Second-generation (GenII) honeypots

Bojan Zdrnja
CompSci 725, University of Auckland, Oct 2004.
b.zdrnja@auckland.ac.nz

Abstract

Honeypots are security resources which trap malicious activities, so they can be analyzed and monitored. During the last couple of years they have become a very important part of the security assets of an organization. Evolution of honeypots led to GenII honeypots which, compared to plain GenI honeypots, allow improved and flexible data control, and capture. Data control prevents attackers from using a compromised honeypot system to attack other external computer systems. Capturing data allows the honeypot administrator to examine in detail all information regarding activities on the honeypot system. This paper gives an introduction to the architecture and usage of GenII honeypots, their features and possibilities for future development.

1. Introduction

By increasing the network connectivity around the world, the Internet has increased the risk of potentially malicious activities being conducted against various organizations and their assets. According to the statistics by the Computer Emergency Response Team (CERT) [7], the number of reported security incidents per year is rising and malicious users are increasingly using automated attack tools.

Year	2000	2001	2002	2003
Incidents	21,756	52,658	82,094	137,529

Table 1: *Reported incidents per year [7]*

In order to detect and stop malicious activities, and protect their assets, organizations implement various security tools and methods. Two of the most common security tools that are used today to protect organizions' network are firewalls and Intrusion Detection Systems (IDS).

Firewalls are most often implemented at the network perimeters where they control network traffic. This control is employed according to a set of rules which define allowed and denied network traffic.

IDS monitor network traffic and alert the administrator when a known malicious activity is detected. In order to detect a malicious activity, an IDS will use two methods: signature detection and anomaly based detection.

These security tools have some inherent shortcomings [1]. A firewall cannot stop malicious users exploiting a new vulnerability in a service to which access is allowed by the firewall rules. IDS cannot reliably detect a previously unknown attack, especially if only signature detection is used. If anomaly based detection is used, it is based "on the assumption that intrusive activities are necessarily different from non-intrusive activities at some level of observation." [6] None of these methods of detection can guarantee that the IDS will report all attacks, so false negative detections will exist. In cases where an attacker has adopted encryption [5], network IDS cannot detect any activities.

Honeypots present an additional security tool which should be implemented in parallel with firewalls and IDS in order to raise the overall security level. Honeypots can be used to detect attacks or to capture and analyze malicious users' behavior, activities and tools.

2. Honeypots basics

Lance Spitzner, a founder of the Honeynet Project, defined honeypots as "a security resource whose value lies in being probed, attacked or compromised." [3] Honeypots are usually implemented as a separate network, which is strictly controlled and monitored. Although honeypots can be implemented on separate machines, which are a part of a organizations' network, it is advisable to physically separate their network in order to fulfill the requirements described in the following chapters.

All activities in this environment, including the network traffic coming into the honeypot and leaving it, are recorded.

The most attractive feature of a honeypot is the detection of malicious activities. As honeypots have absolutely no production value, there must be no activities on them. In this case a honeypot does not depend on any mechanism to differentiate between malicious and legitimate activities because, by definition, all traffic into it is malicious.

2.1 Data control and capture

Data control and capture are two critical requirements for a honeypot [2]. Once a honeypot is compromised, a malicious user can try to attack other systems from the honeypot. These attacks can range from the scanning of remote systems, exploitation of vulnerabilities to running Denial of Service attacks. Data control ensures that a malicious users' activity will be limited and no attacks can be conducted on a remote system, therefore the risk of operating the honeypot is reduced.

Data capture is very important in order to study a malicious users' activity and the attacks committed on the honeypot. Captured data has to be stored securely to ensure a malicious user will not be able to modify or delete it once the honeypot has been compromised. Attackers often use various methods to hide their activities and try to encrypt or obfuscate their data. [2] Therefore, capturing network traffic is not enough. Advanced honeypots will have to capture information at different layers in order to provide the honeypot administrator with the full picture of all the actions performed by the malicious user.

2.2 Production and research honeypots

Honeypots can be classified according to their usage [3]. Production honeypots are usually deployed within organizations with the main purpose of decreasing the overall risk. As the main role of production honeypots is in detecting malicious activities and alerting the security administrator, they are simpler to setup as in this case the interaction with the attacker can be low level. Services that these honeypots offer are usually simulated as they should only lure the attackers into thinking that they are trying to compromise a real, production machine. In this setup, the honeypot administrator has only limited possibilities to analyze attackers' behavior and activities, which will be restricted due to the fact that the service is simulated; however, as the main purpose is just to detect potential threats, this will be sufficient.

Research honeypots, on the other hand, are focused on gathering as much information as possible about malicious users' activities, behavior, methods and tools. Setup of research honeypots can be complex, depending on the level of interaction they offer to malicious users. In order to study malicious users' activities, services that the research honeypot offers cannot be simulated. The honeypot must be deployed on a real operating system with real, and therefore potentially vulnerable, services. Once the honeypot is

compromised, malicious users can use it to attack other systems, so the risk in deploying a research honeypot increases. Requirements for research honeypots add to the complexity as well. It is more difficult to properly implement data control, as the malicious users have practically unlimited options in running various attacks from the compromised honeypot. In addition, the data capture requirement is also more difficult to implement because not only must it collect as much information as possible, but it also has to be invisible to the attacker.

Research honeypots are frequently called honeynets. [4] Honeynets are separate networks of multiple honeypots which are used only to capture and analyze malicious users' activities. Honeynets usually consist of replicas of production systems, in order to lure the attacker.

2.3 Evolution of honeypots

Development of honeypots began in 1999. [2] The first honeypots to be deployed are now referred to as GenI (first-generation) honeypots. These honeypots served as a proof of concept and were very simple to deploy. They had only basic mechanisms for fulfilling data control and capture requirements. The architecture of GenI honeypots is shown in ***Figure 1***.

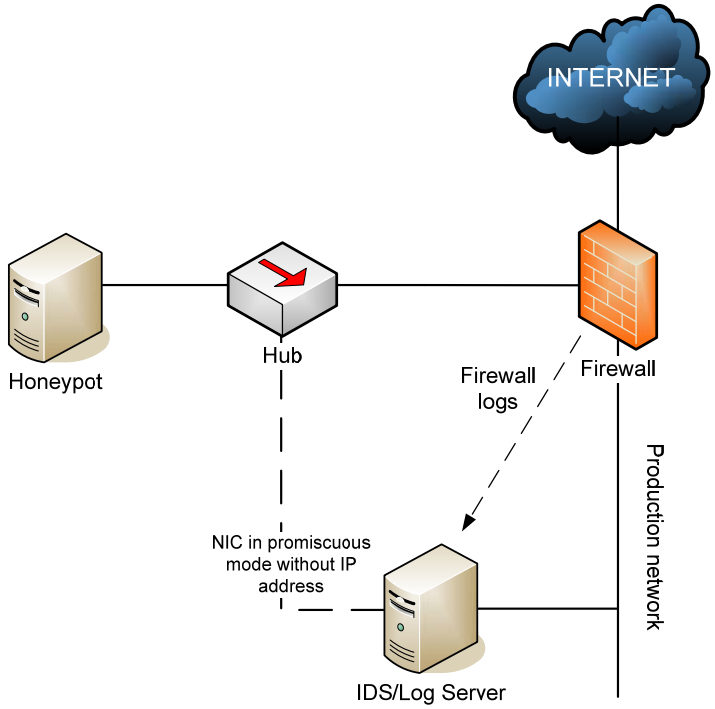

Figure 1: *GenI honeypot architecture (after [3])*

The data control requirement in GenI honeypots is provided by a reverse firewall. This firewall is simple to setup as it has to allow almost all inbound communication to the honeypot, while at the same time it has to deny outbound communication, in the case of a compromised honeypot. In order to decrease the risk of attacking remote systems, if the malicious user succeeds in compromising a honeypot, outbound rules on the firewall must be very strict. Besides setting up strict firewall rules, it is very common to limit "[the] number of connections per minute" [3] on outgoing connections, to prevent potential Denial of Service attacks being launched.

The data capture component in GenI honeypots is done by an IDS which has two main tasks. The first task is to capture all network traffic traversing through this firewall, so that later analysis can be conducted. The second task is the standard IDS operation, which is to parse network traffic in order to detect malicious activities and alert the honeypot administrator accordingly. The malicious user should not be able to detect the data capture component of the honeypot, so the IDS is usually implemented on a system with dual network interfaces [1]. One network interface is defined without an IP address, in promiscuous mode, so that it can be used for sniffing network traffic to and from the

honeypot. As there is no IP address, even a malicious user who compromises the honeypot cannot detect the IDS. The other network interface is connected to a physically separate network, usually a production network, and is used to administer the IDS or collect captured data.

GenI honeypots should be low interaction in order to decrease the risk as much as is possible. Due to the lack of advanced logging capabilities, a malicious user can use encryption or another type of obfuscation to hide his activities from the IDS, which operates only on the network layer.

3. GenII honeypots

GenII honeypots development started in 2002. [2] After the proof of concept with GenI honeypots was successful, the Honeynet Project started work on the second generation, which improves a lot of honeypot features. GenII honeypots aim to provide a high level of interaction with a malicious user. This level of interaction increases the overall risk, so advanced methods of data control and capturing must be available. ***Figure 2*** shows GenII honeypots architecture.

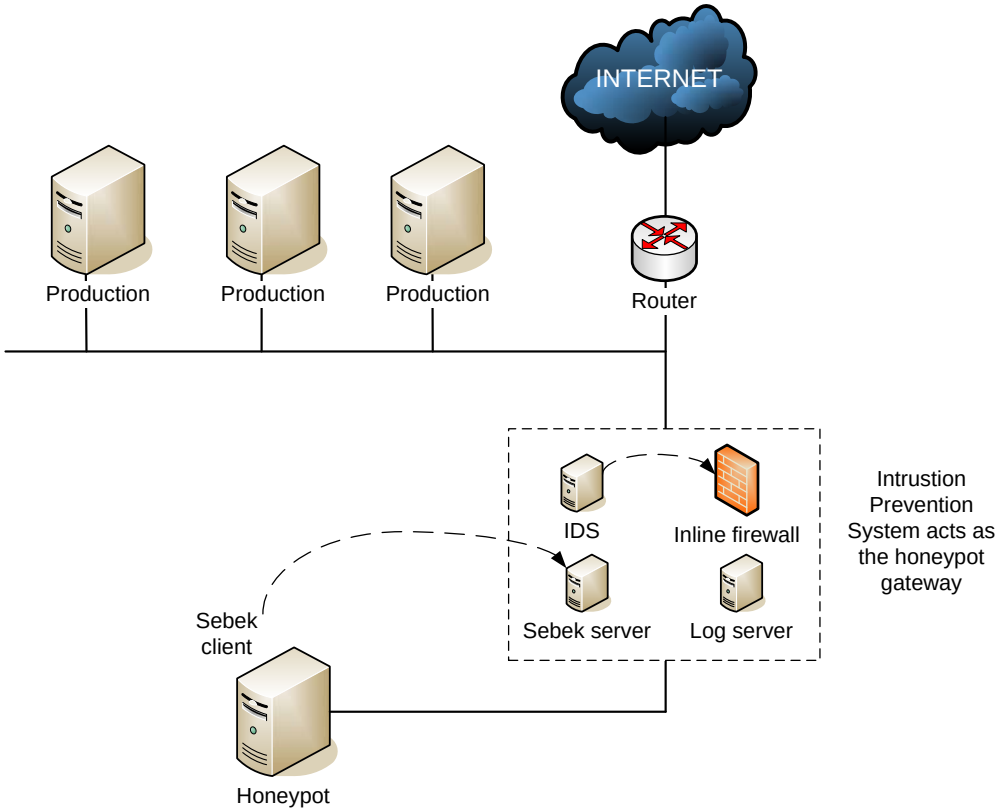

Figure 2: *GenII honeypot architecture (after [2])*

The main difference between GenI and GenII honeypots is the gateway, which is "the key element of any Honeynet." [5] As all network traffic to or from the honeypot must pass through the gateway, it is the perfect place for the implementation of data control and capture mechanisms.

3.1 Data control

Data control is a critical requirement for GenII honeypots. Once the honeypot is compromised, a malicious user may try to attack remote systems from the honeypot. While GenI honeypots offered simulated services, GenII honeypots run on real operating systems with real applications. Therefore, once a GenII honeypot is compromised, it is safe to assume that an attacker has full control over it and that the network traffic going outbound from the honeypot is malicious.

In order to limit that traffic, the gateway consists of an Intrusion Prevention System (IPS). This system basically consists of an inline firewall and an IDS.

The inline firewall operates at network layer two, as a bridge device. While this firewall can be implemented as a network layer three device (the same as in GenI honeypots), the implementation of a bridge device makes detection by the attacker much harder because an inline firewall does not change network packets when they are being processed. Inline firewalls will not decrease the time-to-live (TTL) values of a packet and do not offer means for attackers to detect them, such as MAC addresses. [2]

As in GenI honeypots, the firewall is configured to limit the rate and number of outgoing connections from the honeypot. This is done in order to prevent an attacker from running a Denial of Service attack against a remote system. The firewall is configured to block any connection if their rate exceeds a certain number of connection requests per second, so DoS attacks are effectively prevented.

The IDS implemented in an IPS is a typical IDS, which means that all the shortcomings of an IDS apply here as well. This IDS is, however, configured so that it can change firewall rules when malicious activity is detected.

Once an attack is detected, the IPS can dynamically modify firewall rules so detected packets, and any future packets of same type, will be blocked by the firewall, or changed in order to render them benign.

This feature is of interest as it allows further monitoring of malicious users' activities, while the immediate threat to the remote system(s) is eliminated. The Honeynet Project proposed deployment of the Snort_inline[9] security tool. Snort is an open source IDS [8], and Snort_inline, which is a modified version of Snort, allows dynamic changes of detected attacks by modification of firewall rules, as shown in ***Figure 3***.

```
alert tcp $HONEYNET any -> any 53
msg:"DNS EXPLOIT named"; flags: A+;
content: "|CD80 E8D7 FFFFFF|/bin/sh";
replace: "|0000 E8D7 FFFFFF|/ben/sh";
```

Figure 3: *Snort_inline signature which changes detected attack [2]*

The possibility of replacing the contents of packets which were detected as malicious increases the level of interaction with the attacker. From the attackers' point of view, the malicious packets which were part of his attack on the remote system were successfully sent and even received by the remote system, but as the IPS changed their content, they

were benign. At this point the attacker cannot easily determine why the attack didn't work, unless he has a means of inspecting network traffic at the destination system.

3.2 Data capture

In order to study malicious users' activities and capture their tools, GenII honeypots offer several methods for data capturing. These methods operate at different layers in order to capture as much information as possible.

The first layer of data capturing is at the gateway, which is configured to capture all network traffic coming into or going out of the honeypot; same as is the case in GenI honeypots. However, as GenII honeypots have an IPS at the gateway, this has additional benefits besides the possibility for data analysis. Once a new attack is detected, the detection signature for the IPS can easily be added so that in the future the same attack will be blocked at the gateway level.

The second layer of data capturing are the firewall logs. These logs can provide the honeypot administrator with valuable information about blocked malicious activities. Once the honeypot is compromised, an attacker can, among other things, try to run a Denial of Service attack on a remote system. These logs will show what kind of communication the attacker attempted to establish as well as what the targets were. This layer is present in GenI honeypots as well.

The third layer, which was introduced in GenII honeypots, captures an attackers' keystrokes on the compromised honeypot. Usage of encryption to protect network communication from unauthorized sniffing is very common today in many legitimate services. Secure Shell (SSH) is the most common remote terminal service today and it has almost completely replaced the old and insecure telnet, which sent data in plain text. As attackers today use SSH as well, it is impossible to gather any information about their session by looking at the network traffic alone.

The Honeynet Project developed Sebek [10], which is a set of kernel modules for various operating systems. Sebek works in client-server mode, where the server is installed on the gateway and the client is installed on the honeypot. Sebek is used to capture keystrokes from all remote terminal sessions. As this information has to be logged securely, the Sebek client will send it to the gateway, running the Sebek server. In order to hide this activity from the attacker, captured logs are sent as UDP packets to the gateway with an

encrypted payload. To prevent the attacker from seeing this traffic, the Sebek client will disable the honeypot from sniffing "any packets with a predesignated magic number and UDP port." [5] This effectively hides logging traffic from the attacker, even in the case when he gains full control over the compromised honeypot.

Developed kernel modules can capture files copied by the scp program, which is a remote copy program distributed with SSH. Scp enables user to securely copy files to the remote system, as all network traffic will be encrypted. In order to attack further machines, malicious users often upload exploits and various tools to the compromised honeypot [2]. By collecting uploaded files, the honeypot administrator can analyze them later and, if needed, reverse engineer them, to determine their purpose. This method allows the capture of yet unknown exploits, often referred to as 0-day exploits, which cannot be detected by IDS which rely on signature detection.

3.3 Future development

It is obvious that GenII honeypots can be improved and optimized with respect to data capture and control mechanisms. One of the goals of the Honeynet Project is also to support as many platforms and operating systems as possible.

GenII honeypots are the foundation for future development. The Honeynet Project identified several phases [2] for future work in this area. The first phase was to create a bootable CD-ROM to ease deployment of honeypots in organizations.

The main area of development is covered in the second phase, which is related to the data collection system that will offer centralized collection across multiple distributed honeypots. This will allow correlation of data gathered by multiple honeypots which offers better possibilities for trend analysis and various early warning systems.

4. Conclusion

GenII honeypots offer improvements over GenI honeypots in the two critical requirements: data control and capture.

Controlling network data by an IPS offers various benefits, besides the typical allowing or denying of network traffic at the firewall. The honeypot administrator can change the content of packets which were detected as malicious in order to render them benign. This increases interaction level of GenII honeypots, as malicious users will have a false sense

of working on a fully compromised network yet their further attacks will not succeed. By having the ability to introduce modified or new signatures at the gateway, which will be used by the IPS, the honeypot administrator has better control of which traffic to deny and what to modify.

Capturing the data on multiple layers ensures that enough information about malicious activities is gathered, so subsequent analysis can be completed. A major improvement that GenII honeypots introduced is the ability to capture keystrokes of remote sessions on the honeypot. This way honeypot administrator can monitor malicious users' behavior.

As GenII honeypots are highly interactive in comparison to GenI honeypots, the risk of their deployment increases as well. Once a malicious user compromises the honeypot, they have full control over it and data control relies on proper setup of the gateway which should deny further attacks by the intruder. There is a lower risk with GenI honeypots, because services they offer are simulated and therefore it is very complicated, if not impossible, for a malicious user to take full control over GenI honeypot.

The decision of whether to deploy GenI or GenII honeypots depends on their purpose. In an environment in which a production honeypot is needed, and the main goal is to detect malicious activities and their origins, GenI honeypots will satisfy all requirements due to their easier deployment and decreased risk. GenI honeypots have proved to be excellent in the detection of fast spreading worms. [1] In cases like this it is more important to detect the source of the infection than to analyze malicious activities.

On the other hand, when research honeypots are being deployed, and the main goal is to analyze malicious users' activities, behavior and tools, GenII honeypots offer superior data capture methods and are the only reasonable choice. When implementing this type of honeypots, data control must also not be ignored, as the malicious user has more freedom in their actions. With an IPS in place, GenII honeypots are again superior when compared to GenI honeypots.

5. References

[1] J. Levine, R. LaBella, H. Owen, D. Contis, B. Culver, "The Use of Honeynets to Detect Exploited Systems Across Large Enterprise Networks," IEEE Systems, Man and Cybernetics Society, 18-20 June 2003, pp. 92-99.

[2] L. Spitzner, "The Honeynet Project: Trapping the Hackers," IEEE Security & Privacy Magazine, Volume 1, Issue 2, Mar-Apr 2003, pp. 15-23.

[3] A. Chuvakin: "Honeypot essentials," Information Systems Security, Volume 11, Number 6, Jan-Feb 2003, pp. 15-20.

[4] F. Zhang, S. Zhou, Z. Qin, J. Liu, "Honeypot: a Supplemented Active Defense System for Network Security," Proceedings of the Fourth International Conference on Parallel and Distributed Computing, Applications and Technologies, 2003, pp. 231-235.

[5] The Honeynet Project, "Know Your Enemy: GenII Honeynets," http://www.honeynet.org/papers/gen2/index.html, 2003.

[6] J. McHugh, "Intrusion and intrusion detection," International Journal of Information Security 1, 2001, pp. 14-35.

[7] Computer Emergency Response Center (CERT), "CERT/CC Statistics 1988-2004," http://www.cert.org/stats/cert_stats.html, 2004.

[8] Snort, The Open Source Network Intrusion Detection System, http://www.snort.org/

[9] Snort_inline, http://snort-inline.sourceforge.net/

[10] Sebek, Data capture tool, http://www.honeynet.org/tools/sebek/