Understanding Threats: a Prerequisite to Enhance Survivability of Computing Systems

F. Pouget, M. Dacier, V.H. Pham
Institut Eurécom
B.P. 193, 06904 Sophia Antipolis, FRANCE
Email: {pouget, dacier, pham}@eurecom.fr

Abstract

This paper aims at showing the usefulness of simple honeypots to obtain data that can be used to derive analytical models of the attack processes present on the Internet. Built upon an environment which has been deployed for 18 months, we provide figures and analyses that enable us to better understand how attacks are carried out in the wild. Key contributions of this paper include a critical review of geographical information provided by NetGeo, a study of the aftermath of the Deloder worm and an in-depth analysis of the interaction between the populations of compromised machines devoted to scan the Internet and the ones in charge of actually running the attacks.

1. Introduction

The mere existence of the WORM Symposium indicates that Internet-wide infectious epidemics have emerged as one of the leading threats to information security and service availability. Important contributions have been made in the past that have proposed propagation models [1, 2, 3] or that have analyzed, usually by reverse engineering specific worms, their modus operandi [4, 5, 6, 7]. A few initiatives have been taken to monitor real world data related to worms and attacks propagation. The Internet telescopes and the DShield web site are among them. These approaches are extremely valuable but we will show in this paper that it is also worth using much simpler mechanisms, namely low interaction honeypots, to complement the type of information they provide.

In this paper, we will show the usefulness of the data collected by simple honeypots to formulate and validate assumptions regarding the propagation not only of worms but also of other types of malicious tools. Thanks to the data gathered in an environment which has been deployed for more than 18 months, we will give a few examples of the kind of findings that can be derived from this data set. We also want to point out that the collection of systematic data on which various detection and reaction mechanisms can be tested is a prerequisite for the evaluation of novel techniques, and thus for enhancing survivability of IP networks.

The long term goal of our research is to deploy similar setups in many different places to see if identified threats are identical on a worldwide basis or if, on the contrary, specificities exist. Results presented in this paper justify such a deployment by the richness of information that can be obtained. The simplicity of the setup makes it easy to deploy it at almost no cost in a large number of places. The results presented here are based on a VMWare environment which runs on a recent machine with at least one GB of memory [8]. This setup can not be replicated at no cost. However, as our results indicate that most attacks do not make use of known os fingerprinting techniques, we can, with a very low probability of introducing a bias in our experiments, replace this expensive platform by another one, based on honeyd which is easier to fingerprint but which runs on an old PC equipped with only 256 MBytes of memory [9]. We have deployed such an environment, in parallel to the VMWare one, to successfully confirm this assumption. In this paper though, as we have accumulated a much longer period of data with the VMWare platform, we present the results obtained with that one.

This paper is a follow up to three others we have published in the past [10, 11, 12]. Earlier publications have used a 10 months data set (March 2003 until December 2003) while this one is based on a 16 months data set (February 2003 until May 2004). With respect to previous work, we present four novel key contributions, namely:

1. The most recent months of observation confirm the findings discussed in previous publications. This long term stability is a very strong argument in favor of those who try to build analytical models of the Internet threats.

2. Previous results have presented Australia as the main source of attacks. We found out that this was an artifact caused by the tool chosen to obtain the geographical information, namely NetGeo [13]. Revised results obtained by means of another tool, MaxMind, are proposed.

3. We investigate the aftermath of the Deloder worm which appears to be very atypical and which, as far as we know, has never been discussed publicly so far. This analysis opens some avenues for further investigation.

4. We present an experiment carried out to confirm the assumptions made in our previous publications: two distinct sets of compromised machines are used to run attacks. The first set is in charge of scanning the net without attacking machines while the second set uses the information provided by the first one to run the attacks. The results of this recent experiment are discussed.

The structure of the paper is as follows. Section 2 presents two other solutions (Internet telescopes and DShield) to collect data about ongoing attacks and discuss the added benefits of our approach. Section 3 briefly presents our own environment and summarizes previous results. Section 4 presents the four novel contributions of this paper. Section 5 concludes the paper.

2. State of the Art

For the sake of conciseness, we refer the reader, interested in a complete treatment of the state of the art concerning honeypots and data collection, to our previous publications [10, 11, 12]. Still, we briefly cover here below two major ongoing initiatives related to the collection of data related to real world attacks: Dshield and the Internet telescopes.

2.1. DShield Project

The main idea of the DShield project [14] is to gather in a central place logs from a large number of firewalls. A freely available web site offers various representations of the database which can also be queried thanks to a user friendly interface. DShield operates in association with the SysAdmin, Audit, Network, Security (SANS) Institute which hosts the Internet Storm Center and a similar web site [15]. Another noticeable project is myNetWatchman (see [16]), a free service which aggregates firewall log records, backtraces the activity to its source (if possible) and automatically sends escalation emails to the responsible party (ISPs for instance). However, all these projects present similar restrictions that are explained below, taking DShield as a concrete, illustrative, example.

These approaches deliver trends of attacks occurring in the Internet at large but our experience shows that these macroscopic values can hide important local differences. Therefore, these data must be used with care. As an example of local vs. global differences, we present in Table 1 two different views of the 10 most attacked ports in France during the first week of June 2004, by decreasing order of appearance. The first column shows data obtained on the DShield site while the second one is based on what our honeypots have observed. This leads to the two following comments:

1. Both environments consider port 445 as the top 1 attacked port.

2. They only have 6 out 10 ports in common and the ports that appear in both lists are ranked very differently. This highlights the fact that DShield only provides global trends that can be different from one place to another, depending on many factors (geography, network type, domain name, etc) that remain to be identified and quantified.

3. Firewalls logs miss some important information that are needed to identify attack tools. As explained in [14], different tools target the same ports or even the same sequences of ports. Furthermore, some ports will only be targeted if some others are open[1]. This explains the difference observed in the table for, e.g., the port 4444 which is only scanned by the Blaster worm if port 135 on the same machine is open. Since that port is closed on most firewalls, Dshield records do not show many hits against port 4444. This is different for our honeypots where port 135 is open and where, as a consequence,

[1] See Section 5.3 of [17] for more details on Blaster Worm attacks. The infection always follows the same general pattern: a small set of attack packets obtain initial results, and further network traffic follows, either from the egg development, or from subsequent scans.

Dshield trends	Local trends
445	445
135	139
139	137
1433	135
9898	4444
3127	1026
1434	1027
5554	1433
1025	4899
137	9898

Table 1: Dshield *Port trends* compared to our local observations: ports are listed in decreasing order

many requests are sent to port 4444 as well. This highlights the fact that, in order to analyze data, the collecting environment must be precisely defined and tuned. Collecting logs from a variety of firewalls which are configured in many different ways, in very diverse environments, limits the analysis to very wide spread phenomena, in the best case, and refrains us from seeing others in the worst case.

2.2. Internet Telescopes

Another approach concerns *large telescopes*. They consist basically of a large piece of globally announced IPv4 addresses (most of them are unused) and a dedicated engine that monitor all traffic to these addresses. The most well known one is developed by the CAIDA Project (see [18] for a good application of telescopes on the analysis of the Witty worm). This project was the very first one, to our knowledge, to start investigating rigorously some of the threats found on the Internet. Their seminal work on the widespread usage of DDoS attacks [19] was a great source of inspiration to design our own approach.

This telescope, and others such as the one described in [20], provide very large amount of information. These data sets are very interesting to analyze worms propagation [5]. However, this large volume forbids them from maintaining detailed information about each packet (such as the payload, flags, sequence numbers, etc.). Unfortunately, such information might be important to differentiate traces due to different tools. Also, the generalization of the results obtained thanks to a given telescope is subject to controversy. How can we be sure that the phenomena observed on a given set of addresses are representative of those happening elsewhere? Since the trends provided by DShield indicate that differences exist between continents and countries, one should deploy telescopes in many different places in the world to come up with a good representation of the ongoing attacks. This is clearly not feasible due to the complexity and cost of such telescopes.

As we can see, both types of approaches have merits and drawbacks. The setup we are working on aims at complementing the data already available. By having a simple honeypot setup deployed in many places in the world, we can provide some refined analysis of the attacks observed and, hopefully, correlate or provide explanations for data sets collected by the other projects.

In the following, we detail the kind of results that can be derived from a single setup to justify the usefulness of deploying similar ones, yet not using VMWare anymore, on a large scale.

3. Experimental Setup

3.1. The Observation Platform

Our environmental setup consists in a virtual network built on top of VMWare [8] to which three virtual machines, or guests in the VMWare terminology, are connected (a Windows 98 workstation, a Window NT server and a Linux RedHat 7.3 server). It is very important to understand that the setup is such that these machines can only be partially compromised. They can accept certain connections on open ports but they can not, because of a firewall, initiate connections to the outside world. Furthermore, as they are built on non-persistent disks [8], changes are lost when virtual machines are powered off or reset. We do reboot the machines very frequently, almost on a daily basis, to clean them from any infection that they could have had. The three machines are attached to a virtual Ethernet switch. ARP spoofing is implemented so that they can be reached from the outside world. A fourth virtual machine is created to collect data in the virtual network. It is also attached to the virtual switch[2] and tcpdump is used as a packet collector [21]. In addition, a dedicated firewall controls the access to all machines (iptables [22]) and tripwire regularly checks integrity of the host files. More detailed information about the setup can be found in [10].

[2] what VMWare calls a switch is in fact a hub

3.2. Data Collection and Storage

In the following, we will make use of the expressions *attack source* and *ports sequence* which we define as follows:

- *Attack Source* : an IP address that targets our environment within one day. The same IP address seen in two different days counts for two different *attack sources* (see [12] for more on this).
- *Ports Sequence*: an ordered list of ports targeted by an attack source. For instance, if source A sends requests on port 80 (HTTP), and then on ports 8080 (HTTP Alternate) and 1080 (SOCKS), the associated *ports sequence* will be {80;8080;1080}.

All network packets are stored in a database and enriched with geographical information of the sources as well as their OS fingerprints. The database is made of several tables that are optimized to efficiently handle complicated queries. In other words, the computing cost of querying the database is marginally influenced by the number of logs in it. Of course, the price to pay is increased disk and memory space to handle redundant meta information (keys, counters, etc.) in several tables. We report the interested reader to [12] for more information on the design of the database.

3.3. Attack Tools identification

In [12], we have presented a clustering technique which enables us to identify clusters of traces caused by different attack tools.
The different steps are basically:

1. Group *attack sources* per *ports sequences*
2. For each group, gather important parameters in a DB table
3. Some parameters are generalized by means of a hierarchy algorithm
4. Clusters are extracted thanks to data mining techniques (association rules)
5. Validation of clusters consistency is made thanks to the distance phrase algorithm (Levenshtein); inconsistent clusters are splitted and checked again until they satisfy the validation criteria.
6. Output: one cluster corresponds to one attack tool
7. Find the name of the tool associated to the cluster (tools fingerprinting)

The algorithm used lies beyond the scope of this paper. In the following, when we use the notion of clusters, we refer to the results obtained by applying this algorithm to our database. It is important to keep in mind that each cluster is due to only one attack tool but also that one given attack tool can generate different traces which could be grouped into several clusters. In other words, the relation that links clusters to attack tools is a N to 1 relationship.

4. New Results

4.1. The big picture

In the previous publications, we have shown and discussed early results obtained by querying our database and applying the clustering algorithm on a database containing 10 months of data. With 50% more data, we are still able to confirm all results obtained so far. This surprising stability over a long period of time is extremely encouraging for those who wish to obtain macroscopic analytical models of the Internet threats. Of course, as we will see here after, things are much more complicated and less stable when one tries to dig into the data to better understand the root causes of these stable processes.

To highlight this regularity, we provide a few key figures obtained over this 16 months period and we refer the reader to our previous work ([11, 10, 12]) to verify by himself how stables these numbers are:

- 4102635 packets from/to our virtual machines have been stored in the database.
- 28722 attack sources have been observed.
- Only 205 different ports have been probed.
- Only 604 different ports sequences have been observed
- All attack sources, but a few very rare exceptions, have sent requests during less than 30 seconds.
- In 69% of the cases, attacks sources have sent requests to the 3 honeypots, always in the same order.
- In 6%, attack sources have contacted only two out of the three machines.

- In 25% they have focused on only one of the three honeypots.

- 99.5% of the IP addresses have been seen in only one day.

Last but not least, Figure 1 shows the amount of attack sources observed per country per month. The important thing to note here is the impressive uniformity of the distribution. A few countries systematically compose the top 7 of the attacking countries: China, the USA, Japan, Taiwan, France, Germany and Korea.

Though, it is important to stress that we do note an important discrepancy during the last two months for the attacks originating from China and the USA. A dramatic increase is observed for these data sets which cannot be easily explained by the appearance of a single worm. Furthermore, a deeper analysis reveals that a large number of these attacks target only one of our three honeypots. This is a very new pattern of attack that we had not observed before. On one hand, this seems to contradict our claim that regular patterns exist and that analytical models could therefore be proposed for the existing threats. On the other hand, we have been able to identify this new type of attack pattern because it did not fall within the regular pattern. Thus, despite this new unstable activity, we maintain that data sets obtained by means of simple honeypots are amenable for analytical modeling of the attack processes present on the Internet.

4.2. NetGeo vs. MaxMind

We discussed in [10] some information on attacking machines, and more specifically on their geographical location. In that paper, we have presented results obtained thanks to the NetGeo utility, developed in the context of the CAIDA project [13]. NetGeo is made of a database and a collection of perl scripts that map IP addresses to geographical locations. This utility is open-source and has been applied in several research papers among which [23, 24, 25, 26]. In our case, running NetGeo scripts on our data base indicated that 70% of the attacks originated from only three countries: Australia (27.7%), the USA (21.8%) and the Netherlands (21.1%). This was quite surprising but very consistent month after month. As a result of these first publications and discussions with our peers, we have decided to use another system that also provide geographical location, namely MaxMind [27], to double check the results obtained with NetGeo.

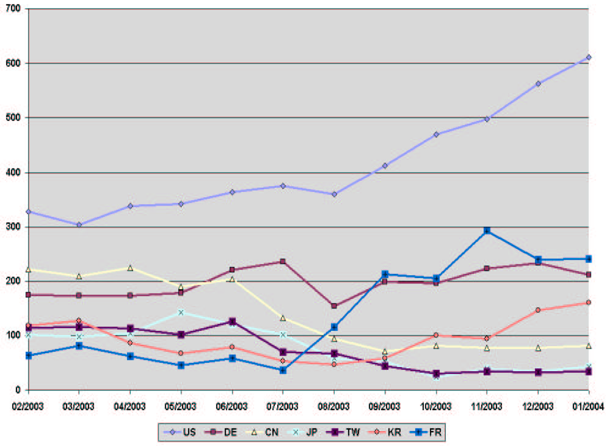

Figure 1: # of attack sources per country and per month

Results are presented in Figure 2 for the top 4 attacking countries according to NetGeo: Australia, Netherlands, USA and 'unknown'. For all IP addresses that NetGeo indicated to us as belonging to one of these countries we have checked the results returned by MaxMind. The Figure must be read as follows: In the upper left pie chart, one can see that 29% of the IP addresses that NetGeo has located in Australia are located in China by MaxMind.

If we consider the results of MaxMind instead of those of NetGeo, our top 4 countries are now replaced by a group of 7 countries, namely: USA, Germany, China, France, Japan, Taiwan and Korea (see table 2). This seems to better fit with the expectations of the security community. For Australia and the Netherlands, MaxMind gives completely different results than NetGeo. According to MaxMind, almost no attacks are originating from Australia anymore but they, instead, seem to come from four other countries: China, Japan, Taiwan and South Korea. Moreover, the Netherlands are replaced by two main countries: Germany and France. It is also worth pointing out that, in our data set, both tools disagree for 61% of all IP addresses. Also, even if they agree, for the total amount of IP addresses located in the USA, 21.8% vs. 22.3%, there are important differences regarding the specific state within the USA where they locate the IP addresses.

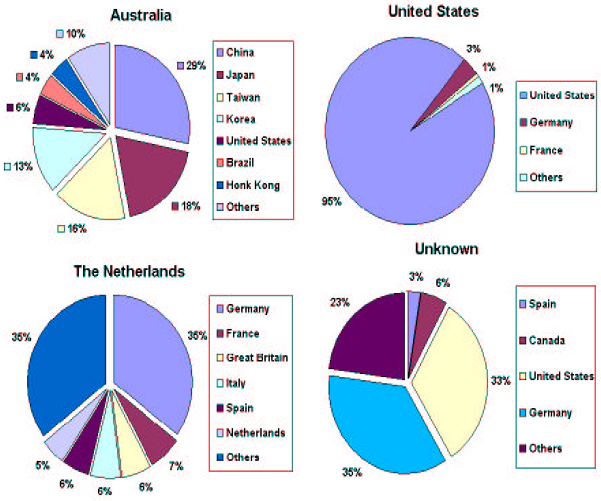

Figure 2: top 4 attacking countries according to NetGeo compared to MaxMind results

The explanation of these differences lies in the precise understanding of the information returned by both tools. The structure of Internet is based on inter-connected autonomous systems (ASs), where each AS is administrated by a single authority with its own choice of routing protocols, configuration, and policies. For a few large, geographically dispersed ASs, NetGeo returns the geographical location of the authority of the AS to which belongs the IP instead of the real location of the IP itself. This is what happens for RIPE (Netherlands) and APNIC (Australia). However, this is not what NetGeo is supposed to return, as described in [13]. At this time of writing, this apparent erroneous behavior is not clear to us and might lead to misleading information[3]. It is perhaps due to some simple coding error that the people in charge of the NetGeo scripts could hopefully fix relatively easily. On the other hand, according to one of the MaxMind representatives, their system uses user-entered location data aggregated with 'whois' data to better estimate the locations of the end users. These data are obtained thanks to data exchange agreements with some of their customers.

Countries	NetGeo	MaxMind
Australia	27.7	0.6
China	1.5	10.1
France	4.1	6.1
Germany	1.9	12.0
Great Britain	0.0	2.5
Italia	1.1	2.7
Japan	0.2	5.7
Netherlands	21.1	1.1
South Korea	0.7	4.8
Spain	0.5	2.0
Taiwan	0.7	5.5
US	21.8	22.3
Others	18.7	24.6

Table 2: NetGeo vs. MaxMind against our honeypot data (% of observed attack sources per country)

The most important consequence of using the information returned by MaxMind instead of NetGeo is that Australia, which was, by far, our top attacking country now disappears from the map.

4.3. Deloder worm and aftermath

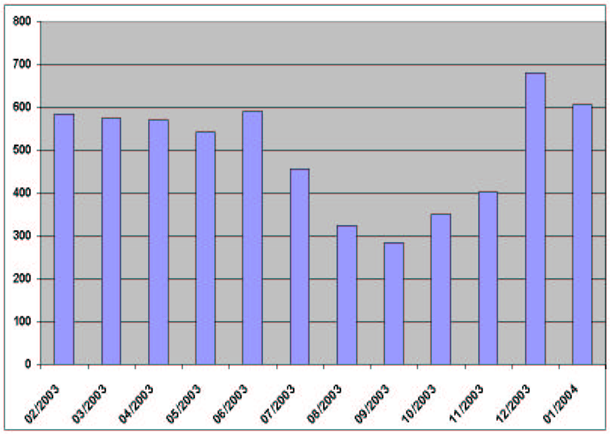

Figure 3: Australian attacks observed each month, based on the NetGeo utility

One of the questions left unanswered in [10] was the apparent decrease of attacks coming from Australia around July 2003. This is represented in Fig-

[3] 06/28/2004, Bugtraq mailing list: information on the *Scob Trojan* indicates that IP addresses of infected machines are mostly located in the USA and Australia [28]. The author indicates that the information is based on APNIC; it is quite likely that here too Australia is blamed for no good reason ...

ure 3. Figure 4 represents the same curves, based on MaxMind data, for all addresses identified as Australian ones by NetGeo.

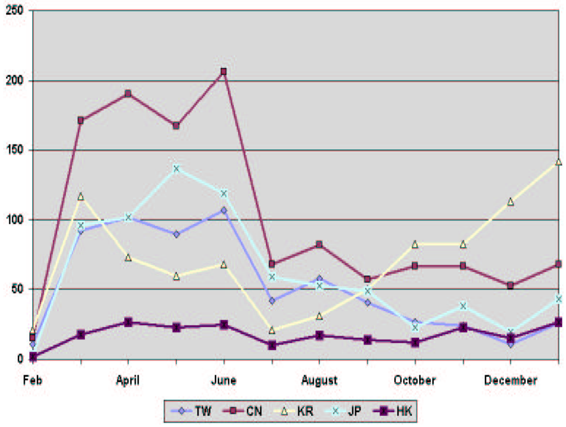

Figure 4: Addresses identified as Australian by Netgeo: Repartition by country over months

We notice that attacks are coming from four Asian countries, respectively China, Japan, Taiwan and South Korea. We were expecting to find the explanation of the 'Australian' decrease in a change due to one of these countries, as a consequence, for instance to some increased control of traffic flowing out of the country. However, Figure 4 shows that the decrease exists for all countries in mid-2003. In addition, we have applied the clustering algorithm presented in [12] which shows that the decrease is mainly due to a few specific clusters, all of them involving ports sequence {445}. The tool associated to these clusters has been identified as being the worm Deloder [6, 7]. In Figure 5, we represent, per month and per country, the amount of attack sources compromised by the Deloder worm that have tried to propagate to our honeypots.

Each represented cluster can be linked to one of its variants. This worm, which spreads over Windows 2K/XP machines, attempts to copy and execute itself on remote systems, via accessible network shares. It tries to connect to the IPC$ share[4] and uses specific passwords. According to antivirus websites like [6, 7], this worm, which was initially detected in March 2003, originates from Asia, and more especially China. This is consistent with our curves.

The surprising fact comes from the rapid decrease of its propagation around July 2003. [5] mentions that the shutdown of CodeRedII was preprogrammed for October 1, 2001. [29] mentions that Welchia worm will self terminate on June 1st, 2004, or after running 120 days. A similar mechanism could have been used for Deloder but, as far as we know, no one has ever made mention of it publicly. In the absence of such a mechanism, it is worth trying to imagine the reasons for such a sudden death. We have come with the following scenario:

1. Deloder is still active but our virtual machines are not scanned anymore, for some unknown reasons. Statistically speaking, this seems unlikely and should be validated by means of other similar platforms.

2. All machines have been patched. Deloder has been eradicated. This is another unlikely scenario since Deloder targets a large number of platforms, many of them being personal computers which will probably never be patched. Newer successful worms targeting the same port (eg Sasser, Welchia, the Korgo family, etc.) tend to confirm this.

3. Deloder bots are listening on IRC channels for commands to run attacks. One of these commands might have told them to stop the propagation process. In this case, the Deloder worm is not visible anymore but its botnet remains as dangerous as before.

At this point in time, unless if a pre programmed shutdown is included in the Deloder worm code, we consider the third option as the most plausible one. A definitive answer to that question could be brought forward by someone who has access to the Deloder worm code, which we have not. If our assumption holds true, this would imply that worms writers have developed a new strategy. Instead of continuously trying to compromise more machines, they have decided to enter into a silent mode when the size of their botnets is sufficient. By doing so, they dramatically reduce the likelihood of seeing an in-depth study of their worm being done as invisible worms are definitely less interesting to the security community than virulent ones. The bottom line of our findings is that such an in-depth analysis of that

[4] or ADMIN$, C$, E$ shares depending on the Deloder variants

worm is probably worse being done if it has not been done yet. Sleeping worms might actually be more sophisticated and nefarious than active ones.

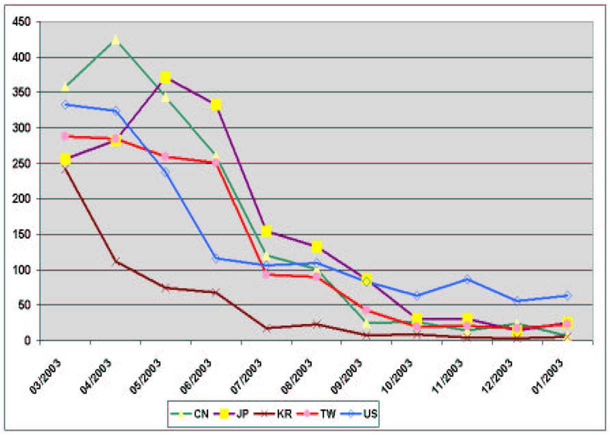

Figure 5: Deloder Activity (# asociated attack sources) observed over months

4.4. Scanners and Attackers

4.4.1. Preliminary Note

In [11], we have shown that approximately 70% of the observed IP addresses have sent requests to our 3 honeypots while 25% have sent packets to only one honeypot with an unexpected success rate. Apart from some identified exceptions (backscatters essentially), all IPs belonging to this category have sent packets to ports that were actually open! Statistically speaking, it is very unlikely to see this phenomenon. As a consequence, we have postulated that among the 70% of machines talking to our three honeypots, the role of some of them was restricted to scan our machines without trying any kind of attacks against them. The result of this information gathering process was then later used by machines that we had never seen before, enabling them to hit systematically our machines on open ports.

In order to validate that claim, to have a better understanding of the ratio of machines involved in the scanning-only process and to try to figure out how many populations of scanners we are facing, we have designed an experiment the results of which are presented here below.

4.4.2. Experiment

Starting mid October 2003, we have opened port 1433 on our linux virtual machine[5], which is the traditional port for the MS SQL service on Windows machines. Before that day, we had never observed a machine talking to that sole linux box sending packets to that port. We were interested in verifying that the situation would change once the 'hypothetical' scanners would have figured out that this port was now open.

The results we have obtained are given in Figure 6. It represents the number of sources that targeted the sole port 1433 on that unique machine. We note that they are all new IPs. None had been observed previously. Point 1 in the Figure corresponds to the date we opened this port. Point 2 shows the first explicit attack observed on that sole machine. It reveals that it takes around 15 days for such a precise attack to happen. Also, it is worth pointing out that port 1433 has been opened on a Linux machine while this port is normally used by a Microsoft service. Thus, this indicates that scanning machines provide some basic information regarding the opened ports but fail in fingerprinting the OS of the machine they have just probed. This might be too costly in regards of the probability of having a Windows port opened on a Linux box. Moreover, the increasing number of specific attacks show that we are facing different attackers communities. Otherwise, they would not try to attack the same machine without success again and again. Clearly, these attackers did not share the experience of their failures.

At the end of December 2003, the number of attacks still increases but less abruptly. Thus, we have decided to close this port on january 12th 2004. This is indicated by the point 3 in Figure 6. We note then a very fast decrease of such observed attacks. They almost totally disappear in February and there is none of them in March. This simply means that some scanners have updated the *shared information*, and it takes less than two weeks for the attackers' community to update their information and to stop the attacks.

In summary, this experiment allows deducing four major results:

1. The claim about scanners-only machines is validated.

[5] We started a daemon listening on that port. There is no service attached to it

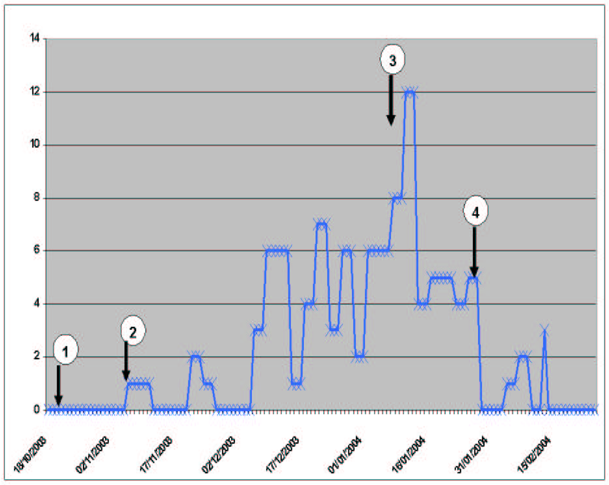

Figure 6: # attack sources having targeted port 1433 only on the sole Linux virtual machine

2. The *shared information* is not as good as it could be. Many tools are now implementing OS fingerprinting techniques (actively [30] or passively [31, 32, 9]): either scanners having targeted our machines are not using them, or the attackers communities do not check this information before launching their attacks. The second hypothesis seems to be the more correct. Indeed, we observe many scans on port 1433 following other Windows ports like 139, 445 or 135. As a consequence, such scanners know whether the machine is a Windows station or not.

3. The previous point justifies the usage of simpler honeypots than VMWare ones to replicate our environment. Indeed, if attackers do not bother distinguishing between Windows and Linux boxes, it is quite likely that are not more interested in detecting honeypots. We had initially build our environment on a VMWare platform to avoid introducing any bias in the data collection process. Now armed with these results, we have good reasons to shift to a cheaper and simpler environment, despite its limitations with respect to fingerprinting. We will, though, maintain a few VMWare machines to verify that observations obtained in both environments keep being consistent.

4. This experiment gives a rough estimate of the number of communities of attackers. Indeed, we can assume each attack is independent and comes from a different person or group of persons. It seems unlikely that someone would repeatedly try an unsuccessful attack. In our case, 76 independent communities have attacked our machines without sharing any piece of information. We are in the process of refining these numbers thanks to a larger number of platforms over a longer period of time.

Having validated the four previous points, we can analyze a little bit further this phenomenon by using our clustering algorithm in order to determine potential scanners that have made this *shared information* available. We can reasonably assume that the information about open ports is maintained up to date by a single tool. In other words, the same scanning tool is responsible for having identified the ports as open and, later for having changed the information regarding that port when it found it closed again. With this assumption in mind, we observe the following facts:

1. From the date we opened the port (point 1 in Figure 6, October 20, 2003) to the date we observed the first explicit attack (point 2 in Figure 6 November 6th, 2003), the clustering algorithm shows that five different sets of machines have targeted port 1433 among others on our 3 honeypots.

2. We observe only two of these five clusters between the 'closing date' of the port (point 3) and the decrease of the attacks (point 4).

The scanning tool which shares information with attack comunities is quite likely one of these two clusters (if not both). Figure 7 gives the observation of these two clusters from October 2003 to March 2004. Scans grouped in the first cluster (Cluster 1: the upper curve) are observed frequently and regularly. Those in the second cluster (Cluster 2) appear, at the contrary, rather sporadically.

Instances of Cluster 1 are seen every day. Thus, if Cluster 1 is the one that contains the scans that have led to the following direct attacks, it is difficult to understand why it took two weeks to see the first attack launched. On the other hand, the first attack is observed less than 5 days after the first scan belonging to Cluster 2. Moreover, the port has been closed on January 12th (point 3) and the first Cluster 2 scan has been observed a dozen days later. The decrease in the number of attacks was noticeable a few days after that specific scan, at the end of January, as can be seen on Figure 6.

Cluster 2 seems thus to be a good candidate. This is confirmed by looking at packets corresponding to both clusters. Some packets associated to Cluster 1 contain a 42-byte data payload sent to our honeypot. In other words, these machines not only look for open ports but also try to send some data to them. On the other hand, packets associated to Cluster 2 are simple TCP SYN, half open scans.

Similar experiments on different platforms would enable us to determine more precisely the modus operandi of these scan-only machines but, so far, this refined analysis leads to the fifth and final result of this experiment:

5. Information gathered by a given scan is shared between several communities of attackers. Indeed, we do not see a one-to-one relationship between a scan-only IP and an attacker. On the contrary, we have more attacks than scans. 76 different but precise attacks have been performed from October 20th, 2003 to January 12th, 2004 (points 1 and 3 on the Figures respectively). Scans belonging to cluster 2 have only been observed five times for the same period. This tends to indicate that more than one community is using the information provided by a scan-only machine. Here to, we need more machines to get a better insight on the interactions between scanner-only and attackers. This is part of our ongoing work.

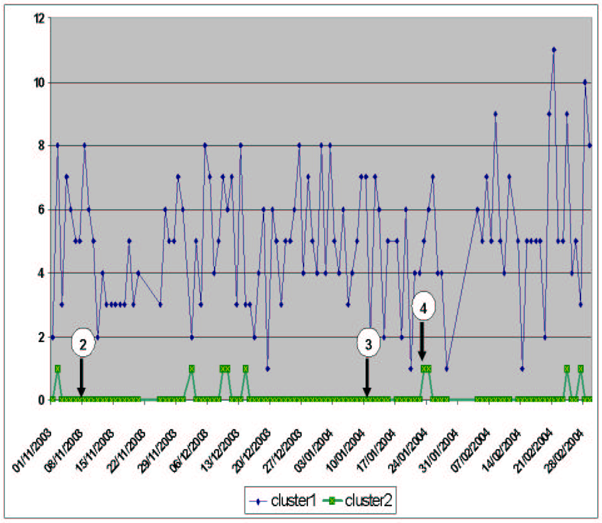

Figure 7: Clusters activity (#attack sources) per day

5. Conclusion

Honeypots are very promising sensors to monitor local attack processes and to complement current tools which look at the malicious activity at a higher level. These local observations bring different and relevant information that needs to be carefully analyzed. We have shown in this paper that they allow to get a better understanding of the attack processes. This knowledge is currently lacking but is necessary for the design of efficient security systems.

More precisely, we have managed in this paper to explain some weird phenomena, such as the Australian activity which was dominant at a first glance. We have also validated that machines are facing multiple independent communities of attackers, and that it is possible to estimate their number, as well as the kind of information they might exchange together. Finally, we have justified the usage of honeypots as an interesting platform to collect data in oder to build analytical models of Internet threats.

Some other questions arise and require more expertise. Numerous honeypot platforms placed in various places will help finding their answers. We are now deploying such honeypots and we hope this work will open new avenues for the ongoing work related to honeypots. We invite all teams interested in using our full data set for analytical purposes to contact us. We have defined a simple model to share our data : we grant access to all partners that accept to put one honeypot, which we will configure remotely, in their premises. At the time of this writing, 12 environments are up and running, in various countries in Europe but also in two other continents. We expect more to join in the next few weeks.

References

[1] S. Staniford, V. Paxson, and N. Weaver. How to own the internet in your spare time. In *Proceedings of the 11th USENIX Security Symposium*, pages 149–167. USENIX Association, 2002.

[2] Z. Chen, L. Gao, and K. Kwiat. Modeling the spread of active worms. In *IEEE INFOCOM*, 2003.

[3] C.C. Zou, W. Gong, and D. Towsley. Worm propagation modeling and analysis under dynamic quarantine defense. In *ACM WORM 03*, October 2003.

[4] E. Spafford. An analyis of the internet worm. pages 446–468, September 1989.

[5] D. Moore, C. Shannon, G.M. Voelker, and S. Savage. Core-red a case study on the spread and victims of an internet worm. In *ACM/USENIX Internet Measurement Workshop*, November 2002.

[6] McAFee Security Antivirus. Virus profile: W32/deloder worm. URL:http://us.mcafee.com/virusInfo/.

[7] F-Secure Corporation. Deloder worm analysis. URL:http://www.f-secure.com.

[8] VMWare Corporation. User' s manual. version 4.1. URL:http://www.vmware.com.

[9] Honeyd Virtual Honeypot from N. Provos. URL:http://www.honeyd.org.

[10] M. Dacier, F. Pouget, and H. Debar. Attack processes found on the internet. In *NATO Symposium IST-041/RSY-013*, April 2004.

[11] 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.

[12] F. Pouget and M. Dacier. Honeypot-based forensics. In *AusCERT Asia Pacific Information Technology Security Conference 2004 (AusCERT2004)*, May 2004.

[13] CAIDA Project. Netgeo utility - the internet geographical database. URL:http://www.caida.org/tools/utilities/-netgeo/.

[14] DShield Distributed Intrusion Detection System. URL:http://www.dshield.org.

[15] The SANS Institute Internet Storm Center. The trusted source for computer security trainind, certification and research. URL:http://isc.sans.org.

[16] myNetWatchman. Network intrusion detection and reporting. URL:http://www.mynetwatchman.com.

[17] X. Qin, D. Dagon, G. Gu, W. Lee, M. Warfield, and P. Allor. Worm detection using local networks. In *Proceedings of the Recent Advances of Intrusion Detection RAID'04*, September 2004.

[18] CAIDA Project The UCSD Network Telescope. The spread of the witty worm. URL:http://www.caida.org/analysis/security/-witty/.

[19] D. Moore, G. Voelker, and S. Savage. Infering internet denial-of-service activity. In *The USENIX Security Symposium*, August 2001.

[20] K.E. Giles. On the spectral analysis of backscatter data. In *GMP - Hawai 2004*, 2004. URL:http://www.mts.jhu.edu/ priebe/FILES/-gmp_hawaii04.pdf.

[21] TCPDump utility. URL:http://www.tcpdump.org.

[22] IPTables Netfilter Project. URL:http://www.netfilter.org.

[23] A. Rosin. Measuring availability in peer-to-peer networks. September 2003. URL:http://www.wiwi.hu-berlin.de/ fis/p2pe/paper_A_Rosin.pdf.

[24] A. Zeitoun, C.N. Chuah, S. Bhattacharyya, and C. Diot. An as-level study of internet path delay characteristics. Technical report, 2003. URL:http://ipmon.sprint.com/pubs_trs/trs/RR03-ATL-051699-AS-delay.pdf.

[25] S.H. Yook, H. Jeong, and A.L. Barabasi. Modeling the internet's large scale topology. In *PNAS -vol.99*, October 2002. URL:http://www.nd.edu/ networks/PDF/Modeling.

[26] T.S. Eugene Ng and H. Zhang. Predicting internet network distance with coordinates-based approaches. In *INFOCOM 2002*, 2002. URL:http://www-2.cs.cmu.edu/ eugeneng/papers/INFOCOM02.pdf.

[27] MaxMind GeoIP Country Database Commercial Product. URL:http://www.maxmind.com/app/products.

[28] H. Hubbard. Scob infection statistics, etc..., June 2004. URL: http://www.securityfocus.incidents.

[29] Symantec. Symantec security response w32.welchia.worm, 2004. URL: http://response.symantec.com/avcentr/venc/-data/w32.welchia.b.worm.html.

[30] NMap utility from insecure.org. URL:http://www.insecure.org/nmap.

[31] p0f Passive Fingerprinting Tool. URL:http://lcamtuf.coredump.cx/p0f-beta.tgz.

[32] Disco Passive Fingerprinting Tool. URL:http://www.altmode.com/disco.