Robert H. Deng
Feng Bao
HweeHwa Pang
Jianying Zhou (Eds.)

LNCS 3439

Information Security Practice and Experience

First International Conference, ISPEC 2005
Singapore, April 2005
Proceedings

Springer

Robert H. Deng   Feng Bao
HweeHwa Pang   Jianying Zhou (Eds.)

Information Security Practice and Experience

First International Conference, ISPEC 2005
Singapore, April 11-14, 2005
Proceedings

Springer

Volume Editors

Robert H. Deng
Singapore Management University
469 Bukit Timah Road, Singapore 259756
E-mail: robertdeng@smu.edu.sg

Feng Bao
HweeHwa Pang
Jianying Zhou
Institute for Infocomm Research
21 Heng Mui Keng Terrace, Singapore 119613
E-mail: {baofeng, hhpang, jyzhou}@i2r.a-star.edu.sg

Library of Congress Control Number: 2005923658

CR Subject Classification (1998): E.3, C.2.0, D.4.6, H.2.0, K.4.4, K.6.5

ISSN 0302-9743
ISBN-10 3-540-25584-2 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-25584-0 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springeronline.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11407959 06/3142 5 4 3 2 1 0

Preface

The inaugural Information Security Practice and Experience Conference (ISPEC) was held on April 11–14, 2005, in Singapore.

As applications of information security technologies become pervasive, issues pertaining to their deployment and operation are becoming increasingly important. ISPEC is intended to be an annual conference that brings together researchers and practitioners to provide a confluence of new information security technologies, their applications and their integration with IT systems in various vertical sectors. The Program Committee consisted of leading experts in the areas of information security, information systems, and domain experts in applications of IT in vertical business segments.

The topics of the conference covered security applications and case studies, access control, network security, data security, secure architectures, and cryptographic techniques. Emphasis was placed on the application of security research to meet practical user requirements, both in the paper selection process and in the invited speeches.

Acceptance into the conference proceedings was very competitive. The Call for Papers attracted more than 120 submissions, out of which the Program Committee selected only 35 papers for inclusion in the proceedings.

This conference was made possible only through the contributions from many individuals and organizations. We would like to thank all the authors who submitted papers. We also gratefully acknowledge the members of the Program Committee and the external reviewers, for the time and effort they put into reviewing the submissions.

Special thanks are due to Ying Qiu for managing the website for paper submission, review and notification. Patricia Loh was kind enough to arrange for the conference venue, and took care of the administration in running the conference.

Last but not least, we are grateful to the Institute for Infocomm Research, and also the School of Information Systems, Singapore Management University for sponsoring the conference.

February 2005

Robert H. Deng,
Feng Bao, HweeHwa Pang,
Jianying Zhou

ISPEC 2005

First Information Security Practice and Experience Conference

Singapore
April 11–14, 2005

Organized by

Institute for Infocomm Research, Singapore

Sponsored by

Institute for Infocomm Research, Singapore
and
Singapore Management University, Singapore

General Chair

Robert H. Deng	Singapore Management University, Singapore

Program Chairs

Feng Bao	Institute for Infocomm Research, Singapore
HweeHwa Pang	Institute for Infocomm Research, Singapore

Publication Chair

Jianying Zhou	Institute for Infocomm Research, Singapore

Program Committee

Tuomas Aura	Microsoft Research, UK
Elisa Bertino	Purdue Univ., USA
Colin Boyd	QUT, Australia
Chin-Chen Chang	CCU, Taiwan
Kefei Chen	Shanghai Jiaotong Univ., China
Liqun Chen	HP Bristol Labs, UK
Xiaotie Deng	City Univ. of Hong Kong, China
Dengguo Feng	Chinese Academy of Sciences, China
Dieter Gollmann	TU Hamburg-Harburg, Germany
Hideki Imai	Univ. of Tokyo, Japan
Sushil Jajodia	GMU, USA
Pradeep K. Khosla	CMU, USA
Dong Hoon Lee	Korea Univ., Korea
Javier Lopez	Univ. of Malaga, Spain

David Naccache	Gemplus, France
Masahiro Mambo	Univ. of Tsukuba, Japan
Chris Mitchell	Univ. of London, UK
SangJae Moon	Kyungpook National Univ., Korea
Reihaneh Safavi-Naini	Univ. of Wollongong, Australia
Kouichi Sakurai	Kyushu Univ., Japan
Ravi Sandhu	GMU, USA
Shiuhpyng Shieh	NCTU, Taiwan
Dawn Song	CMU, USA
Dan Suciu	Univ. of Washington, USA
Rahul Telang	CMU, USA
Vijay Varadharajan	Macquarie Univ., Australia
Victor Wei	Chinese Univ. of Hong Kong, China
Moti Yung	Columbia Univ., USA
Jianying Zhou	I2R, Singapore

External Reviewers

Issac Agudo, Joonsang Baek, Lujo Bauer, Eric Brier, Julien Brouchier, Kisik Chang, C.I. Chen, Shiping Chen, Xi Chen, Benoit Chevallier-Mames, Eun Young Choi, Jean-Sebastien Coron, Guerric Meurice de Dormale, Y.J. Fu, Juan Gonzalez, Huiping Guo, Helena Handschuh, Yvonne Hitchcock, Yoshiaki Hori, Shih-I Huang, Changho Jung, Lea Kissner, Caroline Kudla, Anantharaman Lakshminarayanan, Fu-Yuan Lee, Kwangsoo Lee, Zhou-Yu Lee, Feiyu Lei, Shiqun Li, Tieyan Li, Xiangxue Li, Ya-Jeng Lin, Becky Liu, Changshe Ma, Jose A. Montenegro, James Newsome, Jose A. Onieva, Alina Opera, Pascal Paillier, Joseph Pamula, Young-Ho Park, Kun Peng, Angela Piper, Kyung-Hyune Rhee, Rodrigo Roman, W. Shin, Yuji Suga, Toshihiro Tabata, Yoshifumi Ueshige, Lionel Victor, Guilin Wang, Lingyu Wang, Shuhong Wang, Claire Whelan, Hongjun Wu, Hsiao-Chan Wu, Yongdong Wu, Yi Xu, G.M. Yang, Tzu-I Yang, Jungjae Yoo, Kee-Young Yoo, T.H. Yuen, Ruishan Zhang, Xuan Zhou, Bo Zhu, Huafei Zhu

Table of Contents

Network Security

Cryptographic Techniques I

Secure Architecture I

Access Control

Intrusion Detection

Applications and Case Studies

Secure Architecture II

Data Security

Cryptographic Techniques II

Risk Assessment of Production Networks Using Honeynets – Some Practical Experience

Stephan Riebach, Erwin P. Rathgeb, and Birger Toedtmann

Computer Networking Technology Group
Institute for Experimental Mathematics and Institute for Computer Science and
Business Information Systems, University Duisburg-Essen
{riebach, erwin.rathgeb, btoedtmann}@exp-math.uni-essen.de

Abstract: Threats for today's production networks range from fully automated worms and viruses to targeted, highly sophisticated multi-phase attacks carried out manually. In order to properly define and dimension appropriate security architectures and policies for a network, the possible threats have to be identified and assessed both in terms of their impact on the resources to be protected and with respect to the probability and frequency of related attacks. To support this assessment, honeynets, i.e. artificial networks set up specifically to monitor, log and evaluate attack activities, have been proposed. In this paper, experiences and results gained with setting up, deploying and operating such a honeynet are reported together with some comments on the effectiveness of this approach.

1 Introduction

It is well known that today's networks are subject to numerous threats ranging from blind, automated worm and virus attacks via prefabricated "standard" attacks by using readily available exploits to highly sophisticated, expert attacks. It is also obvious that securing a network involves an intelligent tradeoff between cost (in terms of equipment, expertise, usage restrictions and manpower) and the required level of security. To properly balance this tradeoff, current data on types, frequency and impact of attacks is required. Although some general information on worm and virus threats as well as on known system vulnerabilities is readily available, more specific information related to the customized mix of hardware, operating systems and software used in a network is difficult to obtain.

Larger networks are typically secured by using firewall systems filtering out "evil" packets and traffic on the network, transport and application layer. In addition, Intrusion Detection Systems (IDS) are typically deployed as additional safeguard to detect attacks and anomalies behind the firewalls. Reports and log files from these systems can provide some information on the frequency of attacks. However, since their purpose is to suppress any suspicious activity as early as possible to protect the production network and its data, this information is clearly biased as, e.g. multi phase attacks are blocked at an early stage. To some extent, IDS log information reveals

Robert H. Deng et al. (Eds.): ISPEC 2005, LNCS 3439, pp. 1 – 12, 2005.

also the type of attack. However, as a typically rule based system, e.g., the IDS can only recognize known patterns. Moreover, these logs provide only a limited basis to correlate different attack activities.

Therefore, it has been proposed to set up dedicated, artificial networks, called honeynets [1,1a], specifically for the purpose of monitoring, observing and analyzing malicious activities. Since honeynets are typically not hidden behind firewalls and are tightly controlled with all activity logged on packet level, they provide an unbiased view of the threat situation and at the same time allow performing in depth forensic analysis offline. Since honeynets don't have real users and, thus, don't hold information that has to be protected, malicious activity can be allowed in a controlled way to be able to observe the impact of successful intrusions.

In order to gain practical experience with the honeynet approach, we have implemented and deployed a honeynet. We have operated it over a period of four months and have used it for gathering detailed statistical data on attack activity on one hand and for in depth forensic analysis of specific attacks on the other. This paper summarizes our experiences with respect to the effectiveness of the honeynet approach and also highlights some of our findings.

2 Generic Honeynet Architecture

The term "honeynet" was coined by a group of security experts organized in the "Honeynet Project" (www.honeynet.org). This group promotes the development and application of honeynet concepts and is the main source of the definitions used in this section.

The basic idea that led to the development of honeynets was to detect, observe and document the activity of hackers inside a computer network. Therefore, honeynets are highly specialized networks which have to be kept strictly separate from the actual production networks, have no real users – and thus no real traffic activity – and don't contain any real information (user data). To be able to observe attacks, honeynets have to be vulnerable to a certain extent which means that they cannot be strictly protected by firewalls and that their systems should at least show some of the common vulnerabilities. Honeynets are highly controlled which means that elaborate monitoring and logging facilities capture and document all activity to provide comprehensive data for forensic analysis. All honeynets are "artificial" in a sense that they have no real users and, therefore, no real traffic. Therefore, all traffic in a honeynet is per definition suspicious and traffic originating from a host in a honeynet is an indication that this system has likely been taken over.

In addition to the "honeypot" computers to be scanned, probed or attacked, a "data capture" function is required to make the honeynet useful. In addition to storing all data packets for offline forensic analysis, online monitoring with host and network based Intrusion Detection Systems (IDS) is useful to provide immediate notification about ongoing attacks as well as a basis for targeted forensic analysis. The data capture function can be distributed among several computers or concentrated in a centralized device. Because honeynets are intentionally vulnerable, so called "data control" mechanisms must be implemented to ensure that intruders can not misuse compromised honeypots for further attacks. There are several ways to perform data control, e.g., limiting the outgoing bandwidth, restrictive outgoing packet filtering or, adding packet loss and high delays to outgoing connections [1,1a]. It is particularly important that only the honeypots are visible and accessible for intruders. Therefore, the data control and capture functions have to be hidden from intruders in order not to reveal the honeynet character of the network. In addition they have to be protected against any manipulation.

The honeynet concept has evolved significantly over the past few years, in particular with respect to implementing data capture and control functions [1,1a]. There is a broad spectrum of realization options for honeynets ranging from software emulating specific aspects of operating systems, applications and services (e.g. Honeyd, see www.honeyd.org) to real networks with hosts providing real services and applications. Whereas simple emulations allow only limited interaction; honeynets with live systems allow full interaction. However, the latter ones require significantly more effort for setup, configuration and maintenance.

3 A Practical Honeynet Setup

In our project, the honeynet architecture shown in Fig. 1 was used with 5 honeypots connected via a 100baseT hub. The honeynet was connected to the internet via a router (dual homed Linux machine) providing the data control function. We used packet filtering and additional bandwidth limitations for the outgoing traffic.

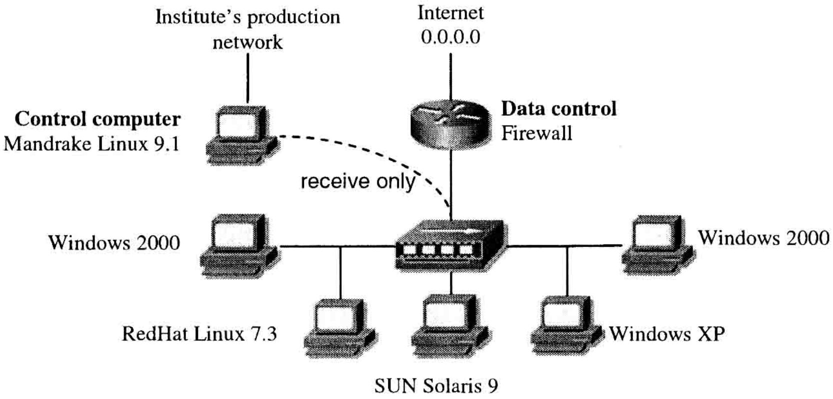

Fig. 1. Honeynet setup for this case study

3.1 Configuration Aspects

The control computer was set up with two network interfaces. A modified network cable was used to connect one of these interfaces to the honeynet in receive only mode [2]. Therefore, it would have been fairly impossible also for expert intruders to detect the presence of this machine and subsequently to attack it. Furthermore, the log data and reports collected on this machine could not be modified from the honeynet avoiding that attackers could cover their tracks. Due to these precautions, it was possible to connect the other interface of the control computer to a production network for maintenance and remote data retrieval. The control computer was responsible for monitoring and capturing all network traffic in the honeynet. For traffic monitoring, the Network Intrusion Detection System (NIDS) "SNORT" [3] was installed to identify known attack signatures in the honeynet traffic continuously and in real time. The SNORT log files were automatically archived once a day and automatically processed locally on the control computer by "SnortSnarf" [4] which produced formatted statistical output. These statistics presented in HTML were automatically published on a web server on the control computer and could be remotely inspected from the production network. In addition, the major statistics files were automatically sent to the honeynet operator once a day. For data capturing the software "tcpdump" [5] was deployed. With tcpdump all data traffic occurring in the honeynet was saved into daily dump files including all protocol overhead (addresses, etc.) from OSI layer 2 upwards. To assure the permanent availability of these vital honeynet components, SNORT and tcpdump were monitored by using "Daemontools".[6]

As indicated in Fig. 1, two of the honeypots were configured with the Windows 2000 operating system, one with Windows XP, one with RedHat Linux 7.3 and the last one with Solaris 9. This mix of operating systems was chosen because it is fairly typical for our production networks. With respect to the vulnerability of the honeypots we updated to a patch level which was fairly typical for an environment where there is only limited central administration of the systems and the users have to take responsibility for their systems themselves. All honeypots were equipped with Host Intrusion Detection Systems (HIDS). Since the honeypots were not accessible remotely from the production network for security reasons, the log files of all HIDSs were collected manually in regular intervals. Complete images of the software installations of all honeypots were saved. Therefore, a compromised system could be restored to its original state with minimum effort. Before cleaning up a compromised system, we also saved a complete image for offline analysis and possible reinstallation for further observation.

3.2 Deployment Aspects

In order to allow for unbiased measurement of the Internet, our honeynet was located outside the firewalled university network. Since the first day the honeynet was running, activity could be detected confirming the statement [7] that a honeynet will be found and attacked without further actions needed.

In order to find out how the attractiveness of the honeynet can be influenced by its configuration, we carried out a phased deployment study [8] increasing the visibility of the honeynet in every step. The first observation was that the attack frequency doesn't significantly depend on the lifetime of the honeynet, i.e. it neither increases because the network becomes known over time nor decreases because it has been identified as honeynet. As a consequence, no significant "warmup" phase seems to be required before starting statistical measurements. The most significant increase in attack frequency which could be clearly attributed to a configuration change was observed after the full activation of a DNS server making the network fully visible in the Internet. This effect can be attributed to the fact that DNS lookups are used by the spreading mechanisms of internet worms as shown in section 4.1. Activation of various popular services (http, ftp) on the honeypots did increase the frequency of non-automated attacks as well as their diversity (cf. section 4) with the result that all services provided were eventually attacked. However, the attacks were still unspecific in a sense that a significant part of the attacks on the web server were targeted towards the Microsoft IIS although only an Apache server was running. Actively generating traffic by having honeypots participating in a P2P file sharing network (providing fake content only) proved to be counterproductive. The P2P search and download processes produced an enormous amount of data traffic, but only limited correlation could be detected between the P2P traffic and attack signatures. None of the IP addresses used in the P2P communication was involved in any non-P2P signature. Furthermore there was no temporal correlation between attacks and P2P traffic; also the overall attack frequency didn't increase significantly. From our study it can be concluded that making the honeynet known in the DNS and providing a range of popular services on the honeypots is useful whereas actively generating traffic is not worthwhile and also clogs the log files with irrelevant data.

3.3 Operational Aspects

A honeynet is a highly specialized instrument to detect, observe and analyze attacks in detail. To produce meaningful results, honeynets require daily administration and maintenance. During normal operation, the honeynet generated at least 1 Mbyte of SNORT log data and 75 Mbyte (average) of tcpdump logs per day. This raw data was completely archived for statistical and forensic analysis. The statistical evaluation was highly automated as described above.

The portscan log files were transformed into the CSV-format for detailed analysis in MS Excel. The automatic formatting and publishing of the SNORT logs allowed for a quick inspection and gave indications about potentially successful attacks which were then followed up. In addition, the HIDSs of the honeypots had to be collected and inspected on a daily basis so a compromised honeypot could be identified rather quickly. Furthermore, sporadic in depth control and analysis of the honeypots was necessary to minimize the probability of undetected attacks. These routine tasks amounted to a maintenance effort ranging from a minimum of one hour up to several hours per day.

A manual forensic analysis was performed in several cases where successful (non-automated) attacks could be detected. For manual inspection of the tcpdump log data, the program "tcptrace" [14] was used to identify successful TCP connections related to successful attacks. "Ethereal" [15] was then used to fully decode the packets of interesting connections up to the application level. Due to the size of the logs and the need to scan several of them for various attributes, e.g. specific IP addresses, this was a rather significant and time consuming effort.

4 Results Relevant to Risk Assessment

Statistical analysis of attack activity was mainly based on the SNORT log files. SNORT classifies attack signatures into severity levels. In the following we distinguish between

- Alarm: dangerous and harmful attacks (SNORT priority 1)
- Warning: suspicious signatures potentially preparing attacks (priority 2)
- Notice: unusual traffic not identified as dangerous (priority 3)

As Fig. 2 shows, there was no obvious correlation between the number of alarms and the number of warnings. From this we concluded that the majority of attacks were blind attacks which were not being prepared by intensively scanning and probing the network first.

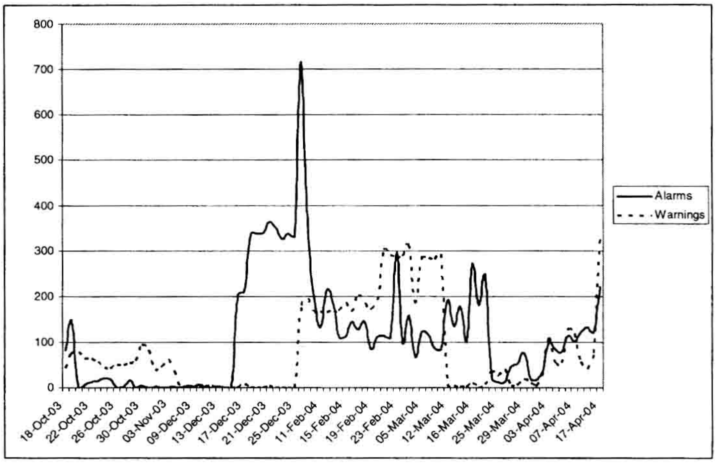

Fig. 2. Number of alarms and warnings during the study

When analyzing the targets of attacks, we found that nearly 97% of all attack signatures were specifically targeted towards the Windows systems. This was not unexpected since it makes sense to concentrate the effort to develop attacks on the clearly dominating operating systems. However, the obvious conclusion that windows systems are significantly more threatened would be misleading here, because a more detailed analysis of the alarms showed that 81.25% of all alarm signatures were of the type "NETBIOS DCERPC ISystemActivator bind attempt" [16]. This signature is generated by an attack against Microsoft's DCERPC interface on port 135/tcp and can clearly be linked to worm spreading mechanisms. As a consequence, it is sufficient to apply one single patch to the Windows systems to counter the vast majority of attacks and bring the remaining attack frequency into the range observed for other operating systems. During the study period, we didn't identify any worm related activity targeted towards non-windows systems. However, even if the attack frequency towards the other operating systems was still lower after filtering out worm activity, attack sophistication seemed to be even higher and also resulted in successful takeovers. The Solaris honeypot, e.g., was compromised by a classical multi-phase attack eventually exploiting a known vulnerability of the "cachefsd" service.

After filtering out the worm activity from the analysis, the impact of providing increasingly more services on the honeypots on the attack frequency became obvious as shown in Figure 3. Furthermore, a more significant correlation between alarms and warnings becomes visible indicating a higher share of more elaborate, multi-phase attacks.

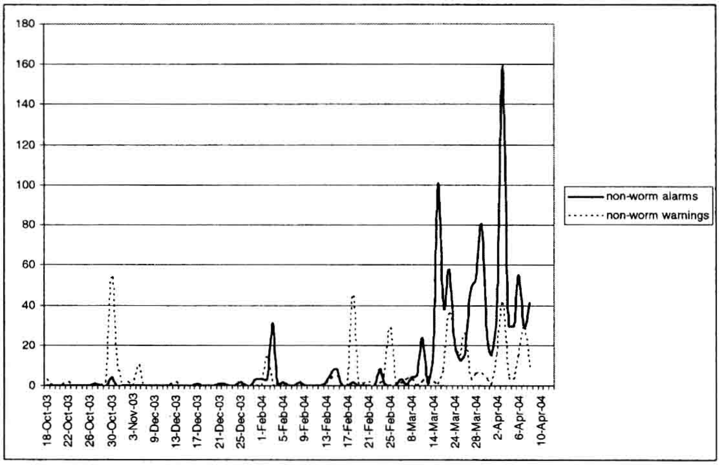

Fig. 3. Non-worm related attack activity

4.1 Analysis of Worm Related Activity

After identifying worms as the dominant source of attacks, a more detailed evaluation could be performed to establish correlations providing some insight into the mechanisms involved in worm activity. We also analyzed the priority 3 notices logged indicating unusual packets not identified as dangerous, like ICMP PING's, and their correlation to worm related alarms as shown in Figure 4. Since Snort differentiates ICMP ECHO packets by their generating operating system, we found that 95% of all ICMP ECHO packets were generated by Windows systems. Because of the high volume of these ICMP scans in the first half of the study period, we assumed that they were directly related to worm activity.

However, this scanning activity only resulted in a corresponding increase in the frequency of actual attacks starting on Dec. 16th 2003 with an instantaneous rise. This coincides explicitly with the moment at which we configured the local DNS server in the honeynet to perform DNS reverse lookup. This indicates that the Windows worms produced an enormous amount of ICMP packets to identify possible targets, performed a DNS lookup for the identified IP addresses and then attack the systems.