A Pointillist Approach for Comparing Honeypots

Fabien Pouget[⋆1] and Thorsten Holz[⋆⋆2]

[1] Institut Eurécom, BP 193, 06904 Sophia-Antipolis Cedex, France
[2] Laboratory for Dependable Distributed Systems, RWTH Aachen University, 52056 Aachen, Germany

Abstract. Our research focuses on the usage of *honeypots* for gathering detailed statistics on the Internet threats over a long period of time. In this context, we are deploying honeypots (sensors) of different interaction levels in various locations.
Generally speaking, honeypots are often classified by their level of interaction. For instance, it is admitted that a high interaction approach is suited for recording hacker shell commands, while a low interaction approach provides limited information on the attackers' activities. So far, there exists no serious comparison to express the level of information on which those approaches differ. Thanks to the environment that we are deploying, we are able to provide a rigorous comparison between the two approaches, both qualitatively and quantitatively. We build our work on an interesting classification of the observed attacks, and we pay particular attention during the comparison to the bias introduced by packet losses.
The proposed analysis leads to an interesting study of malicious activities hidden by the noise of less interesting ones. Finally, it shows the complementarities of the two approaches: a high interaction honeypot allows us to control the relevance of low interaction honeypot configurations. Thus, both interaction levels are required to build an efficient network of distributed honeypots.

1 Introduction

Many solutions exist for observing malicious traffic on the Internet. However, they often consist in monitoring a very large number of IP addresses like a whole class A network or a large range of unused IPs. Several names have been used to describe this technique, such as *network telescopes* [1, 2], *blackholes* [3, 4], *darknets* [5] and *Internet Motion Sensor* (IMS) [6]. Some other solutions consist in passive measurement of live networks by centralizing and analyzing firewall logs or IDS alerts [7, 8]. A few websites report such trends like DShield,

⋆ Work by F. Pouget is partially supported by the French ACI CADHO in collaboration with LAAS-CNRS and CERT Renater
⋆⋆ Work by T. Holz is supported by the Deutsche Forschungsgemeinschaft (DFG) as part of the Graduiertenkolleg "Software for mobile communication systems"

SANS/ISC or MyNetwatchman [7, 9, 10]. Coarse-grained interface counters and more fine-grained flow analysis tools such as NetFlow [11] offer another readily available source of information.

So far, nobody has investigated the possibility of using a large number of local and similar sensors deployed all over the Internet. However, we strongly believe that local observations can complement the more global ones listed above. A direct analogy can be made here with weather forecast or volcanic eruption prediction, where both global and local approaches are applied. As a consequence, we are on the way to deploying many small honeypot environments in various locations thanks to motivated partners, as part of the Leurre.com Project. The main objective is to gather statistics and precise information on the attacks that occur in the wild on a long-term perspective. We have initially used high interaction honeypots. Then, because of the incoming and increasing number of participants in addition to the hard constraints imposed by their implementation, we have considered the idea of deploying low interaction honeypots. At the time of writing, some environments of different interaction levels are running. We invite the interested reader to have a look at the existing publications for more information on that point [12–14].

An important issue that must be addressed with such deployment is the bias introduced by the choice of low interaction platforms. The environmental setup we present here gives us the opportunity to make a rigorous comparison of two different interaction approaches, both qualitatively and quantitatively. So far, such a comparison did not exist. Honeypots have been classified in interaction categories without concrete justification [15]. For instance, it is admitted that a high interaction approach is suited for recording hacker shell commands, while a low interaction approach provides limited information on the attackers' activities. This paper intends to show that this classification is too restrictive. As far as our research objectives are concerned, both approaches present value. The contributions of this paper are the following:

- We show that both approaches provide very similar global statistics based on the information we collect.
- A comparison of data collected by both types of environments leads to an interesting study of malicious activities that are hidden by the noise of less interesting ones.
- This analysis highlights the complementarities of the two approaches: a high interaction honeypot offers a simple way to control the relevance of low interaction honeypot configurations and can be used as an effective etalon system. Thus, both interaction levels are required to build an efficient network of distributed honeypots.

The rest of the paper is structured as follows: Section 2 describes and justifies the setup of the distributed honeypot. This environment has been implemented in two different ways corresponding to two distinct interaction levels. The analysis is then built on these two approaches. Section 3 introduces a comparison of global statistics obtained by means of these two distinct implementations. in

particular, we show the similarity of the information provided by the two environments. In Section 4 we take a closer look at some activities that are apparently different between platforms. This in-depth study of both platforms leads to the discovery of strange attack scenarios that require particular attention. We finally explain to what extent high interaction honeypots can be used as reference systems to optimize the configuration of low interaction ones. These two last Sections provide rationales for the Leurre.com project that we are deploying. Finally, Section 6 concludes this paper.

2 Environment Setup: two different levels of interaction

2.1 High Interaction Experimental Setup - H_1

We have presented in previous publications [12,16] some experiments based on so called "high interaction honeypots". This environment, called in the following H_1, is a virtual network built on top of VMware (see Figure 1) [17]. Three machines are attached to a virtual Ethernet switch [3] 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 [17]: changes are lost when virtual machines are powered off or reset. We perform regular reboots to guarantee that the virtual machines are not compromised, as the objective is to gather statistical data in a long-term perspective. 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 [18]. This machine and the VMware host station are as much as possible invisible from the outside. Both mach0 and mach2 run an ftp server; in addition, mach1 provides a web server. Logs are collected daily and transferred to a centralized and secure place.

We have also made some comparisons with another deployed "high interaction" honeypot called GenII [19]. However, the collected data were based on snort-inline [4] alerts. First, alerts provide different information than raw data (see Section 2.3 to find explanations on the information we can extract) and are quite likely false positives. Second, snort-inline drops packets based on the way it *estimates risk*. These two reasons have prevented us from making interesting comparisons at this stage. Thus, we do not refer to this architecture in the following.

2.2 Low Interaction Experimental Setup - H_2

We have deployed a platform called H_2 similar to H_1 presented before, but with emulated operating systems and services. We have developed it based on several open source utilities. Indeed, it consists in a modified version of honeyd [20]. The

[3] A switch in the VMware jargon actually behaves like a hub

[4] snort-inline is an open source Intrusion Prevention System (IPS)

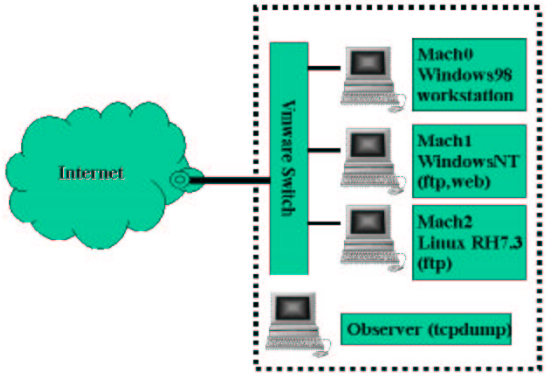


Fig. 1. H_1 Environment scheme

platform only needs a single host station, which is carefully secured by means of access controls and integrity checks. This host implements a proxy ARP. This way, the host machine answers to requests sent to several IP addresses. Each IP is bound to a certain *profile* (or *personality* in the honeyd jargon). Thus, H_2 emulation capacity is limited to a configuration file and a few scripts. It emulates the three same Operating Systems as H_1 for mach0, mach1 and mach2. We have scanned the open ports in H_1 and opened the very same ones in the honeyd configuration file for each of the three virtual machines. Some service scripts that are available in [20] have been linked to open ports, like port 80 (web server) or port 21 (ftp). As a consequence, H_2 can be seen as offering a similar yet simplified behavioral model of H_1. In the same manner, we connect every day to the host machine to retrieve traffic logs and check the integrity of chosen files.

2.3 Information Extraction

As previously explained, dump files are periodically collected from H_1 and H_2 and are stored in a centralized database. There, they are analyzed by means of other utilities and additional information is brought in, such as IP geographical location, domain name resolution, passive OS fingerprinting, TCP stream analysis, etc. For the sake of conciseness, we do not want to detail the database architecture and the way we obtain information in this paper; we invite the interested reader to look at our previous publications, where we have described the setup in detail [14, 21].

3 Global Statistics Analysis

3.1 Introduction

Honeypots can be seen as *black boxes*: they describe a system whose internal structure is not known. All what matters is that the device transforms given inputs into predictable outputs.

In our case, incoming malicious requests are the input and provided replies are the output. Let I_1 be the quantity of information from Honeypot H_1 (the high interaction honeypot). In the same way, let I_2 be the quantity of information provided by Honeypot H_2 (the low interaction honeypot). Intuitively, we expect $I_2 \lessapprox I_1$. However, it is more difficult to estimate to which extent I_2 brings less information. The following Sections intend to qualify and quantify this information difference $I_1 - I_2$.

The initial setting is the following: environments H_1 and H_2 are both placed in the same network. The virtual machines mach0, mach1 and mach2 have three adjacent IPs in H_1, say X.X.X.1, X.X.X.2, X.X.X.3. In a similar way, virtual machines mach0, mach1 and mach2 have in H_2 contiguous addresses, resp. X.X.X.6, X.X.X.7, X.X.X.8.

H_1 has been running since February 2003. Environment H_2 started running on July 2004. A technical problem prevented us from collecting the whole month of November 2004. Thus, we will focus on data collected on both environments from August 2004 to October 2004, that is 10 continuous weeks.

We propose in the following Section to study the differences between the two platforms in that period, thanks to the information stored in the database (see Section 2.3).

3.2 Attack Categories

Both environments H_1 and H_2 are targets of attacks. Each environment contains three virtual machines running different services and different OSs. They are not equally targeted. This leads us to define three major categories of attacks:

- The ones which target only one machine. They are called attacks of Type I.
- The ones which target two out of three virtual machines. They are called attacks of Type II.
- The ones which target all three virtual machines. They are called attacks of Type III.

Table 1 represents the distribution (in percentage) of these 3 categories on each environment H_1 and H_2. Values are very similar. This attack classification is used in the following to start comparing environments.

3.3 Type III Attack Analysis

We propose in this Section to look at Type III attacks. They stem for around 35% of the total attacks. Figure 2 represents the number of associated sources

Attack Type	H_1 Environment	H_2 Environment
Total	7150	7364
Type I	4204 (59%)	4544 (62%)
Type II	288 (4%)	278 (4%)
Type III	2658 (37%)	2542 (34%)

Table 1. Different Attack Types observed on H_1 and H_2

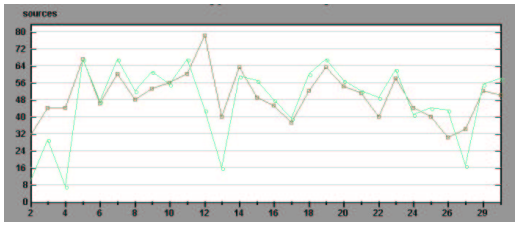


Fig. 2. Attacks of Type III on the two French platforms H_1 and H_2

observed on environments H_1 (dark curve) and H_2 (light curve) every 2 days. Curves have the same general shape. We do not expect any difference for the reason that attacks targeting the three virtual honeypots are likely to be broad-sweeping scans [13]. Thus, those scans should be observed independently on the platform. In other words, there should be the same number of scans on both platforms. This is not exactly the case in Figure 2 where curves present small dissimilarities.

A closer look at the attacks confirms that almost all IP sources associated with Type III attacks have been observed on both environments. For those which are not included in one curve, it appears that they are classified as attacks of type III in one environment, and in attacks of Type II in the other one. In a few cases, they are even classified as attacks of type I. An analysis of the corresponding packet traffic reveals that they often consist of a single TCP packet sent to one target. It might happen that packets are lost due to congestions in the Internet and we can imagine that such packets are not retransmitted by the attacker. To validate this assumption, we check that there is no bias in the loss observation, that is, we observe an equal number of packet losses on platform H_1 and on platform H_2. In addition, the number of supposed scan packet losses is distributed among all virtual machines without apparent preferences. As a first approximation, the value we observe can also be linked to the estimated TCP

packet loss value in the path between the attacking machine and the honeypot environment at a given date. If for a period of time $\Delta(t)$ the estimated packet loss between the attacking source and the honeypots environment is p_loss, then the probability Pr of getting an incomplete scan on the six virtual machines becomes:

$$Pr = 1 - (1 - p\_loss)^6 \tag{1}$$

In this experiment, we identify 92 such losses over a total of 2851 distinct type III attacks during the two-month observation (observed on both environments or only one). According to the previous equation, this is equivalent to an average packet loss of 0.6%, which remains coherent with actual traffic monitoring [22]. This is even quite low if we compare with the global average 2-5% observed on the Internet Traffic Report web site [23]. However, we also note on their site high differences between continents. European traffic seems less susceptible, in average, to packet losses than other continents such as Asia.

A first assertion based on our experiment is:

Assertion 1 *It is not necessary to deploy honeypots using hundreds of public IP addresses in order to identify scan activities against large block IPs. Three addresses contained in that block are sufficient. Large-scale scans will be attacks on the three honeypot machines. We may observe only two attempts in case of packet losses, as it appears that not all scanning engines do implement packet retransmission processes.*

To complete the analysis, we also observe another interesting property common to H_1 and H_2 based on the fact that virtual machines have been assigned contiguous IP addresses. The main scanning technique consists in issuing requests to IP address by incrementing their IP value by 1. To quantify the importance of this scanning method, we represent in 3.3 the six possible orders of scanning that have been observed. We give for each of them their frequency (in percentage), that is, the number of IP sources which have targeted the three virtual machines over the total number of IP sources associated to Type III attacks.

Type III Attack Order	Percentage
Order 1: Mach0, Mach1, Mach2	79%
Order 2: Mach0, Mach2, Mach1	5%
Order 3: Mach1, mach0, Mach2	4%
Order 4: Mach1, Mach2, Mach0	5%
Order 5: Mach2, Mach0, Mach1	3%
Order 6: Mach2, Mach1, Mach0	4%

Table 2. Scanning order for Type III attacks

The figures remain quite constant when computing it on a monthly basis. Attacks targeting machines by increasing IP numbers correspond to 79% of the

total. The other values are more or less equal. It is important to point out that all attacks which have targeted the three machines of one platform in a different order than Order 1 have, instead, respected this Order 1 when sending packets to the three machines of the other platform.

This highlights the fact that all scans are done according to Order 1 but some packets may arrive in a different order on the platform, creating the illusion of other scanning orders. This remark is also validated by studying the source ports used by the attacking machine, and more specially, their sequence over the scans on the honeypot virtual machines. It consists in 80% of the cases in an arithmetic sequence with a common difference of 1. These simple observations of two different but correlated sequences (targeted virtual machines and attacking source ports) leads to three major remarks:

- We observe scan activities that sweep through IP addresses sequentially in decreasing order in very few cases.
- Almost all scans that target three consecutive IPs are programmed to hit them sequentially in increasing IP order. It might happen, however, that the order is slightly disrupted because of some packet retransmissions. A study of the different source ports used by the attacking machine confirms this (the non-privileged ports are used sequentially).
- Scanning machines do not wait for a scan to be finished in order to target the next IP. Scanning threads are not blocking. In other words, we observe that temporal periods of scanning activities against two virtual machines from a same source can overlap.

Finally, we intend to have a closer look at some scanner implementation options in order to build relationships with the observed traces. For instance, the advscan Sourceforge Project allows parametering some variables such as the number of concurrent threads, the delay or the scanning duration [24].

3.4 Type II Attack Analysis

Attacks of Type II represent a very small fraction of all observed attacks on H_1 and H_2. As we explain in the previous Section, some scanning activities that target a large block of IPs can *miss* some addresses insofar as the tools do not retransmit lost packets. It has been observed that 88% of the attacks of type II are residues of scanning attacks on both environments H_1 and H_2, and thus, are incomplete Type III attacks. The remaining 12% are more interesting:

- *For 9% of Type II attacks:* The IPs have been observed against two virtual machines on one environment, namely mach0 and mach2. The attacking IPs have also been observed on the other environment. A closer look at the source ports used by the attacking machines leads to the conclusion that these attacks scan one out of two successive IPs. Indeed, all these IPs which have targeted mach0 (X.X.X.1) and mach2 (X.X.X.3) on H_1 have targeted mach1 (X.X.X.7) only on H_2. Inversely, all these IPs which have targeted

mach0 (X.X.X.6) and mach2 (X.X.X.8) on H_2 have only targeted mach1 (X.X.X.3) on H_1. This can be seen as a limitation of our local honeypot platforms. Indeed, we will not be able to distinguish attacks with larger scan hops. We are not aware of any tool using this strategy. However, a complementary analysis can be performed by means of large telescopes and darknets.

– *For 3% of Type II attacks:* They concern attacks on the sole two Windows machines mach0 and mach1 on both environments H_1 and H_2. They are for instance attack attempts on port 5554 (Sasser Worm FTP Server [25]) or port 9898 (Dabber Worm backdoor [26]). It is clearly not the usual propagation techniques of these worms. We face attacks that have acquired some knowledge of the existence of Windows machines on both environments, and that have made some random-like attempts on them. Indeed, we do not observe attempts on both ports but only one on each machine. The attacking IPs are also not observed on both environments, unlike the others.

This leads to a second assertion:

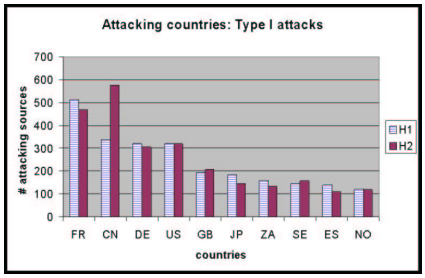
Assertion 2 *Attacks targeting two out of three machines can be specific to the two victim machines, but are with high probability residues of scanning activities.*

3.5 Type I Attack Analysis

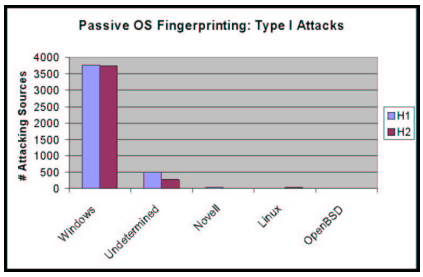
Categories of type I are far more difficult to compare between environments H_1 and H_2. They account for around 60% of all attacks on each machine. Figure 3 represent some global characteristics of these attacks on both environments. To be more precise, Figure 3(a) presents the geographical location of the attack sources corresponding to Type I attacks. On the horizontal axis are presented the top 10 countries. The vertical axis gives the number of associated attacking sources for each environment. Figure 3(b) gives the estimated attacking OS, based on passive OS fingerprinting techniques [27]. The vertical axis gives also the number of associated attacking sources for each environment.

As a general remark, there is no important differences between environments H_1 and H_2. For instance, both are targeted by 4 main countries with the same order of magnitude (France FR, China CN, Germany DE, United States of America US)[5]. The other country participations are more variable over months but remain coherent between both environments. The passive fingerprinting analysis confirms this similarity between attacks on the two environments too. The IP sources which attack the platforms are essentially running on Windows. To complete this comparison, Figure 4 lists the 10 most targeted ports on each platform H_1 and H_2. The vertical axis shows the number of associated attacking sources for each environment. The order is identical and the number of attacks on those 10 ports are very similar on both environments.

[5] The geographical location has been obtained by means of the Maxmind commercial utility [28]



(a) Attacking Countries



(b) Passive Fingerprinting

Fig. 3. Global statistics Comparison between H_1 and H_2

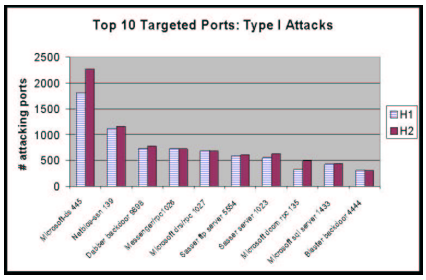


Fig. 4. Top 10 Targeted Ports for Type I attacks on each platform H1 and H2

In summary, Type I attacks represent lots of common characteristics between platforms H_1 and H_2. On the other hand, the amount of information collected on both environments is totally different. The high interaction platform H_1 has received 480684 packets sent to its virtual machines. This is 40 times as many as what H_2 has received. This is quite normal, since many attacks target *talkative* services like Microsoft ones (see Figure 4) which are not emulated on the low interaction honeypots. The following Section intends to present a refined analysis of the differences which are mainly due Type I attacks.

4 Refined Analysis

4.1 Different Type I categories

As illustrated by the previous Section, Type I attacks present very similar global statistics between the two environments H_1 and H_2 (see Figures 3 and 4). Thus, if we intend to limit the analysis to these figures, we can clearly use a low

interaction honeypot instead of a high interaction one. The complexity of the last configuration is not justified, according to the comparison we made. On the other hand, the number of collected packets is totally different. At this stage, we cannot guarantee that type I attacks observed on H_1 are exactly the same as the ones observed on H_2. Since the previous statistics tend to indicate this property, we propose in this Section to refine the Type I attack analysis, in order to check that they indeed present very similar characteristics between both platforms. Thanks to our setup, we are able to distinguish two distinct phenomena that are correct explanations for some observed type I attacks. We group all the remaining *non classified* attacks in a third category. These three categories of type I attacks are discussed in the following Sections.

4.2 Sequential Scans residue

This is the first category of Type I attacks. They are to be compared with the same large scanning activities than we presented in Section 3.3. This case can be rare but we can also imagine that two losses can happen on the same environment. It is simply identified by looking at common IP addresses on both environments which have targeted one machine on one environment and three virtual machines on the other one, during a short period of time. We find the same number of corresponding sources on H_1 and on H_2, 1 out of 1000 Type III attacks in average. To validate that it correctly corresponds to packet losses, we consider that if for a period $\Delta(t)$ the estimated packet loss between the attacking source and the honeypots environment is p_loss, then the probability Pr to observe two losses out of three scans becomes approximatively:

$$Pr = 3 * p\_loss^2 * (1 - p\_loss) \tag{2}$$

This remains coherent with the low number of cases we observe. This category has been observed thanks to the complementarities between H_1 and H_2. Indeed, a single environment cannot allow identification of such attacks.

4.3 Random Propagation Activities

This is the second category of Type I attacks we can imagine. Many tools choose random IPs during their propagation process. They can be worms or bots (Sasser, W32/Agobot, Bobax, etc [25, 29]). As they choose their victims randomly (or at least randomly in a certain IP class, for instance a class B if they favor local propagation), it is quite normal to observe a given IP source only once if it belongs to such an attack process.

To identify these Type I attacks, we have decided to build a technique upon the work already published: we have presented in [13] a clustering algorithm that allows identifying root causes of frequent processes observed in one environment. Due to space limitations, we refer the interested reader to [13] for a detailed description of the clustering technique. In brief, we basically gather

all attacks presenting some common features (duration of the attacks, number of packets sent, targeted ports...) based on generalization techniques and association-rules mining. The resulting clusters are further refined using "phrase distance" between attack payloads. In summary, we gather within a cluster all attacking sources that are likely to have used the same attack tool to target a given machine.

As a consequence, tools propagating through random IPs have similar characteristics, even if they are not observed twice on the environments, so they should belong to the very same cluster. These Type I sources are more precisely characterized by clusters where all IP sources have targeted only one virtual machine, and where the attacks within a single cluster are equally distributed among virtual machines. If the distribution of the attacks per virtual machine is homogeneous (which means we do not observe a significant number of attacks on a few virtual machines only), we consider that the attack belongs to this category which we call *Random Propagation Strategy Category*. We have systematically verified this property for all clusters, with the algorithm presented in Table 3.

If we consider the 240 clusters associated with attacks on H_1, only 54 correspond to type I attacks. In addition, 43 out of these 54 clusters have *random propagation strategies*. The remaining 0.5% of the observed clusters that are associated with type I attacks are discussed in the next category. Finally, we want to point out that attacks on that category can be identified as easily on platform H_1 as on H_2.

4.4 Targeted Attacks and Opened Issues

This is the third category of Type I attacks. It gathers all Type I attacks which cannot be classified in the two previous categories. They are not numerous, as explained above. They are represented by 0.5% of the clusters and imply a few dozen attacking sources. This category regroups various attacks of interest, due to their originality. These attacks have always targeted the same virtual machine in only one environment. The reasons why some attacks focus on one machine only are really worth being investigated to determine if a specific service is targeted or if this is due to another phenomenon. In the following, we give two illustrative examples:

- *Example 1: Attacks on port 25666 target virtual machine mach0 on H_1.* This attack has been observed 387 times from 378 different IP addresses between August 2004 and February 2005. Each attack source sends on average three packets to mach0. A closer look reveals that all packets have 80 or 8080 (http) as TCP source port and RST-ACK flags set. They are replies to DoS attacks against web servers, also known as *backscatters* ([2]). In summary, we have observed for 6 months DoS attacks against different web servers, and these attacks always spoofed mach0 IP address with source port 25666. Such regular processes have been observed in other platforms we developed. Up to now, we have observed 15 of these processes on H_1 and H_2.

Surprisingly enough, these attacks occur very regularly, day after day. It seems also surprising that DoS tools choose to use static spoofed addresses: either spoofed (IP,port) are somehow hardcoded in a tool used by different people (which would be more than bizarre), or these DoS attacks, observed during 6 months, are part of a unique process launched against several targets over a very long period of time. This means that the spoofed address list has been generated once, and has then been used for multiple attacks. The regularity of such a process also indicates that a common cause is the underlying reason for all these attacks. Finally, these *periodic backscatters* come to ports that are likely close on both environments (usually very high non-privileged ports in the range $[1025, 65535]$). Thus, we would get the same amount of information, whatever the targeted environment is.

– *Example 2: Targeted port 5000 Attack on mach1 on H_2.* Two very different worms are mainly responsible for port 5000 scans. The first, *Bobax*, uses port 5000 to identify Windows XP systems. Windows XP uses port 5000 (TCP) for 'Universal Plug and Play (UPnP)', which is open by default. The second worm, *Kibuv*, uses an old vulnerability in Windows XP's UPnP implementation to break into these systems. This vulnerability was one of the first discovered in Windows XP and patches have long been made available. However, we observe a cluster that is associated to that port. It gathers 73 distinct IP sources that have targeted only one virtual machine on port sequence 5000. Surprisingly enough, the 73 attacks have targeted the very same virtual machine within two months. This does not match the Bobax and Kibuv worm propagation scheme, as it has been found that they rather scan machines randomly. In addition, it is important to note that the port is closed on that machine. Packets contain no payload. They are limited to half a dozen TCP SYN packets. This attack cannot be considered as random insofar as it always implies the same virtual target.
At the time of writing, we have no concrete explanation of such a phenomenon. It has also been noticed by other administrators in Incidents mailing lists [30]. The Michigan Internet Motion Sensors group notifies in [31] that the observed activities do "not support the theory of Kibuv entirely". This might be due to revived threats such as *Sockets de Troie (Blazer 5)* or *1998 Trojan ICKiller* or Yahoo Chat or non-referenced tools based on the UPnP exploit [32,33]. A closer look at the received packets is required at this stage to determine the attack. However, as the port 5000 is close in both platforms H_1 and H_2, we would get the same amount of information, whatever the targeted environment is.

Type I attacks are very interesting. We have identified backscatters related activities and tools with widespread random propagation. A few numbers of attacks remain unclassified. They seem to be specific to the platform itself, so some precautions must be required to understand them. At the time of writing, they are hidden in the noisy permanent activities and thus, they do not really trigger lots of attention. Simple honeypots emulating a few IPs allow their identification. This is a preliminary but necessary step to start their in-depth analysis.

For each Cluster C_j of type I:

Preliminaries :

Compute the number N_j of attacks associated to C_j on the Environment
Compute the number $N_{j,0}$ of attacks associated to C_j on the virtual machine mach0
Compute the number $N_{j,1}$ of attacks associated to C_j on the virtual machine mach1
Compute the number $N_{j,2}$ of attacks associated to C_j on the virtual machine mach2
We check that $N_{j,0} + N_{j,1} + N_{j,2} = N_j$
$Threshold = 0.1N_j$

Test on Cluster C_j:

Mean $= \mu = \frac{N_j}{3}$

variance $= \sigma^2 = \frac{\sum_{0 \leq k \leq 2}(N_{j,k} - \mu)^2}{3}$

IF $\sigma < Threshold$
THEN
$res = 1$
Cluster C_j associated to random propagation tools
ELSE
$res = 0$
Cluster C_j associated to targeted attacks
A closer look at packet contents is required.

Table 3. Simple algorithm associated to Type I tools having random propagation strategies

Then, more interaction on that port would bring valuable information on that attack. As the attack is very specific and we have no preliminary knowledge on it, writing a simple script to H_2 is not the correct choice. A controlled environment like H_1 must be built to observe the attack details when launched against real interactive systems. In a second step, a script can be developed for H_2.

We show here that high interaction honeypots are very complementary to low interaction honeypots as they can indicate which services are not currently interactive enough on low interaction honeypots. We intend in the last Section to make this analysis more automatic so that we can determine which services must be developed (by means of scripts) on the low interaction honeypot to get a similar amount of information.

4.5 Interaction Differences and Improvements

The platforms are globally targeted in the same way, as has been detailed in the previous Sections. However, it is also clear that we collect more data on a high interaction honeypot, as real services exchange more packets witht the attackers. In average, 40 times more packets are collected with H_1 than with H_2. Based on these observations, this Section intends to show where the information is lacking, and how this can be handled.

As specified in Section 2, platforms H_1 and H_2 have similar configurations. All open ports on machines in H_1 are also opened in H_2, and vice-versa. On the H_2 side, it can be sufficient to open a port in order to get attack information. It can also be necessary to develop simple emulation scripts in order to enhance

the environment interaction. Thus, the idea is the following: The more attacks interact with a port, the more important it is that honeyd runs an interactive script behind. In other words, if the amount of information we obtain on attacks through a given port on H_1 is a lot higher than the one captured on H_2 against the same port, one of the two following actions must be undertaken:

- A script must be implemented to emulate the associated service if any.
- The script interaction should be brought to a higher level if the script already exists.

Obviously enough, each attack may require different interaction levels. For instance, scans do not require high interaction and an open port on both environments will give the same amount of information.

Furthermore, the error would be to consider here only packets from/to a given port to compare the amount of information between the two environments. For instance, if a source sends a request on port A and then waits for the answer to communicate with port B, the missing information if port A is closed on the other environment is a lot more important than just considering the simple request/answer on port A. We miss all the communication with port B as well.

As a consequence, we use the clusters presented in [13] and introduced in Section 4 to avoid these problems and to determine what services should be enriched on H_2. Each cluster groups together all IP Sources sharing strong characteristics in their attack processes. These attacking sources have exchanged the same amount of information on one environment. The interaction we get on a virtual machine must be weighted by the frequency of the attacks on the involved ports, as we explain above. The interaction is quantified by considering the number of exchanged packets. This can be refined by taking payload length into account, but we limit this analysis on this simple assumption. This leads to the algorithm presented in Table 4:

The algorithm has been launched on each platform for a 2-month period. We get the following results:

- For ports where simple scripts are already attached to H_2, it appears they behave correctly compared to the real services running in H_1.
- For Netbios ports (135, 139 and 445 specially), the ratio $\frac{I(H_2)}{I(H_1)}$ is equal to 1.5%. No script emulates these services in H_2. This is clearly not acceptable, insofar as H_2 is missing a large quantity of information in comparison to H_1. We are in the process of writing scripts to emulate these services.
- For other ports like 111, 515,..., the operation of opening these ports provides as much information as the real services in H_1 at this time. There is no need to emulate these services.

The algorithm gives an important hint of which ports are not correctly configured on the low interaction environment. It also provides a priority list of these services the emulation of which should be improved as fast as possible. The result confirms that most of the missing information comes from the Microsoft

Preliminaries :

FOR the two Environments H_1 and H_2:
FOR each Virtual Machine M_j and each associated port $p_{j,k}$:

Gather the list of Clusters $C_{l,k}$ corresponding to attacks on Virtual Machine M_j against at least port $p_{j,k}$
Be N the total number of IP Sources having targeted Virtual machine M_j
Be η the threshold to compare interactions between environments. $\eta = 0.7$
FOR each Cluster $C_{l,k}$
Compute the number n_l of Sources belonging to Cluster $C_{l,k}$
Compute P_l, the total number of exchanged packets between Sources belonging to Cluster $C_{l,k}$
Compute the *frequency* of Cluster $C_{l,k}$ as

$$f_l = \frac{n_l}{N}$$

Interaction Estimation:

The interaction estimation is for H_1

$$I(H_1) = \sum_{l \geq 1} P_l . f_l$$

The interaction estimation is for H_2

$$I(H_2) = \sum_{m \geq 1} P_m . f_m$$

Analysis:

IF $\frac{I(H_2)}{I(H_1)} \leq \eta$
The current implementation on port $p_{j,k}$ for Virtual Machine M_j in H_2 is not correct
The Interaction on this port is not satisfactory. The associated script should be enhanced.

Table 4. Comparing Interactions between H_1 and H_2

services. To conclude, this algorithm highlights the important complementarities that can be obtained by using both a high interaction and a low interaction honeypot.

5 Leurre.com Project

We have presented in previous publications some experiments based on a high interaction honeypot [13, 34]. These experiments have shown 1) that most of the attacks are caused by a small number of attack tools and that some very stable processes occur in the wild, and 2) that some processes have not been noticed by more global observations from darknets and telescopes. Thus it is worth deploying local sensors to complement the existing approaches.

The major objective consists in getting statistical information from the attacks. Therefore, low interaction honeypots represent a suitable solution. Indeed, we only want to observe the first attack steps in order to get a better understanding of current malicious activities. This paper provides another strong motivation, as it shows that low interaction honeypots brings as much information as high interaction ones when it comes down to global statistics on the attacks. In addition, some regular comparisons between the two types of environments (the high interaction environment being the etalon system) lead to an optimization of the low interaction configuration.

Leurre.com project aims at disseminating such platforms everywhere thanks to motivated partners, on a voluntary basis. Partners are invited to join this project and install a platform on their own. We take care of the installation by furnishing the platform image and configuration files. Thus, the installation process is automatic. In exchange, we give the partners access to the database and its enriched information [6]. We are also developing a dedicated web to make research faster and more efficient. The project has started triggering interest from many organizations, whether academic, industrial or governmental. We hope the number of partners will keep on increasing in the near future.

6 Conclusion

This paper presents a very important contribution to the Leurre.com project. Indeed, it shows on one hand that high interaction honeypots are somehow superfluous in the context of large-scale deployment of sensors, since global statistics remain very similar. On the other hand, it shows that they are vital for controlling the configuration relevance of low interaction honeypots. This leads to the conclusion that complementarities between high and low interaction honeypots can increase the accuracy of information collected by simple environments deployed in different places. Besides, this comparison has led to an interesting analysis of collected data. First, it allows identifying very specific attacks and weird phenomena, as has been shown through some examples. Second, it highlights the need to take into account packet losses in the analysis of malicious data. Otherwise, this can lead to misled conclusions.

Last but not least, we hope this paper will be an incitement for other partners to join the open project Leurre.com that we are deploying.

Acknowledgments

The authors wish to thank Prof. M. Dacier and V.H. Pham for their helpful comments.

References

1. CAIDA, the Cooperative Association for Internet Data Analysis. Internet: `http://www.caida.org/`, 2005.
2. D. Moore, G. Voelker, and S. Savage. Infering internet denial-of-service activity. In *The USENIX Security Symposium*, August 2001.
3. D. Song, R. Malan, and R. Stone. A global snapshot of internet worm activity. Technical report. URL:http://research.arbor.net/downloads/snapshot_worm_activity.pdf.

[6] A Non-Disclosure Agreement is signed to protect the confidentiality of the names of the partners

4. B. Gemberling C. Morrow. How to allow your customers to blackhole their own traffic. URL:http://www.secsup.org/CustomerBlackhole/.
5. Team Cymru: The Darknet Project. Internet: `http://www.cymru.com/Darknet/`, 2004.
6. E. Cooke, M. Bailey, Z.M. Mao, D. Watson, F. Jahanian, and D. McPherson. Toward understanding distributed blackhole placement. In *Proceedings of the Recent Advances of Intrusion Detection RAID'04*, September 2004.
7. The SANS Institute Internet Storm Center. The trusted source for computer security trainind, certification and research. URL:http://isc.sans.org.
8. V. Yegneswaran, P. Barford, and S. Jha. Global intrusion detection in the domino overlay system. 2004.
9. DShield Distributed Intrusion Detection System. URL:http://www.dshield.org.
10. myNetWatchman. Network intrusion detection and reporting. URL:http://www.mynetwatchman.com.
11. Cisco Systems. Netflow Services and Applications (1999).
12. M. Dacier, F. Pouget, and H. Debar. Attack processes found on the internet. In *NATO Symposium IST-041/RSY-013*, April 2004.
13. F. Pouget and M. Dacier. Honeypot-based forensics. In *AusCERT Asia Pacific Information Technology Security Conference 2004 (AusCERT2004)*, May 2004.
14. F. Pouget, M. Dacier, and V.H. Pham. Leurre.com: On the advantages of deploying a large scale distributed honeypot platform. In *E-Crime and Computer Evidence Conference(ECCE 2005)*, March 2005.
15. L. Spitzner. *Honeypots: Tracking Hackers*. Addison-Wesley, 2002.
16. M. Dacier, F. Pouget, and H. Debar. Honeypots, a practical mean to validate malicious fault assumptions. In *The 10th Pacific Ream Dependable Computing Conference (PRDC04)*, February 2004.
17. VMWare Corporation. User' s manual. version 4.1. URL:http://www.vmware.com.
18. TCPDump utility. URL:http://www.tcpdump.org.
19. The Honeynet Project. Know Your Enemy: GenII Honeynets, 2003. `http://www.honeynet.org/papers/gen2/`.
20. honeyd Homepage. Internet: `http://honeyd.org/`, 2004.
21. F. Pouget, M. Dacier, and H. Debar. Honeynets: Foundations for the development of early warning systems. 2005. Publisher Springler-Verlag, LNCS, NATO ARW Series.
22. Stanford Linear Accelerator Center. Tutorial on internet monitoring and pinger, 2001. URL: http://www.slac.stanford.edu/comp/net/wan-mon/tutorial.html.
23. Internet Traffic Report, 2005. URL: http://www.internettrafficreport.com/main.htm.
24. The AdvanceSCAN advscan utility, 2005. URL: http://advancemame.sourceforge.net/doc-advscan.html.
25. Symantec Security Response. W32-sasser.worm, 2004. URL: http://securityresponse.symantec.com/ avcenter/venc/data/w32.sasser.worm.html.
26. LURHQ. Dabber worm analysis, 2004. URL: http://www.lurhq.com/dabber.html.
27. p0f: Passive OS Fingerprinting Tool. Internet: `http://lcamtuf.coredump.cx/p0f.shtml`, 2004.
28. MaxMind: Geolocation and Credit Card Fraud Detection. Internet: `http://www.maxmind.com`, 2004.
29. SOPHOS. Sophos virus analysis: W32/agobot-pq, 2004. URL: http://www.sophos.com.au/virusinfo/analyses/w32agobotpq.html.
30. 5000 spike? Internet: `http://lists.sans.org/pipermail/list/2004-May/048192.html`, 2004.

31. TCP port 5000 syn increasing. Internet: http://seclists.org/lists/incidents/2004/May/0074.html, 2004.
32. Security Port Scanner, Trojan Port List: ICKiller. Internet: http://www.glocksoft.com/trojan_list/ICKiller.htm, 2005.
33. 2003 UPnP Exploit. Internet: http://www.packetstormsecurity.org/0112-exploits/XPloit.c, 2003.
34. Fabien Pouget and Marc Dacier. Honeypot-based forensics. In George Mohay, Andrew Clark, and Kathryn Kerr, editors, *Proceedings of AusCERT Asia Pacific Information Technology Security Conference 2004*, pages 1–15, 2004.