Honeypots:
A Practical Mean to Validate Malicious Fault Assumptions

Practical Experience Report

Marc Dacier, Fabien Pouget
Eurecom
2229, Route des Crêtes ; BP 193
06904 Sophia Antipolis Cedex
France
Email: {dacier, pouget}@eurecom.fr

Hervé Debar
France Télécom R&D
42 rue des Coutures ; BP 6243
14066 Caen Cedex 4
France
Email: herve.debar@francetelecom.com

Abstract

In this paper, we report on an experiment run with several honeypots for 4 months. The motivation of this work resides in our wish to use data collected by honeypots to validate fault assumptions required when designing intrusion-tolerant systems. This work in progress establishes the foundations for a feasibility study into that direction. After a review of the state of the art with respect to honeypots, we present our test bed, discuss results obtained and lessons learned Avenues for future work are also proposed.

1. Introduction

It is well agreed upon by the dependability community that dependable systems have to be designed with respect to certain fault assumptions. These assumptions define, among other things, the classes of faults that can occur, their rates of occurrences, the amount of simultaneous faults that can be injected into the system etc. Intrusion tolerant systems are nothing else but classical fault tolerant systems dealing with a special class of faults: malicious ones. However, as opposed to accidental faults, no sound and representative set of data has been accumulated to validate fault assumptions made in the design of intrusion tolerant systems..

We claim that honeypots can be used to accumulate data sets concerning the attack processes. They should play an important role in the future design of intrusion tolerant systems. We have conducted an experiment during 6 months to validate that claim. The results are given hereafter and avenues for further research are considered.

The paper is organized as follows. Section 2 proposes a general introduction to the notion of honeypots. Section 3 offers a survey of existing work. Section 4 describes the set up we have used for our experiments. Section 5 presents some of our results and Section 6 concludes the paper.

2. Honeypots: Introduction

2.1. Definitions

Honeypots, honeytokens and honeynets have been used for some time in computing systems even if the use of this terminology is recent. In the late 80's, Clifford Stoll [28] had the idea of placing 'interesting' data in appropriate places to lure hackers. This idea is now formalized as a "honeytoken" by Lance Spitzner [30]. In the 90's, Cheswik implemented and deployed a real "honeypot" [7]. Bellovin discussed the very same year the advantages and problems related to its usage [3]. In 98, Grundschober and Dacier ([15], [14]) introduced the notion of "sniffer detector" (see also [1]), one of the various forms of what is called today a "honeytoken". Lance Spitzner has proposed the following definition for a honeypot[1]:

[1] It is worth noting that a) this definition is different from the one given by the same author in his book [29], ii) this new definition has been discussed at length on the honeypot mailing list [17] but no final consensus has been reached among the participants.

«A honeypot is an information system resource whose value lies in unauthorized or illicit use of that resource.» [30].

During the last 2 years, many different implementations of the concept of honeypots have been proposed. Some attempts, yet not convincing, have been made to classify them (see for instance [29], [17], [8]). We report the interested reader to [26] for a more detailed presentation of these tools as well as for a discussion of the classification issues.

3. State of the art

3.1. Research on data collected by honeypots

Honeypot data serves three main purposes:

Post mortem Analyses: in this case, honeypots machines are expected to be fully compromised by hackers [25]. Once this stage is reached, the machine is halted and analyzed by means of tools such as the « coroner toolkit » [13] or its successor, the « Sleuth kit » [6]. The purpose of these analyses is to discover out new software attack tools not yet found in the wild.

Identification of new threats: it has been claimed that honeypots could also be used as early warning systems. They could be designed to quickly identify new types of threats. However, as far as we can tell, no published work has investigated this problem in some depth. Some anecdotic report exists, such as the ones done with wireless honeypots ([9], [22], [33]) that aim at showing the risk of leaving wireless networks unattended.

A Statistical data gathering tool: the rapid evolution of existing platforms might be the reason for the surprising lack of publication of data collected by honeypots over a long period of time. The most visible project, the honeynet project has published a first document in 2001 [18] but, since then, seems to have focused on implementations issues. The Irish team appears to be the only member of the Honeynet Research Alliance to offer such data on its web site [16] but this concerns their sole environment and it does not provide any kind of analysis. A noteworthy exception in the field of statistical data analysis can be found in [10] where the authors analyze, thanks to their honeypots, the propagation of the NIMDA worm. Thus, as of today, only one team seems to have investigated the possibility of using data from honeypots to model attack processes.

3.2. Research in network monitoring

As early as 1993, Bellovin [4] has shown the interest of studying real packets passing on the networks. He showed the existence of anomalous behaviors, packets that did not indicate an attempted break-in but that, nevertheless, were worthy of attention. The museum of broken packets [34] offers a survey of such weird packets. There is now a conference [21] (previously a workshop [19], [20]) where results of work dedicated to the analysis of real data streams are presented. Security issues have been studied during the last two editions, indicating a growing interest of that community for these problems and highlighting the usefulness of gathering real world data to study them.

4. Testbed Description

We have used three different environments to find out which kinds of honeypots bests suit our needs. The first one consists in a single machine that passively collects data with tcpdump [31] and has no single port open. In the second environment, the Honeyd tool [27] is used to simulate three virtual machines. Similarly, the third environment is a virtual network built on top of Vmware [32]. In the following, for the sake of conciseness, we have decided to focus on the results obtained with this sole last environment Three machines are attached to a virtual Ethernet switch[2] supporting ARP spoofing. The VMware commercial product enables us to configure them according to our specific needs. *mach0* is a Windows98 workstation, *mach1* is a Windows NT Server and *mach2* is a Linux Redhat 7.3 server. The three virtual guests are built on *non-persistent disks* [32]: changes are lost when virtual machines are powered off or reset. In A fourth virtual machine is created to collect data in the virtual network. It is also attached to the virtual switch and tcpdump is used as a packet gatherer [31]. This machine and the VMware host station are totally invisible from outside. Both Mach1 and Mach2 run an ftp server; in addition Mach1 also provides a static web server. Logs are collected daily and transferred to a safe place where they are enriched with some external data, as discussed below, and inserted into a database.

5. Results

[2] A switch in the Vmware jargon but it actually behaves as a hub.

5.1. Introduction

In the following we report results based on data collected between March and June 2003 (4 complete months). We have observed a total number of 970718 packets coming from 6285 different IP sources. As such, we have seen more than 2 new attacking sources per hour. The vast majority of the observed packets were TCP ones (97.9 %). Others were ICMP (1.4 %) and UDP packets (0.7 %). Attacks were directed towards a very limited number of ports, 164 in total. In 70.4 % of the cases, an attacking source has sent requests to the three honeypots in a very short period of type (typically in a few seconds). In only 5.6 % they have contacted only 2 out of the three machines. However, and this is something important we discuss later, in 24% they have focused on only one of the three honeypots. Interestingly enough also, attacking sources do not seem to come back: they have never been observed for more than one day. This could partially be explained by the fact that machines owned by individual users obtain a different, temporary address, whenever they are connected to the Internet.

In the context of this paper, we show two things. First, it is indeed possible to learn something about the attack processes and the threats that we are facing. The data highlight the existence of some stable processes that could and should be modeled. Second, the observations we have made show the need for designing a more global set up, at the international level, to answer questions that are left open.

The results proposed are divided into three main categories that characterize i) the attacking machine ii) the attacked machine iii) the attacked port.

5.2. Information on the attacking machine

5.2.1 Geographical location. In order to find out where the attacks were coming from, we have taken advantage of the Netgeo utility [24], developed in the context of the CAIDA project [5]. It consists in a database and a collection of sophisticated Perl scripts that map IP addresses and AS numbers to geographical locations.

Surprisingly enough, most attacks originate from Australia (33 %), the Netherlands (21 %) and the USA (17 %). These countries are definitely not the usual suspects that would be quoted by security experts [12]. As of the time of this writing, we have not yet figured out the reason for that difference. More data from French and European honeypots would be required to get a better understanding. We have observed attacks from 85 other countries but these three ones account for more than 70 % of the total.

These ratios are fairly stable over the four months period. This was quite surprising to us as well but fortunate since it indicates the existence of some stable process that is at the origin of this phenomenon. Its modelization must then be possible.

5.2.2 Operating System of the attackers. In order to find out what kind of operating system was used on the attacking machine, we have used the Disco utility ([11], [2]) to passively fingerprint TCP packets. Since they represent the majority of the observed packets the restriction to this kind of packets is not a problem. Our analysis indicates that 71% of the attacking machines were running a Windows operating system. 22 % of all the attacking machines where not recognized by Disco. This might either indicate that many attacking machines have been configured to defeat fingerprinting or that Disco's coverage in terms of OS is incomplete.

5.2.3 Timing of the attacks. The surge in attacks has been attributed to compromised personal computers permanently connected to the Internet through home broadband connections. If automated robots are responsible and if they have continuous access to the net, we were expecting to see them attacking all day long. However, our analyses do not confirm this but rather indicate a small increase in the attacks during the late afternoon and evening. This indicates that either the attacks are not originating from automated robots or that a majority of attacking machines is not connected continuously. We speculate the latter and we expect to see this trend to disappear as more and more machines get permanent Internet access. This assumption is confirmed by the fact that more attacks are observed during week ends. This non homogeneity indicates the need of some human intervention to make an attack possible.

5.3. Influence of the OS of the target

Each target has been probed approximately by a third of all attacking IP addresses. This distribution is stationary over the four months. This seems to indicate that targets are chosen completely at random. According to our investigation, this does not appear to be completely true though. Two distinct attack processes are apparently taking place and none of them is choosing IP addresses randomly.

The first process is in charge of scanning machines. In that case, we observe that attacking sources are scanning the three machines sequentially. This happens, as written before, in more than 70 % of the

cases. It is worth noting that, whenever a source attacks our three honeypots, the sequence is always the same, namely mach0 then mach1 then mach2. We have never observed any other sequence during the 4 months period. The nature of scans has also changed. It was frequent, a few years ago, to see systematic scans, i.e. scans where all ports between 1 and 1024 were probed. Nowadays, scans are targeted against a very limited number of ports. The largest scan we have observed has probed only 9 different ports on each machine.

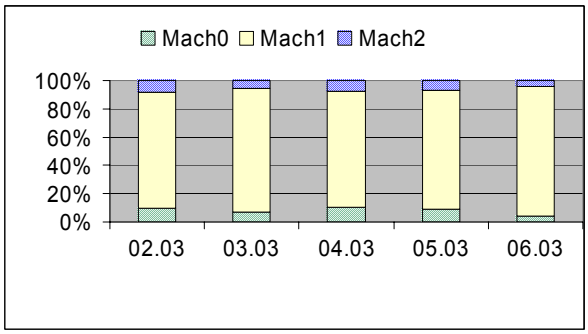

Figure 1 shows the percentage of packets received per honeypot. The unequal repartition is mainly due to the different OS behaviors. Mach1, the NT server, is a lot more responsive than mach0 and mach2. We are also surprised to observe how constant the packets repartition remains over the months.

The second process concerns the 24 % of the cases where an attacking machine is probing only one of our honeypots. In these cases, the machines have never been seen doing a port scan. At the contrary, and much to our surprise, they were always, systematically, sending requests to ports that were open on the machine they were talking to.

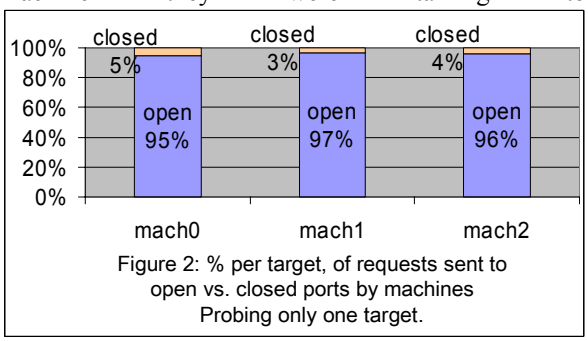

Figure 2: % per target, of requests sent to open vs. closed ports by machines Probing only one target.

This phenomenon is clearly highlighted in Figure 2 where we have represented the percentage of those 24 % of requests that have been sent to an open port. To obtain these values we listed the opened ports of our machines and we compared the destination ports of the requests to our list. In over than 95 % of the cases, the attackers have made no mistake. The 5% remaining requests are identified as traces of some Denial-of-Service activity. The IP addresses of our machines were spoofed and attack targets (victims) answered by sending typical packets to our addresses, such as: TCP (RST ACK), TCP (RST), TCP (SYN ACK), ICMP (Host Unreachable)… [MoVS01].

This is to be compared to the general case in Figure 3 where we see that the percentage of requests sent to closed ports vary between 23 and 53 %, depending on the target.

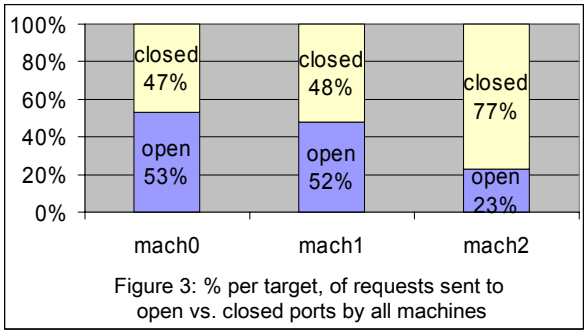

Figure 3: % per target, of requests sent to open vs. closed ports by all machines

The variation is due to the fact that each machine had a different number of open ports. Mach2, especially, is a Linux machine which had all Windows specific ports close (in particular ports 445, microsoft-ds, and 139, netbios-ssn). Since these ports were very frequently requested by machines performing scans, this machine exhibits a different pattern than the two others which had these ports open.

5.4. Analysis of targeted ports

164 different ports have been probed by attacking machines. Figure 4 shows the distribution of packets received by each port. Port 139 stands clearly out with more than 50 % of the received packets.

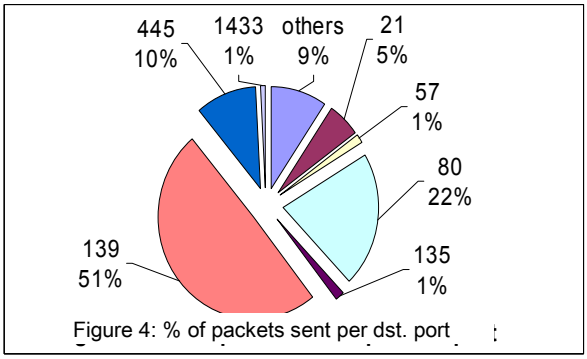

Figure 4: % of packets sent per dst. port

The service running on that port is *netbios-ssn* for Windows network shares. Whenever an attacking machine discovers that this port is open, it sends a large number of requests aimed at gathering information about the target. A similar thing happens with port 80 (http) for which a large number of attacks are known.

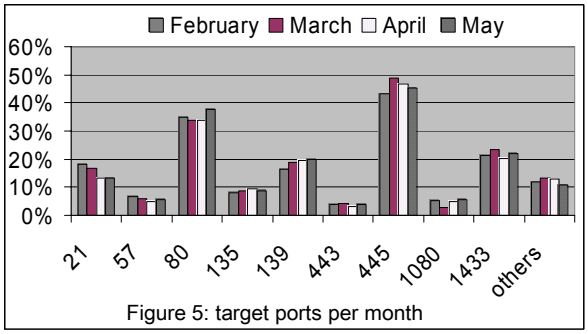

Figure 5: target ports per month

Attack sources try many of them systematically and this results in a much higher percentage for these two ports.

The attack process against these ports seems to be fairly regular. Indeed, if we observe how many sources have sent requests to the most popular ports on a month per month basis, we find out, as shown in Figure 5, that i) certain sources, such as 445 (Microsoft-ds) and 80 (http), are requested by more sources but that ii) the "popularity" of each port is very stable over the 4 months period. This is counter intuitive as we were expecting to find peaks of activities against specific ports whenever a new attack tool is published in underground mailing lists. This phenomenon, at first glance, seems to be lost in the noise of the regular attacks but more data is required to draw a conclusion.

It is worth noting that port 139 is probed by less than 20 % of the attacking sources but, as seen here above, packets sent to that port account for more than 50 % of the observed attacking packets. In other words, the most dangerous threats (in terms of number of attackers) are not necessarily the most visible (in terms of number of packets).

6. Conclusions

In this paper, we have presented data obtained by means of three honeypots being attacked over a period of four months. Three main lessons can be learned from the presented data. First of all, the regularity exhibited by the data indicates that there is some real value in using data from honeypots to model attack processes and threats. We postulate that honeypots should therefore be much more used by the community interested in providing rationales for its fault assumption models.

Secondly, our observations appear to be sometimes different from what authors in other locations have reported. This is, for instance, the case with Australia being our main source of attacks or scans being limited. There clearly is a need for more identical test beds put in diverse locations to validate and complete our analysis. Diversity must be not only be considered in terms of geographical locations but also in terms of target types (education, government, private sectors, etc...). This motivates the need for a truly international collaboration in that space.

Last but not least, our current results have opened some avenue for further research. Some questions have been left unanswered. The unexpected behavior of machines knowing exactly which port was opened on which machine is something that must be clarified. If there exists a collaboration process taking place between scanning machines and attacking machines, we need to design new, dynamic, environments not only to find out how long it takes for new information to be collected and shared but also to figure out if we are facing one or several populations of collaborating attackers.

7. Bibliography

[1] H. AbdelallahElhadj, H. M. Khelalfa and H. M. Kortebi, "An experimental sniffer detector: SnifferWall", SEcurité des Communications sur Internet Workshop (SECI'02), Tunisia, Sept. 2002.

[2] O. Arkin, "*ICMP Usage in Scanning - The complete Know-How*", The Sys-Security Group. Version 3.0. June 2001. (www.sys-security.com)

[3] S. Bellovin, "There Be Dragons", *Proc. of the Third Usenix Security Symposium*, Baltimore MD. Sept. 1992.

[4] S. M. Bellovin, "Packets Found on an Internet", *Computer Communications Review* 23:3, pp. 26-31, July 1993.

[5] Cooperative Association for Internet Data Analysis, home page: www.caida.org

[6] *The sleuth kit V1.62* (previously known as TASK), Brian Carrier, 2003, www.sleuthkit.org/

[7] B. Cheswick, "An evening with Berferd in which a cracker is lured, endured and studied", *Proc Winter USENIX Conference*, San Francisco, Jan 20, 1992.

[8] F. Cohen, D. Lambert, C. Preston, N. Berry, C. Stewart and E. Thomas, "*A Framework for Deception*", Tech. Report, July 2001, http://all.net/journal/deception/Framework/Framework.html

[9] P. Cracknell, "*The wireless honeypot project: A brief look at how wireless networks are used and misused in the City of London*", RSA Security UK Limited (RSA SUL), CISSP, Tech. Report, http://www.rsasecurity.com/worldwide/downloads/honeypot_report2003.pdf

[10] H. Debar and D. Lefranc, "Observations on the Internet traffic reaching broadband-connected users", *EICAR Conference*, Copenhagen, Mai 2003.

[11] *The Disco tool* home page: http://www.altmode.com/disco

[12] J. Evers, "*Experts: Most Code Red attacks coming from Asia*", IDG News Service, 2001. Available on line: www.computerworld.com/securitytopics/security/story/0,10801,62730,00.html

[13] Coroner toolkit, D. Farmer, W. Venema, home page: http://www.porcupine.org/forensics/tct.html

[14] Stéphane Grundschober, "*Sniffer Detector Report*", ,Internship Report from IBM Zurich for the Eurecom Institute , June 1998, 50 pages, ref. Eurecom : CE-98/IBM/GRUN - Document number: 1914. Available on line: http://www.eurecom.fr/~nsteam/Papers/grundschober98.ps

[15] S. Grundschober, M. Dacier, "Design and Implementation of a Sniffer Detector", *Recent Advances on Intrusion Detection Workshop (RAID98)*, 1998. www.raid-symposium.org/raid98/

[16] Irish Honeynet Alliance members. Collected data available on line: www.honeynet.ie/results.htm

[17] *Honeypot mailing list*, FAQ; www.securityfocus.com/popups/forums/honeypots/faq.shtml

[18] "*Know Your Enemy: Statistics Analyzing the past ... predicting the future*", Honeynet Project, July 2001. Available on line: http://project.honeynet.org/papers/stats

[19] *Internet Management Workshop 2001*, home page: http://www.icir.org/vern/imw-2001/

[20] *Internet Management Workshop 2002*, home page: http://www.icir.org/vern/imw-2002/

[21] *Internet Management Conference 2003*, home page: http://www.icir.org/vern/imc-2003/

[22] E. Jacksch, "*Tenebris Wireless Honeypot Project: Assessing the threat against wireless access points.1.0*", CISSP, Tenebris Technologies Inc, 2002. www.tenebris.ca/docs/TWHP20021119.pdf

[23] D. Moore, G. Voelker et S. Savage. "*Inferring Internet Denial-of-Service Activity*", 2001 USENIX Sec. Symp.

[24] *Netgeo Utility*, available online at http://netgeo.caida.org/perl/netgeo.cgi

[25] A. Neville, "*IDS Logs in Forensics Investigations: An Analysis of a Compromised Honeypot*", March 2003. Available on line: http://www.securityfocus.com/infocus/1676

[26] F. Pouget, M. Dacier, H. Debar. "*Honeypot: a comparative survey*", Eurecom Report, RR-03-81

[27] *Honeyd* Home page, Niels Provos, http://www.citi.umich.edu/u/provos/honeyd/

[28] C. Stoll, "Stalking the Wiley Hacker", *Communications of the ACM*, Vol. 31 No 5. May 1988.

[29] L. Spitzner, "*Honeypots: Tracking Hackers*", Addislon-Wesley, ISBN from-321-10895-7, 2002.

[30] L. Spitzner, "*Honeytokens: The Other Honeypot*", 2003. www.securityfocus.com/infocus/1713

[31] *Tcpdump* home page: http://www.tcpdump.org/

[32] *VMWARE*, User's manual. Version 3.1, home page: http://www.vmware.com

[33] *WISE*, Wireless Information Security Experiment, http://www.incident-response.org/WISE.htm

[34] "*The museum of broken packet*", M. Zalewski, http://lcamtuf.coredump.cx/mobp/