

Smallpot: Tracking the Slapper and Scalper Unix Worms

Costin Riau 2003-02-04

Introduction

Fueled by the old myth that "you can't get a virus in Unix" and by the increasing popularity of Linux and FreeBSD, Unix viruses passed an important milestone in 2001 and continued by receiving even more attention during 2002.

It all begun with the [Ramen](#) worm, then continued with [Adore](#), [Lion](#), [Cheese](#), [RST.B](#) and many, many more. Some of them even became widespread, culminating with the inclusion of OSF.8759 in the [May 2002 Wildlist](#)

Unfortunately, while there are lots of charts and distribution statistics available for Win32 viruses, the same cannot be said about Unix viruses. Actually, at the time of writing, there is no reliable (and constantly updated) source of information regarding the distribution of Linux/FreeBSD malware, not to mention other Unix flavors which are less appealing to virus writers.

Given the relative lack of information from this point of view, it would be very interesting to compare the spreading of Linux and FreeBSD malware with their Win32 counterparts as a means of evaluating the security and "virus-proofness" of these platforms. Such data may also help to develop a model for predicting the spreading of future Unix malware.

But how exactly can this be done? For Win32 viruses, the large amount of AV software deployed at workstation and server level provides a reliable source of reports, giving antivirus companies a simple way to create statistics on viral spread. Unfortunately, the same is not true for Unix viruses. First of all, these are not commonly distributed as infected e-mails, so gateway reports are almost non-existent. Similarly, the Unix versions of AV products rarely have remote reporting capabilities, thus making the source of Unix infections scarce. So, again, how does one track the proliferation of Unix viruses?

Well, two interesting examples of this recent chain of worms and viruses - [FreeBSD/Scalper](#) and [Linux/Slapper](#) - may offer some insights. Since both exploit a popular service (HTTP) and both probe random machines on the Internet with specific packets of data, it would be possible to use a simple honeypot to build a list of machines presumably infected by Slapper or Scalper, and analyze their distribution and proliferation.

Unfortunately, a single honeypot will not provide accurate results, as attacks of both Scalper and Slapper are not fully random, but instead go on predefined patterns. However, by using a set of honeypots spread around the world, the results will provide a broader view of the number of infected machines and the geographical distribution of attacks. Therefore, in order to build any reliable statistic regarding the distribution of these two viruses, one will need access to the data collected by a reasonable large honeypot network, covering most areas of the world.

Here's where the [Smallpot Project](#) comes in handy. The Smallpot Project is a generic honeypot that was initially designed to track the spreading of the [CodeRed](#) worm on Win32 systems, slowly grown into a means of monitoring almost any kind of Internet malware (tracking e-mail worms is one thing not covered by the Smallpot project) or even hacking attempts. Smallpot nodes are distributed all around the world, in countries such as the United States, South Korea, Romania, Germany, the Philippines, and Taiwan. Moreover, since all Smallpot reports are collected on a daily basis on a central node, it is relatively easy to process and analyze all the logs.

Thus, by combining the results provided by the Smallpot network over the past year, we can attempt to track the spread of Slapper and Scalper since their initial release. Building a geographic distribution map will also provide a simple view of the countries and areas that have been most affected by these two worms, allowing us to compare them to popular Win32 worms such as CodeRed, [Nimda](#) or [Spida](#), drawing a parallel between Win32 and Unix malware, and for comparing their general spreading behavior.

Scalper and Slapper

Our two subjects of interest, Scalper and Slapper, rely on different vulnerabilities to replicate, but probe potential victims in a similar manner. Both viruses will first send the following HTTP request on port 80, and then scan the reply for various known Apache versions. Here's how the initial HTTP request string looks like:

```
GET / HTTP/1.1\r\n\r\n
```

If the probed host looks vulnerable, Scalper will send a 32314-byte packet that is a common buffer overflow exploit. Likewise, Slapper connects on port 443 (SSL) and tries to exploit a bug present in OpenSSL versions older than 0.9.6e and 0.9.7-beta.

Just a small note regarding the initial probe used by Slapper and Scalper: a host can be hit by port 80 requests coming from a variety of tools, browsers, agents, and viruses. The tools usually include security scanners and HTTP exploits used by hackers to find and compromise machines for use as DDoS bots, storage space, or stealing data. Fortunately, the request used by Slapper and Scalper on port 80 is not one of these, and to my knowledge, is not used in any security scanner or stand-alone exploit. The same applies for browsers, which usually identify themselves with tons of other HTTP parameters and information requests, while for viruses, (the major offender here obviously being Nimda) they don't send such plain HTTP/1.1 requests either. Because of that, one can assume within a reasonable error threshold that all the requests of the form mentioned above were due either to Slapper or Scalper. Of course, we can further refine the process by taking the second packet for Scalper and the port 443 data for Slapper, but unfortunately, because of time-outs and network errors, they do not always arrive after the initial HTTP request.

The Smallpot Project

In the early July, 2002, the Net witnessed the emerge of the first fileless automotive Windows viral code sequence, now known as CodeRed. Due to its "fileless" nature, CodeRed brought at least two new problems for the antivirus developers. Firstly, of course, detection, which requires more than the usual file scan methods. Secondly, it created the need to implement tools to capture and study the movements of such things, directly on the Internet.

As has already been mentioned, Smallpot, short for "Small Honeypot", was designed with exactly the latter purpose in mind: to simplify the collection and classification of Internet malware, as well as tracking their spread and studying hacker attacks. It should be noted that besides HTTP, Smallpot also tries to fake various other Internet services such as FTP, POP3, SMTP, SUN-RPC, Telnet, UPnP, MS-SQL, SSH and backdoor servers such as NetBus or SubSeven. However, in this discussion of Slapper and Scalper, we will only take into consideration the HTTP packets, which should provide an acceptable degree of precision. It should also be mentioned that the data collected by Smallpot on Slapper and Scalper begins in September 2002, when the first reports (and samples) started to arrive.

A Problem: Mapping IP Addresses to Countries

The IP addresses around the world hitting Smallpot nodes rarely have a reverse DNS lookup entry. Because of that, constructing a country-based statistic on the spread of Slapper/Scalper

is no easy task. To further complicate the problem, at the time of writing there is no absolute authority on the Internet that can be automatically queried for the geographical position of a machine with a given IP address. Even worse, sometimes the reverse DNS lookup entry for an IP address will not tell much on the location of the attacking machine. For instance, a machine with a name ending in .com can be anywhere.

Fortunately, there are a couple of solutions that solve this problem by maintaining a database of known networks and associated country codes. Amongst these services, MaxMind's GeoIP (*) and JufSoft's ActiveTarget (*) seem to be easiest to automate and operate with a large list of IP addresses. Unfortunately, because they operate on static lists and perform no further checks on the exactness of the location guess, both tools have a small error factor in the precision of their reports. Still, during my experimental tests, the error proved small enough not to affect the results by any relevant measure, therefore, the final results and figures included in this article should reflect the reality with a degree of precision of about 97%.

Results

The following list of results has been obtained by parsing all Smallpot reports caused by Slapper and Scalper between September 2002 and January 2003 (the two-letter country codes used in the following tables refer to the standard [English ISO 3166 country codes](#)):

Country	% of Attacks
US	20.57
TW	14.06
CN	7.81
JP	7.55
KR	5.73
PL	4.43
IN	3.12
MX	3.12
HK	3.12
BR	2.86
CA	2.60

FR	2.60
DE	2.60
IT	1.82
GB	1.56
RU	1.56
CL	1.56
NL	1.56
RO	1.30
CO	0.78
AU	0.78
PH	0.78
SE	0.52
LK	0.52
IS	0.52
BO	0.52
IE	0.52
DK	0.52
ES	0.52
SK	0.26
FI	0.26
ID	0.26
SN	0.26
AR	0.26
PA	0.26
GR	0.26
TH	0.26
BJ	0.26
SG	0.26

VE	0.26
BG	0.26
YU	0.26
SV	0.26
CU	0.26
HU	0.26
CZ	0.26

As the list shows, the United States, Taiwan, China, Japan, and Korea accounted for more than half of all reports worldwide. Also worth noticing, we can see that other than the United States, all of the countries are located in Asia. As a means of comparison, the following chart was computed for CodeRed (variants .A and .B).

Country	% of attacks
US	28.59
CN	14.08
KR	8.39
DE	4.13
TW	3.98
GB	3.56
BR	3.41
ES	3.13
FR	2.99
CA	2.13
IT	1.99
IN	1.85
NL	1.28
AU	1.00
JP	1.00
TR	1.00

MX	1.00
RU	0.85
CL	0.85
IL	0.85
AT	0.85
TH	0.71
AR	0.71
PL	0.71
BE	0.57
RO	0.57
SE	0.57
DK	0.57
HK	0.57
KW	0.57
UY	0.43
CO	0.43
GR	0.43
VE	0.43
MA	0.28
KE	0.28
CH	0.28
UA	0.28
NO	0.28
IR	0.28
SI	0.28
unknown	0.28
NZ	0.14
PT	0.14

GH	0.14
PA	0.14
PR	0.14
BS	0.14
FI	0.14
PE	0.14
GE	0.14
GM	0.14
BD	0.14
TN	0.14
SG	0.14
PH	0.14
TZ	0.14
EU	0.14
CR	0.14
CU	0.14
MY	0.14
CI	0.14
VN	0.14
HT	0.14
YU	0.14
EG	0.14
CZ	0.14

First of all, we should note that in both cases the United States received the highest percentage of attacks, most likely due to its huge computer base. Also, Germany is noteworthy for its absence from the top of the Scalper/Slapper list. On the other side, due to its inclusion on the Scalper/Slapper list, we may be able to conclude Linux/FreeBSD machines are quite popular in Poland.

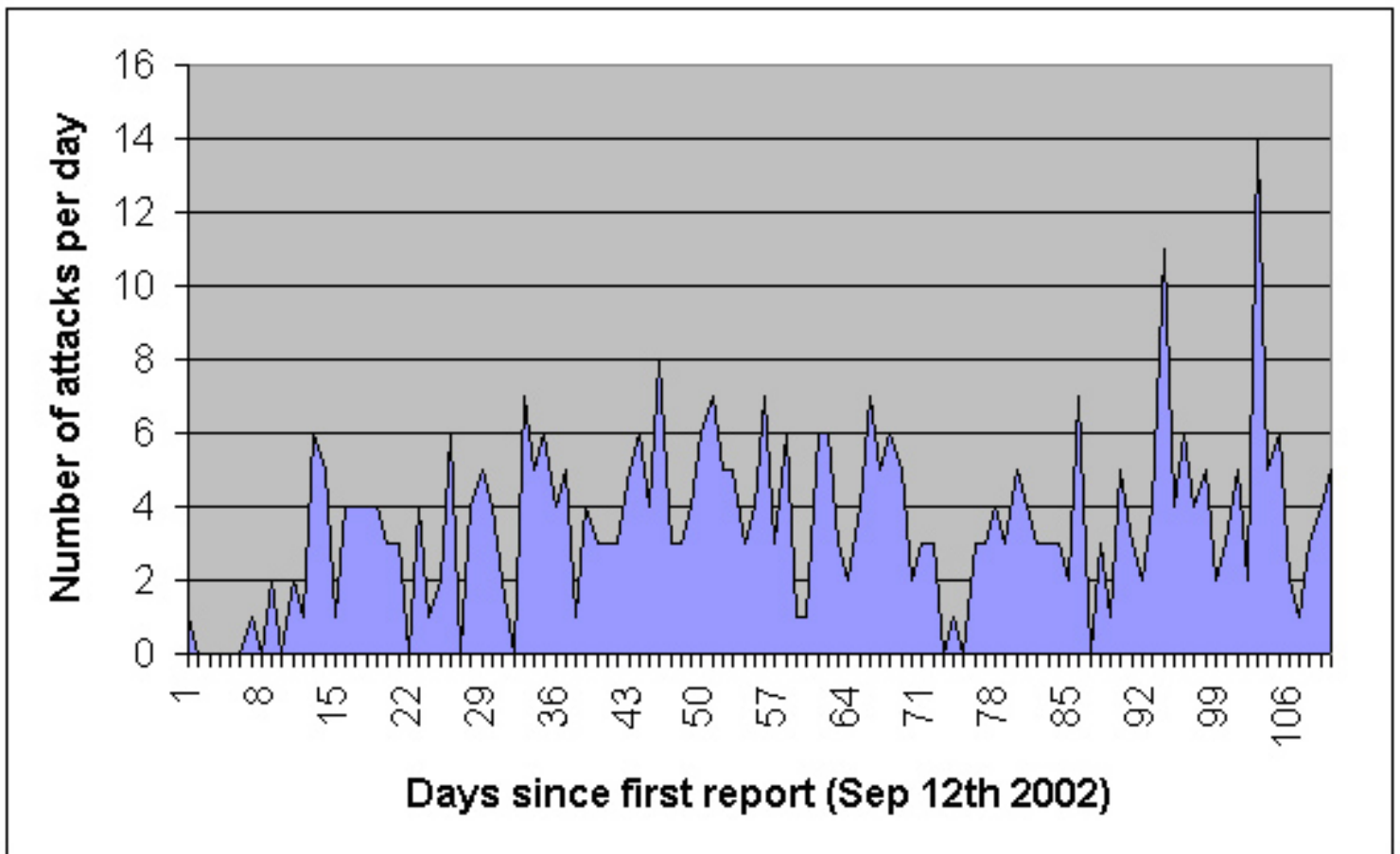
Another very interesting position is Japan. While not present on the CodeRed list amongst numerous other Asian countries, Japan holds the fourth position on the Slapper/Scalper top list. That can be interpreted in two ways: either most of the machines in Japan are running Unix clones or the security of Windows machines is at an extremely high level. It is more likely to be a combination of both.

Another way to look at the results received by Smallpot would be by sorting them according to month:

Month (2002)	Number of attacks:
August	0
September	36
October	114
November	115
December	122

Since Scalper started spreading in July 2002, but the first Smallpot report didn't come in until September, immediately after Slapper infection reports started to arrive from users, it's very likely that most, if not all, Smallpot hits are caused by Slapper-infected machines. This is also supported by the number of Linux machines deployed on the Net, which is believed to be at least one order of magnitude higher than FreeBSD installations.

On a per-day basis, the distribution of Slapper/Scalper attacks is as follows:



From the graph above we can see that it took about one week for a reasonable number of machines to get infected (and therefore start hitting Smallpot nodes) and then about one more week to reach the saturation level. For comparison, it took about one day for CodeRed.C to reach the saturation level, similarly for the MS-SQL infecting worm Spida, which gives some figures regarding the speed with which Linux viruses may spread.

Conclusions

Although not as fast spreading as their Win32 counterparts, it's obvious that Linux viruses (and especially worms) are becoming increasingly common, while FreeBSD malware is not too far behind. Couple this with older figures regarding the spread of the Solaris worm "SadMind", and we can conclude that Unix malware isn't likely to be disappearing any time soon.

One notable difference between Win32 and Unix viruses is that while most of today's Win32 malware relies on e-mail to replicate, Unix malware is still dependent on buffer overflows and popular exploits to become widespread. On the other side, given recent examples such as Slammer (on Win32), it's obvious that buffer overflows are making headway in the Windows world as well. As such, it is not unreasonable to assume that Unix viruses that replicate via e-

mail, possible using buffer overflows in known mailing programs, will also start to make their first steps on this yet unexplored territory.

[Privacy Statement](#)

Copyright 2006, SecurityFocus