The sys admin√s (and the professor's) daily grind: Do-it-yourself antispam blacklists

GARBAGE INCINERATOR

At the Niederrhein University future admins implement spam defense mechanisms by attracting the attention of the Viagra Mafia. The results are pertinacious blacklists and expert knowledge of methods for combating the menace. **BY JÜRGEN QUADE AND CHARLY KÜHNAST**

A project at Niederrhein University [1], Krefeld, Germany, prepares students for their working lives, teamwork, and the daily madness, part of which is the inflationary emergence of spam. Spam can be fought by the use of various methods, and one of them is the spam blacklist (SBL). Now students at the university are working on implementing and maintaining an SBL.

Following the idea of "Fight Spam with Spam," we deliberately set up IMAP and POP3 mail accounts that were not protected from spam. The accounts acted as honeypots to catch spam mail. To attract spammers, the students spread the honeypot email addresses as widely as they could. To do so, they ignored all the rules concerning responsible use of email addresses and published the addresses on websites in social networks; they also posted in test newsgroups such as *de.test* and visited the darkest corners of the web they could find.

It didn't take long to achieve satisfactory results: The accounts soon filled up with tons of spam. The students' assignment was to set up a system to determine the origin of the incoming messages as quickly as possible (by identifying the IP address of the sending server) and to add the spam to the blacklist for a defined period of time.

Listing 1: Policyd-weight Configuration

```
## DNSBL settings
@dnsbl_score = (
#HOST,                    BAD SCORE,  GOOD SCORE,   LOG NAME
'list.dsbl.org'            3.5,         0,           'DSBL_ORG',
'ix.dnsbl.manitu.net'      3.5,         0,           'IX_MANITU',
'sbl.hsnr.de',             3.5,         0,           'HSNR_DE',
);
```

What this aimed to achieve was to allow the mail server to compare the delivering servers' IP addresses with the blacklist when it checked its regular accounts. If the mail server noted that a delivering server had been involved in spam distribution recently, it refused to accept the email (see Figure 1).

The first step was to query the IMAP accounts automatically at regular intervals. Routines then extract the details of the mail server that delivered the spam from the mail headers. The relevant strings are located in the *Received* lines:

```
Received: from bhixhv
(wsip-70-183-106-183↩
.sd.sd.cox.net [70.183.106.183]) by ↩
islay.kuehnast.com (Postfix) with ↩
ESMTP id B3B1F5D7A3; Tue, 21 Oct 2008 ↩
20:17:43 +0200 (CEST)
```

The published address, *islay.kuehnast.com*, resolves to one of the honeypot mail servers. It is thus fairly obvious that any host that delivers mail to it will be a spam distributor – in many cases it turns out to be a compromised machine that is being misused as a drone on a botnet [2]. The name the machine uses to identify itself, *bhixhv* in this case, is spoofed – this is true of nearly all spam mail.

A reverse lookup of the IP address, *wsip-70-...*, is not necessary; the IP is just fine. The IP address is listed in the Postfix log in square brackets and thus is extracted easily with a regular expression. The same line contains the mail server timestamp, which is relevant for automatically extending entries for repeat offenders or for automatically removing an IP from the list if it does not deliver any spam within a defined period and is thus deemed to be clean.

Setting up DNS Zones

From a technical point of view, the SBL is a DNS zone combined with a DNS query to discover whether or not a server is listed on the SBL.

Now, the setup adds the IP address it has discovered to the SBL zone on the DNS:

```
183.106.183.70.sbl.hsnr.de  ↩
IN A      127.0.0.10
                              IN TXT    ↩
"Spam from this IP received: ↩
2008-10-21 20.17h"
```

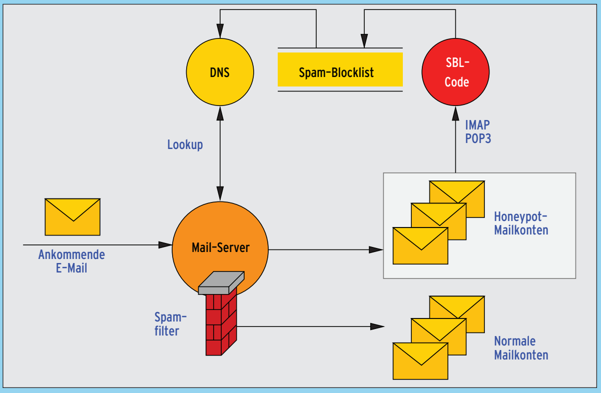

Figure 1: Mail delivered by a listed server is filtered.

SYSADMIN

Security Lessons 66
Attack of the cross-site scripting request forgeries!

Surviving NFS 3 68
We explore some of the security issues you might encounter with NFS 3.

Because queries to the SBL zone involve a reverse lookup, you need to enter the individual octets in reverse order. DNS will resolve the name for the IP *127.0.0.10*; the last octet *.10* was chosen arbitrarily, the main thing being that it is greater than *1*. Different numbers here could be used to classify the results – for example, to evaluate which honeypot provided the entry.

The string in the TXT record is stored in the mail logfile when the mail server refuses to accept an incoming email because of an SBL query. In the simplest of all cases, you would tell the mail server to drop the communication link to the sending server in the case of an SBL hit. The Postfix configuration looks like this:

```
smtpd_recipient_restrictions = ➘
[Andere_Regeln], reject_rbl_client ➘
sbl.hsnr.de, permit
```

No matter how much faith you have in the effectiveness of your do-it-yourself SBL, it is still not a good idea to drop email just on account of a list entry.

Tools such as Policyd-weight [3] for Postfix offer a more comprehensive approach by applying multiple spam detection criteria. For example, Policyd-weight can query multiple SBLs and perform other (header) checks. The Policy daemon generates a score from the results and compares the score with a configurable threshold to decide whether to accept or drop the mail.

Listing 1 shows a Policyd-weight configuration with three SBLs. If the threshold value is set to, say, *8.0*, a server has to appear in all three lists for Policyd-weight to classify it as a spammer.

Equality of Arms

Incidentally, we gave the students a free choice of weapons in implementing the SBL. The variety of solutions the students proposed just goes to prove the old saying "ask 10 computer scientists and get 11 solutions." For example, the routines to extract the relevant data from the honeypot mail included such widely disparate solutions as Bash script, PHP, DotNet, and C.

Bind 9 won the nameserver contest because of its simple configuration and multiple platform support.

But again, some participants wanted full control of the code and set about programming a miniature nameserver that offered the required functionality, but no more.

The garbage incinerator project is still running. I just wonder what the students will dream up when we come to the final step, "Monitoring and Reporting"? Can't wait to see. ■

INFO

[1] Niederrhein University, Department of Electrical Technology and Computer Science: *http://www.hs-niederrhein.de/fb03.html* (in German)

[2] "Bot Posse: An insidious spam botnet attacks Charly" by Charly Kühnast, *Linux Magazine*, August 2006, *http://www.linuxpromagazine.com/issues/2006/69/bot_posse/*

[3] Policyd-weight: *http://www.policyd-weight.org*