ERNW
providing security.

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

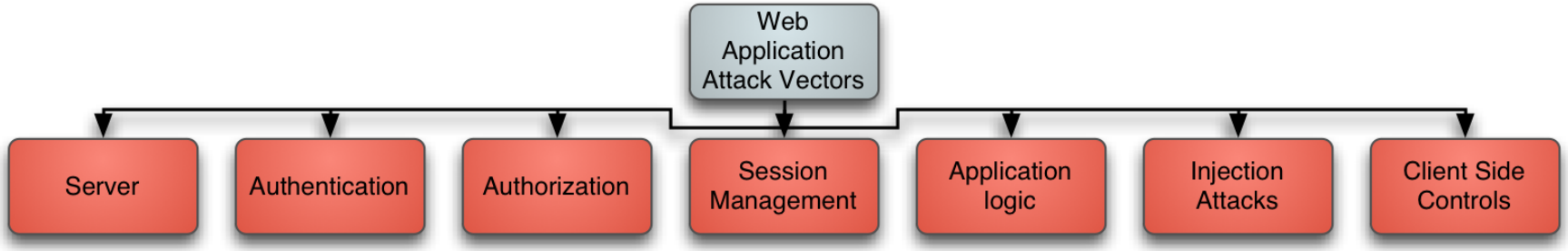
Head of Application Security & Chief Security Officer, ERNW GmbH

Contact:
ERNW GmbH
Michael Thumann
Breslauer Str. 28
69124 Heidelberg
Email: mthumann@ernw.de

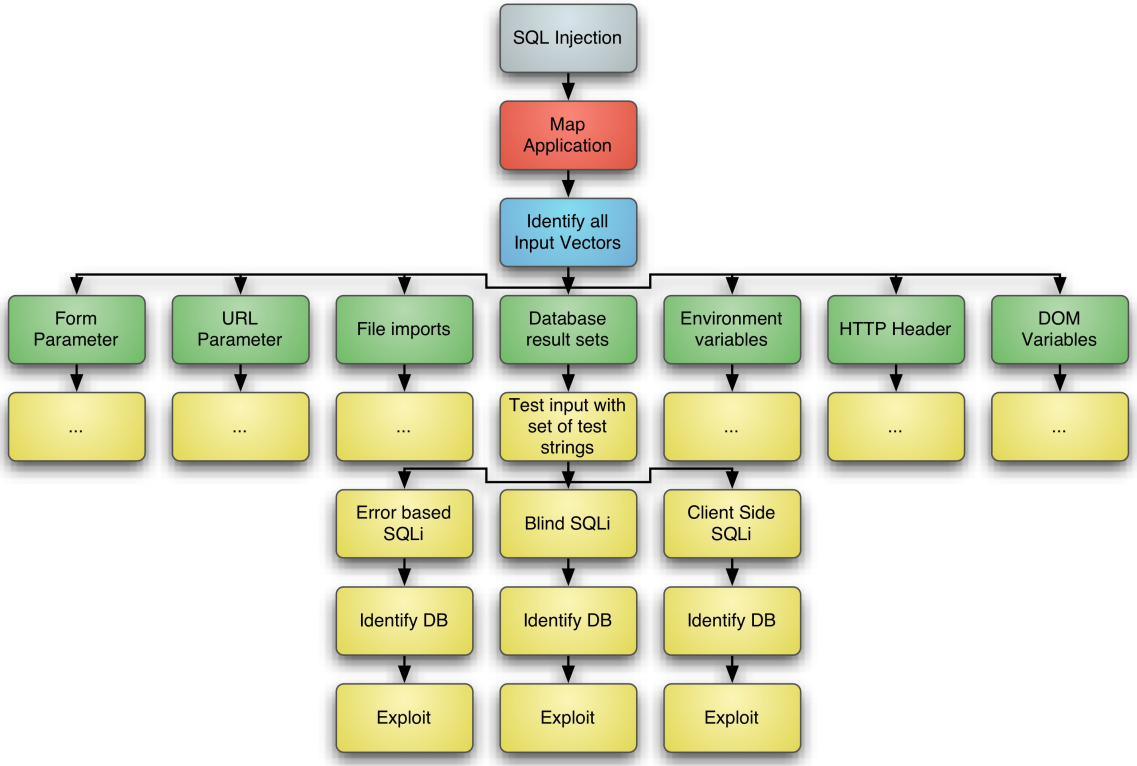
Attack Vectors

Web Application

Web Application Attacks



SQL Injection Example



ERNW
providing security.

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

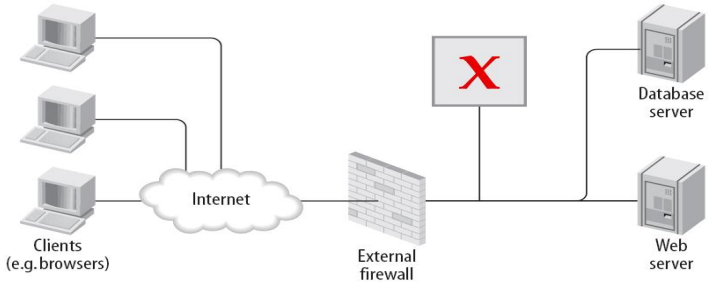
Web Application Firewalls

- Detection mechanisms
  - Negative Detection Model
  - Positive Detection Model
  - Additional Mechanisms
- Possible deployments
  - Reverse Proxy
  - Layer 2 Bridge
  - Network monitor
  - Server based
  - Embedded

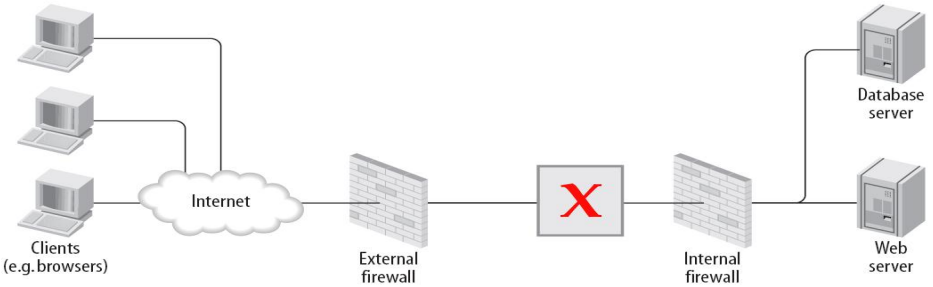
Detection Mechanisms

- 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

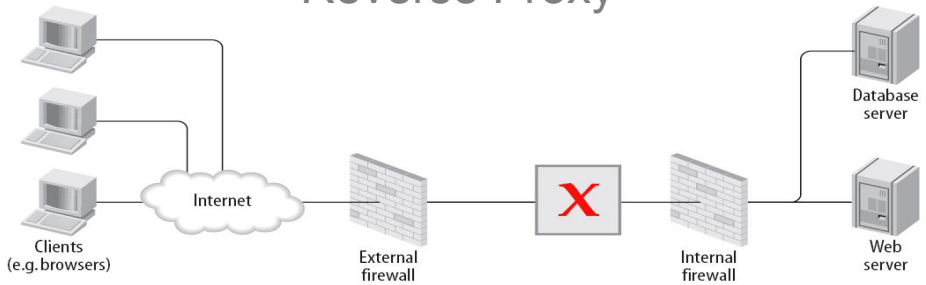


Network Monitor



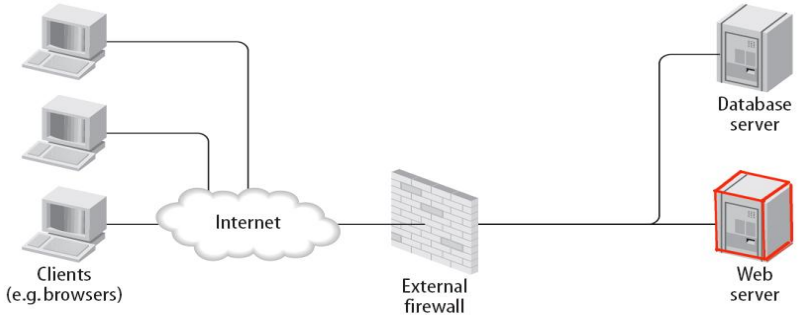
Layer 2 Bridge

Reverse Proxy



WAF deployments

Server based

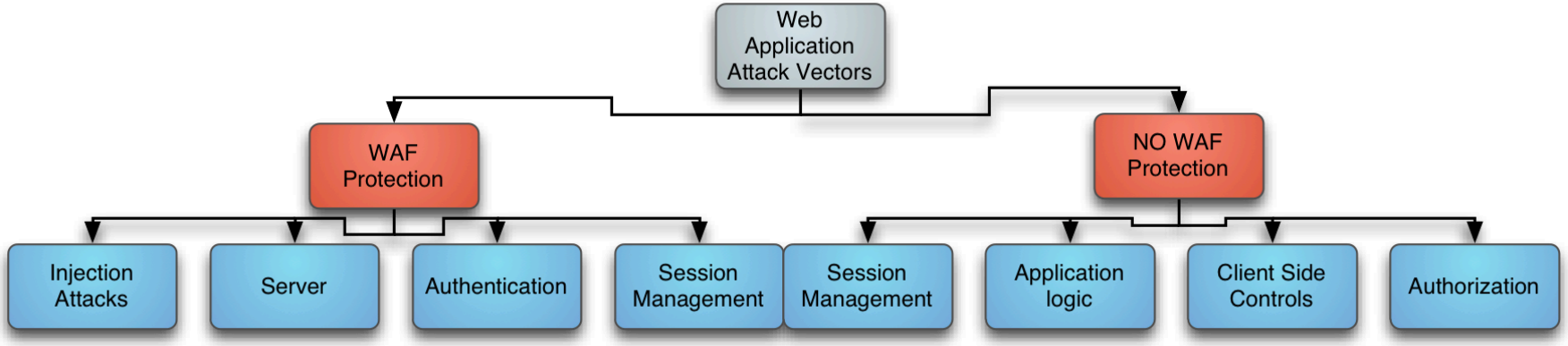


Embedded

but ...

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

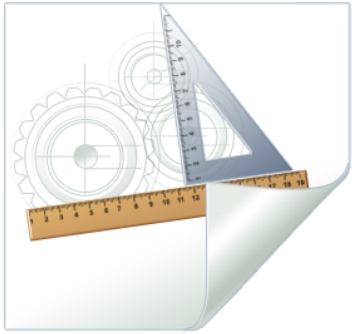
Vulnerabilities that could be mitigated by a WAF



Fingerprinting

Web Application Firewalls

WAFs covered in research



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

Fingerprinting

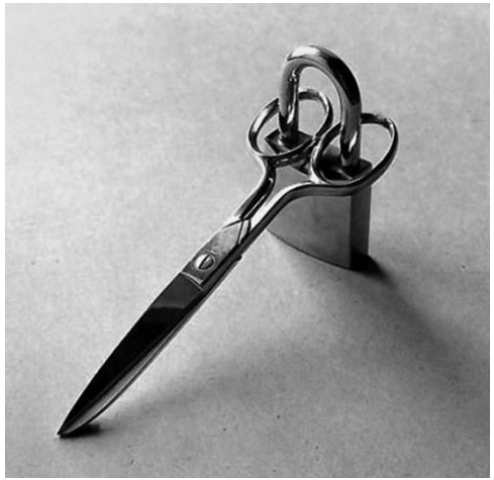
- Phase one: 10 Teststrings
- Phase two: "Specialties"

ModSecurity

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

GET /print.php?message=a HTTP/1.1\r\n
Host: modsec.waf\r\n
User-Agent: libwww\r\n
Accept: a\r\n
\r\n

PHPIDS



- No global Filtering

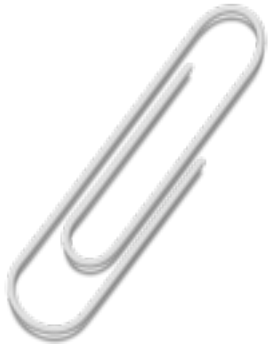
http://example.com/pfyq/hohkfs/$BAD_STRING

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
```

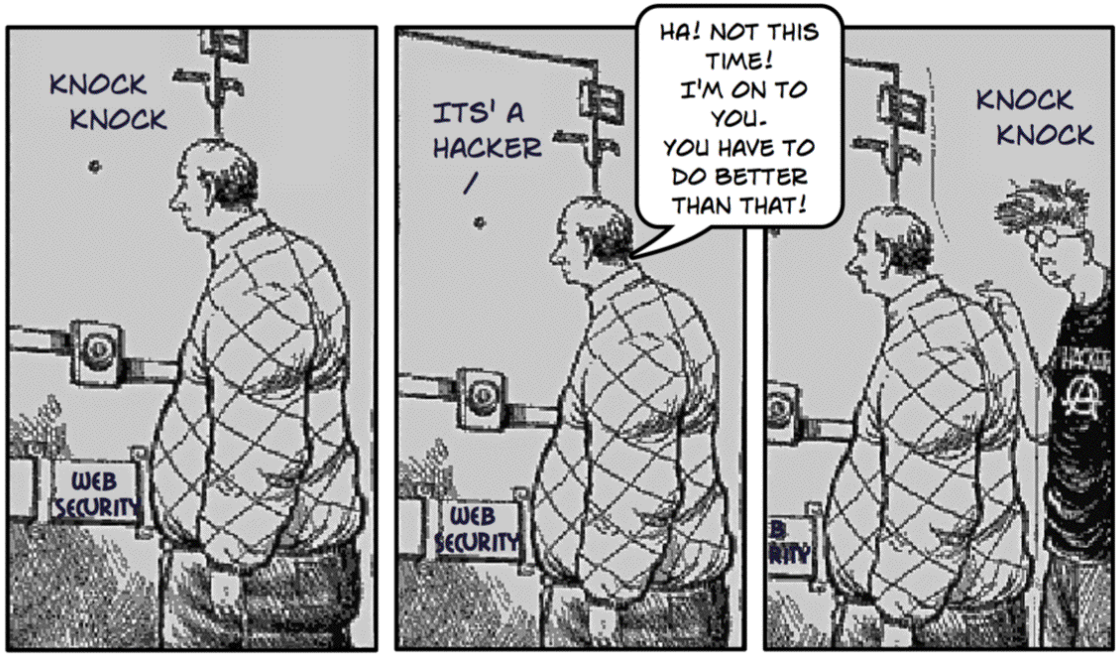
Web Application Firewall 460

- 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


KNOCK KNOCK
WEB SECURITY
ITS' A HACKER /
WEB SECURITY
HA! NOT THIS TIME! I'M ON TO YOU. YOU HAVE TO DO BETTER THAN THAT!
KNOCK KNOCK

WAF secured ☺

SQLi based Circumvention

- 'or 'a='='a=
- 'or 'a'='a'--
- 'or 1 --
- ...' --

ERNW
providing security.

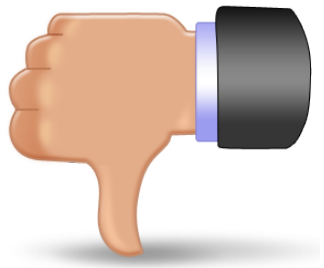
But these statements can't be used to extract data, hm?

- 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

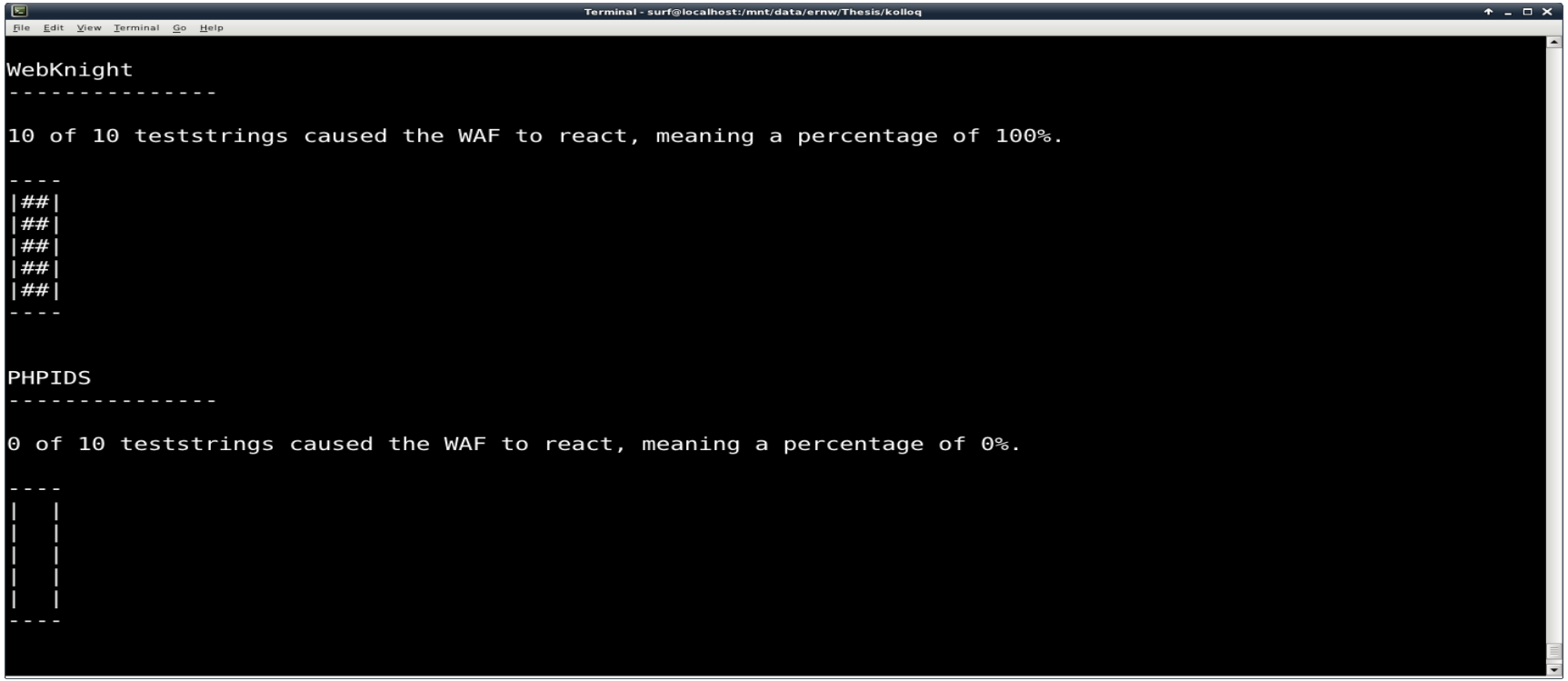
Supported functions

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

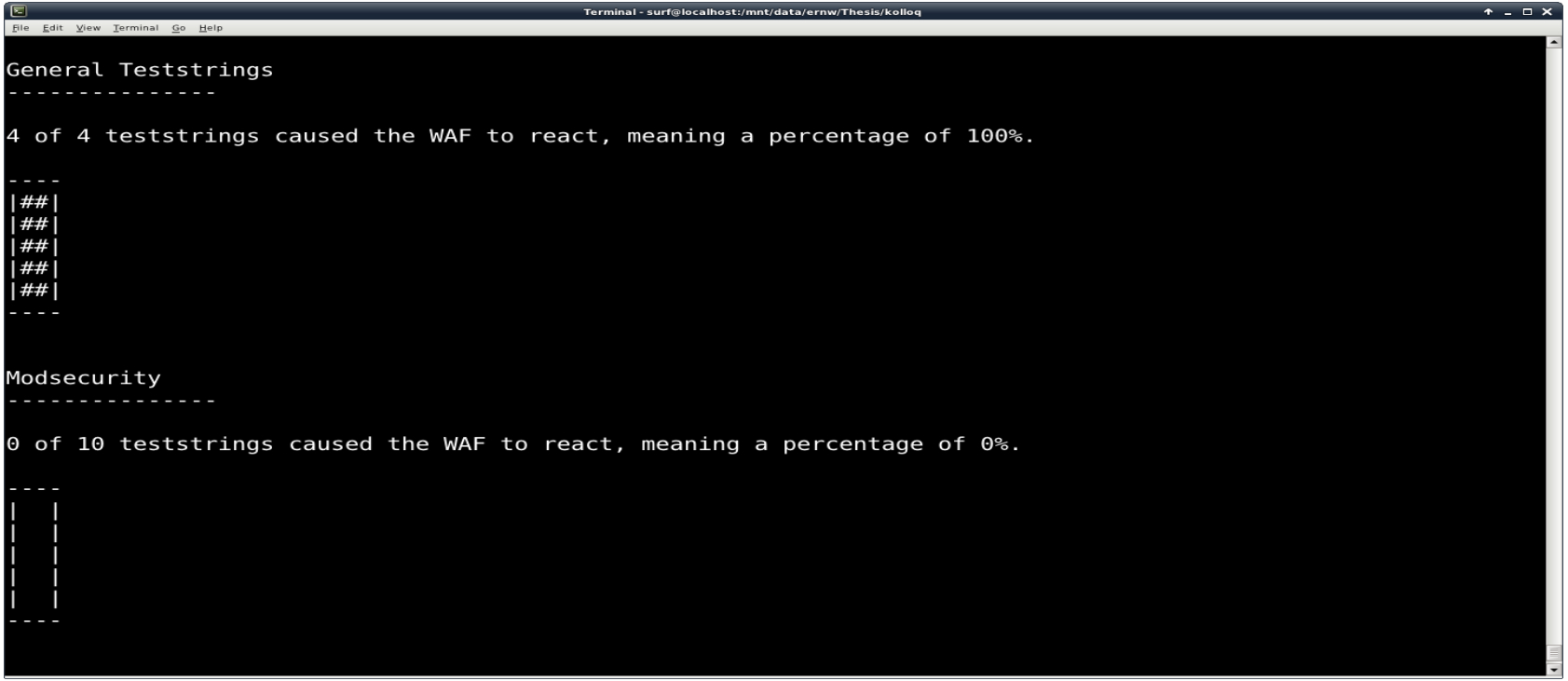
time for a demo?

tsakwaf

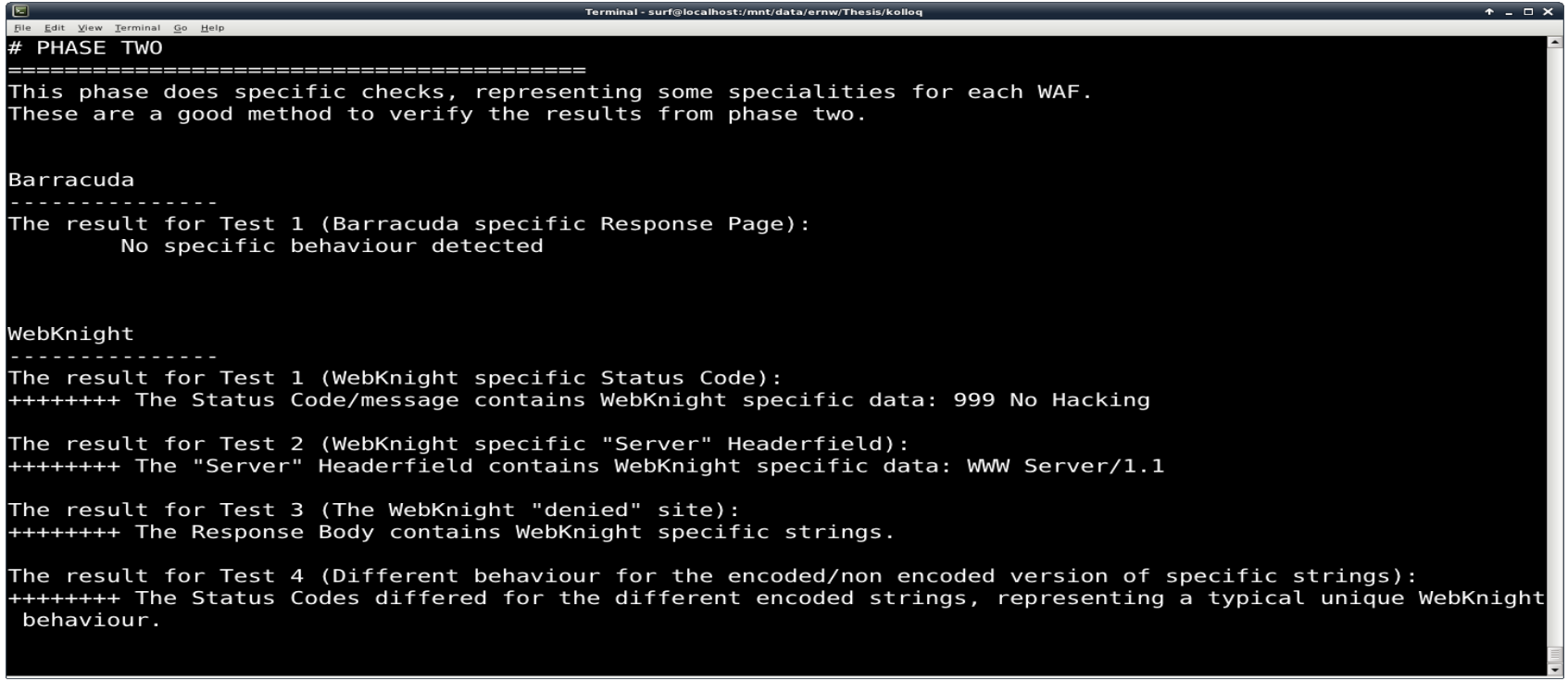
Fingerprinting



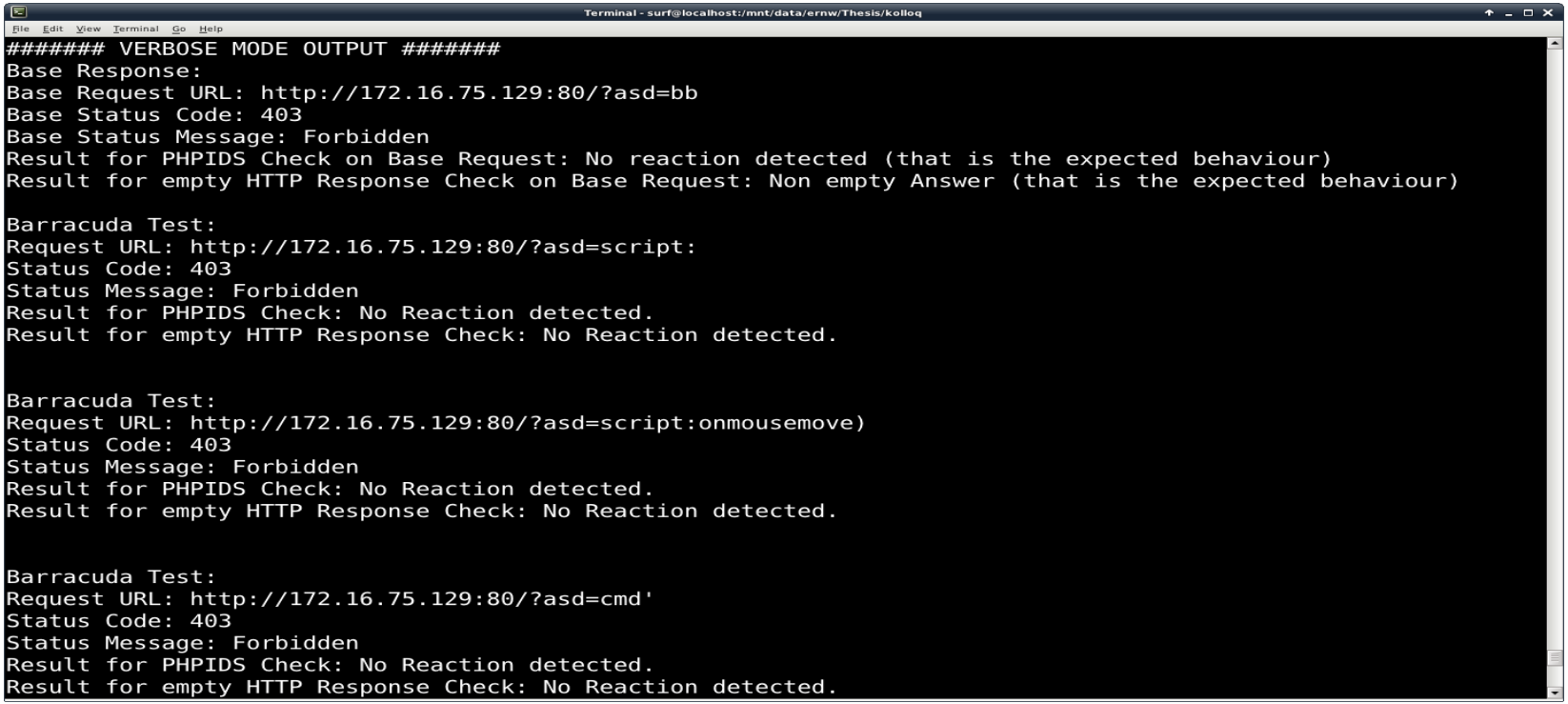
Fingerprinting



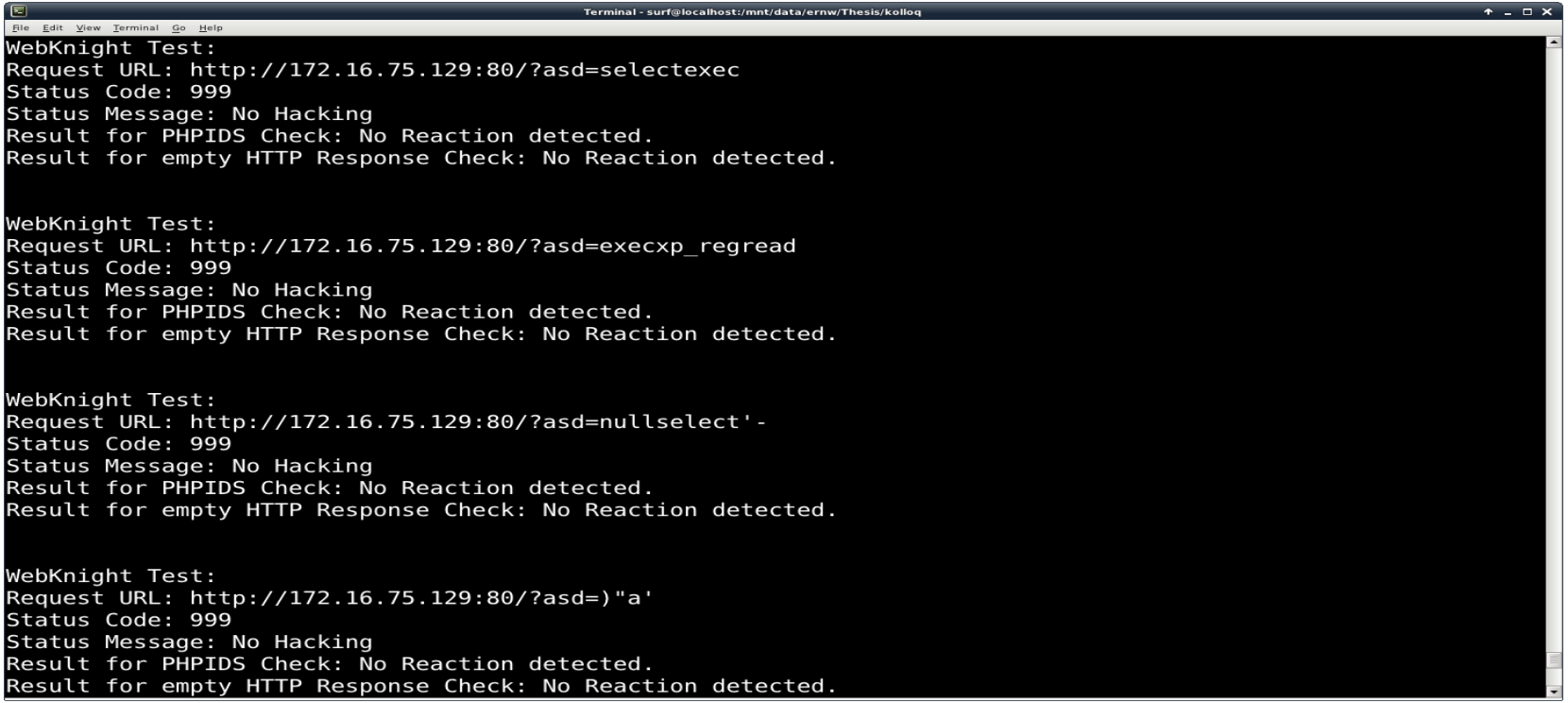
Fingerprinting



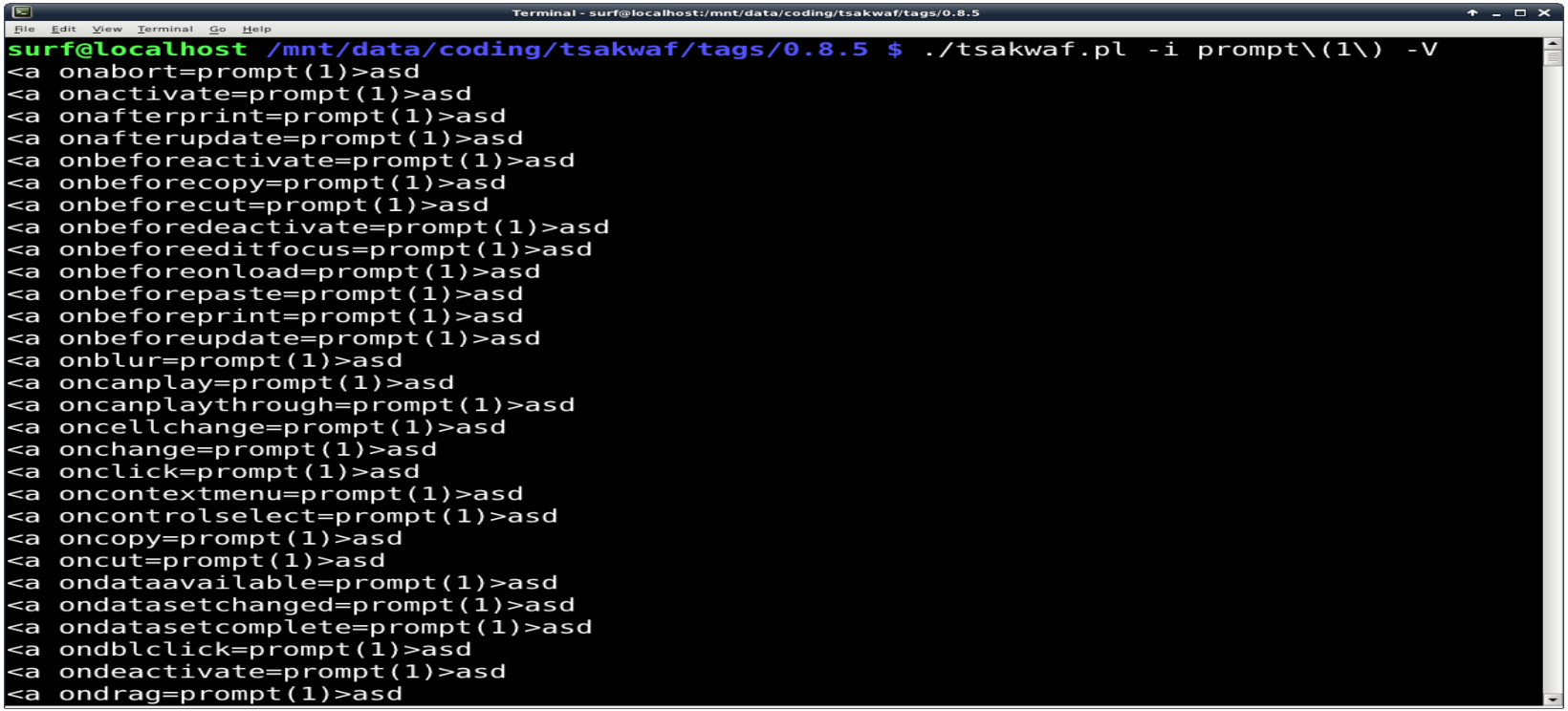
Fingerprinting



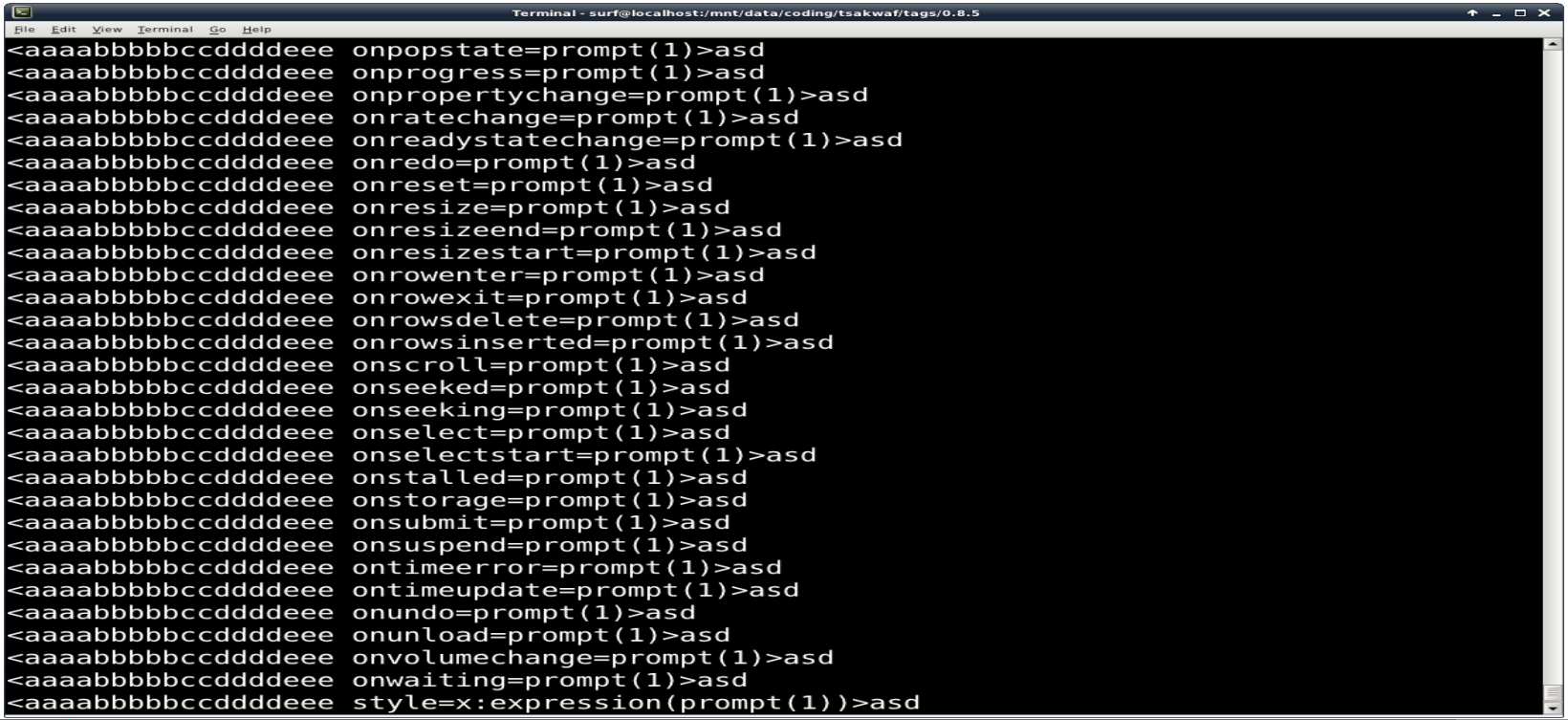
Fingerprinting



XSS Code Generator



XSS Code Generator



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?