Web Application Risk Awareness with High Interaction Honeypots *

Sergio Nunes[1] Miguel Correia[2]

[1] Novabase [2] Universidade de Lisboa

Abstract. Many companies are deploying their business on the Internet using web applications. Risk awareness allows to mitigate the security risk of these applications. This paper presents an experiment with a collection of high interaction web honeypots in order to analyze the attackers' behavior. Different security frameworks commonly used by companies are analyzed to evaluate the benefits of the honeypots security concepts in responding to each framework's requirements and consequently mitigating the risk.

Resumo. Muitas empresas estao a lançar o seu negócio na Internet usando aplicações web. O *risk awareness* permite mitigar o risco associado a essas aplicações. Este trabalho apresenta uma experiência com um conjunto de honeypots web de alta interação, de modo a analisar o comportamento dos atacantes. Diferentes *security frameworks* utilizadas por empresas são analisadas para avaliar os benefícios do uso de honeypots web no contexto da mitigação de risco.

1 Introduction

Nowadays, most of the traffic circulating in the Internet is web traffic, traveling over the HTTP and HTTPS protocols. As multiple applications are moving to the web with the Web 2.0 phenomenon, this type of traffic tends to increase. The Web provides unified access to dynamic content with a simple browser, being able to encapsulate and integrate multiple technologies. There are multiple web ramifications divided among multiple browsers, webservers, web languages and databases that must all function flawlessly together, despite the involved complexity. The development of such web applications in its own is a complex task. Developers suffer pressure regarding time to market minimization and this leads to time sparing in software testing procedures. Without adequate security testing, web applications are deployed with multiple vulnerabilities. The data accessed through web applications is becoming more and more critical, containing private information that enables financial transactions in multiple online businesses. This vicious cycle is growing and organizations are unable to foment the necessary risk awareness to be able to analyze these new web threats.

* This work was partially supported by the FCT through the CMU-Portugal partnership and the Large-Scale Informatic Systems Laboratory (LaSIGE).

This new massification of web technologies poses multiple questions regarding information security: What is the role of security with this significant change? Is there an improvement in the confidentiality, integrity and availability of information with this new situation? Are there any new security threats that put information at risk?

The objectives of this paper are to address these questions by implementing a *high-interaction honeypot environment* composed of several common web applications used in the Internet that have reported vulnerabilities. By exposing these vulnerable web applications in a monitored honeypot architecture, the attacks can be captured and investigated, along with the tools and actions of the attacker after the intrusion. The proactive honeypot deceptive techniques record as close as possible the attackers' behavior to minimize his/her advantage, instead of relying in the common prevention, detection and reaction security approach, in the usual situation of waiting to be attacked. The careful analysis of the detailed gathered attack data and the know-how gained by managing honeypots, provides an insight about the modus operandi and motives of the attacker, classifying him according to a pre-established profile.

Having the attacker profile defined, the threat model can be specified in order to develop the necessary risk awareness and risk mitigation controls. Risk mitigation is accomplished in organizations by employing a variety of information security, compliance and risk *frameworks* that address multiple domains across the wide information technology environment. The paper considers three frameworks: ISO/IEC 27001 , Cobit and PCI-DSS. These frameworks present a major focus in security guidelines by providing specific control requirements and objectives to control risk in organizations integrating people, processes and technology as a whole. These frameworks present most of the time general guidelines that do not descend to specific security technologies, so it is important to evaluate how common security technology concepts adapt to these frameworks. Honeypots can bring added value to such frameworks by satisfying multiple enumerated control requirements.

In a nutshell, the paper tackles its objectives in a sequence of three steps:

1. Recollection of attack data using a high-interaction honeypot environment with several common web applications;
2. Web application attackers profiling based on the data obtained in step 1;
3. Analysis of the honeypots' benefits to the security guidelines provided in common risk assessment frameworks, based on the results of steps 1 and 2.

2 Context and Related Work

The honeypots' main function is to be probed and attacked [15,1,3,13]. The value of this security mechanism relies on monitoring the real steps and tools of a real attack and learning where the unknown vulnerabilities lie and how to protect the critical information assets. These monitoring and decoy capabilities aid the security professional in developing the required know-how of the modus operandi

of the attacker and infer the security situational awareness of his network to plan for the adequate safeguards and effective incident responses [16].

Web honeypots are means for gathering web attack information and develop situational risk awareness [6,14]. The Google Hack Honeypot (GHH) [10] reveals a new use for honeypots as it simulates vulnerable web applications that are commonly searched by attackers over search engines. The attacking search procedure uses careful placed search queries that are able to find vulnerable applications by matching specific strings in the previous indexed information. Mueter et al. developed a toolkit for converting automatically PHP applications into high-interaction honeypots [11]. They tested the Honeypot-Creator against a wide variety of applications and analyze the results using their high interaction analysis tool (Hihat).

What is the risk to business operations of an attack happening? Most of the time, this question remains unanswered in organizations that have services and do business over the Internet. It is crucial to mitigate the security risk using common frameworks of risk management and compliance. The regulatory compliance that organizations must meet should be dealt with due care by the upper business management, so it is necessary to have an effective way of controlling and securing information technologies. Nowadays there are multiple compliance and risk frameworks so the question remains which to use and where to direct its efforts to achieve adequate risk mitigation.

The ISO/IEC 27001 is an international standard that provides a model for establishing an Information Security Management System (ISMS) as a strategic organization decision [8]. The objective of an organization by being certified in this standard is the compliance that it has put effective information security processes in place, instead of applying non repeatable ad-hoc procedures. The certification issued by an independent third party serves as evidence that the security controls exist and function according to the standard requirements. This evidence can serve as advantage against competitors, can respond to the compliance requests of some costumers and assures business security following best practices which generate a trust relationship.

The Information Systems Audit and Control Association published the Control Objectives for Information and Related Technology (Cobit) to help information technology governance professionals to align technology, business requirements and risk management [7]. Cobit is positioned at the higher business management level dealing with a broad range of IT activities and focuses on how to achieve effective management, governance and control.

The Payment Card Industry Data Security Standard (PCI-DSS) was developed to assure cardholder data security and unify consistent data security measures globally [12]. It was created by American Express, Discover Financial Services, JCB, MasterCard Worldwide and Visa International to establish requirements for the security of the payment card industry affecting everyone that stores card payment data, including common online commercial transactions.

3 Web Application Honeypots

3.1 Honeypot environment

This section deals with the planning, implementation, configuration and analysis of the high interaction honeypot environment. The main requirement for this environment was the ability to gather detailed attack and malicious action information that provided a real situational risk awareness regarding web attacks. The environment had to be similar to a real production deployment. The option chosen was to deploy a virtual high interaction honeynet, because it does not limit the attacker's actions. The testbed was composed by real operating systems, webservers, databases and web applications constrained by virtualization.

The honeypots network, also known as honeynet, had to be managed remotely under secure conditions due to the high monitorization that this sort of high interaction honeypots needs. The solution relied on the use of a management station with SSH access over the Internet [4].

Minimize the management burden was another requirement that is tackled with the deployment of VMware Server that allows transparently copying and moving of honeypot virtual machines. The possibility of emulating ISO images as a virtual cd-rom also accelerates the installation process. VMware Server also provides the possibility of deploying checkpoints to be able to return to previous states if the honeypots are compromised or intermediate state forensic analysis is needed.

There is the risk of the attacker targeting other systems after honeypot compromise, so this situation must be controlled and safeguarded as a requisite. The response was the use of Honeywall, a layer 2 bridge with filtering, attack detection and connection limiting capabilities between the honeynet and the Internet and the possibilities of monitorization of the virtual honeypot in the host operating system employing the principle of security layering by employing multiple approaches [2].

The hardware used was composed by 12 Dell and Fujitsu Siemens Pentium 4 and Core 2 Duo PCs with 512MB to 2GB of RAM. One PC was used as the Honeywall bridge, another was the management console and the remaining ones were the VMware Server hosts used for the honeypots. The management and honeypot networks used two dedicated HP Procurve 2600 series switches physically separated. The software used for the honeypot host systems was a minimal installation of Ubuntu 8.04 which is the most recent version supported by VMware Server 2.0.1.

The honeypot host systems had two network interfaces (NICs): one configured static IP address for management and the other configured with access to VMware without IP address. The management is performed over SSH and via the VMware management console over the management NIC. The xtail command line utility was installed and configured for watching the VMware virtual disk files. Monitoring of the honeypots was done using Sebek, a kernel module designed by the Honeynet project for that purpose [15,5].

The honeypots implemented used different operating systems, different webservers, different databases and different web applications developed in different languages, as can be seen in Table 1. The operating system choice division was based on compatibility with Sebek and representativeness in the Internet hosts commonly used as webservers. The name of the honeypots represent the operating system installed with "webserver" for the Linux machines, "xp" for Windows XP machines and "win2003" for Windows 2003 machines.

Honeypot Name	Operating System	Webserver	Database	Application
Webserver1	Ubuntu 7.10	Apache 2.2.4	Mysql	PHPbb
Webserver2	Ubuntu 7.10	Apache 2.2.4	-	Wordpress
XP1	Windows XP	Apache	Mysql	EasyPHP
Win2003	Windows 2003	IIS 6.0	SQLServer	Snitz Forum
Webserver3	Ubuntu 7.10	Apache 2.2.4	Mysql	PHPNuke
Webserver4	Ubuntu 7.10	Apache 2.2.4	Mysql	PHPmyadmin
Webserver5	Ubuntu 7.10	Apache 2.2.4	Mysql	PHP-fusion
XP2	Windows XP	IIS 5.1	-	ASP-CMS
XP3	Windows XP	Tomcat	-	JSP Examples

Table 1. Honeypots specification

3.2 Experimental results

This section presents an overall statistical analysis of the results gathered from the honeypot environment from June to September of 2009 with the analysis of the attack information across different detailed graphs.

Figure 1 shows that during this time frame our environment suffered a total of 8858 attacks. It can also be observed that the first honeypot named "webserver1" (see Table 1) suffered more attacks that the other honeypots. This can be explained by its position in the IP address range as the first host serving HTTP requests as a webserver. Detecting the availability of a webserver, the attacker starts by targeting automatically this host with all his arsenal of web exploits without checking the installed web application and gives up without probing sequentially the next IP address.

The large majority of the attacks detected were not specific to the applications installed, but randomly or sequentially scanned across the honeypot IP address range for multiple specific vulnerabilities. The number of targeted attacks is 498 representing only 6% of the total of targeted and untargeted attacks.

The diversity of operating systems and webservers present in our honeypot environment does not influence the attack number results as there is no significant distinction on attack rate by operating system or when comparing webserver technologies as it can be observed in Figure 1.

Attacks to web applications (Figure 2) reveal that PHP is the most attacked web language with PHPMyAdmin as the most attacked application, while the

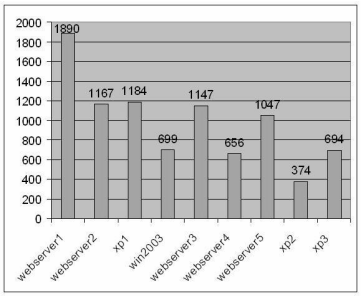

Fig. 1. Number of attacks by honeypot (8858 total)

other installed applications present no significant number of attacks with the exception of the tomcat manager. There is a significant amount of blind attacks to commonly used Internet web applications that were not installed in the environment like Horde, Roundcube or Zencart. These web applications are widely deployed over the Internet so attackers prefer to conduct random or sequential exploitation in order to compromise the highest number of machines possible with little target search and information gathering procedures.

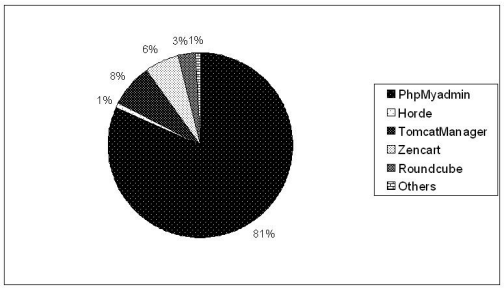

Fig. 2. Percentage of attacks by application

As it can be seen in Figure 3, there is a large amount of URL bruteforcing attacks, trying to find hidden applications with known vulnerabilities by enumerating default locations and version numbers. Direct command execution is also tried across multiple known vulnerable applications, because of the simplicity in compromising vulnerable hosts in this manner. Code Injection was accomplished against a known vulnerability in PHPMyadmin and remote file include was tried in requests to non existent vulnerable web applications in our environment. Authentication bruteforce attacks were performed against the tomcat manager application.

Figure 4 shows the worldwide origin attack distribution that probed our environment based on source addresses using GeoIPlite country mapping database by Maxmind [9]. There were 272 different attack sources detected with an average of 32 attacks by country. The United States was the main source of attack of the environment followed by China as the new rising star in hacking attempts with their huge evolution in technological resources. The addition of

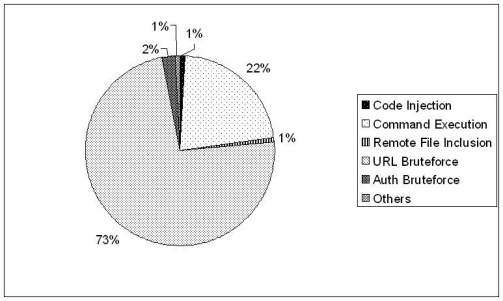

Fig. 3. Percentage of attacks by type

both these sources represents more than half of the attacks verified in the honeypot testbed. The diversity of attacking countries captured by the environment shows that there are attackers almost everywhere that try to intrude systems over the Internet bypassing any geographical borders, language barriers and cultural issues. There were only 9 attacks detected from Portuguese sources, which consisted only of web server fingerprinting attempts.

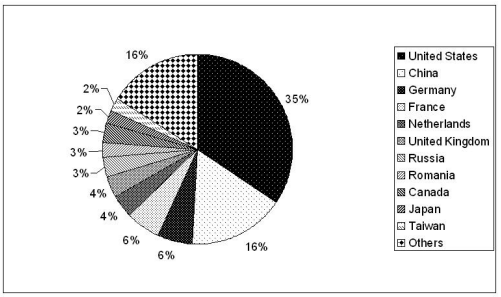

Fig. 4. Top attacking countries

Some of these attack sources can be innocent hosts that were previously intruded and are used as remote headquarters for conducting further attacks. The wide search for open proxies verified in the honeypot testbed also shows these resources are being used to masquerade the real source of attacks.

Comparing these results with the statistics of recent web attacks, we can conclude that that there was no attempt to exploit multiple cross site scripting and SQL injection vulnerabilities present in our environment, as these vulnerabilities require more knowledge to adapt to the attacker's final objective. The major threat of information leakage was not verified in our environment as it does not present real sensitive information. It can be verified that our environment suffered a high number attacks that show a rise of web threats, but as the number of targeted attacks is low it is impossible to see a wide variety of attack and vulnerability types. The high number of untargeted attacks suffered by our environment dictates that there is a maximization of quick intrusion efforts by probing the entire Internet address space for a recent disclosed vulnerability.

3.3 Attacker profiling

Based on the data and evidence gathered in our honeypot environment, this section deals with profiling the attackers of our environment, describing the characteristics and modus operandi that allow recognizing their behavior.

Most of the attacks that we faced were driven by *script kiddies* testing the latest disclosed exploit globally throughout the Internet, without even first fingerprinting the web server to see if it runs the vulnerable application. They were apparently driven by pure curiosity as most of them replayed the published exploit without any code changes and repeated its execution multiple times when, in some cases, there was no possibility of success. Most of them jumped the necessary information gathering and scanning phase to try directly to get access to the supposed vulnerable system. The intrusion can be easily identified as most of these individuals do not have sufficient skills to erase effectively their tracks or remain undetected inside the host. Their attacks are untargeted as they sweep multiple host ranges using the disclosed exploit sequentially with no focus on the system as a whole or its data value, but only as a single IP address inside the range chosen. Others performed enumeration tasks in the scanning phase looking for specifically unprotected administration components using published scripts and tools. When those components were found with authentication requirements, they conducted default and common user and password enumeration. This behavior reveals a more practical knowledge with proficiency in the use of malicious attacking tools, being able to analyze the results provided by them. As the results show failure in exploitation or take to much time to complete, they jump to the next system without analyzing further ways of intrusion.

A minority of attacks has evidence of *bot owners* as they have a modus operandi similar to script kiddies, but their main motivation is to install a bot to control the target remotely. They also start directly in the gaining access phase by searching for a specific vulnerability along a predefined range of IP addresses to maximize the intrusions and consequently the number of bots installed. After identifying a successful intrusion they upload, install and hide the bot automatically using an automated deployment script. The remote bot management is performed using an alternative protocol such as IRC, having possibilities of upgrading the bot software and of performing manual commands on the compromised host. Another difference in this modus operandi when comparing with script kiddies is that they are worried in hiding the bot and remain undetected, by for example disabling the anti-virus or installing a rootkit, in order to maintain the access to their zombies active and continue increasing the botnet power and size. This botnet power and size are the main factors that influence the profit when selling the botnet in the black market, if financial gain is the attacker's major motivation.

Our honeypot infrastructure is installed in a university IP range and has no real challenge regarding data value. The honeypot applications installed tried to simulate confidential data value such as students' forums, blogs and administration panels with predefined known vulnerabilities. Any knowledge attacker will first gather information about the target and conclude that it is situated in a

university and unless he has specific reasons to attack that host, he will continue his challenge elsewhere. The only event for which we can conclude that the attacker gathered information about the IP range ownership was the attempt to proxy requests to a scientific subscription article site. The attacker researched that multiple universities have access to scientific subscription article sites and some of those sites authenticate the subscription with the universities source IP address providing access to paid articles. The motive of this attack can be classified as profit to save money by not buying the individual articles directly onsite or selling this privileged information to other individuals looking to access the scientific subscription articles for less money than the online subscription.

4 Risk Awareness

There are multiple frameworks commonly used by organizations that help us to organize an information security system measuring the risk involving IT assets. This section analyzes how the honeypots can contribute to the risk awareness concerning threat and vulnerability identification by looking at multiple frameworks in a methodological critical approach. Using the knowledge gathered from the honeypot testbed experience and the profiling of the attacker's mindset, an evaluation is performed to research how the honeypot concepts adapt to each framework's objectives and controls, bringing added value to the organization's risk mitigation requirements.

The ISO/IEC 27001 standard mandates to monitor and review the ISMS to identify attempted and successful security breaches and incidents. The honeypots could bring to this requirement increased added value when compared to traditional intrusion detection systems, because of the detailed information gathered about an attack, which enables gaining real know-how and situational awareness of the risk that the asset faces. The usual intrusion detection systems deployed in organizations commonly match attack signatures with attacking procedures full of false positives and deviate the time of security personnel from protecting the critical assets.

In ISO/IEC 27002, the supporting standard for ISO/IEC 27001, there are some controls that can be adapted to the added value of honeypots. The control for protection against malicious code (27001 Annex A.10.4.1) can be complemented with a honeypot by performing evaluation of malicious code using client honeypots and by having a honeypot infrastructure capable of monitoring malicious code spreading mechanisms. The use of multiple different malware analysis is suggested in the standard as a vector to improve the effectiveness of malicious code protection.

The ISO/IEC 27002 standard suggests that is necessary to reduce risks from exploitation of technical vulnerabilities (27001 Annex A.12.6). The control defines that timely information about technical vulnerabilities of information systems being used should be obtained, the organization's exposure to such vulnerabilities evaluated and appropriate measures taken to address the associated risk. This is the main focus of the honeypot technology and by adequate use of

honeypots it is possible to accomplish this goal of establishing an effective management process for technical vulnerabilities that responds to the requirements.

The ISO/IEC 27002 standard details the need to ensure a consistent and effective approach to the management of information security incidents (27001 Annex A.13.2.2). It suggests defining the responsibilities and procedures to deal with the incidents collecting forensic evidence for internal problem analysis. The forensic evidence can also be used to pursue a legal action preserving the chain of custody that assures the admissibility in court. This collection of evidence can be gathered using honeypots or honeypot data gathering mechanisms. It can be seen that the chain of custody has multiple requirements to be admitted in court, so training how to collect and preserve the evidence should be an exercise first performed on decoy systems such as honeypots, to prepare for a real incident on production systems.

The ISO/IEC 27002 standard states that there should be a learning experience from information security incidents allowing the incidents to be monitored and quantified. The information gained from the evaluation of information security incidents should be used to identify recurring or high impact incidents. This learning can be developed with the risk and threat awareness delivered with the continuous use and analysis of honeypots. Honeypots were created precisely as a mechanism for learning about the modus operandi of attackers.

In the ISO/IEC 27002 standard there is a section concerning the correct processing in applications (27001 Annex A.12.2) detailing components such as input and output data validation that are the cause of multiple web attacks like those analyzed in this paper. Although honeypots are no direct defense against those attacks, they provide the necessary learning and research capabilities necessary for secure programming and correct evaluation of the risk that results with the lack of validation in applications. The attacked decoy web applications can measure the threat level and serve as case studies for future applications developed.

The protection of organizational records is also a subject detailed in the ISO/IEC 27002 standard regarding its loss, destruction or manipulation (27001 Annex A.12.5.4). Organization information disclosure attacks happen frequently in an enterprise and they are difficult prevent or even to detect. The concept of honeytokens can help in the detection of disclosure of critical data by placing careful bogus monitored records in such datastores and track those records while they travel through the network serving as a warning that the data is being disclosed.

A similar analysis has been done to COBIT and PCI-DSS, but it is not possible to show it for space reasons. Table 2 summarizes the results of the analysis done for the three frameworks.

It can be observed in the table that the honeypots can bring benefits to multiple requirements in each framework. More generically, the major benefits of using honeypot concepts when dealing with risk frameworks are:

- The creation of a risk awareness culture being able to correctly identify the threats to IT and evaluate the impact to business of attacks;

Honeypot Concept	**ISO/IEC 27001**
Risk Awareness	4.2 Establishing and managing the ISMS
Secure Coding	A.12.2 Correct processing in applications
Malicious Code Detection	A.10.4.1 Controls against malicious code
Information Disclosure Detection	A.12.5.4 Information leakage
Vulnerability Management	A.12.6 Technical vulnerability management
Incident Response	A.13.2.2 Learning from information security incidents
Honeypot Concept	**COBIT**
Risk Awareness	PO9 Assess and manage IT risks
Secure Coding	AI2 Acquire and maintain application software
Malicious Code Detection	DS5.9 Malware prevention, detection and correction
Information Disclosure Detection	DS11.6 Security requirements for data management
Vulnerability Management	DS5.5 Security testing, surveillance and monitoring
Incident Response	DS5.6 Security incident definition
Honeypot Concept	**PCI-DSS**
Risk Awareness	12.1.2 Identify threats and vulnerabilities, conduct risk assessment
Secure Coding	6.5 Develop all web applications with secure coding guidelines
Malicious Code Detection	5.1.1 Detect, remove and protect against malware
Information Disclosure Detection	3.1 Keep cardholder data storage to a minimum
Vulnerability Management	6.2 Identify newly discovered security vulnerabilities
Incident Response	12.9 Implement an incident response plan

Table 2. Honeypot benefits to three frameworks studied

- The promotion of secure coding by learning from the application attacks suffered, evaluating the coding vulnerabilities that were explored and developing the safeguards necessary to correct them;
- The detection of malicious code due to monitorization of propagation attempts and unusual activity, along with the testing of suspicious webpages and binaries in a test decoy environment;
- The detection of disclosure of information with the monitorization of decoy bogus items (honeytokens);
- The creation of an accurate and timely vulnerability management framework being able to identify, analyze and patch with a minimum time delay recently disclosed exploits and malicious tools used by attackers;
- The creation of an incident management and response system capable of identifying, classifying and addressing security problems;

5 Conclusion

In this paper an evaluation of web attack threats is presented focusing in the importance of developing risk awareness to mitigate them. To gather this attack information, a high-interaction web honeypot environment was installed, configured and monitored during approximately 4 months. This research confirmed our previous belief that honeypots are useful for companies but underestimated

by them, probably mainly because of a lack of knowledge regarding this technology, its uses and benefits. The fear of challenging the attacker and being unable to control the consequences of the intrusion is also a deterrence factor in the use of honeypots by companies. These issues are never balanced with the possibility of developing the necessary risk awareness within the company using these decoy systems to be able to defend the critical assets when a real attack emergency happens. We believe this is a critical factor enhanced by the use of honeypots: the possibility of being familiar with the modus operandi of the attacker and being prepared to respond to a real situation. Readiness only becomes effective with adequate training and this training is done using a test honeypot environment.

References

1. Anagnostakis, K.G., Sidiroglou, S., Akritidis, P., Xinidis, K., Markatos, E., Keromytis, A.D.: Detecting targeted attacks using shadow honeypots. In: Proceedings of the 14th USENIX Security Symposium (2005)
2. Chamales, G.: The Honeywall CD-ROM. Security & Privacy, IEEE 2(2), 77–79 (March-April 2004)
3. Dagon, D., Qin, X., Gu, G., Lee, W., Grizzard, J., Levine, J., Owen, H.: Honeystat: Local worm detection using honeypots. In: Proceedings of the 7th International Symposium on Recent Advances in Intrusion Detection (RAID). pp. 39–58 (2004)
4. Hatch, B.: SSH port forwarding. SecurityFocus http://www.securityfocus.com/infocus/1816 (January 2005)
5. Honeynet-Project: Know your enemy: Sebek (November 2003)
6. Honeynet-Project: Know your enemy: Web application threats (April 2008)
7. ISACA: Cobit framework 4.1. http://www.isaca.org (2007)
8. ISO/IEC 27001: Information technology - security techniques - information security management systems - requirements
9. Maxmind: Geoiplite. http://www.maxmind.com (2009)
10. McGeehan, R.: Ghh http://ghh.sourceforge.net/
11. Mueter, M., Freiling, F., Holz, T., Matthews, J.: A generic toolkit for converting web applications into high-interaction honeypots. University of Mannheim (2008)
12. PCI-DSS: Payment card industry data security standard version 1.2. http://www.pcisecuritystandards.org (October 2008)
13. Portokalidis, G., Slowinska, A., Bos, H.: Argos: an emulator for fingerprinting zero-day attacks for advertised honeypots with automatic signature generation. In: Proceedings of the ACM SIGOPS/EuroSys European Conference on Computer Systems (EuroSys). pp. 15–27 (2006)
14. Riden, J., Oudot, L.: Building a PHP honeypot. InfoSecWriters http://www.infosecwriters.com (April 2006)
15. Spitzner, L.: Honeypots: Tracking Hackers. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (2002)
16. Yegneswaran, V., Barford, P., Paxson, V.: Using honeynets for internet situational awareness. In: Proceedings of the 4th Workshop on Hot Topics in Networks (HotNets) (2005)