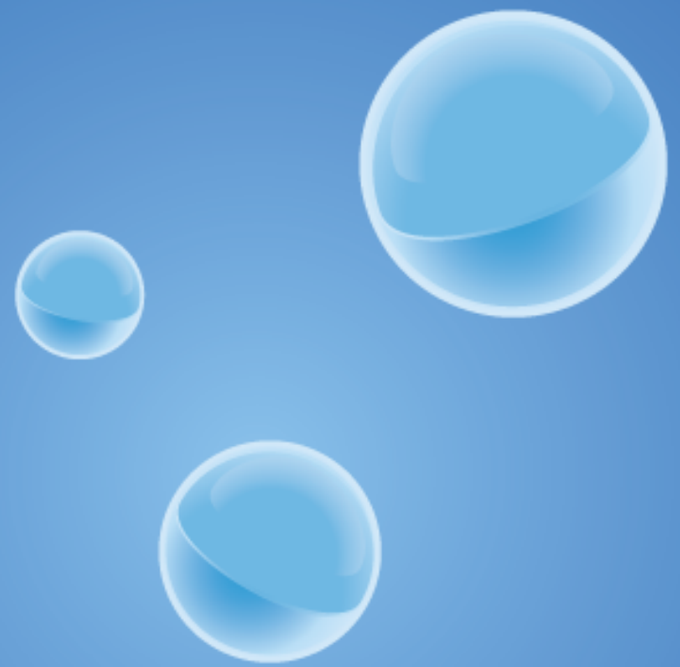
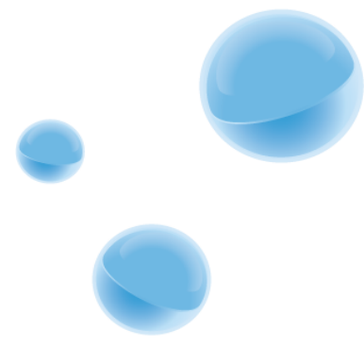


VARNISH

Makes Websites Fly

Securing your web stack with Varnish

Kacper Wysocki



slides from by Eduardo Scarpellini

hop to other slides,

we will come back here

soon enough

VFW / Sec Rules / **SQLi**

- Based on filtering of SQL queries.
  - Insensitive;
  - Encoding: ASCII, querystring;
  - space (%20, +) or tabs (%09) as spaces;

- Ex.: prevention of “OR [0-9]=[0-9]” injection.

```
if (req.url ~ "(?i)(O|%4F|%6F)(R|%52|%72)(%20|%09|\+)+(%22|%27)?(%20|%09|\+)*\d+(%20|
%09|\+)*(%22|%27)?(%20|%09|\+)*(%3D|%3C|%3E)+(\%20|%09|\+)*(%22|%27)?(%20|%09|\+)*\d+")
{


    error 403 “SQL-Injection";


}
```

VFW / Sec Rules / **XSS**

- Based on filtering of *JavaScript* functions and HTML “bad”-tags (script, img, embed, object, applet, etc).
  - Insensitive;
  - Encoding: ASCII, querystring, UTF-8;
  - *JavaScript*/browsers offers a lot of possibilities combining different sintaxes and encodings – http://ha.ckers.org/xss.html;
    - spaces, tabs and \n (CRLF) inside the tags – between chars;
  - Muuuch more complex!

- Ex.: prevention of “<script>” tag injection:

```
if (req.url ~ "(?i)(<|(%|&#x)3C)?((%|&#x)(09|10|13|20))*(S|(%|&#x)[57]3)((%|&#x)(09|10|13|
20))*(C|(%|&#x)[46]3)((%|&#x)(09|10|13|20))*(R|(%|&#x)[57]2)((%|&#x)(09|10|13|20))*(I|(%|&#x)
[46]9)((%|&#x)(09|10|13|20))*(P|(%|&#x)[57]0)((%|&#x)(09|10|13|20))*(T|(%|&#x)[57]4)((%|&#x)
(09|10|13|20))*(%3A|(>|(%|&#x)3E))") {

 error 403 ”Cross-site Scripting";

}
```

VFW / **Tests** / Scenarios

- *OWASP Broken Web Applications* used to construct 2 identical sets, each witch 20 vulnerable apps.
  - *WordPress*, *Gallery2*, *Joomla*, *AWStats*, *TikiWiki*, etc.

- Scenario 1: *Varnish* configured **without** the sec-filters.
  - As seen on most ISP's.
- Scenario 2: *Varnish* configured **with** sec-filters.

- *OWASP Zed Attack Proxy* used as a pentest tool in both scenaries.

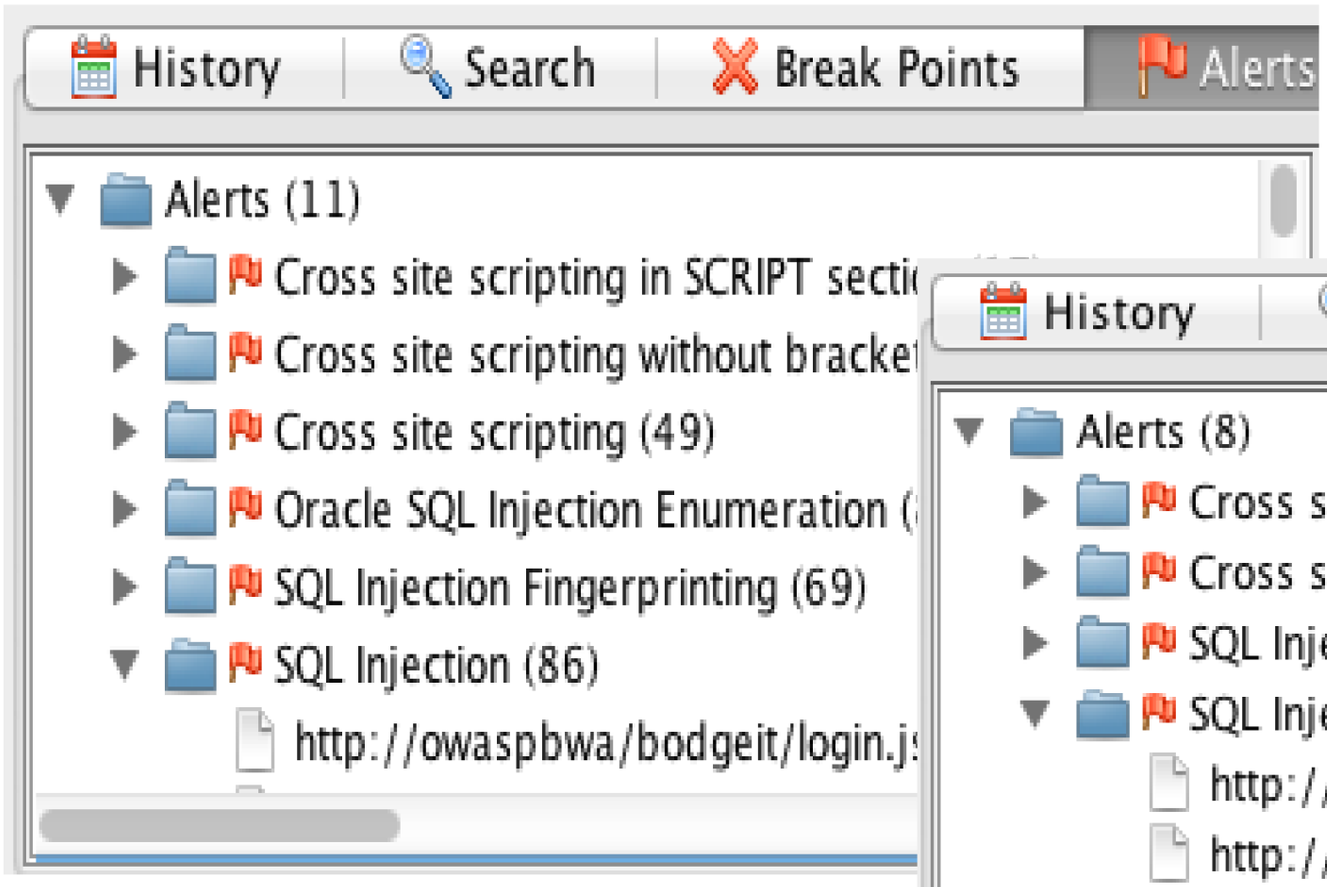
VFW / **Tests** / Numbers

- **Scenario 1** (235 suspected vulnerabilities):
  - XSS: 72 vulnerabilities;
  - SQLi: 163 vulnerabilities;
- **Scenario 2** (63 suspected vulnerabilities):
  - XSS: 19 vulnerabilities;
  - SQLi: 44 vulnerabilities;
  - **Decrease of 73%**, approximately;
  - Some types of attacks were completely neutralized by security filters.

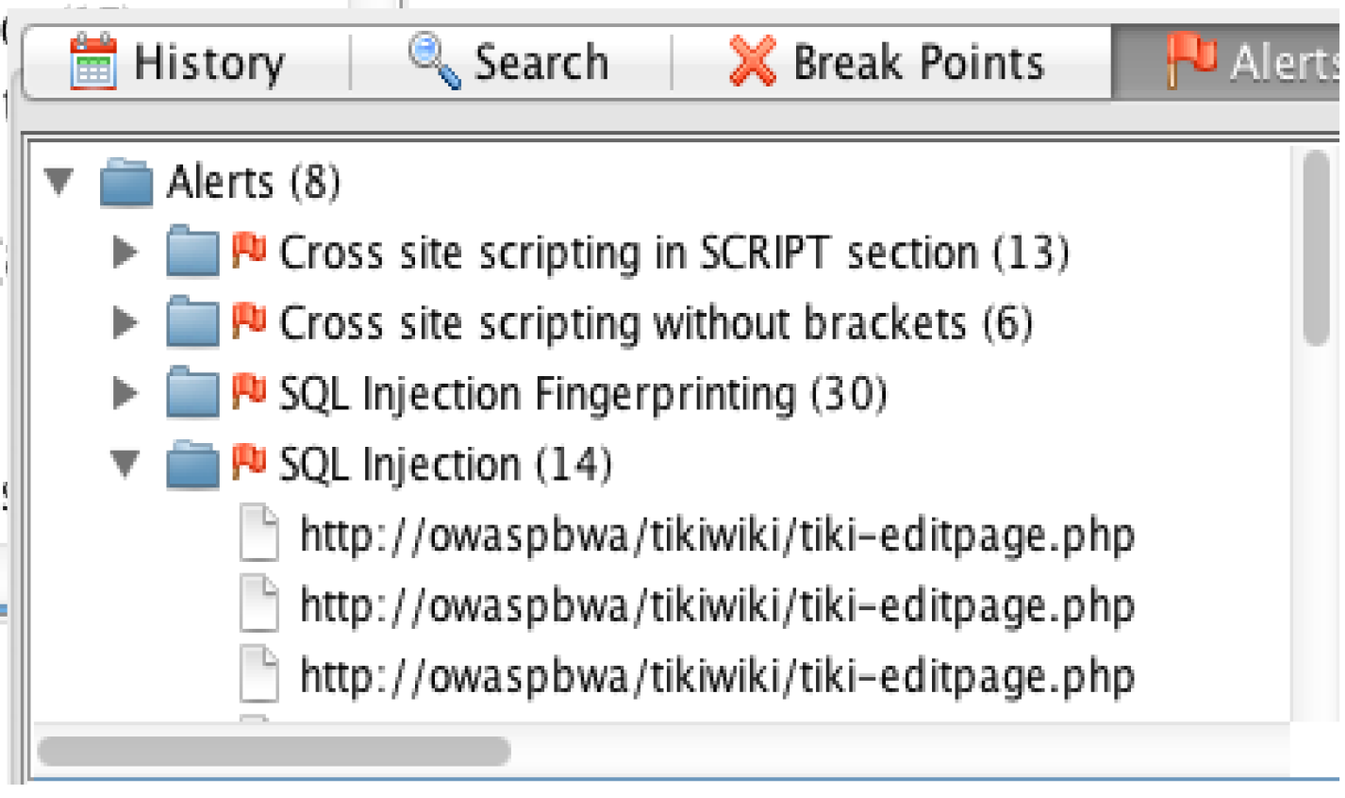
VFW / **Tests** / Numbers

***OWASP Zed Attack Proxy* - Screenshots**

Scenario 1



Scenario 2



VFW / Results / **Considerations**

- **Weaknesses**:
  - The approach (pattern-match) is susceptible to generating false positives;
  - High complexity of sec-filters for XSS (infinite possibilities of encodings and syntaxes);
  - Analysis of just 8Kb of HTTP POST (only for “application/x-www-form-urlencoded”);

- **Improvements**:
  - Standardization/normalization of request data before the application of filters (simplification of XSS regexps);
  - **We really need an “official” way to inspect the POST data – provided by Varnish.**
    - POST.vcl is quick’n’dirty hack around HTC_Read(), sp->htc->fd;