Tarpit (networking)

A **tarpit** is a service on a computer system (usually a server) that purposely delays incoming connections. The technique was developed as a defense against a computer worm, and the idea is that network abuses such as spamming or broad scanning are less effective, and therefore less attractive, if they take too long. The concept is analogous with a tar pit, in which animals can get bogged down and slowly sink under the surface, like in a swamp.

The original tarpit idea

Tom Liston developed the original tarpitting program *LaBrea*.[1] It can protect an entire network with a tarpit run on a single machine.

The machine listens for Address Resolution Protocol requests that go unanswered (indicating unused addresses), then replies to those requests, receives the initial SYN packet of the scanner and sends a SYN/ACK in response. It does not open a socket or prepare a connection, in fact it can forget all about the connection after sending the SYN/ACK. However, the remote site sends its ACK (which gets ignored) and believes the 3-way-handshake to be complete. Then it starts to send data, which never reaches a destination. The connection will time out after a while, but since the system believes it is dealing with a live (established) connection, it is conservative in timing it out and will instead try to retransmit, back-off, retransmit, etc. for quite a while.

Later versions of LaBrea also added functionality to reply to the incoming data, again using raw IP packets and no sockets or other resources of the tarpit server, with bogus packets that request that the sending site "slow down". This will keep the connection established and waste even more time of the scanner.

SMTP tarpits

One of the possible avenues that were considered to battle bulk-spam at one time, was to mandate a small fee for every submitted mail. By introducing such artificial cost, with negligible impact on legitimate use as long as the fee is small enough, automated mass-scale spam would instantly become unattractive. Tarpitting could be seen as a similar (but technically much less complex) approach, where the cost for the spammer would be measured in terms of time and efficiency rather than money.

Authentication procedures increase response times as users attempt invalid passwords. SMTP authentication is no exception. However, server-to-server SMTP transfers, which is where spam is injected, require no authentication. Various methods have been discussed and implemented for SMTP tarpits, systems that plug into the Mail Transfer Agent (MTA, i.e. the mail server software) or sit in front of it as a proxy.

One method increases transfer time for all mails by a few seconds by delaying the initial greeting message ("greet delay"). The idea is that it will not matter if a legitimate mail takes a little longer to deliver, but due to the high volume, it will make a difference for spammers. The downside of this is

that mailing lists and other legitimate mass-mailings will have to be explicitly whitelisted or they will suffer, too.

Some email systems, such as sendmail 8.13+, implement a stronger form of greet delay. This form pauses when the connection is first established and listens for traffic. If it detects any traffic prior to its own greeting (in violation of RFC 2821) it closes the connection. Since many spammers do not write their SMTP implementations to the specification, this can reduce the number of incoming spam messages.

Another method is to delay only known spammers, e.g. by using a blacklist (see Spamming, DNSBL). OpenBSD has integrated this method into their core system since OpenBSD 3.3,[2] with a special-purpose daemon (spamd) and functionality in the firewall (pf) to redirect known spammers to this tarpit.

MS Exchange can tarpit senders who send to an invalid address. Exchange can do this because the SMTP connector is connected to the authentication system.

A more subtle idea is greylisting, which, in simple terms, rejects the first connection attempt from any previously unseen IP address. The assumption is that most spammers make only one connection attempt (or a few attempts over a short period of time) to send each message, whereas legitimate mail delivery systems will keep retrying over a longer period. After they retry, they will eventually be allowed in without any further impediments.

Finally, a more elaborate method tries to glue tarpits and filtering software together, by filtering e-mail in realtime, while it is being transmitted, and adding delays to the communication in response to the filter's "spam likeliness" indicator. For example, the spam filter would make a "guess" after each line or after every x bytes received as to how likely this message is going to be spam. The more likely this is, the more the MTA will delay the transmission.

Background

SMTP consists of requests, which are mostly four-letter words such as MAIL, and replies, which are (minimally) three-digit numbers. In the last line of the reply, the number is followed by a space; in the preceding lines it is followed by a hyphen. Thus, on determining that a message being attempted to send is spam, a mail server can reply:

```
451-Ophiomyia prima is an agromyzid fly
451-Ophiomyia secunda is an agromyzid fly
451-Ophiomyia tertia is an agromyzid fly
451-Ophiomyia quarta is an agromyzid fly
451-Ophiomyia quinta is an agromyzid fly
451-Ophiomyia sexta is an agromyzid fly
451-Ophiomyia septima is an agromyzid fly
451 Your IP address is listed in the DNSBL. Please try again later.
```

The tarpit waits fifteen or more seconds between lines (long delays are allowed in SMTP, as humans sometimes send mail manually to test mail servers). This ties up the SMTP sending process on the spammer's computer so as to limit the amount of spam it can send.

IP-level tarpits

The Linux kernel can now be patched to allow tarpitting of incoming connections instead of the more usual dropping of packets. This is implemented in iptables by the addition of a TARPIT target.[3] The same packet inspection and matching features can be applied to tarpit targets as are applied to other targets.

Mixed SMTP-IP level tarpits

A server can determine that a given mail message is spam, e.g. because it was addressed to a spam trap, or after trusted users' reports. The server may decide that the IP address responsible for submitting the message deserves tarpitting. Cross-checking against available DNSBLs can help to avoid including innocent forwarders in the tarpit database. A daemon exploiting Linux libipq can then check the remote address of incoming SMTP connections against that database. SpamCannibal is a GPL software designed around this idea;[4] Stockade is a similar project implemented using FreeBSD ipfirewall.

One advantage of tarpitting at the IP level is that regular TCP connections handled by an MTA are *stateful*. That is, although the MTA doesn't use much CPU while it sleeps, it still uses the amount of memory required to hold the state of each connection. On the opposite, LaBrea-style tarpitting is *stateless*, thus gaining the advantage of a reduced cost against the spammer's box. However, making use of botnets, spammers can externalize most of their computer-resource costs.

Criticism

It is known that a tarpitted connection may generate a significant amount of traffic towards the receiver, because the sender considers the connection as established and tries to send (and then retransmit) actual data. In practice, given current average computer botnet size, a more reasonable solution will be to drop suspicious traffic completely, without tarpitting. This way, only TCP SYN segments will be retransmitted, not the whole HTTP or HTTPS requests.[5]

Commercial implementations of tar-pitting

As well as MS Exchange, there have been two other successful commercial implementations of the tar pit idea. The first was developed by TurnTide, a Philadelphia-based startup company, which was acquired by Symantec in 2004 for $28 million in cash.[6] The TurnTide Anti Spam Router contains a modified Linux kernel which allows it to play various tricks with TCP traffic, such as varying the TCP window size. By grouping various email senders into different traffic classes and limiting the bandwidth for each class, the amount of abusive traffic is reduced - particularly when the abusive traffic is coming from single sources which are easily identified by their high traffic volume.

After the Symantec acquisition, a Canadian startup company called MailChannels released their "Traffic Control" software, which uses a slightly different approach to achieve similar results. Traffic Control is a semi-realtime SMTP proxy. Unlike the TurnTide appliance, which applies traffic shaping at the network layer, Traffic Control applies traffic shaping to individual senders at the application

layer. This approach results in a somewhat more effective handling of spam traffic originating from botnets because it allows the software to slow traffic from individual spam zombies, rather than requiring zombie traffic to be aggregated into a class.

See also

- Turing tarpit
- Anti-spam techniques (e-mail)
- Mail-sink

References

1. Tom Liston talks about LaBrea (http://labrea.sourceforge.net/Intro-History.html)
2. Spamd's man page (https://man.openbsd.org/spamd.8)
3. [1] (https://archive.today/20130222025643/http://xtables-addons.sf.net/)
4. [2] (https://web.archive.org/web/20031014025402/http://www.spamcannibal.org/)
5. Sebastian, Walla (2019). "MALPITY: Automatic Identification and Exploitation of Tarpit Vulnerabilities in Malware". *IEEE European Symposium on Security and Privacy (EuroS&P) Security and Privacy (EuroS&P)*: 590–605.
6. "Symantec snaps up antispam firm - CNET News" (https://archive.today/20120716044720/http://news.cnet.com/Symantec+snaps+up+antispam+firm/2100-7355_3-5266548.html). Archived from the original (http://news.cnet.com/Symantec+snaps+up+antispam+firm/2100-7355_3-5266548.html) on 16 July 2012.