The Impact of Honeynets for CSIRTs

Jan Kohlrausch and Jochen Schönfelder
DFN-CERT Services GmbH
Heidenkampsweg 41
D-20097 Hamburg
[kohlrausch|schoenfelder]@dfn-cert.de

Abstract

For the daily work of a CSIRT it is of major importance to know which vulnerabilities are currently abused to compromise computers and to timely warn the constituency if a zero-day exploit is found. Besides the traditional incident response work, honeypots have shown to become more important to follow these aims.

In this paper we give an overview on the NoAH project and related projects devoted to the deployment of distributed honeypots and show how CSIRTs and other security teams can benefit from the deployment of their infrastructure.

1 Introduction and Motivation

All CSIRTs are dependant on different sources of information for their daily work. Traditionally, these sources are manual reports from the constituency, mailing-lists, and other CSIRTs e.g. in the FIRST community. Before the year 2001, most compromised systems in the German research network were UNIX or Linux servers which had been manually compromised. Often, these systems were abused for port-scans trying to identify other vulnerable systems. Considering the number of these incidents, the CSIRTs were able to handle them without need for automation including sending manual e-mail reports to the affected sites.

However, with the rise of the well-known internet worms and interconnected networks of compromised systems the situation has changed. Related to this, the statistical data give evidence of a dramatically grown number of reconnaissance activity (e.g. port-scans) looking for vulnerable systems.

To cope with this new situation, CSIRTs have to react and adopt their workflow to the new situation. In this article we show, how sensor networks and honeypots can help in this progress.

The remaining of this article is organized as follows. In the following section we give a brief repetition of the basic services of CSIRTs. Here, we also describe the current situation concerning these services we have observed over the last years. Chapter 3 summarizes some relevant projects deploying network of low-interaction as well as high-interaction honeypots. In the following chapters we point out how the previously discussed services of CSIRTs can benefit from these projects and summarize the main results.

2 CSIRT Services and Current Situation

Almost all computer security incident response teams (CSIRT) provides services concerning handling of security incident and disseminating security advisories to their constituency. The first service applies to the reaction to security incidents reported by the constituency, other CSIRTs, and other sites. A security incident is here understood as an incident affecting the confidentiality, integrity, authenticity, and availability of data or systems at a site. For example, this kind of incident includes compromised machines, theft of confidential research data, modification of sensitive administrative data, and denial of service attacks against web or IRC servers. This service is comprised of the following tasks:

Incident Handling This task includes an analysis of the incident in order to find answers to the following questions:

- What has happened?
- What is the impact of this incident?
- How was the attacker able to succeed in his intended actions (e.g. how did he compromise a machine)?
- What did the attacker do after the initial break-in?
- What other machines and sites are involved in this incident?
- How can the attacker be tracked down?
- How to recover from the incident?

Incident Coordination A large number of security incidents affect multiple sites. For example the analysis of a compromise machine can reveal other compromised machines at different sites. The notification of all involved sites and the coordination of the activities concerning the incident is done by this task.

As a tendency, the CSIRTs observe a massive increase in incidents affecting a large number of interconnected machines. The strategy is to massively and non-selectively attack machines in order to build large networks of compromised machines controlled by a central mechanism. In the later, we refer to these networks as *botnets*. One of the reasons might be that in the past years, ISPs started to offer cheap high-speed internet access to home-users. As a consequence home-user systems have become very attractive for attackers because an increasing number of home-users use the internet access for home banking applications and commercial use (e.g. ebay.com). In addition, a large number of home-users is not experienced in computer security and these systems can be assumed to be less secure than workstations in universities or company networks. Incident statistics of the last years confirm that an increasing number of botnets is directly abused for criminal intent like *phishing* for bank account data, other fraudulent intent or abused for relaying spam emails. Moreover, botnets can be effectively abused to attack other systems by causing denial of service conditions.

Commonly, the IRC protocol is used to control all compromised machines. In detail, a program called *IRCbot* is installed on each compromised machine connecting to an IRC server under the control of the attacker. The attacker can control all machines connected to a specific IRC channel by issuing commands on this channel. These commands are interpreted and executed by the IRCbot. Commands include scanning for other vulnerable machines, denial of service attacks, and updates of the bot software. Common ways to compromise machines include self-spreading internet-worms, zero-day exploits for internet browsers and trojan software attached as attractive content to e-mails. All these ways have in common, that the attack does not require manual interaction.

Moreover, botnets give the attacker the advantage of avoiding single points of failure. Thus, if a compromised machine was taken offline, the attacker can easily switch to a different machine.

In contrast to the aforementioned non-selective attacks selective attacks have a specific aim. This can be stealing research related documents in a university environment or industrial espionage. Commonly, this type of attack is more sophisticated and it can be expected that the attacker has the knowledge of an insider. In most cases the impact of selective attacks is more severe for the victim compared with non-selective attacks. Since the compromised systems are not abused for subsequent criminal activities or port-scans and because the attacker is commonly an insider this kind of attack is usually very difficult to detect. As an additional drawback, detection is complicated by the massive background noise of automated attacks.

A large number of CSIRTs including the DFN-CERT is providing an advisory or alerting service. Main aim is to provide the constituency with information about novel security vulnerabilities and related software updates. As an advantage the constituency receives all information from a single source and therefore they do not need to follow the different sources issuing security advisories. A slightly different aim is followed by alerting services. Major aim is to warn the constituency if new vulnerabilities have been discovered or zero-day exploits for unknown vulnerabilities have been published. While the advisory service focuses on the software update or fix, the focus of the alerting service is on the description of the vulnerability and the detection of attacks. Thus, both services complement each other.

As a tendency, the number of zero-day exploits for vulnerabilities especially in the Microsoft Windows operating systems is increasing. One of reason can be expected in the improved protection of most home-user systems which are now commonly protected by personal firewalls and virus scanners. This makes it is more difficult for attackers to compromised these systems. Therefore, zero-day exploits that are not detected by virus scanners are very effective to compromise a very large of systems in a short time interval as shown e.g. by the well-known WMF-image exploit. It has pointed out that especially web-browser are intensively investigated for unknown vulnerabilities allowing zero-day exploitation.

3 Honeynet Projects

3.1 eCSIRT.net Project

The eCSIRT.net project in [1] is an European research project which is funded through the 5th EU Framework. Major aim is to deploy a widespread network of distributed IDS sensors and to analyse the captured data. The IDS are mainly deployed by the European CERT community. However the eCSIRT.net initiative is open for participation by all European teams that have been shown to follow established best practices by joining the TI accreditation framework

The architecture consists of a distributed network of IDS sensors and a central server. All data which is captured by the IDS sensors is sent to this server and stored in a relational database.

The data is captured by a snort network sensor (Snort-NIDS) and an argus daemon. To be able to capture data sent to services, a honeypot daemon (honeyd) is installed that emulates specific services (e.g. webserver) and therefore, allows the attacker to establish TCP-connections to the sensors. Without this daemon the network sensors would not be able to capture any data contained in TCP-connections.

The Snort-NIDS captures all attacks that match the corresponding Snort signature or that can be detected by a Plug-in. Therefore, captured data is limited by the features of the Plug-ins and the available Snort signatures. Since only the basic responses of a service are emulated, the service itself is not compromised by an attacker or worm. As a consequence, the honeypots are pretty save from being compromised. This is in contrast to most high-interaction honeypots intended to being compromised. However, because of the limitation of interaction any actions of an attacker or worm that follows the initial compromise cannot be monitored. In addition, if the attack requires interaction of a service (e.g. if the attack requires a certain protocol state) the attack can only be detected if this interaction is emulated by the honeyd.

For data transport the IDMEF format is chosen which is based on the XML language.

All data is stored in a relational database on the central eCSIRT server. This data is used to create statistics which reflect the number and distribution of attacks.

3.2 LEURRE.COM Project

The LEURRE.COM project in [2] is an international project that operates a broad network of honeypots covering more than 20 countries and the 4 continents. Primary aim of the project is to gather data that allows to better understand the current malicious activity. Therefore, the aim of the project and the technical realisation is very similar in comparison with the eCSIRT project. As a consequence, the project does not focus on the in-detail analysis of attacks which is e.g. in contrast to the honeynet project. Anonymous statistical data is presented on a public web-server.

The architecture consists of a distributed network of low-interaction honeypots and a central server. All data which is captured by the honeypots is sent to this server and stored in a central relational database.

Collected data include the full network packets that were captured on the honeypots and additional data including a guess for the operating system from which the attack originated. These additional data is obtained by the `b0f` or `Disco` tools for passive fingerprinting based on TCP/IP packet header information.

As mentioned above, the primary aim of LUERRE.COM is to better understand the current malicious activity. Therefore, the project focuses on the identification of automated attack tools for whose the interaction given by low-interaction honeypots is sufficient. In [7] it is shown that the use of low-interaction honeypots is well-suited for that task.

Basically, a clustering algorithm is used to identify clusters of data that are mainly characterised by port sequences to which the malware connects to (see [6] for more details). Since each cluster is assumed to correspond to a specific attack tool (*root cause*) this approach allows to automatically identify these tools.

3.3 Nepenthes

The tool nepenthes simulates the basic behavior of services or common back-doors of trojan software in such a way that malware treats the system as if it is vulnerable. Thus, only the behavior that is necessary to lure known malware is emulated.

Therefore, the tool can be classified as a low interaction honeypot that simulates services. However, in contrast to the honeyd, nepenthes uses directly the TCP/IP stack of the native operating system and focuses on the capture and analysis of malware.

Technically, nepenthes is based on a modularised architecture. Modules exist for the simulation of vulnerable services and back-doors, for parsing and analysing shell-code, for downloading files from HTTP and FTP servers, and for logging the results. The data flow between the modules is provided by three dispatchers that invoke a function of the appropriate module and supply the data to this module. For example, if shell-code has been captured the dispatcher for shell-code passes this data successively to all shell-code Parsing Modules until a module is able to successfully parse the shell-code. Therefore, additional modules can be easily added to the nepenthes framework.

List of Nepenthes Modules:

Vulnerability modules Vulnerability Modules simulates the basic behavior of services or back-doors and accept network connections. Modules are available for various vulnerabilities and trojan backdoors.

Shellcode handler A generic module exists that parses shell-code using regular expressions. The module is able to detect code fragments that decode XOR-encoded data and to decode this data. In addition, code fragments are detected that invoke command line programs (CreateProcess). The common use of the CreateProcess function is to invoke commands that download programs using the protocols HTTP and FTP.

Download handler These modules are able to download programs using the protocols HTTP and FTP. The URLs for downloading these programs are supplied by either the Vulnerability Modules or the shell-code parsing modules.

Submission modules Submission modules allow to store the captured malware as file in the filesystem ('submit-file') or into a PostgreSQL database.

For a central logging infrastructure for distributed nepenthes clients is provided by the SURFnet IDS project.

3.4 NoAH Project

NoAH (European Network of Affined Honeypots) [3] is a a three-year European research project which is funded through the 6th EU Framework. Major aims include:

- Design a state–of–the–art infrastructure of honeypots which will gather and correlate data on cyberattacks. Focus is on the detection of zero-day vulnerabilities and internet worms.

- Develop techniques for the automatic identification of attacks, and for the automatic generation of their signatures. Mechanisms to distribute these signatures to firewalls and other containment systems will also be investigated.

- Install and operate a pilot honeypot infrastructure to demonstrate the usefulness and effectiveness of distributed security monitoring systems.

To improve the chance for capturing zero-day exploits it is crucial to monitor as much IP address-space as possible. Therefore, NoAH uses a hierarchical architecture comprised of a first layer of low-interaction honeypots followed by a second layer of high-interaction honeypots. Primary objective of the low-interaction honeypot is to monitor a large number of IP addresses and relay established connections to an appropriate high-interaction honeypot. Moreover, the low-interaction honeypots have the potential to identify known attacks. Since the focus of the project is on the analysis of unknown attacks it is important to prevent pollution of the high-interaction honeypots by known attacks.

The traffic to the low-interaction honeypots is either directly coming from the internet or sent through a tunnel. A tunnel can e.g. be realized by traffic redirection provided by a router (GRE tunnel). As a third alternative the low-interaction honeypots can receive data from *honey@home* systems. These systems are intended to be deployed by home-users to also cover these targets. Those systems are deployed on an unused IP address and relay connections to a low-interaction honeypot via an network which anatomizes the IP addresses of both systems.

Primary method to detect zero-day exploits is given by the argos containment environment developed by the Vrije Universiteit Amsterdam ([5]). Argos takes advantage of the way by which almost all exploits for buffer overflow and related vulnerabilities gain control over the attacked system. These exploits overwrite critical memory structures with data which is under the control of the attacker. For example, many buffer overflow exploits overwrite the return address of a function frame on the stack segment with an address pointing to injected shell-code. Argos's key idea is to tag all data coming from the network and to monitor the use of this data. If this data is used in an illegitimate form an alarm is raised. Illegitimate use includes direct execution of the tagged data and if such data is loaded in the instruction pointer of the CPU. As a major advantage of this approach, the attack is stopped before it gains control. Therefore, the guest operating system running in the argos environment does not need to be reinstalled after a successful attack.

Argos is implemented as a modified qemu virtual machine. Modifications include the tagging and tracking of data by the virtual machine monitor sent to the guest operating system running inside the virtual machine. In addition, the argos virtual machine allows to analyse the attack and to automatically generate a signature for the attack.

4 Benefits for CSIRTs

The quality of the CSIRT's services discussed in chapter 2 is heavily dependant on the workflow and sources of the information related to these services. In this chapter we discuss how the deployment of sensor networks and honeypots can positively influence this workflow and the information sources.

4.1 eCSIRT, LEURRE.COM, and Nepenthes Projects

As discussed in chapter 2 the number of attacks intended to non-selectively compromise machines has dramatically grown. Fortunately, the attacks itself as well as the abuse of the compromised machines can be easily detected. For the detection we can take advantage of the property that these attacks are non-selective and the attacker does not know anything about the attacked machine. Vulnerable systems are commonly identified by port-scans. These scans are often done either by continuously stepping through a network or by selecting a random IP address for the scan. Both methods cannot avoid that most network connections hit IP addresses that are not assigned to a physical machine. Therefore, scanning machines can very reliably be detected by monitoring dark IP address space. Additional information concerning the attacked vulnerability can be obtained by deploying honeypots on formerly unassigned IP addresses. For example, the sensors of the eCSIRT network employ a snort NIDS sensor to identify the attacked vulnerability. The LEURRE.COM project use custom honeyd plug-ins to emulate current vulnerable network services. Although both projects chose different technical solutions, they can very reliably identify known attacks and exploits as being used by the common internet worms and automated attacks.

The eCSIRT sensor is one of the building blocks of the *CarmentiS* project. Aim of this project is to provide the German CSIRT alliance (CERT-Verbund) with an early warning system. The early warning system includes mechanisms to import and process sensor data of multiple sources in such a way that the data can be stored in a central database in a consistent format. One of the major aims of the processing and the data format is to be well-suited for the incident handling service of CSIRTs. In detail this enables:

- Automatic import of data concerning compromised machines (IP address, etc.).
- Support for the identification of correlations between multiple incidents. E.g. if a single machine is abused to attack multiple sites or joins multiple botnets.
- Support for providing automated notification of compromised machines to the constituency.
- Identification of trends in computer abuse.

Complementary to the eCSIRT and LEURRE.COM projects is the *mwcollect alliance*. Focus of the alliance is to deploy a network of distributed honeypots to capture malware like exploit code and trojan software (e.g. IRC bots). In analogy to the aforementioned projects low-interaction honeypots are used to emulate vulnerable services. However, the emulation is optimized to be able to capture malware. To be more precise, the emulation of the service considers the protocol state at which the exploit data is sent. As soon as the exploit data has been received it is analysed by nepenthes shellcode modules to identify code fragments which try to subsequently load malware from the internet. If such a code fragment is found the malware is downloaded and stored in a relational database. As a result, a database consisting of different malware can be build covering the input from a distributed network of nepenthes sensors. Analysis of this malware reveals information concerning the detection and their properties. Because the attacks are fully automated it can be expected that the malware will be installed on many other compromised machines. Therefore, this information is very helpful if a machine is investigated for signs of intrusion (incident handling service).

In chapter 2 we mentioned that most compromised machines are controlled by a central mechanism (e.g. IRC server). CSIRTs can take advantage of this mechanism to track down other compromised machines connected to the server. If the IRC protocol is used to control a botnet this process can be partly automated by using nepenthes.

4.2 NoAH Project

While the aforementioned approaches are well-suited to effectively detect known attacks and capture known malware they commonly fail to detect unknown attacks. Basically, this is due to the lack of interaction offered by the type of honeypot used by these projects. The interaction needed for unknown attacks to succeed is unknown and for that reason, this interaction cannot be emulated in advance by low-interaction honeypots. Even if the emulation is sufficient, emulated services as provided by low-interaction honeypots does not allow to distinguish between attacks and garbage data. This is primarily because the vulnerability itself does not exist in an emulated service. However, reliable identification of unknown attack requires triggering of the related vulnerability. For example, a zero-day exploit for an unknown buffer overflow vulnerability cannot be detected until the vulnerable buffer has been overwritten.

The advisory and alerting services of most CSIRTs traditionally rely on information provided by mailing-lists. However, the recent zero-day exploits (e.g. WMF) for Microsoft Internet Explorer and Mozilla Firefox have pointed out that these public information are often incomplete and not sufficient for a helpful warning. In addition, without having effective methods to detect the specific attack, an alert concerning the attack is nearly worthless. Therefore, other sources of information are of increasing importance closing this gap of information.

As discussed in section 3, the *argos* virtual machine is especially designed to detect zero-day exploits concerning buffer overflow and related vulnerabilities. In addition, methods for automatic signature generation are developed in the NoAH project based on argos. For the early detection of zero-day exploits it is crucial to monitor a large network of heterogeneous honeypots. We expect that the likelihood of detecting zero-day exploits depends on the monitored address space and the position of the honeypots. The first requirement has been considered by the hybrid architecture consisting of low-interaction as well as high-interaction honeypots. To fulfill the second requirement, NoAH low-interaction sensors can be deployed in arbitrary networks including home-user machines in ISP networks and dynamically using unused IP address space of ISPs. Therefore, we believe that the NoAH project will provide an effective approach to support alerting services of CSIRTs.

As an additional drawback of the eCSIRT, LEURRE.COM, and Nepenthes projects they fail to detect and analyse selective attacks. Detecting is challenging because the attacker can be expected to have a specific knowledge about the attacked target. Therefore it is unlikely that the attacker can get tricked to attack a low-interaction honeypot.

However, the strategy of detecting these attacks can take advantage of the specific aim of the attacker. For example, if an attacker is assumed to steal research results, a promising approach for detection is to lure the attacker to a honeypot providing faked data attracting the attacker. Since the NoAH project allows to employ advertised honeypots, the project supports this strategy. As an additional advantage of the NoAH architecture, only low-interaction honeypots relaying all traffic to the NoAH core are deployed at the cooperating site. This limits the danger of the honeypots exposed to the deploying site.

5 Summary

Basic services of CSIRTS include incident handling and dissemination of vulnerability and patch notifications. Since the number of incidents has dramatically grown the past few years, CSIRTs need to adopt this service to the new situation. In this paper we have pointed out how the technical resources provided by different projects devoted to honeypot and sensor networks can support this process. Moreover, we have shown how the NoAH project supports in the detection of zero-day exploits.

References

[1] The european csirt network. *http://www.ecsirt.net/* .

[2] Leurre.com honeypot project. *http://www.leurrecom.org/* .

[3] Noah project homepage. *http://www.fp6-noah.org/* .

[4] K. G. Anagnostakis, S. Sidiroglou, P. Akritidis, K. Xinidis, E. Markatos, and A. D. Keromytis. Detecting targeted attacks using shadow honeypots. In *Proceedings of the 14th Usenix Security Symposium*, Baltimore, MD, USA, July 31 – August 5 2005.

[5] G. Portokalidis, A. Slowinska, and E. Markatos. Argos: an emulator for fingerprinting zero-day attacks. In *Proceedings of ACM SIGOPS Eurosys 2006*, April 2006.

[6] F. Pouget and M. Dacier. Honeypot-based forensics. In *Proceedings of the AusCERT Asia Pacific Information technology Security Conference 2004*, 23rd - 27th May 2004.

[7] F. Pouget and T. Holz. A pointillist approach for comparing honeypots. In *Proceedings of the Conference on Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA05)*, July 2005.