

# Spam Control on the Web

Paul M. Winkler  
PyGotham II  
June 2012

Me: Paul Winkler  
slinkp@gmail.com  
programmer at OpenPlans  
<http://openplans.org>



# The Problem

- Comment / post spam evolves
- There will never be a silver bullet
- Old projects in maintenance mode get increasingly hard to keep spam-free

"An Internet service cannot be considered truly successful until it has attracted spammers."

- Rafe's law

<http://rc3.org/2006/10/20/rafes-law/>

# The Story of Btwlzyq

Tracking a resourceful spammer on  
<http://communityalmanac.org>.

# Existing Solutions

A brief taxonomy / survey

See also appendix B - prior art.

# Existing Solutions: Form Modifiers

- Captcha (obtrusive)
- Other problem solving (obtrusive)
- Honeytrap Fields (unobtrusive)

By design, only useful against bots.

# Content Filtering Services

Remote APIs:

- Bayesian filters
- IP blacklist

Good: Unobtrusive. Large training set.

Bad: Network overhead. Reliability.



# Content filters: Local

- Bayesian filters and IP blacklists, but also:
- IP throttling
- Other metadata filtering: link counting, admin users, ...

Good: Unobtrusive. Low I/O overhead. Reliable.

Bad: Requires training.

# Summary

- Lots of solutions
- None are sufficient alone
- Complement each other
- $n$  solutions ==  $n$  APIs
- $n$  APIs integrate into  $m$  web apps == aaargh

# Trac SpamFilter plugin - a flexible approach

- "all of the above" approach
  - 14 filters and 3 captchas
- extensible, easy to code new filters
- highly configurable
  - select filters
  - assign karma scores to filters (positive = good)
  - set minimum karma needed for posting

# Trac SpamFilter plugin (cont'd)

- Records possible spam in database for moderation
- Moderation UI: rough but useful
- Moderation includes training
- Lots of tests

- Home
- Download
- Documentation
- Mailing Lists
- License
- FAQ

# Administration

## General

- Basic Settings
- Logging
- Permissions
- Plugins

## Spam Filtering

- Akismet
- Bayes

## Configuration

## Monitoring

## Ticket System

- Delete Changes
- Components
- Delete
- Milestones
- Priorities
- Severities
- Ticket Types
- Versions

## Spam Filtering: Configuration

### Karma Tuning

Minimum karma required for a successful submission:

*Content submissions are passed through a set of registered and enabled filter strategies, each of which check the submitted content and may assign karma points to it. The sum of these karma points needs to be greater than or equal to the minimum karma configured here for the submission to be accepted.*

Strategy	Karma points	Description
AkismetFilterStrategy	<input type="text" value="1"/>	By how many points an Akismet reject impacts the overall karma of a submission.
BayesianFilterStrategy	<input type="text" value="5"/>	By what factor Bayesian spam probability score affects the overall karma of a submission.
ExternalLinksFilterStrategy	<input type="text" value="2"/>	By how many points too many external links in a submission impact the overall score.
IPBlacklistFilterStrategy	<input type="text" value="5"/>	By how many points blacklisting by a single server impacts the overall karma of a submission.
IPThrottleFilterStrategy	<input type="text" value="2"/>	By how many points exceeding the configured maximum number of posts per hour impacts the overall score.
RegexFilterStrategy	<input type="text" value="5"/>	By how many points a match with a pattern on the BadContent page impacts the overall karma of a submission.
SessionFilterStrategy	<input type="text" value="9"/>	By how many points an existing and configured session improves the overall karma of the submission. A third of the points is granted for having an existing session at all, the other two thirds are granted when the user has his name and/or email address set

## CVS Subversion CVSNT

Transform them  
into a secure  
real time  
multisite  
development  
solution

[www.wandisco.com](http://www.wandisco.com)

[Advertise on this site](#)

Home  
Download  
Documentation  
Mailing Lists  
License  
FAQ

# Administration

## General

[Basic Settings](#)  
[Logging](#)  
[Permissions](#)  
[Plugins](#)

## Spam Filtering

[Akismet](#)  
[Bayes](#)  
[Configuration](#)  
[Monitoring](#)

## Ticket System

[Delete Changes](#)  
[Components](#)  
[Delete](#)  
[Milestones](#)  
[Priorities](#)  
[Severities](#)  
[Ticket Types](#)  
[Versions](#)

## Spam Filtering: Logs

Viewing entries 61-75 of 466.

[← Previous Page](#) | [Next Page →](#)

	Path	Author	Karma	IP Address	Date/time
<input type="checkbox"/>	<b>Putaspannerintheworks</b>	/ticket/2324	<b>-31</b>	202.101.6.85	11/01/2006 06:26:28 PM
	<ul style="list-style-type: none"> <li>IPBlacklistFilterStrategy (-5): IP 202.101.6.85 blacklisted by bsb.empty.us</li> <li>ExternalLinksFilterStrategy (-14): Maximum number of external links per post exceeded</li> <li>BayesianFilterStrategy (-5): SpamBayes determined spam probability of 100.00%</li> <li>AkismetFilterStrategy (-2): Akismet says content is spam</li> <li>RegexFilterStrategy (-5): Content contained blacklisted patterns</li> </ul> <p><i>Putaspannerintheworks [url=http://cvbxcvb.cv.funpic.de]master ...</i></p>				
<input type="checkbox"/>	<b>DishfitforthegodsA</b>	/ticket/1830	<b>-22</b>	202.101.6.85	11/01/2006 06:23:42 PM
	<ul style="list-style-type: none"> <li>IPBlacklistFilterStrategy (-5): IP 202.101.6.85 blacklisted by bsb.empty.us</li> <li>ExternalLinksFilterStrategy (-10): Maximum number of external links per post exceeded</li> <li>BayesianFilterStrategy (-5): SpamBayes determined spam probability of 100.00%</li> <li>AkismetFilterStrategy (-2): Akismet says content is spam</li> </ul> <p><i>DishfitforthegodsA [url=http://telefonare.te.funpic.de]finanza ...</i></p>				
<input type="checkbox"/>	<b>qerrosasde</b>	/ticket/222	<b>-26</b>	67.94.174.220	11/01/2006 06:22:15 PM
	<ul style="list-style-type: none"> <li>ExternalLinksFilterStrategy (-14): Maximum number of external links per post exceeded</li> <li>BayesianFilterStrategy (-5): SpamBayes determined spam probability of 100.00%</li> <li>AkismetFilterStrategy (-2): Akismet says content is spam</li> <li>RegexFilterStrategy (-5): Content contained blacklisted patterns</li> </ul> <p><i>qerrosasde [url=http://cvbxcvb.cv.funpic.de]master ...</i></p>				

**CVS**  
**Subversion**  
**CVSNT**

Transform them  
into a secure  
real time  
multisite  
development  
solution

[www.wandisco.com](http://www.wandisco.com)

[Advertise on this site](#)

# About the Trac plugin

<http://www.cmlenz.net/archives/2006/11/managing-trac-spam>

SpamAssassin has a similar multi-filter strategy, but is designed for use with email, not web:

<http://wiki.apache.org/spamassassin/BlogSpamAssassin>

# Filters - remote

stopforumspam.py

akismet.py

blogspam.py

defensio.py

extlinks.py

httpbl.py

ip\_blacklist.py

linksleeve.py

typepad.py



# Filters - local

regex.py

bayes.py

extlinks.py

ip\_blacklist.py

ip\_regex.py

ip\_throttle.py

session.py

# Captcha

recaptcha.py

image.py (uses PIL)

expression.py ("what is three plus twelve") ... looks unfinished

It only works with Trac.

# Hamage Control!

- Goal: Decoupling SpamFilterPlugin from Trac
- Prototype
- <http://github.com/slinkp/hamage>

# Hamage Control: modes of operation

- Python library API
- WSGI middleware
- Hybrid
- Native integration with every framework
  - Nooo.

# Python API: Filters

```
class MyFilter(object):  
    def test(self, req, author, ip):  
        "return (score, 'reason')"
```

Positive score = ham, negative =  
spam.

# Python API: FilterSystem

```
>>> from hamage.filter import FilterSystem
>>> config = {
...     'options': {'min_karma': 1},
...     'filters': ['hamage_extlinks']}
>>> config['options']['backend_factory'] =
'django_orm'
>>> filtersys = FilterSystem(config)
```

# Python API: FilterSystem

```
>>> filtersys.strategies
```

```
[<hamage.filters.extlinks.ExternalLinksFilterStrategy object at ...>]
```

```
>>> filtersys.backend_factory
```

```
<class 'hamage.backends.django_hamage.models.DjangoBackendFactory'>
```



# Python API: FilterSystem

```
>>> from hamage.filter import Request
>>> req = Request.blank('/foo',
...remote_addr='10.20.30.40')
>>> filtersys.test(req, author='fred',
...    changes=[('Old content', 'New content')])
Traceback (most recent call last):
...
hamage.filter.RejectContent: Submission rejected as
potential spam
```

# Python API: FilterSystem

```
>>> filtersys.min_karma = 0
>>> filtersys.test(req,
...     author='fred',
...     changes=[('old content', 'New
content')])
(0, [])
```

# Python API: FilterSystem

```
>>> lotsa_links = 'http://somewhere.org ' * 100
```

```
>>> filtersys.test(req, author='fred',  
...               changes=[(None, lotsa_links)])
```

```
Traceback (most recent call last):
```

```
...
```

```
hamage.filter.RejectContent: Submission rejected as  
potential spam (Maximum number of external links per  
post exceeded)
```

# Python API: Registering filters

```
# Put entry points in your setup.py
setup(name='hamagecontrol',
      entry_points={
          'hamage_filters': [
              'hamage_extlinks =
hamage.filters.extlinks:ExternalLinksFilterStrategy',
          ],
          'hamage_backends': [
              'django_orm
=hamage.backends.django_hamage.models:DjangoBackendFactory',
          ],
      },
      ...)
```

# WSGI Middleware

Request: Client POST → hamage (filtering) → application

Response: application response → hamage (form field and error message injection) → client

# Demo

- Django running via wsgiref server, behind WSGI middleware

# Wish List: RESTful web service?

- Use with any language
- Scale independently
- Would love to do this
- ... later

# Performance & Scaling

- Run cheapest filters first; allow them to short-circuit.
- Parallelize slow filters (eg. network IO)
  - How?
- Asynchronous operation



# More about async

- Use case: speed
- Use case: moderation
  - Integration just got tougher
  - Feedback just got really hard
    - Don't bother?

# More on logging

- Remember Btwlzyq?
- Consistency matters

# Parting shot

```
<a href="http://example.com" rel="nofollow">  
  buy my cheap replica rolexes</a>
```

No page rank for you buddy

# Appendix A. Links

- Hamage Control: <https://github.com/slinkp/hamage>
- Django integration demo code:  
[https://github.com/slinkp/pygotham\\_hamage\\_demo](https://github.com/slinkp/pygotham_hamage_demo)
- These slides:  
[https://github.com/slinkp/pygotham\\_hamage\\_demo/blob/master/pygotham2\\_hamage\\_slides.odp?raw=true](https://github.com/slinkp/pygotham_hamage_demo/blob/master/pygotham2_hamage_slides.odp?raw=true)
- Trac plugin: <http://trac.edgewall.org/wiki/SpamFilter>
- About WSGI: <http://lucumr.pocoo.org/2007/5/21/getting-started-with-wsgi/>
- About entry points: <http://stackoverflow.com/a/9615473/137635>

# Appendix B. Prior Art

Python packages related to spam. Too many for one slide, see [https://gist.github.com/2896944#file\\_prior\\_art.txt](https://gist.github.com/2896944#file_prior_art.txt)