Learning More About Attack Patterns With Honeypots

Thorsten Holz

thorsten.holz@informatik.uni-mannheim.de

Abstract: *Honeypots are information system resources, whose value lies in unauthorized or illicit use of these resources. In this paper, we present a project that has established a world-wide distributed sensor system of honeypots. Within this system, each platform has the same configuration, thus allowing us to compare the collected data of each platform. And since all platforms send all logging data to a central database, this enables us to correlate all data and draw conclusions from it.*

Besides presenting the project, we show how the collected data can be used to learn more about attack patterns. In addition, we illustrate how we can learn more about root-causes of attacks, i.e., specific tools or techniques used by attackers.

1 Introduction

Today, almost all aspects of our life, e.g., mobile communication and finance, depend heavily on computer systems. Due to the growing pervasiveness of computers (for example eHomes or Personal Area Networks) and ubiquitous mobility of users and devices, this dependence is steadily increasing. Nevertheless, there are more and more security threats in communication networks: we are flooded with unsolicited bulk e-mail (UBE – "spam"), we have huge problems with viruses, worms and other malware, and crackers are often able to break into systems. Furthermore, Denial-of-Service (DoS) attacks, electronic fraud, and other abuses of communication systems show the downsides of the progressive interconnection. In the area of information security, researchers all over the world try to find ways to stop at least some of these threats.

An approach to learn more about attacks and attack patterns is based on the idea of electronic decoys, called *honeypots*. A honeypot is an information system resource whose value lies in unauthorized or illicit use of that resource. It allows us to learn more about attacks in communication networks. Honeypots can also be combined into networks of honeypots (*honeynets*) to learn more about the diverse proceeding of attackers. A detailed introduction to honeypots and honeynets can be found in [Pro05a, DGH04a, DGH04b].

With a classical honeynet, we can learn more about attacks against a single network and a single environment, respectively. So we get a rather local view of attacks and presumably miss "the big picture". To change this, we participate in a project entitled *leurre.com*. This project is a distributed approach in honeypot-based research and we will describe the architecture and some of the results we obtained in this section. More results have been published as [PH05] and more information about leurre.com can be found in [DPP05].

This paper is outlined as follows: Section 2 gives an overview of related work in the field. An overview of the network and system setup is given in Section 3 and we present several results in Section 4. Finally, we conclude this paper with Section 5.

2 Related Work

Today, many solutions exist to observe malicious traffic on a large-scale base, e.g., on the whole Internet. However, they often consist in monitoring a very large number of unused IP address spaces to monitor malicious activities. Several names have been used to describe this technique, such as *network telescopes* [Cai05, MVS01], *blackholes* [SMS01, CBM+04], *darknets* [Cym04], or *Internet Motion Sensor* (IMS) [BCJ+05]. All of these projects have the same approach: they use a large piece of globally announced IPv4 address space and passively monitor all incoming traffic. For example, the network telescope run by the University of California, San Diego, uses 2^{24} IP addresses. This is 1/256th of all IPv4 addresses. The telescope contains almost no legitimate hosts, so inbound traffic to nonexistent machines is always anomalous in some way, i.e., the principle of honeynets is also used in this context. By analyzing all packets, they are able to infer information about attackers. Since the network telescope contains approximately 1/256th of all IPv4 addresses, it receives roughly one out of every 256 packets sent by an Internet worm with an unbiased random number generator. Thus the monitoring of unexpected traffic yields a view of certain remote network events. This can for example be used to study the threats posed by Denial-of-Service attacks [MVS01].

Another approach is to passive measure live networks by centralizing and analyzing firewall logs or IDS alerts [Ins05a, YBJ04]. Especially the *Internet Storm Center* (ISC) / *DShield.org* [Ins05b, Ins05a] is a well-known project in this area. In this project, the collected data are simple packet filter information from different sources all around the world and no "high-level" data are included. Reports are published on a daily basis and include information about attack patterns and takes a closer look at unusual events. A report combines 8 – 20 million records per day with 200,000 – 400,000 source and 300,000 – 450,000 target IP addresses per day. The results are nevertheless only simple queries like "Most Attacked Port". Moreover, the data contain no detailed information about the source who has collected the packet. So a comparison of different attacks is not easily possible.

Coarse-grained interface counters and more fine-grained flow analysis tools such as *NetFlow*/*cflow* offer another readily available source of information. A *flow* is defined as IP traffic with the same source IP, destination IP, source port and destination port, since this quadruple can describe the IP traffic between two devices on the Internet. A router which is capable of flows will only output a flow record when it determines that the flow is finished, e.g., either by explicit connection shutdown or timeout. The flows are stored in a central database and can be analyzed from a high-level point of view. With this aggregation of data, it is often possible to draw conclusions about unusual events within a network.

Finally, another approach is similar to the one we use for our project: in the context of the project *eCSIRT.net*, several European Computer Security Incident Response Teams

(CSIRTs) set up a network of IDS sensors across Europe [GDK04]. This network collected data about attacks in a central database for further analysis and helped in vulnerability assessment. After the project ended, some teams decided to continue the then established sensor network across Europe, which provides information about network attacks since September 2003.

3 The Project Leurre.com

After having introduced several other approaches for network-wide monitoring, we can now start to describe the project we are involved in. Under the project name *leurre.com*, we have established a world-wide network of honeypots. The setup of this project is depicted in Figure 1 and will be described in the following paragraphs.

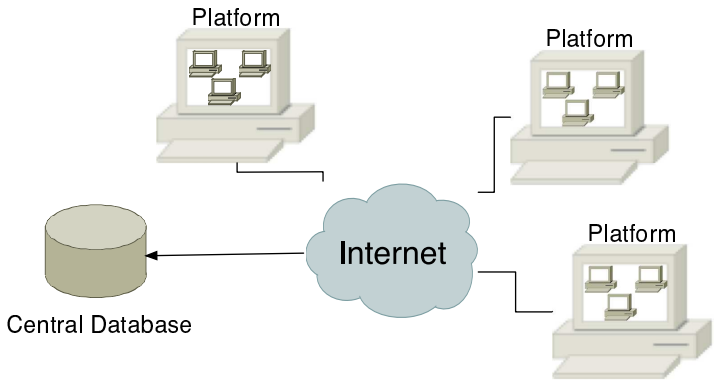

Figure 1: Setup of leurre.com

The setup is based on the honeypot software *honeyd* [Pro05b, Pro04]. *Honeyd* is a small daemon which creates virtual hosts on a network. It simulates the TCP/IP stack of arbitrary operating systems and can be configured to run arbitrary services. This tool enables a single computer system to claim multiple addresses by intercepting Address Resolution Protocol (ARP, [Plu82]) requests and redirecting them to *honeyd*.

Honeyd allows us to easily set up several virtual hosts on a network with several services. This is used to build an easy to deploy low-interaction honeynet. For this project, a bootable CD-ROM has been developed that does exactly this: each participant has to provide a common off-the-shelf (COTS) computer system. The system is booted with the provided CD-ROM and a floppy, which contains customizations for each platform. During the fully automated installation process, a low-interaction honeynet with three virtual hosts is set up. This platform P is the basic building block of the *leurre.com* project. Each platform consist of three virtual hosts h_{1-3}, with the following configuration:

- h_1 and h_2: Personality is "Microsoft Windows 2000 SP3" and several well-known TCP and UDP ports are configured as being open. These ports include FTP, Telnet, HTTP, and NetBIOS. Some of these ports also have scripts associated to them in order to actually simulate a service.
- h_3: Personality is "Linux Kernel 2.4.20" and more than five TCP ports are configured as being open. Besides common services like SSH, FTP, and HTTP, also several uncommon services like line printer spooler or caching service are simulated.

Each platform has the same configuration file, in order to ease the comparison of malicious activities between some of them. This helps to avoid the main drawback of projects like *DShield.org*: since we know that all platforms have the same configuration and thus simulate the same services, we can easily compare attacks against different platforms.

In August 2005, there are 32 platforms deployed all around the world. Platforms cover more than 16 countries on four continents. All of these platforms are located in different IP ranges. Thus we are currently able to cover some part of the whole Internet, although we have deployed only a limited amount of sensors with only three IP addresses each.

To learn more about attack patterns, we want to learn more about the actual root-cause of network traffic we receive. To help in that, we want to learn more about the *attack source*. An attack source (in short: *source*) is defined in the following way [PD04]:

> "*Attack Source*: it defines an IP address that targets our honeypot environment within one day. This time constraint is arbitrary and based only on our observation: so far, observed attacks have always been limited to short time periods (no more than 1 minute). Thus, if the same IP address is sending packets to one of our honeypots on the 12th of January and then on the 4th of February, we consider that they come from two distinct attack sources. In addition, this definition is motivated by the fact that some IP addresses are dynamically allocated and they change frequently, in terms of days [...]"

To learn more about the attack source, we use passive techniques for fingerprinting the remote operating system. Since the TCP/IP stack of all operating systems has some minor differences, it is possible to conclude which system the attacker is running if we just analyze the packet that we receive. With this completely passive analysis we do not generate any additional or unusual network traffic. Nevertheless, we are often able to give an educated guess of the remote operating system. For this purpose we use the tools *p0f* and *disco* which implement this technique.

To help in analyzing the attack source, we also use several tools to determine the geographical location of the source. Currently we use the tools *NetGeo* and *Maxmind GeoIP*.

With the help of these tools, we can enrich the collected data about an attack source with several other information, e.g., the remote operation system or the geographical location. This helps us to learn more about the actual attack. All data are collected in a central database which can be queried by all participants. This database is the central source for data analysis and allows everyone who contributes to the project to learn more about the

Platform	Number of sources	Average sources/day
P_1	309626	894.87
P_2	45407	574.77
P_3	76578	238.56
P_4	30528	160.67
P_5	14911	55.23
P_6	9080	39.65

Table 1: Number of unique source and average number of sources per day for different platforms

data collected at the other platforms. The database can be queried via a web interface that allows each participant to execute arbitrary SQL queries. Since some of the collected data are sensitive information (e.g., the IP addresses of all platforms), every participant has to sign a Non-Disclosure Agreement (NDA). This NDA helps to protect all sensitive data and ensures that all of this information stays private. Besides this, it allows everyone to use the collected information for his own purposes. In the following, we will present several results we have obtained by our analysis. If not stated otherwise, the analysis is based on the platform we have deployed at the Laboratory for Dependable Distributed Systems. Please note that data is sanitized if necessary so that it does not allow one to draw any conclusions about specific attacks against a particular system, and protects the identity and privacy of those involved.

4 Results

At first, we will take a look at some statistical numbers we have collected in the past months. We the help of the web interface, we can easily query the database to get a quick overview of peaks in the data set that we have collected:

- Average number of attack sources per day is 184.94
- Maximum number of attack sources per day was 2022 and happened at November 15, 2004.

The two number show that there is a high variation in the collected data about the number of unique attack sources per day. To take a closer look at this phenomenon, we present in Table 1 the number of unique sources for six different platforms. In addition, the table presents the average number of sources per day, which shows a high variation across different platforms as well.

Currently it is unclear why we have this high variation in the number of average source per day. One possible explanation for the high number of average sources for the first platform is the following: since this system is deployed within the network with the first octet 192, it presumably receives many packets from broken systems which use Network Address Translation (NAT). Such a system often use the IP range 192.168.0.0/16 (defined in RFC

	Windows	Others	Unknown
Week 1	7235	18	10
Week 2	6839	26	5
Week 3	6475	38	-
Week 4	7766	89	-
Week 5	6594	24	64
Week 6	3599	5	58
Week 7	4640	11	92
Week 8	6247	20	83

Table 2: Operating system of attack source on weekly basis between January and February 2005

1918). If this system is infected by autonomous spreading malware, the malware can try to propagate further within the same network within the IP range 192.0.0.0/8. Thus it could be that we receive many packets from autonomous malware that just tries to attack other systems. But we do not have an explanation why several sources receive on average traffic from less than 60 sources per day. But we conclude that there are IP ranges in the Internet that are "more quiet" than others.

An analysis of the remote operating system of an attack source reveals the following result: almost all attack sources are running Microsoft Windows. Table 2 shows the remote operating systems that has been detected during the period January 1, 2005 until February 26, 2005 on a weekly basis for one specific platform. This information is based on the analysis by *p0f*. The statistic shows that clearly most of the attack sources are running Microsoft Windows. The presumable reason for this is: most autonomous spreading malware is propagating with the help of machines that run Windows as operating system. This autonomous spreading malware generates a lot of traffic that we receive with our platforms. At other platforms, the observed distribution of operating systems is slightly less biased, but nevertheless Windows clearly dominates on all platforms.

With the help of the tool *Maxmind GeoIP*, we can also determine the country in which the attack source is located. In Figure 2 we give an overview of the distribution of attack sources by country. This data is based on the information we have collected between January 1, 2005, and June 30, 2005, on our platform. In this period, we could observe 153,791 attack sources in total. We used Maxmind GeoIP to analyze the mapping between IP addresses and country. As we can see in the figure, most attack sources are located in the United States (24 %) and China (18 %). Germany (7 %), Japan (6 %), and Canada (5 %) follow on the next positions. With 33 %, the number of attack sources located in other countries is high. In total, we have observed attacks from 183 countries. For example, only one source could be located in Zambia and Netherlands Antinelles.

For other platforms, the collected data show similar results. In table 3 we give an overview for the same period of time (January 1 – June 30, 2005) for three different platforms $P_i, i = 1 \ldots 3$ that are deployed in different IP ranges.

Nevertheless, there are platforms that show different results. For example, platform P_3 monitors a huge amount of traffic originating from the United States. In addition, another

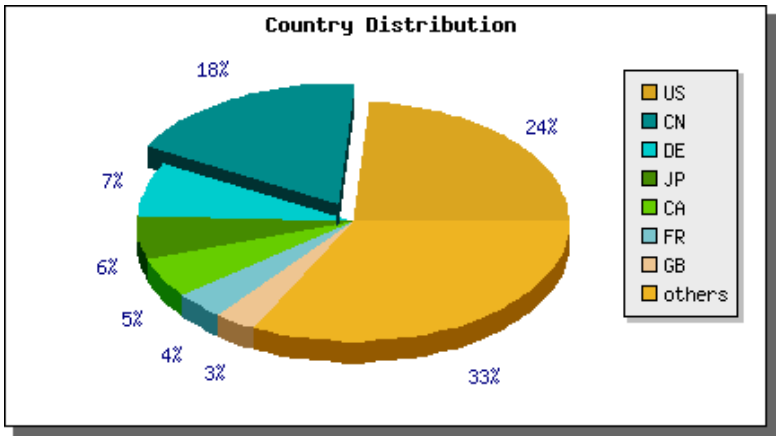

Figure 2: Country of attack source, monitored at one platform between January and June 2005

	P_1	P_2	P_3
US	23.76	20.91	47.78
CN	18.15	5.45	11.11
DE	7.38	8.73	2.55
JP	5.89	2.10	3.30
CA	5.37	1.57	9.01
FR	3.66	4.68	2.20
ES	1.54	6.40	1.88
IT	1.78	5.80	1.30

Table 3: Distribution of country in percent for attack sources between January and July 2005

platform received a significant amount of traffic origination from one single country that could not be observed by the other platforms. Currently, we do not have an explanation for these uncommon events. It could be that these spikes are due to spoofing attacks, i.e., the attacker tries to hide his true identity by specifying another TCP source address during his attack.

One of the first results that we want to present is the following observation:

> **Observation 1**: Let $\mathfrak{R}$ be a specific root-cause of an attack, P_i be a platform with the configuration described above, and n be the total number of platforms deployed. Then the traffic, that is caused by $\mathfrak{R}$ and monitored at each platform $P_i, i \in \mathbb{N}$, is correlated, unless the traffic at P_i is below a certain threshold.

A root-cause $\mathfrak{R}$ of an attack is a specific tool or specific technique, with which an attacker is able to execute a certain attack against a system. Thus the root-cause exactly describes the attack pattern that can be observed by a monitoring system. An example of a root-cause is an *exploit*, with which an attacker can take advantage of a vulnerability. Often the

root-cause of an event can be narrowed down to only one tool that has been used by the attacker. But there are also cases in which this is not possible and in which we can confine the root-cause to only a few possible tools that could have been used by the attacker.

Observation 1 states that similar attacks can be observed at all environments, if the specific environment receives at least enough traffic to be over a certain threshold. This constraint is necessary since platforms with low attack profile can certainly not monitor every kind of attack. In addition, random noise that is generated by portscans or other systems, should not be taken into account. In the following, we will show that there is indeed a correlation of root-causes of attacks that can be monitored network-wide. We will do this on the basis of several examples, that show the correlation and thus substantiate our observations.

We will start with an analysis of the total number of unique sources monitored at each platform per day. Figure 3 displays this number for three different platforms that are deployed at three different locations and in three different IP ranges. The plotted data starts at April 1, 2005, and lasts until July 1, 2005. From the Figure it is obvious that these three data sets are correlated since they follow a similar shape. For example, we can see that all three platforms have a low number of unique sources at the days 25, 61, and 75. Moreover, all platforms have a spike at days 7, 27, and 81. Nevertheless, the correlation is not perfect. For example, on day 38, two platforms have an overall low number of unique sources, whereas the third platform received more traffic than usual. Please notice that one of the platforms had three days downtime between day 70 and 72, during which no data was collected.

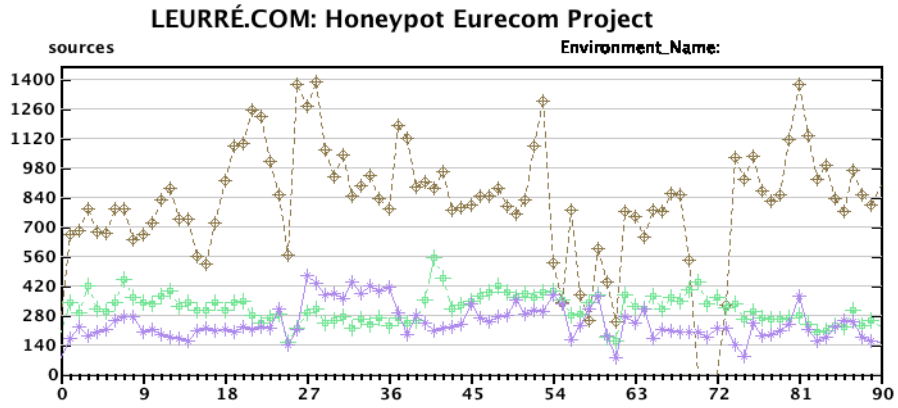

Figure 3: Number of unique sources per day on all TCP ports, monitored at three platforms within three months

If we just plot the unique sources for one specific TCP port, we also see that the data sets are correlated. Figure 4 shows the number of unique sources for TCP port 445 between April 1, 2005 and July 1, 2005. Again, we observe spikes an low levels of unique sources at the same days as noted above. Thus we can conclude with the help of this figures, that the data sets are correlated. Of course, we cannot give a hard proof of this observation since we take measured data into account. Nevertheless, we hope that the presented examples

support our observation.

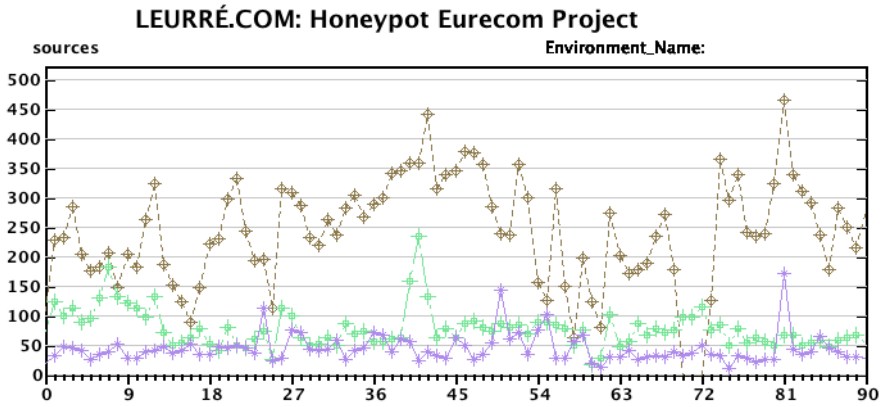

Figure 4: Number of unique source per day for TCP port 445, monitored at three platforms within three months

The second import results that we want to present is the following observation:

> **Observation 2**: Let P be a platform with the configuration described above, and $n \in \mathbb{N}$ be the total number of platforms deployed. If a new root-cause $\mathfrak{R}$ appears (i.e., a new exploit is used by attackers or attackers use a new technique), we can observe it at least at a subset of all platforms $P_i, i = 1..n$.

Again, no hard proof of this observation can be given due to the missing mathematical foundations of the whole observations. Nevertheless, we can give again some examples to substantiate our observation. We will start with a new root-cause $\mathfrak{R}_1$ that was at first observed at the end of November 2005. We will take a look at TCP port 42, that is commonly used by Windows Internet Naming Service (WINS). WINS provides a service similar to DNS for systems that use NetBIOS. With the help of this service, NetBIOS names can be mapped to IP addresses and vice versa. Figure 5 displays the number of unique sources per day observed at TCP port 42 from November 15, 2004 until January 4, 2005, for all platforms. At all days before November 5, we observed almost no traffic at this port. This port is commonly not contacted by systems located in other networks and thus rather quiet. Only sometimes one or two unique sources could be observed, but most of the time no traffic is received on this port.

At November 26, 2005 (day 11 in the figure), Immunity [Ait05] released an advisory for a vulnerability at TCP port 42. The vulnerability can be used to gain remote access to an unpatched system. However, Immunity did not release an exploit for everyone, only Immunity's client base had received a working exploit. In the following days, we could observe an increasing amount of traffic at this port. Probably this traffic was caused due to this new root-cause, resulting from attackers that scanned large parts of the Internet for vulnerable systems. At December 31, 2004 (day 46 in the figure), an exploit for this

vulnerability was released. Again, we can observe an increasing amount of traffic. This traffic is caused by attackers that actively attack other systems and scan large parts of the IP space. A huge increase of the number of unique sources can be monitored shortly after this date, since also a network worm using this vulnerability was released, and presumably the first bots included this exploit in their propagation mechanism to spread further.

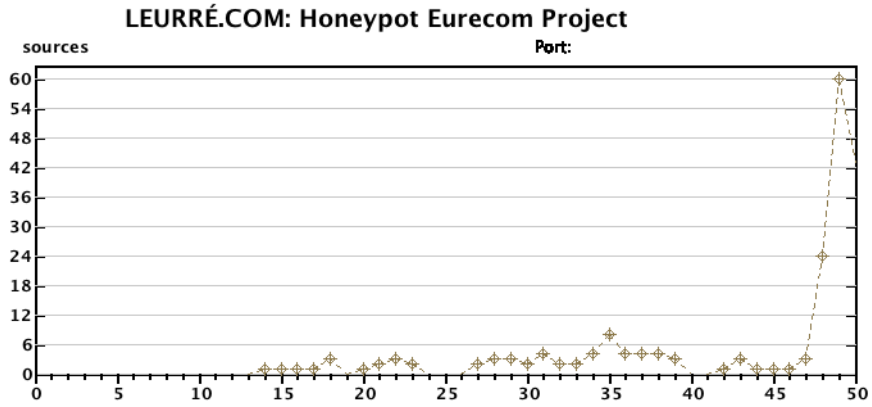

Figure 5: Number of unique sources per day for TCP port 42, monitored by all platforms between November 15, 2004, and January 5, 2005

If we take a look at a large time frame, we can see the impact of the new root-cause to a greater extend. Figure 6 shows the number of unique sources per day for TCP port 42 starting at November 15, 2004, until February 15, 2005. Again, we see clearly that before the actual vulnerability was announced (day 11), nothing could be observed. Between day 11 and day 46 (release of actual exploit), we can monitor the noise that is generated by attackers that scan the Internet for vulnerable systems. After the release of the exploit, we see a huge increase of the number of unique sources per day. This is probably caused due to actual attacks and the network traffic that was generated by autonomous spreading malware that uses this vulnerability to propagate further. In the following months, the amount of traffic at this TCP port stayed on a higher level.

And to give a further substantiation for Observation 1, we have plotted the number of unique sources per day for the three platforms we used in the previous example for TCP port 42 in Figure 7. Again, we see that the number of unique sources per day is correlated, e.g., there is a spike at day 80 visible at all three platforms. This figure confirms our claim and we conclude that there is indeed a correlation of attacks that can be monitored network-wide.

As a second example to endorse Observation 2, we take a look at another root-cause $\mathfrak{R}_2$. At August 3, 2005, a vulnerability for Computer Associate's "BrightStor ARCserve Backup Agent for SQL" was announced. This service listens on TCP port 6070 and provides a service to backup and restore data. Together with the advisory, an actual exploit was released. Similar to the previous example, we will take a look at the number of unique sources per day for this TCP port at all platforms. Figure 8 displays this number for the

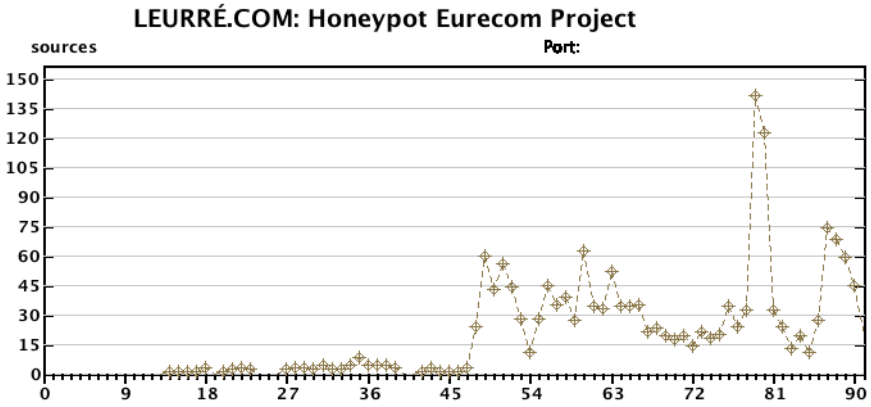

Figure 6: Number of unique sources per day for TCP port 42, monitored by all platforms between November 15, 2004, and February 15, 2005

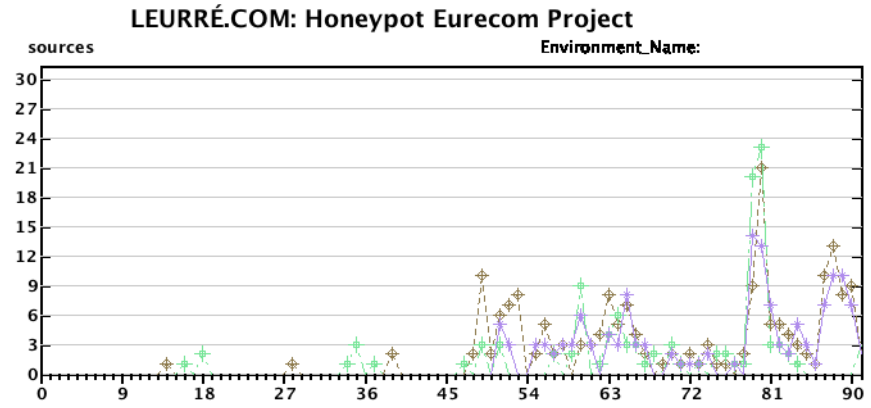

Figure 7: Number of unique sources per day for TCP port 42, monitored by three different platforms between November 15, 2004, and February 15, 2005

time between July 1, 2005, and August 11, 2005. Again, we have chosen a TCP port that normally does not receive much traffic and therefore we did not monitor any sources before August 3, 2005 (day 33). After the release of the vulnerability, we can monitor an increasing amount of attack sources per day. Once more, we conclude that root-cause $\mathfrak{R}_2$ leads to events that can be monitored by our world-wide distributed sensor system.

The previous two examples have shown that there is indeed an effect by each root-cause $\mathfrak{R}$ that can be observed with our distributed approach. Nevertheless, we are not able to monitor a new root-cause each time it appears. For example, if the port on which the root-cause could be monitored is subject to a lot of traffic (e.g., TCP port 445 or TCP port 135), we cannot easily monitor a new root-cause. Similarly, if the root-cause does not cause

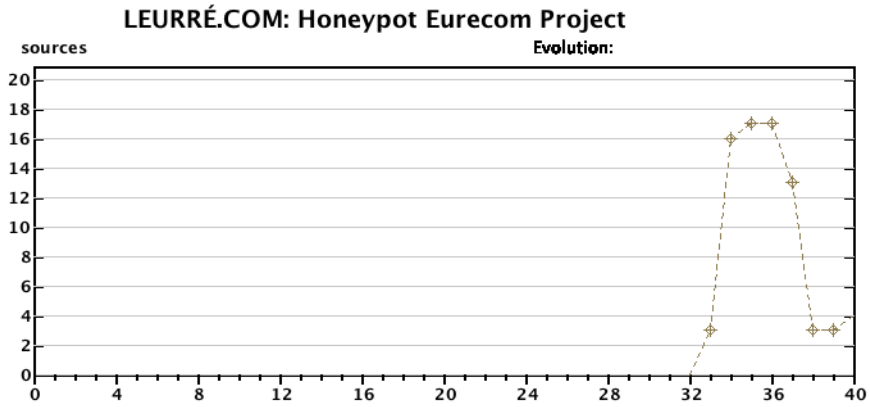

Figure 8: Number of unique sources per day for TCP port 6070, monitored by all platforms between July 1, 2005, and August 11, 2005

much traffic (e.g., only a limited number of individuals use an unknown, new exploit), we can probably not monitor it since we will not be the target of such an attack. Still, our approach can be helpful to learn more about new attack patterns and new root-causes. If we observe a sudden change in the number of unique sources, this is an indication that something suspicious is going on and that we should take a closer look at these events. Our approach can thus be used as some kind of early-warning system that is a burglar alarm for network-based attacks.

Since the collected data are stored in a database, it is easy to search for further correlations within the data. At first we want to take a look a specific *port sequences*. A port sequence $\mathfrak{p} = P_1|P_2|P_3|\ldots$ is a sequence of packets that is observed at the ports $P_i, i \in \mathbb{N}$ from the same attack source within a specific period of time.

Port sequences can be used to learn more about the root-cause $\mathfrak{R}$ of an attack. Often, a root-cause leads to a characteristic sequence of ports that are accessed within a short amount of time. And this characteristic can be used to identify a specific tool or technique that has been used by an attacker. For example, the port sequence $\mathfrak{p}_{Blaster} = 135|4444|$ is a clear sign that we have monitored an attack that is caused by the *Blaster* worm. Blaster is a computer worm that attacks systems running Windows as operating system. It was first observed on August 11, 2003. At first, the worm tries to exploit a vulnerable machine on TCP port 135 and aims at compromising the host by using a vulnerability concerning the Distributed Component Object Model (DCOM) Remote Procedure Call (RPC) of Windows. If TCP port 135 is closed, the worm concludes that it cannot exploit the host and therefore search another target. If the port open, it assumes that it can exploit the host. Afterwards, it contacts TCP port 4444 on the corresponding IP. On this port, the first stage of the worm has bound a command shell. And in the second stage, it tries to download itself and thus completely infecting the other system. So if we observe that an attack source contacts our low-interaction honeypots on TCP port 135 and 4444 in a short amount of time, we can conclude that we have just seen a blaster worm that tries to spread further.

As a second example, we explain how we can identify that the root-cause of an attack is the *Dabber* worm. On May 12, 2004, this worm was observed for the first time. From [Gro04] we know that Dabber tries to propagate in the following way:

> "[...] the worm will connect to port 5554 and send a single byte (an ascii 'C'). If that connection is successful, it will reconnect to port 5554 and send the exploit. After the exploit has been sent the worm will attempt to connect to port 9898 on the target host in order to confirm the infection was successful, again sending an ascii 'C'. If the connection to port 9898 is successful, an internal tally is incremented, presumably so statistics can be retrieved from the backdoor at a later time. Sequential scans on port 5554 and 9898 are an indicator of an infection."

So again, we search in our data for network connections to TCP port 5554 and 9898 within a short period from the same attack source. If we find such a port sequence $\mathfrak{p}_{Dabber} = 5554|9898|$, we can conclude that we have observed the propagation of the Dabber worm. Similarly, we can identify other root-causes of attacks by carefully analyzing port sequences.

In Figure 9 we have plotted the number of attack sources per week with port sequence $\mathfrak{p}_{Dabber} = 5554|9898|$ for the period April 1, 2004 up to November 11, 2004. In the first four weeks, we do not see any source with the port sequence $\mathfrak{p}_{Dabber}$ since at this point of time, this root-cause did not exist. Starting with week five, wee see that this root-cause appears. We see a steady increase of this port sequence and in the following weeks, we can monitor the existence of the Dabber worm.

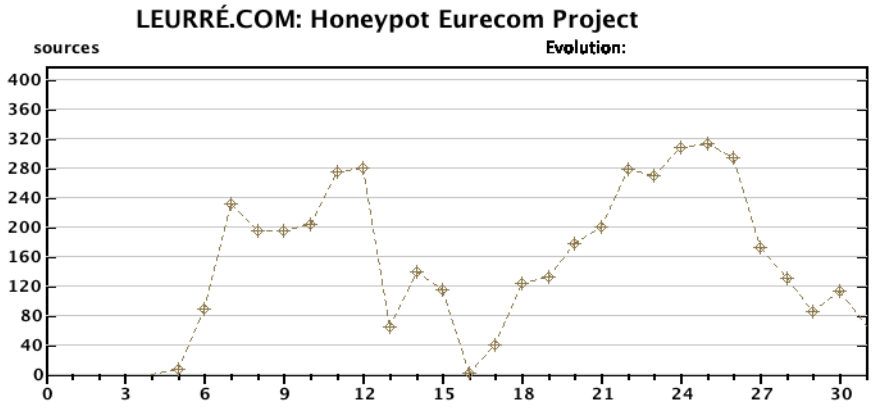

Figure 9: Number of unique sources per week for which the root-cause is Dabber between April and November 2004

Some port sequences clearly dominate the whole database. Since we observe most traffic on TCP ports 135 and 445, the port sequences $\mathfrak{p}_{DCOM} = 135|$ and $\mathfrak{p}_{LSASS} = 445|$ dominate. Other short port sequences like $\mathfrak{p}_{RPC} = 1025|$ or $\mathfrak{p}_{WINS} = 42|$ are also often

observed, but not that interesting from an analysis point of view since we can retrieve such short port sequences with easier methods. Longer port sequence like $\mathfrak{p}_{MyDoom} = 1025|6129|2745|80|3127|$ are more valuable from our point of view. This specific port sequence points us to a variant of the *MyDoom* worm. Similar to Blaster and Dabber, this worm tries to exploit a remote vulnerability and propagate further. Several variants of this worm exist, and one particular one can be recognized by its characteristic port sequence $\mathfrak{p}_{MyDoom}$.

The analysis of a new port sequence can lead to a better insight for a root-cause $\mathfrak{R}$. We can learn more about correlations between packets that we have observed and narrow down a specific root-cause. This is something that is not that easily possible with the information provided by the *Internet Storm Center* (ISC) / *DShield.org* [Ins05a].

5 Conclusion

In this paper we have presented a distributed approach in the area of honeynets. We have introduced the project leurre.com. In the scope of the project, we help to establish a world-wide distributed network of platforms, each platform running the same honeynet setup. All of these platforms have the same configuration and send the collected data to a central database. Hence, we are able to actually compare the observed attacks in more detail and carry out an in-depth analysis of the collected information.

We have presented several results we have obtained by analyzing the collected data. For example, we could show that there is a network-wide correlation of events that can be monitored at the different platforms. In addition, we showed how we can determine a new root-cause $\mathfrak{R}$. This can be achieved through a careful analysis of the attack sources monitored at a specific port. Moreover, we presented the concept of port sequences and how this helps to learn more about root-causes of attacks.

References

[Ait05] Dave Aitel. IMMUNITY : Knowing You're Secure. Internet: `http://www.immunitysec.com/`, Accessed: 2005.

[BCJ+05] Michael Bailey, Evan Cooke, Farnam Jahanian, Jose Nazario, and David Watson. The Internet Motion Sensor: A Distributed Blackhole Monitoring System. In *NDSS '05: Proceedings of the 12th Annual Network and Distributed System Security Symposium*, 2005.

[Cai05] CAIDA, the Cooperative Association for Internet Data Analysis. Internet: `http://www.caida.org/`, Accessed: 2005.

[CBM+04] Evan Cooke, Michael Bailey, Z. Morley Mao, David Watson, Farnam Jahanian, and Danny McPherson. Toward Understanding Distributed Blackhole Placement. In *WORM '04: Proceedings of the 2004 ACM Workshop on Rapid Malcode*, pages 54–64, New York, NY, USA, 2004. ACM Press.

[Cym04] Team Cymru: The Darknet Project. Internet: http://www.cymru.com/Darknet/, 2004.

[DGH04a] Maximillian Dornseif, Felix Gärtner, and Thorsten Holz. Vulnerability Assessment using Honepots. *PIK - Praxis der Informationsverarbeitung und Kommunikation*, 4(27):195–201, 2004.

[DGH04b] Maximillian Dornseif, Felix C. Gärtner, and Thorsten Holz. Ermittlung von Verwundbarkeiten mit elektronischen Ködern. In *Detection of Intrusions and Malware, and Vulnerability Assessment, DIMVA 2004, Dortmund, Germany, July 6-7, 2004, Proceedings*, Lecture Notes in Informatics, pages 129–141, 2004.

[DPP05] Marc Dacier, Fabien Pouget, and Van Hau Pham. LEURRE.COM Honeypot Project. Internet: http://www.leurrecom.org, Accessed: 2005.

[GDK04] Olaf Gellert, Till Dörges, and Klaus-Peter Kossakowski. Ein Netzwerk von IDS-Sensoren für Angriffsstatistiken, 2004.

[Gro04] LURHQ Threat Intelligence Group. Dabber Worm Analysis. Internet: http://www.lurhq.com/dabber.html, 2004.

[Ins05a] The SANS Institute. Distributed Intrusion Detection System. Internet: http://dshield.org/, Accessed: 2005.

[Ins05b] The SANS Institute. Internet Storm Center. Internet: http://isc.sans.org/, Accessed: 2005.

[MVS01] David Moore, Geoffrey M. Voelkeroffrey, and Stefan Savage. Inferring Internet Denial-of-Service Activity. In *Proceedings of the 10th USENIX Security Symposium*, August 2001.

[PD04] Fabien Pouget and Marc Dacier. Honeypot-Based Forensics. In *Proceedings of AusCERT Asia Pacific Information technology Security Conference 2004*, 2004.

[PH05] Fabien Pouget and Thorsten Holz. A Pointillist Approach for Comparing Honeypots. In *Detection of Intrusions and Malware, and Vulnerability Assessment, Second International Conference, DIMVA 2005, Vienna, Austria, July 7-8, 2005, Proceedings*, volume 3548 of *Lecture Notes in Computer Science*, pages 51–68. Springer, 2005.

[Plu82] David C. Plummer. An Ethernet Address Resolution Protocol, November 1982. Request for Comments: RFC 826.

[Pro04] Niels Provos. A Virtual Honeypot Framework. In *Proceedings of 13th USENIX Security Symposium*, pages 1–14, 2004.

[Pro05a] The Honeynet Project. Know Your Enemy. Internet: http://www.honeynet.org, Accessed: 2005.

[Pro05b] Niels Provos. Developments of the Honeyd Virtual Honeypot. Internet: http://honeyd.org, Accessed: 2005.

[SMS01] Dug Song, Rob Malan, and Robert Stone. A Global Snapshot of Internet Worm Activity, 2001. Internet: http://research.arbor.net/downloads/snapshot_worm_activity.pdf.

[YBJ04] Vinod Yegneswaran, Paul Barford, and Somesh Jha. Global Intrusion Detection in the DOMINO Overlay System. In *NDSS '04: Proceedings of the 11th Annual Network and Distributed System Security Symposium*, 2004.