Empirical Study of the Impact of Metasploit-Related Attacks in 4 Years of Attack Traces

E. Ramirez-Silva and M. Dacier

Eurecom Institute
Sophia Antipolis, France
{ramirez, dacier}@eurecom.fr

Abstract. For several years, various projects have collected traces of malicious activities thanks to honeypots, darknets and other Internet Telescopes. In this paper, we use the accumulated four years of data of one such system, the Leurré.com project, to assess quantitatively the influence, in these traces, of a very popular attack tool, the Metasploit Framework. We identify activities clearly related to the aforementioned exploitation tool and show the fraction of attacks this tool accounts for with respect to all other ones. Despite our initial thinking, the findings do not seem to support the assumption that such tool is only used by, so called, script kiddies. As described below, this analysis highlights the fact that a limited, yet determined, number of people are trying new exploits almost immediately when they are released. More importantly, such activity does not last for more than one or two days, as if it was all the time required to take advantage of these new exploits in a systematic way. It is worth noting that this observation is made on a worldwide scale and that the origins of the attacks are also very diverse. Intuitively, one would expect to see a kind of a Gaussian curve in the representation of the usage of these attacks by script kiddies over time, with a peak after one or two days when word of mouth has spread the rumor about the existence of a new exploit. The striking difference between this idea and the curves we obtain is an element to take into account when thinking about responsible publication of information about new exploits over the Internet.

1 Introduction

In this paper, we present a thorough analysis of 4 years of data collected by a number of honeypots distributed all over the world. The initial goal of this effort was to see i) if script kiddies activities were captured by honeypots and, if yes, ii) what relative importance such traffic had in the bulk of the collected dataset. Since, a priori, nothing distinguishes the attack traffic generated by a script kiddie from the one due to a botnet or an organized crime organization, our first task was to formulate the problem in a tractable way. Therefore, we have reduced the problem to the identification and quantification of the traces due to

I. Cervesato (Ed.): ASIAN 2007, LNCS 4846, pp. 198–211, 2007.

a specific tool that most script kiddies, without any expertise at all, could use to run attacks. There is a consensus in the security community to say that the Metasploit Framework is probably "the" tool that matches this criteria. Thus, the analysis presented here after focuses on the identification of attack traces due to that specific tool.

Much to our surprise not only did this attack tool left clear traces on our honeypots, a little bit all over the world but, more importantly, the discussion presented at the end of this paper seems to indicate that this tool is used in a very systematic way by well organized people who use the very latest exploit within the 24 first hours of their release. Such activity profile does not really match the expected behavior of script kiddies and this finding should be taken into serious consideration by the security community at large and by those who produce and publish such exploit code in particular.

The structure of the paper is as follows. In Section 2, we present the Leurré.com environment and the data set used in this experiment. We introduce the key notion of clusters, as defined within the Leurré.com project and we offer a brief presentation of the Metasploit Framework. We invite the reader who would already be familiar with these notions to skip this Section and immediately continue with the next one. Section 3 describes the experimental setup we have built to systematically identify all traces of potential interest in our database. We conclude that Section by explaining why the identified traces are likely to contain traffic not related to the Metasploit attacks. Section 4 proposes various strategies to filter out this noise and discuss the results obtained with this cleaned dataset. Section 5 concludes the paper with some discussion on the most surprising results.

2 Data Collection Environment

2.1 The Leurré.com Project

For almost 4 years, the people coordinating the Leurré.com project [4] have deployed and maintained a distributed system of identical honeypots all over the world. As of today, the system is made of approximately 50 platforms located in 30 different countries. Each platform monitors 3 distinct IP addresses, using the honeyd application developed by Niels Provos [13]. Each platform captures, by means of a tcpdump file, all packets sent to and from these three virtual machines. All captured tcpdump files are parsed and stored in an SQL database, enriched with data such as the geographical location of each attacking IP, the identification of its operating system (obtained thanks to p0f and disco, two passive OS fingerprinting techniques [14],[15]), etc. The interested reader is invited to look at [12],[7],[8],[9],[11] for more information on the various findings obtained thanks to this infrastructure.

2.2 The Leurré.com Notion of "Cluster"

The notion of "cluster of traces", as defined within the Leurré.com project [9],[10],[12], is a key concept used throughout the rest of this paper. To make a

long story short, one can say that a "cluster" is nothing else but a group of IP addresses that have interacted with the virtual machines of a given platform in a very similar way. Therefore, one can imagine that all these traces are likely to be due to the execution of the same attack tool on each of these attacking IPs. In other words, that the same tool has been launched from all IPs found in a cluster, or, similarly, that all IPs found in a given cluster are likely compromised by the same tool. It is clear that the semantic attached to the notion of cluster is, by far, not an exact one. The same tool can leave different traces [10], leading to the creation of several clusters that, actually, relate to a single tool. Similarly, distinct tools may leave the same fingerprint against a platform resulting in impure clusters where IPs corresponding to machines infected by different tools are grouped together. Nevertheless, introduced in [9], this notion has been validated and used in several publications, highlighting the fact that, in many cases, it was a meaningful way to group traces together.

We invite the interested reader to refer to [9] for the details of the algorithm used to build the clusters.

2.3 Metasploit Framework

Metasploit [5], according to the latest survey conducted by Fyodor [2], is the most popular vulnerability exploitation tool [3] and comes at the fifth position for the most popular security tool, according to the same study. Quoting that study: *"Metasploit took the security world by storm when it was released in 2004. No other new tool even broke into the top 15 of this list, yet Metasploit comes in at #5, ahead of many well-loved tools that have been developed for more than a decade. It is an advanced open-source platform for developing, testing, and using exploit code. The extensible model through which payloads, encoders, no-op generators, and exploits can be integrated has made it possible to use the Metasploit Framework as an outlet for cutting-edge exploitation research. It ships with hundreds of exploits, as you can see in their online exploit building demo. This makes writing your own exploits easier, and it certainly beats scouring the darkest corners of the Internet for illicit shellcode of dubious quality. Similar professional exploitation tools, such as Core Impact and Canvas already existed for wealthy users on all sides of the ethical spectrum. Metasploit simply brought this capability to the masses."* [3].

The Metasploit Framework can be invoked in different ways to launch attacks (msfconsole, msfcli interface or the msfweb interface). When using the graphical interface, the user can not easily launch attacks against a large number of hosts but, by using the msfcli command, one obtains a command line interface which is well suited to automatize campaigns of attacks against large numbers of hosts using so called Metasploit plugins, ie vulnerability exploitation tool. This command is simply invoked as follows: *"msfcli match_string options(VAR=VAL) action_code"* where match_string is the plugin (exploit) name to be launched. The action_code is a single letter used to specify what should be done; S for summary, O for options, A for advanced options, P for payloads, T for targets, C to try a vulnerability check, and E to exploit [6].

For instance, to launch the execution of the so called *"backupexec_dump"* plugin against the host 192.168.1.11, one would issue the following command:

"./msfcli backupexec_dump PAYLOAD=win32_exec RHOST=192.168.1.11 TARGET=0 E"

Released at the end of 2003, the framework has evolved over the years incrementally. In early 2007, version 3.0 has been produced which is a complete rewrite of the whole framework using the Ruby language with new features and interfaces that distinguishes it completely from the previous releases. Version 2.x, written in Perl, was also different from 1.0 and has been through 8 releases; each release came with new plugins (exploit modules). For the sake of consistency and also for practical reasons, we restrict ourselves to these 8 versions (version 2.0 to 2.7), out of 10, of the Metasploit Framework to analyze its impact on our dataset.

3 Experimental Setup

3.1 Introduction

The analysis we have carried out is made of two distinct steps. In the first step, we have experimentally produced the partial definition of clusters that a Metasploit attack against our platforms would have left. Then, in a second phase, we have identified in our database all clusters whose definition was matching the one of any of those produced in the first step. Last but not least, we have applied various filters to ensure that the found clusters were, with a very high probably, linked to a real Metasploit attack and not to another attack which would have left the same fingerprint on the attacked platform. In this section, we present, step by step, the process followed to create the various partial definitions of "Metasploit" clusters. Section 4 presents the various filtering strategies applied on them.

Fig. 1 gives a high level description of the process that leads to the creation of potential candidate definitions of Metasploit-related clusters. Two distinct functional modules appear. The one on the left is responsible for launching all possible attacks against one of our platforms, in a laboratory. Traces of the attacks are saved, labeled and provided to the second module, on the right, which extracts, for each attack, the characteristics common to all clusters that would contain the same kind of traces. It also searches the database for all matching clusters, if any, and produces, as an output, a list of clusters found in the database that matches the signature of a Metasploit-related attack.

These two modules are described in more detail here below.

3.2 Launching All Possible Attacks

We wrote a perl script that iterates through all attacks available in the Metasploit Framework and that targets the virtual machines on the platform using

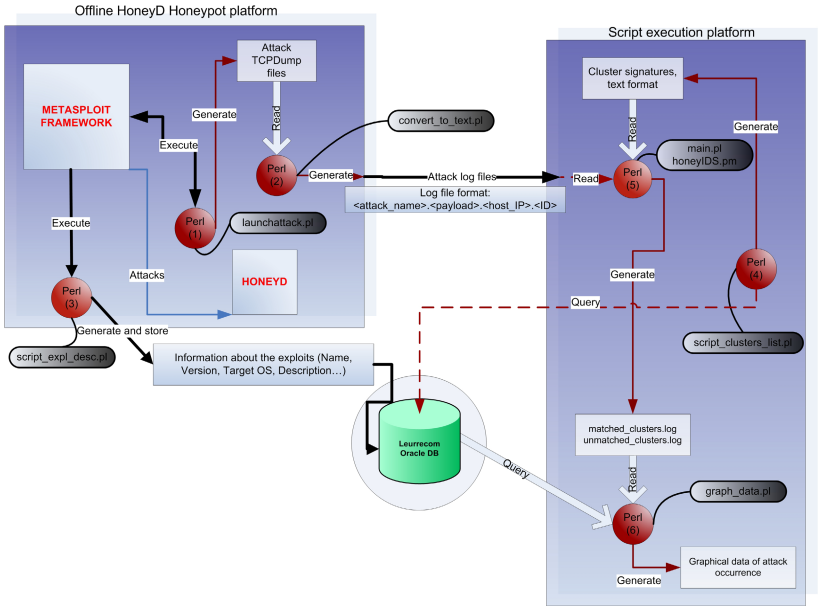

Fig. 1. High level presentation of the signatures generation process

all possible combinations, ie targeting Machine 1 only (resp 2, or 3 only), Machines 1 and 2 (resp. 1 and 3, 2 and 3), machines 1, 2 and 3. The order of the attack, 1-2-3 or 1-3-2 or ..., is one of the seven attributes that defines a cluster, as described in Section 2. We have not taken this element into consideration in our experiment as it would have dramatically increased the number of traces produced without adding any discriminant information, since all sequences must be seen as valid.

Our script invokes the msfcli command to launch the Metasploit attacks on the three honeypot IPs. It consists of iterative loops that start by querying Metasploit for all available attacks and then runs each attack, with all possible payloads, against the various combinations of the three available honeypots. This script also starts and stops the honeyd service and generates a tcpdump in order to be able to generate a tcpdump file for each attack.

The different steps the script goes through are:

1. Query Metasploit for all available attacks
2. For each attack, query Metasploit for all available payloads
3. For each honeypot IP and combination of IPs, launch the attack and the specific payload as follows:
   (a) Start the honeyd service.
   (b) Start the tcpdump monitor.
   (c) Launch the attack using the msfcli shell command with specific attack and payload and default options.
   (d) When the attack is over, stop the honeyd service and stop tcpdump.

(e) Rename the generated log file.
(f) Go to step (a) until all attacks are carried out.

3.3 Data Processing: Labeling Clusters with Attack Signatures

The role of the second functional module is to search for all clusters in the database that contain traces similar to the ones generated in the first phase of the experiment. To do this, we extract from each tcpdump file generated in the first phase, the values of the four first attributes used to define a cluster. As explained before, we ignore the order in which the virtual machines have been hit. We also ignore the total duration of the attack as well the average inter arrival time of the packets as these two factors could vary depending on the way the attacker has automatized the launching of the Metasploit plugin. Indeed, suppose that two attackers are scanning, e.g., the class C where one of our platforms is located. The first one does the scan randomly whereas the other does it sequentially. Both traces will end in clusters that will vary only on the basis of the last 3 attributes. As we are interested in finding these clusters, as well as all the others, we simply ignore the last 3 attributes when generating the signatures of our traces. To do this, we have a script that converts each attack dump file, obtained in 3.1, into an attack signature which has the following format:

```
Attack=<attack name> ports=<ports sequence> T=<No. targeted virtual
machines> N=<Total No. packets sent> n1=<packets sent to machine1>
n2=<packets sent to machine2> n3=<packets sent to machine3>
```

Last but not least, we extract from the Leurré.com database all the cluster identifiers the four first attributes of which match one of the attack signatures generated before. More precisely:

- Compare each attack signature to all the cluster signatures and declare a match if all of the following are true:
  1. $n1(min)Cluster \leq n1Attack \leq n1(max)Cluster$[1]
  2. $n2(min)Cluster \leq n2Attack \leq n2(max)Cluster$
  3. $n3(min)Cluster \leq n3Attack \leq n3(max)Cluster$
  4. $Ports\_sequence\_Cluster = Ports\_sequence\_Attack$
  5. $number\_of\_targeted\_IPs\_Cluster = number\_of\_targeted\_IPs\_Attack$
  6. $N(min)Cluster \leq NAttack \leq N(max)Cluster$

3.4 Preliminary Results

When we ran the attack script with all the exploit modules found in release 2.7, we obtained approximately 4000 distinct tcpdump files. It should be noted that certain Metasploit attacks require a connection from the target (to download a

[1] In the definition of a cluster, the number of packets sent against a given machine is not an absolute value but a range of values -to take into account duplicates and lost packets, among other things.

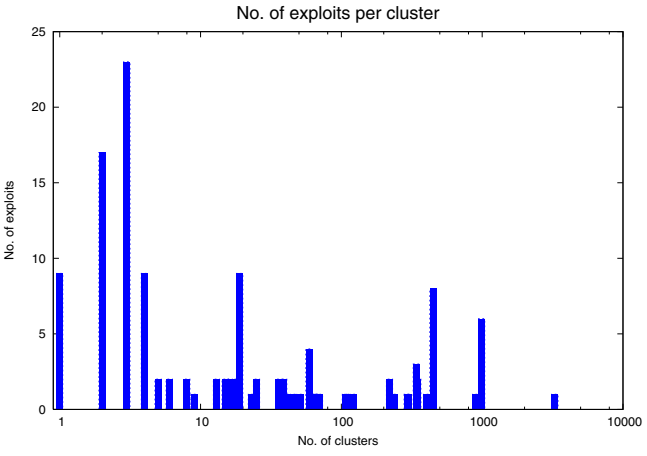

Fig. 2. Distribution of the number of exploits wrt number of clusters

file for example), whereas others wait for a connection from a user (SSL attack). These exploits have therefore been omitted from the analysis since they do not generate any traffic at all.

At the time of the experiment, the Leurré.com database did contain approximately 150.000 distinct clusters.

When we matched the derived 4000 signatures with each of these 150.000 cluster definitions, we end up with around 19'000 distinct cluster IDs. In other words there are 19'000 groups of traces in the database that are similar to traces generated artificially in the laboratory by running Metasploit plugins against a similar platform. Fig. 2 shows the distribution of the amount of exploits "per cluster". The figure shows that among the 132 exploits, there are 9 exploits that have matching characteristics in a single cluster, 17 for which 2 clusters where identified for each, 23 with three clusters, etc. In other words, almost half of the exploits are mapped with a single or a couple of clusters in the DB. We also see that a few exploits are mapped to a very large number of clusters (up to 3287 distinct ones !). It is quite likely that, among these clusters, many are not related at all to the Metasploit attack but simply target the same port in a similar way (e.g., port 445 or 139 or ...). We can, therefore, not rely on this first extraction method to look at the observed activities. In the next Section, we explain how we can filter out all the clusters that are likely due to other phenomena.

4 Analysis Results

4.1 Logic of the Experiment

From the previous Section, it is quite clear that the procedure we have followed may have helped identifying traces in our database that are linked to the manifestation of Metasploit related attacks but it is also clear that these traces are

mixed with a large number of traces that have nothing to do with the phenomena we are interested in.

To isolate the interesting traces, we are going to follow a three stage process. In the first phase, Section 4.2, we filter out a very large number of traces to keep only those for which we are almost certain that they correspond to the phenomena we are interested in. This first result ensures us that there is, indeed, something to be found in the dataset. In the second phase, Section 4.3, we relax some of the constraints used in the first filtering process and we verify that the characteristics of this second result are consistent with the first one. This suggests that we have captured again, in this second filtering, traces related to the Metasploit related phenomena. Last but not least, in the third phase, Section 4.4, we apply some heuristics that we believe could also capture other interesting traces and, hereto, we verify that the characteristics of this new experiment are consistent with those corresponding to well identified Metasploit traces.

4.2 Selection on the Basis of the Original Date

In order to define the traces we are interested in, we impose some reasonable constraints on them and we select only those clusters that fulfill all criteria. The basic underlying idea is that a cluster contains traces related to a given Metasploit plugin if the number of attacks observed for that cluster around the date of the release of the plugin is significantly different than before or after. To select clusters that satisfy this property, we apply the following algorithm:

1. For each of the 19000 selected clusters in the previous phase do:
   - Obtain the original plugin release date corresponding to the cluster under consideration.
   - Compute the number of attacks, per day, observed for that cluster in the period going from -30 days until +30 days after the found release date.
   - If this cluster had never been observed before the release day minus 3 day, select the cluster and go to step 2.
   - Compute the average number of attacks for that cluster for the period [release date - 30 days, release date + 30 days]. Compute the standard deviation for the same period.
   - Select the cluster and go to step 2 if, within the period [release date - 5 days, release date + 5 days], we observe days where the number of attacks is greater than the average value + 2 times the standard deviation.
   - If no such point exists, discard the cluster and move to step 1 with the next cluster in the list.
2. Search for the maximal value of attacks per day observed for the selected cluster over the whole lifetime of the cluster.
3. If the found maximal value does not appear within the period [release date - 5 days, release date + 5 days], discard the cluster as we are interested in clusters that should normally be more active around the period of the plugin release. Continue to step 1 with a new cluster.
4. If the maximal value is within the expect boundaries, mark this cluster as being a good candidate.

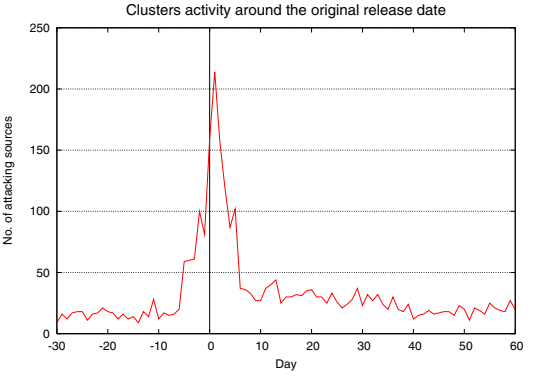

Fig. 3. First phase, number of attacks observed around day 0

The information concerning the original release date we have used is the one published officially in the Metasploit website, and is the date of the first appearance of the exploit module in the Framework. The execution of this algorithm against the 19000 selected lectures leads to the selection of only 700 of them! By having been very selective, we are quite confident that these clusters do indeed correspond to activities linked to the Metasploit Framework.

Fig. 3 represents the number of attacks observed for these 700 clusters where the X axis represents the number of day before and after the original plugin release. It highlights the fact that the peak activity occurs between -1 day and up to 2 days after the exploit release date with a maximal value in day 1. Two conclusions can be derived from this picture:

1. some exploits are tried out in the wild a few days before being officially published
2. the new plugins are very rapidly tried out and abandoned, as highlighted by the burst of attacks observed on day +1.

It is interesting to note that these attacks have been observed against platforms located all over the world and that they did originate from machines found in many different countries as well. This is a general phenomenon, not restricted to some countries or some platforms. This is represented in Fig. 4 and 5. Fig. 4 shows the geographical location of the attack sources. In Fig. 5, the horizontal axis presents the top 10 countries where attackers are coming from, for the selected clusters. The vertical axis gives the number of associated attacking sources. The other countries are grouped in the 'others' category (62 countries).

Fig. 6 shows the distribution of the attacks per environment[2]. We can observe that the attacks are not limited to a particular environment, at the contrary, they

[2] All Leurré.com partners are bound by an NDA that forbids them from communicating to the outside neither the IPs of the attackers or the IPs of the attacked platforms. This is why we anonymize the names of the platforms by replacing them by the name of the country where they are located.

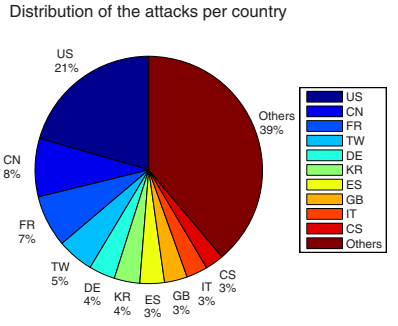

Fig. 4.

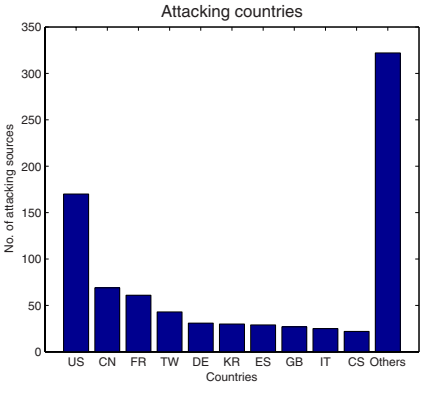

Fig. 5.

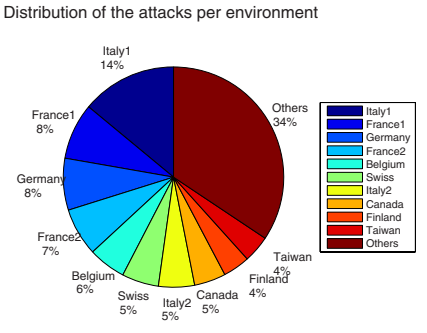

Fig. 6.

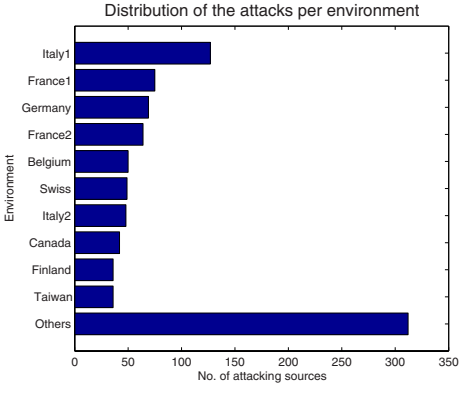

Fig. 7.

are well distributed. We only present the 10 most frequently attacked environments. The data series labelled *others* correspond to all other Metasploit related attacks observed on the remaining 38 environments.

4.3 Selection on the Basis of All Release Dates

So far, the algorithm described before has been applied for each cluster for a single date, the date of the original release of the plugin linked to the cluster under consideration. However, it is reasonable to expect that an "old" plugin published, e.g., in 2005, would suddenly be reused intensively simply because, e.g., a new releases of the framework is published. This could be a side effect of the publicity surrounding the publication of the new release. To take this element into account, we rerun the algorithm on the 19000 clusters, minus the 700 found before, by taking into account not only the original release date of the plugin but all other dates of plugin releases coming after that. In other words,

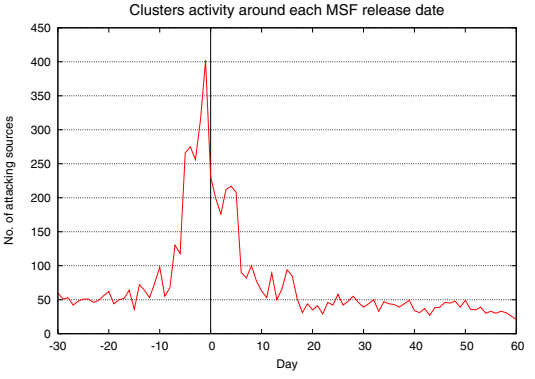

Fig. 8.

if a cluster matches the criteria of the previous algorithm for any of the release date that follows the original release[3], we select that cluster and sum all its activities around the range [release date - 30 days, release date + 30 days] for all periods following the original release date.

Fig. 8 shows the number of attacks corresponding to the ≈ 1300 matched clusters identified with this new method. The shape of this curve shows a striking similarity with the one shown in Fig. 3. There are two major differences though. First, the peak value of the curve appears at day -1 instead of +1, in the previous case. Second, we observe a very high number of hits at day -2. A deeper analysis reveals the explanation of these phenomena:

- A single exploit module appears to be responsible for the burst at day -1: *msasn1_ms04_007_killbill* exploit module from release 2.5. The most significant clusters that matched that specific attack signature have almost 230 attacking sources on that day. The attack has been observed only on two environments: one in Luxembourg and the other one in France. Most of the attacks came from 2 countries: Germany (DE) and Spain (ES).
- A single exploit module appears to be responsible for the burst at day -2: *mssql2000_preauthentication* exploit module from release 2.6. The most significant clusters that matched that specific attack signature have almost 100 attacking sources on that day. The attack has been observed a little bit all over the world and most of the attacks came from 1 country: China (CN).

So, in this case, the filtering has identified new traces that are, quite likely, linked to the Metasploit Framework and that also revealed some specific behavior on behalf of the attackers.

4.4 Clusters Without Activity Before Day -2 Filter

The two preceding filters are very good to select clusters that are, with a very high probability, linked to Metasploit related activities. However, they are probably

[3] Dates of releases 2.1 to 2.7.

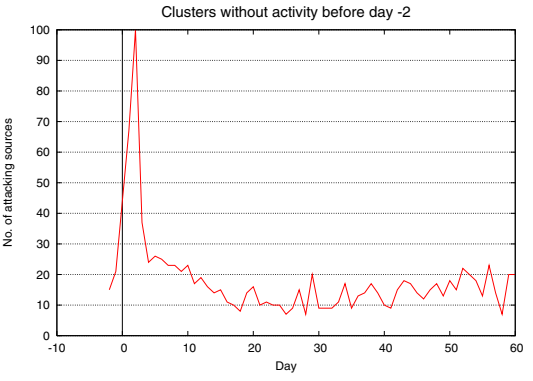

Fig. 9.

too restrictive and may have discarded clusters that could have been of interest. As a sanity check, we have decided to select all clusters for which the very first manifestation had been observed in a window of -2 to +2 days around any of the 8 possible release dates. In this last filtering approach, we do not discard clusters fulfilling this property if their maximal value appears in a period of time unrelated to any important Metasploit dates. Our hope is to identify, by doing so, clusters that are linked to the Metasploit exploit around one or several release dates but that got mixed with another, more important activity, later on.

The application of this algorithm to the remaining clusters not yet selected, we obtain 80 new clusters. Fig. 9, represents the number of attacks per day, reported relatively to any release date. Here to, we obtain a very bursty curve, just after the release which seems to indicate that the clusters we have selected are behaving similarly than the other ones and, therefore are also due to the Metasploit plugin releases.

4.5 Discussion

Fig. 10 offers the sum of all activities linked to the clusters identified in the three previous methods. The refined approach confirmed the first observations made:

1. the exploits are used extremely rapidly once they have been released.
2. some exploits are used in the wild before being made public.

It is also worth noting that the amount of attacks observed is actually fairly small. This, of course, has to be put in relation with the very limited number of addresses we are observing and, furthermore, the fact that these honeypots are low interaction ones. One can assume that we only see attacks that do participate to a very large, potentially worldwide, scan of the internet for specific exploit. Therefore, the hits we see do simply represent the tip of the iceberg and it means that there are people in the world who, as soon as plugins are released, immediately launch a worldwide scale attack against all possible platforms thanks to the new plugin, or new release. It is also important to notice

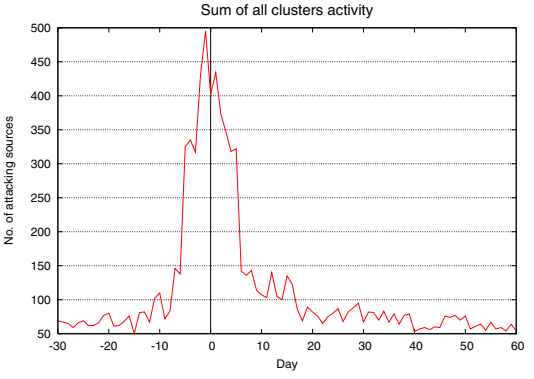

Fig. 10.

that these phenomena last for a very limited amount of time, one or two days. This is not at all what one would expect from a large population of script kiddies, scattered all over the world, with different skills and equipments at their disposal. As for every activity involving a large number of participants, it should rather be represented by a Gaussian curve highlighting the fact that a few ones see the new release immediately, spread the news, more script kiddies try it out as well, reaching a peak and, then, the number of attacks slowly decreases over the course of several days or weeks. The fact that none of our curves matches this description is a strong indication that new Metasploit releases are used by a very different population of users. These ones keep a close eye on new releases and have, probably, bots at their disposals to try them out on a very large scale immediately. Security administrators should be aware of that fact and, similarly, keep the publication of new exploits within Metasploit under close scrutiny as they can represent significant threats for their systems.

Whereas we are certainly not advocating that "security by obscurity" is a paradigm that should be promoted, at the same time we consider that those who publish new exploit plugins for the Metasploit Framework should be made aware of the fact that they help well organized entities who are not maneuvering for the good of the humanity.

5 Conclusion

In this paper, we have proposed a method to systematically identify in a very large dataset all the traces that were likely due to the Metasploit Framework (releases 2.0 to 2.7). We have shown that new plugins and new releases, are used by an important population, all over the world, that seems eager to run these exploits against as many machines as fast as possible. Quantitative examples are given throughout the text that show the validity of the approach as well as the impact of that tool on the community at large.

References

1. Arbaugh, W.A., Fithen, W.L., McHugh, J.: Windows of Vulnerability: A Case Study Analysis. IEEE Computer 33, 52–59 (2000)
2. Fyodor.: Top 100 Network Security Tools (last visited, July 25, 2007), available on line on `http://sectools.org`
3. Fyodor.: Top 3 Vulnerability Exploitation Tools (last visited, July 25, 2007), available on line on `http://sectools.org/sploits.html`
4. Leurré.com Project web page (last visited, July 25, 2007), `http://www.leurrecom.org`
5. Metasploit Project web page (last visited, July 25, 2007), `http://www.metasploit.com`
6. Metasploit Framework User Guide. Version 2.5., `http://metasploit.com/projects/Framework/docs/userguide.pdf`
7. Pouget, F., Dacier, M., Debar, H., Pham, V.H.: Honeynets: foundations for the development of early warning information systems. In: The Cyberspace Security and Defense: Research Issues - NATO Advanced Research Workshop, Gdansk, Poland (September 6-9, 2004)
8. Pouget, F., Dacier, M., Debar, H.: Honeypots, a practical mean to validate malicious fault assumptions. In: PRDC 2004. 10th International symposium Pacific Rim dependable computing Conference, Tahiti, French Polynesia (March 3-5, 2004)
9. Pouget, F., Dacier, M.: Honeypot-based Forensics. In: Proc. AusCERT Asia Pacific Information Technology Security Conference, Brisbane (2004)
10. Pouget, F., Dacier, M.: Honeypot Platform: Analyses and Results. Rapport de recherche RR-04-104 (October 30, 2004)
11. Pouget, F., Dacier, M., H., Pham, V.H.: Leurre.com: on the advantages of deploying a large scale distributed honeypot platform. In: ECCE 2005. E-Crime and Computer Conference, Monaco (March 29-30, 2005)
12. Pouget, F.: Distributed System of Honeypots Sensors: Discrimination and Correlative Analysis of Attack Processes. PhD thesis, Institut Eurecom (2006)
13. Provos, N.: A virtual honeypot framework. In Proceedings of the 12th USENIX Security Symposium, pp. 1-14 (August 2004)
14. Disco tool web page, `http://www.altmode.com/disco/`
15. p0f passive fingerprinting tool web page, `http://lcamtuf.coredump.cx/p0f-beta.tgz`