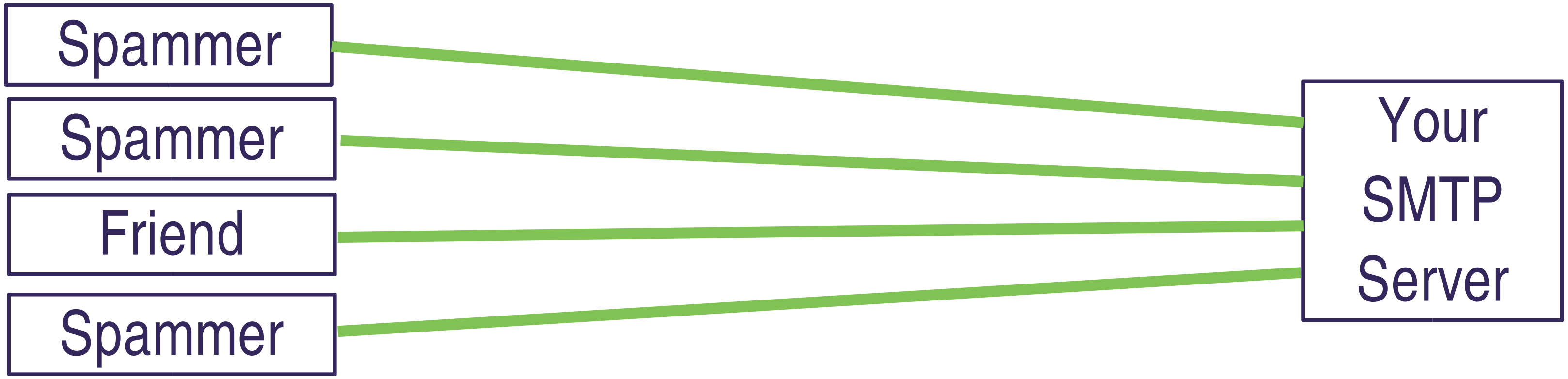
TarProxy: Lessons Learned and What's Ahead

Marty Lamb

Martian Software, Inc.

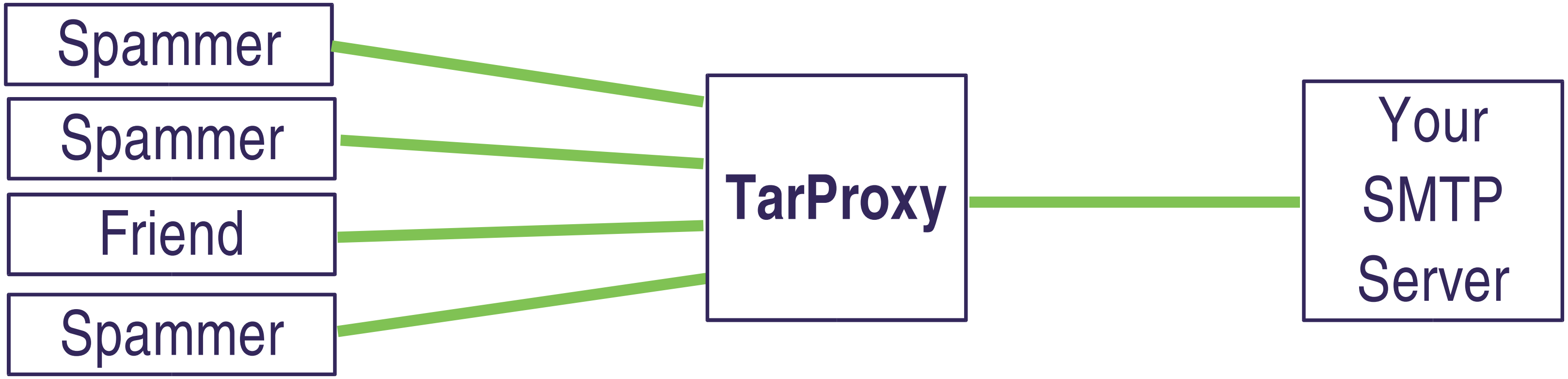
Downingtown, PA

TarProxy is an inbound SMTP proxy...



TarProxy is an inbound SMTP proxy...

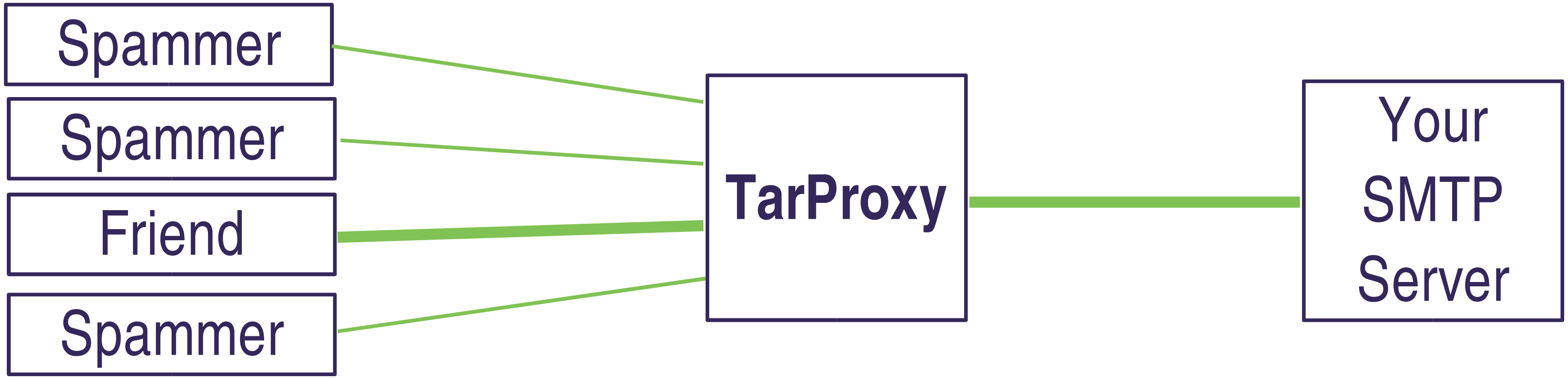
placed between the Internet and your existing SMTP server...



TarProxy is an inbound SMTP proxy...

placed between the Internet and your existing SMTP server...

that throttles the incoming connection if/when it detects spam.



Why?

To slow down spam transmission, hitting spammers where it counts. Fewer spams sent -> fewer dollars received.

To make open relays fill with spam, hopefully waking up the sleeping administrators.

To cause issues at open proxies, hopefully waking up THOSE sleeping administrators.

To take additional actions at SMTP time in order to handle spam as far upstream as possible.

Reject at SMTP Time

```
554 I don't need any Viagra.  Go away.
```

Tempfail

```
451 I'm tired of this.  Spam me later.
```

(Note that this is not very false-positive friendly)

Really Tarpit

```
451-Your spam is important to us.  Please stay on the line...
451-Your spam is important to us.  Please stay on the line...
451-Your spam is important to us.  Please stay on the line...
451-Your spam is important to us.  Please stay on the line...
451-Your spam is important to us.  Please stay on the line...
451-Your spam is important to us.  Please stay on the line...
451-Your spam is important to us.  Please stay on the line...
451-Your spam is important to us.  Please stay on the line...
```

(Note that this is false-positive hostile!)

Launch External Processes

Drop off the face of the earth from the spammer's perspective

```
iptables -A INPUT -s $REMOTE_IP -p tcp –destination-port 25 -j DROP
```

Let others benefit from your analysis

```
cat $MSGFILE | razor-report
```

Poke the spammer in the eye (not yet implemented)

What IS Spam?

Question

- How does TarProxy determine if a message is spam?

Answer

- It doesn't.

Better Answer

- The determination is made by an external classifier/filter

Initial Approach

A straightforward proxy

- 1-1 connection mapping
- SMTP-aware, sent tokens + context to plugins
  - Message body
  - Metatokens
    - Sender attempted nonexistent address
    - Sender attempted to relay
    - Sender used your hostname in HELO/EHLO
    - Sender is on a DNSBL

Ran in a production environment in a small office for approximately two months

Lesson 1: Don't Tarpit Yourself

1-1 connection mapping resulted in HUGE numbers of concurrent connections to production mailserver.

New connections were being made far faster than they were being released.

Solution: Spool and classify messages entirely before forwarding to production server. Forward to internal SMTP server last.

Neat Side Effect: Better scaling. Several TarProxies on cheap servers can proxy to the same production server.

Lesson 2: Expect Classifiers To Do Less

The initial API was powerful, but required extensive customization in classifiers (incremental classification, etc.)

Solution: Only rely on typical features, such as:

- Read a message
- Write a filtered version of the message
- Perform some header gymnastics

Send TarProxy-specific metatokens to classifiers via header insertion.

Neat Side Effect: Immediate availability of compatible classifiers.

Lesson 3: Allow Classifiers To Do More

Initial API did not allow classifiers to modify the message.

Solution: Just use a simple filter interface for classifiers. Message goes in, message comes out.

Neat Side Effect: Classifier Chains.

Another Neat Side Effect: No need to classify again on client side.

Lessons Applied

1. Receive a message and spool it. Don't acknowledge yet.
2. Add SMTP metatokens as TarProxy-specific headers
3. Pass entire message through classifier(s)
4. Process headers on final message
5. Take appropriate action (accept, reject, tempfail, delay acknowledgment, etc.)
6. Go to 1 or quit.

TarProxy-Specific Header Examples

```
X-TarProxy-RemoteIP: 68.82.9.23

X-TarProxy-HELO: mail.yahoo.com

X-TarProxy-MAIL_FROM: sdbwqgc@telpage.net

X-TarProxy-RCPT_TO: someone@your_domain.com

X-TarProxy-RCPT_TO: someone@not_your_domain.com

X-TarProxy-RCPT_TO-failure: 550 we do not relay
```

Taking Action

TarProxy's actions are governed by matching patterns against the post-classified message's headers. These patterns are called triggers.

A trigger consists of a fieldname regex and a field value regex. If both match the same header, the trigger's commands are executed in order.

Trigger Example

```
header “X-Text-Classification” matches “.*spam.*” {
   # reduce connection speed by half
   adjustThrottle -50%
   # don't even ack the DATA command for 30 seconds
   addPenalty 30000
}
header “X-Text-Classification” matches “.*virus.*” {
   drop
   tarpit
}
```

Caveats

- Increases network overhead
  - (SO_RCVBUF/SO_SNDBUF are tiny)
- Even nonspam is received at least a little slower
- Internal SMTP server must disallow relaying from TarProxy's internal IP address!

Challenges

STARTTLS and other SMTP extensions

POP-before-SMTP

Whose corpus should be used in classification?

- Is anyone working on combining corpora?

Certainly several more...

Current Status

Open Source (GPL)

Still under development. Want to help?

- Code
- Docs
- Testing

http://www.martiansoftware.com/tarproxy

tarproxy @ martiansoftware . com

MLamb @ martiansoftware . com

Mailing list instructions on website