High voltage protection for your webapps

Author: Edward Fjellskål & Kacper Wysocki

Date: October 23, 2012

Contents

Who R US?

Who R US?

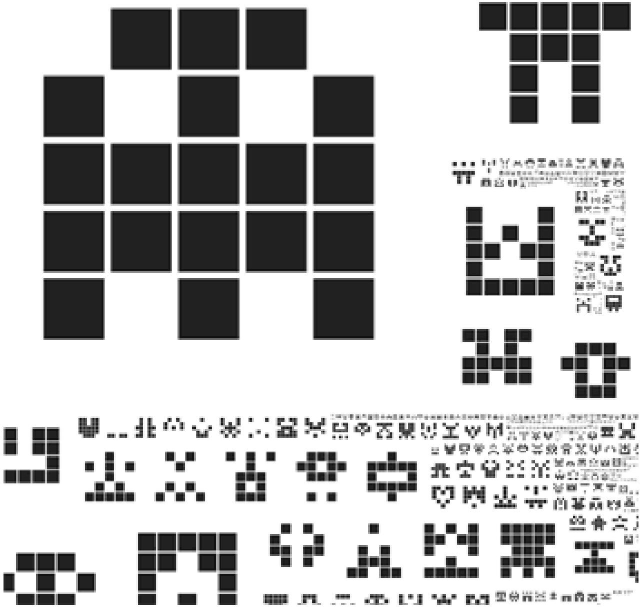
- Edward B. Fjellskål, analyst
- Kacper Wysocki, consultant
- Kristian Lyngstøl, developer
- Eduardo Scarpellini, master student

Storytime

Varnish got prod ready.

- sweet architecture
- good acceleration
- flexible control and transparency
- security barriers

Swiss army katana



B.L

```
I fear not the man
who has practiced 10,000 kicks
once,
but I fear the man who has
practiced one kick
10,000 times.
```

Varnish can haz

- simple hackable rules
- block webscanners etc
- insignificant performance penalty

LOLBUNNY

Hai, I iz in ur kitchin,
hacking teh http
fixing ur bugs
1
ICANHASCHEEZBURGER.COM

There are no silver bullets

Usually we use varnish to:

- scale a website
- hack around application issues

There are no silver bullets!

Aaaand then!

we were under siege!

- Arab News *DDoS*

Solved it!

- with a clever caching hack

```
if(req.http.attack ~ "pattern"){
   set resp.http.Cache-Control =
                "max-age=3312315123166";
}
```

The app stack

- tomcat
- varnish

Mod_security

- add another service? no
- needs monitoring, support, integration
- moar $$$!
- overkill?
- wouldn't save us from the DDoS

Other WAFs?

- not impressed, but why?
- too expensive
- not flexible enough
- no custom app rules

Instead

- write rules
- in varnish..

Instead

- write rules
- in varnish..where
- the ops can get at it

Instead

- write rules
- in varnish..where
- the ops can get at it
- without hurting feelings

Instead

- write rules
- in varnish..where
- the ops can get at it
- without hurting feelings
- and breaking things

Rules

- small rulesets
- custom rules for apps

Rulez

- customer still up

Rulez

- customer still up
- running f!$#% code

Rulez

- customer still up
- running f!$#% code
- hack quick hotfixes

The Approach

The Approach

- security rule framework in VCL
- expert rules for GET
- POST handling

Thwarting attacks

Thwarting attacks

what we do today:

- common malicious access
- SQL injection
- Cross site scripting
- Cloaking

Cloaking

- cloak web stack
- cloak client
- example: http://u.delta9.pl

Cloud

- yes we hate the word
- easy deploy VSF
- enforce rules and ACLs
- push firewall nearer user

diff -u mod_sec VCL

- we show you the diff

mod_security

```
SecRule REQUEST_COOKIES|!REQUEST_COOKIES:/__utm/|
REQUEST_COOKIES_NAMES|ARGS_NAMES|ARGS|XML:/*
"http:\/\/[\w\.]+?\/.*?\.pdf\b[^\x0d\x0a]*#" \
  "phase:2,rev:'2',ver:'OWASP_CRS/2.2.6',maturity:'9',
  accuracy:'9',capture,t:none,t:htmlEntityDecode,
  t:compressWhiteSpace,t:lowercase, ctl:auditLogParts=+E,block,
  id:'950018', setvar:'tx.msg=%{rule.msg}',
  setvar:tx.anomaly_score=+%{tx.critical_anomaly_score},
  setvar:tx.%{rule.id}-OWASP_CRS/WEB_ATTACK/UPDF_XSS-%{matched_var_name}=%{tx.0}"
```

VFW

```
if (req.url ~ "(?i)(S|%[57]3)(\s|%20|\t|%09|\+)*(C|%[46]3)
     (\s|%20|\t|%09|\+)*(R|%[57]2)(\s|%20|\t|%09|\+)*
     (I|%[46]9)(\s|%20|\t|%09|\+)*(P|%[57]0)(\s|%20|\t|%09|\+)*
     (T|%[57]4)(\s|%20|\t|%09|\+)*(>|%3E)") {
   set req.http.X-VFW-Threat = "Cross-site Scripting";
   set req.http.X-VFW-RuleID = "xss.xss-2";
   call vfw_main;
}
```

$@%^&*(!

completely unreadable

- and these are the *short ones*!

VFW

```
if (req.url ~ "(SHOW|DROP|CREATE) (DATABASES|TABLES|PROCESSLIST)") {
  set req.http.X-SEC-RuleName = "SQL Injection";
  set req.http.X-SEC-RuleId = "sql-15"
  call sec_sql_sev1;
}
```

Varnish Security FireWall

Security handlers

- reject, alert
- redirect
- g-line / offensive script
- honeypot
- block

Security handlers

Weaknesses

- WAF evasion
- WAF fingerprinting
- WAF spray

A note on patterns

```
All fixed set patterns
are incapable
of adaptability or pliability.
The truth is outside of
all fixed patterns.

  -- Bruce, again.
```

Future work

- Unicode normalization
- Fuzzing!
- Better GUI

Future

Chronicle / Deanne Fitzmaurice

Questions?

http://github.com/comotion/VSF

edwardfjellskaal@gmail.com kwy@redpill-linpro.com

References:

- http://www.varnish-cache.org/trac/wiki
- http://github.com/comotion/security.vcl
- http://github.com/scarpellini/VFW