (money, ideology, compromise, ego) classification of individuals who commit espionage, to create a classification of blackhats based purely on motives. Their six categories, money, entertainment, ego, cause, entrance to social group, and status (MEECES), also appear in other taxonomies such as Chantler's [Cha96], in which there are three groups, elite, neophytes and losers and lamers, which are defined by a hacker's activities, skill level, knowledge and motivation. Chantler went even further to conclude that of the blackhat community, 30% fell into his elite group, 60% were neophytes, and 10% were losers and lamers. Another taxonomy which builds of previous research was created by Rogers, in which hackers were divided into seven distinct, although not necessarily mutually exclusive, categories based on ability: newbies, cyber-punks, internals, coders, old guard hackers, professional criminals, and cyberterrorists [Rog00]. Taxonomies such as these and an understanding of the blackhat community are essential to profiling and identifying specific hackers or hacker groups.

3 Analysis of Log Files

The honeynet logs which we attempted to analyze were taken from a data set provided by The Honeynet Project. The Honeynet Project is a non-profit organization dedicated to improving computer security by providing information about types of attacks, attackers, and motives. They obtain data from various honeynets deployed by members of the Honeynet Research Alliance. This particular data set was published on The Honeynet Project's website for a Scan of the Month Challenge. The Honeynet Project organizes these monthly challenges so that members of the security community can have the opportunity to examine actual honeynet data and share their methods and findings. To reduce the task of extensively searching through the entire set of log files, we initially read through the results of the challenge to learn what significant events occurred on the honeynet.

In total, four different types of log data were provided: Apache logs, Snort NIDS logs, Linux syslogs, and iptables firewall logs. Each data set has a slightly different starting and ending date, but in general, the data ranges from January 20, 2005 through March 17, 2005. Since the Apache and Snort logs will suffice for the purpose of this paper, discussion of the other two logs will be omitted.

3.1 Apache Logs

The Apache logs contain a record of all user activities and errors on the honeynet. They are separated into requests which produced error messages and requests which were successfully processed by the server. Apache logs contain the IP address of the remote system, the time of request, and also the specific request of the attacker.

The Apache logs of this honeynet reveal that the honeynet was compromised using an AWStats.pl exploit on February 26, 2005. AWStats is a server logfile analyzer that graphically generates all web, mail, or ftp statistics. It can also be run as a CGI in which the program is stored and executed on the web server when requested by a client. In versions 5.7–6.2 of AWStats, the awstats.pl script contained a bug in which a command prefixed and postfixed with the character '—' can be executed on the system. So, if AWStats exists in the cgi-bin directory, running a command such as the one recorded at 21:13:25 on 26/Feb/2005:

- `‘‘GET/cgi-bin/awstats.pl?configdir=%7cecho%20%3becho%20b_exp%3buname%20%2da%3bw%3becho%20e_exp%3b%2500HTTP/1.1’’`

will cause configdir to execute the command:

- ‘echo; echo b_exp; uname -a; w; echo e_exp’

which gives attacker information about the system and also reveals user information such as who is logged on the system and what they are doing. This AWStats exploit appears to be relatively simple to execute for a programmer of even little experience. Regardless, the attacker can use it to gain valuable system information by running commands that would go possibly undetected on any regular system given that they would appear to be harmless AWStats commands.

From our data we can see that the attacker uses this exploit to download a tar file twice in the span of a minute with two different IP addresses, once in Italy and another in Germany (shownq below). Given the specificity of the download and the period of time, it seems safe for us to assume that this is the same attacker. From the IP address, we see the attacker downloaded the tar file from a Romanian website shady.go.ro.

- `213.135.2.227--[26/Feb/2005:14:13:38-0500]‘‘GET/cgi-bin/awstats.pl?configdir=%20%7c%20cd%20%2ftmp%3bwget%20www.shady.go.ro%2faw.tgz%3b%20tar%20zxf%20aw.tgz%3b%20rm%20-f%20aw.tgz%3b%20cd%20.`

`aw%3b%20.%2finetd%20%7c%20HTTP/1.1''200410''-''''Mozilla/4.0(compatible;MSIE6.0;WindowsNT5.1;SV1;FunWebProducts)''`

- `82.55.78.243--[26/Feb/2005:14:14:43-0500]''GET/cgi-bin/awstats.pl?configdir=%20%7c%20cd%20%2ftmp%3bwget%20www.shady.go.ro%2faw.tgz%3b%20tar%20zxf%20aw.tgz%3b%20rm%20-f%20aw.tgz%3b%20cd%20.aw%3b%20.%2finetd%20%7c%20HTTP/1.1''200410''-''''Mozilla/4.0(compatible;MSIE6.0;WindowsNT5.1;SV1;FunWebProducts)''`

Although we have no way of verifying this, the results of this challenge stated that with this line of code, an IRC bot was downloaded and installed. This seems to be a reasonable claim given that IRC activity was recorded on the honeynet the same day as this download. We also observed that another tar file was downloaded using the same technique and website on March 2. The respondents to this challenge also analyzed this file and found it to be a backdoor to port 60666.

Something interesting that occurred on the honeynet was that on March 12, an attacker (possibly the original) attempted to download the same two tar files from the exact same website. This time however, the request failed, indicating that the version of AWStats on the honeynet was updated to a patched version. There appears to be two reasons for this; either the original attacker was attempting to remove any traces of himself on the system, or another attacker found this machine and wanted to close any vulnerabilities to the system in order to 'own' the machine.

3.2 Snort NIDS Logs

Snort is a knowledge-based, rule-driven intrusion detection system aimed at monitoring system use and detecting any malicious network traffic or activities. Knowledge-based intrusion detection systems contain information about known attacks and system vulnerabilities and search through system logs for evidence of attacks which are similar in pattern. Intrusion detection systems can also be behavior based, in which information about normal user behavior is given, and any deviations from that behavior is flagged as an attack. The information acquired from a knowledge-based intrusion detection system is usually more accurate but also less complete than behavior-based systems [DDW].

The snort logs recorded from this honeynet give an idea of the types of attacks and probes that any ordinary computer connected to a network is likely to be repeatedly subject to. In total, 85 unique snort alerts were recorded over the period of February 25 through March 31. We will examine and describe a few common alerts.

3.2.1 RPC Alerts

- RPC portmap status request UDP [Classification: Decode of an RPC Query] [Priority: 2]
- RPC portmap listing TCP 111 [Classification: Decode of an RPC Query] [Priority: 2]
- RPC STATD UDP stat mon_name format string exploit attempt [Classification: Attempted Administrator Privilege Gain] [Priority: 1]

If successful, these scans can reveal to any potential attacker the services that are available on the victim hosts. The first alert requests port information for the status service. If this request is successful, the attacker might then attempt to access this service and gain more information about the system.

RPC Portmapper is a server which assigns port numbers to services and is commonly on port 111. The second RPC snort alert appears to be a request to gain information about the services available that were assigned by the portmapper. While it might be possible that this inquiry is not malicious, an argument could be made that not everyone should be able to access this information or people who need to know this information shouldn't need to inquire for it. Thus, it is reasonable to flag these portmap queries as signs that an attack is about to happen.

The third RPC alert is an attempt to exploit an old string format vulnerability in the rpc.statd service which is sometimes packaged with Linux distributions. The rpc.statd service passes a format string supplied by the user to the syslog() function. The vulnerability in this program was that it neglected to validate the input so that a user could construct a string that would inject machine or executable code into a process address space, which would execute with the privileges of the rpc.statd process, usually root. With these privileges, a malicious user could create or delete any file with the same ease as a root user. This vulnerability in rpc.statd was first noted in 1996 and exploits of it were seen in 2000. The bug has since been fixed and only unpatched RedHat versions 6.2 or older are affected.

3.2.2 MS-SQL Alerts

- MS-SQL Worm propagation attempt [Classification: Misc Attack] [Priority: 2]
- MS-SQL Worm propagation attempt OUTBOUND [Classification: Misc Attack] [Priority: 2]
- MS-SQL version overflow attempt [Classification: Misc activity] [Priority: 3]

The MS-SQL Worm, also known as the Slammer worm, exploits a vulnerability on a Microsoft SQL server, a database management system. It is known as the first Warhol worm, given its capability to infect the entire internet within 15 minutes [SPW02]. In January 2003, the Slammer worm was able to infect more than 90% of computers within 10 minutes and caused denial of service on several Internet hosts. Systems running vulnerable versions of the Microsoft SQL server were susceptible to heap or stack overflows. Once a UDP packet sent to port 1434 successfully infects a host, its code is executed following either a heap or stack overflow. The code randomly generated other IP addresses and targeted them searching for the same vulnerability. Systems not running Microsoft SQL server, or patched versions of this system can not be harmed by this worm propagation attempt. The OUTBOUND alert informs an administrator that there is an infected machine on the system that is sending out the corrupted UDP packets. This indicates that an MS system is on the honeynet but we don't have enough evidence to verify that. The overflow attempt alert signifies that the UDP packet is trying to execute its code and cause a heap or stack overflow. It makes sense then that the first and third MS-SQL alerts are often seen together.

3.2.3 ICMP PING Alerts

- ICMP PING CyberKit 2.2 Windows [Classification: Misc activity] [Priority: 3]

PING is a network tool that sends packets to a particular host to determine whether or not it is reachable and correctly functioning. It can also report how long it took for the packets to get to the host and back and how many packets were dropped. An attacker can send the ICMP echo request packets and listen for a response to determine whether this machine is active and can be compromised. One of the actions of the W32.Welchia.Worm, seen in August 2003, was to PING the IP address it randomly generated to see if the machine was active and able to be infected.

3.2.4 ICMP Destination Alerts

- ICMP Destination Unreachable Port Unreachable [Classification: Misc activity] [Priority: 3]

The system returns "ICMP Destination Unreachable Port Unreachable" alerts when a packet fails to reach its destination. This can happen if the packet is being sent to a port that is currently closed, or not in a listening state, but it can also happen if the gateway finds a shorter route to send the traffic through. Another possibility for receiving this message is that the gateway does not have enough buffering capacity to forward the packet. Because of the fact that this message can appear in multiple ways, a single alert of this kind does not indicate malicious activity. It must be examined with the other kinds of alerts to see if someone is trying to get access to a port that they are not allowed to.

4 Profiling

4.1 Characteristics of the Target

Knowing the characteristics of our target may be significantly helpful when investigating future attacks, since similar systems are likely to be the next targets of the blackhats who attempted to hack into this particular network. From the honeynet logs provided, we can guess that there were three machines on the system: combo, bridge and bastion. Both the names of the machines on the honeynet and the IP addresses (11.11.*.*) were santized by the Honeynet Project. Given that the system deployed is a honeynet, we also believe that it is safe to assume that there was no valuable information (actual or spurious) stored on any of the systems to excessively attract any attackers. It also seems to be a reasonable assumption that there was a relatively low level of security on the honeynet, nothing that would openly try to prevent anyone from attacking the system or try to stop someone once they had compromised the honeynet.

From the snort alerts, it seems reasonable to conclude that many of the attacks or scans attempted were not specific to certain characteristics and services of the honeynet. Commands that were run appear to be relatively simple and easily repeatable across many systems. Thus, the system was probably unaffected by a majority of these scans and worms simply because they were not applicable to the files and services available on the honeynet.

4.2 Characteristics of the Events

The characteristics of an attack might give us insight into the motives of an attacker. They will tell us what caused the attack to take place. As previously stated, a blackhat may try to attempt an attack to gain revenge, status, information, money, or might try a hack simply for the challenge. None of the attempted hacks on the honeynet seemed to be for economic or political reasons, especially given that fact that as a honeynet, the network likely contained little to no information of value to blackhats with these motives.

4.3 Consequences of the Events

Understanding what the consequences of each event are allows us to understand why a blackhat might have chosen this particular time and target to attack. The results of an attack are often beneficial to the attacker and his cause but can also be harmful if he or she is not skillful enough. The Apache logs indicated that many requests involved gaining root access to the honeynet and/or executing commands to learn more about the system. Given that we are fairly certain that no information of value was on the system, it seems reasonable to state that the main consequence of many of the events is to gain control of bandwidth or more systems to carry out further attacks or propagate harmful worms.

A more specific consequence of the AWStats exploit was the ability to install an IRC bot on the compromised machine. IRC bots typically need to be run on systems with long uptimes and a fast and stable connection to the internet. Thus, there are several advantages if a blackhat manages to find a system that is not his own to run the IRC bot.

4.4 Characteristics of the Blackhats

Given that there were often several events taking place on the honeynet simultaneously, it is difficult to pinpoint one attacker in particular and conclude which events he or she is responsible for. Thus, we will discuss generally the types of blackhats who attempted to break into the system and what their motives might be.

From what we saw in the data, even when we could guess that multiple actions were likely performed by the same blackhat, different IP addresses were logged, indicating that the attacker likely had multiple systems under his or her control. So we conclude that, although helpful, IP addresses are not likely to be conclusive regarding the nationality or location of our attackers. While we also choose not to rely on the times of attacks because of the numerous attackers, we can consider the duration of attacks to determine the amount of resources necessary to carry them out. Resources can be viewed in terms of time and money. None of the attacks attempted required any type of funding other than needing an actual machine to connect to the target. Although it did appear that some attackers used multiple machines to carry out their attacks, they were not ultimately necessary for success. They merely aided in the anonymity of the attacker. The attacks made on the honeynet also did not seem to require much time and dedication. We did not have evidence of any attackers spending an extended, continuous amount of time attacking the honeynet or an attacker consistently returning to the honeynet.

5 Conclusions and Future Work

The profile we created from our data shows that most attacks on the honeynet were done by neophytes, hackers with a basic level of knowledge and experience, but still learning. We came to this conclusion after finding that most attacks on the honeynet were unoriginal, older exploits for which most systems are no longer vulnerable to. There was also no evidence to support that the honeynet was specifically targeted as acts of vengeance or greed. Most events on the system were simply acts of network or application reconnaissance to find services or vulnerabilities.

It appears that the honeynet provided only basic services and had a limited amount of information, if any. Thus, we conclude that it is unlikely for systems similar to this one to attract hackers above the neophyte level. Although this information is valuable, it is also important to obtain information about attackers of all levels, including the elite level. It is clear though, that elite attackers are unlikely to attack basic honeynets that have no additional means of attracting blackhats. Additional work complementing this project might include deploying honeynets which would attract elite attackers in order to obtain a more complete database of knowledge of the hacker community.

References

[Cha96] N. Chantler. *Profile of a Computer Hacker.* Infowar, 1996.

[Che92] B. Cheswick. An evening with Berferd, in which a hacker is lured, endured, and studied. Proceedings of the Usenix Winter '92 Conference, 1992.

[DDW] Herve Debar, Marc Dacier, and Andreas Wespi. *Towards a taxonomy of intrusion-detection systems.*

[KAS04] Max Kilger, Ofir Arkin, and Jeff Stutzman. *Know Your Enemy.* 2nd edition, 2004.

[RBBK04] Frederic Raynal, Yann Berthier, Philippe Biondi, and Danielle Kaminsky. Honeypot forensics. Proceedings of the 2004 IEEE Information Assurance Workshop, 2004.

[Rog00] Marc Rogers. A New Hacker Taxonomy. 2000.

[Spia] Lance Spitzner. Know Your Enemy: A Forensic Analysis.

[Spib] Lance Spitzner. The Honeynet Project: Trapping the Hackers.

[SPW02] Stuart Staniford, Vern Paxson, and Nicholas Weaver. How to 0wn the Internet in Your Spare Time. Proceedings of the 11th Usenix Security Symposim, August 2002.