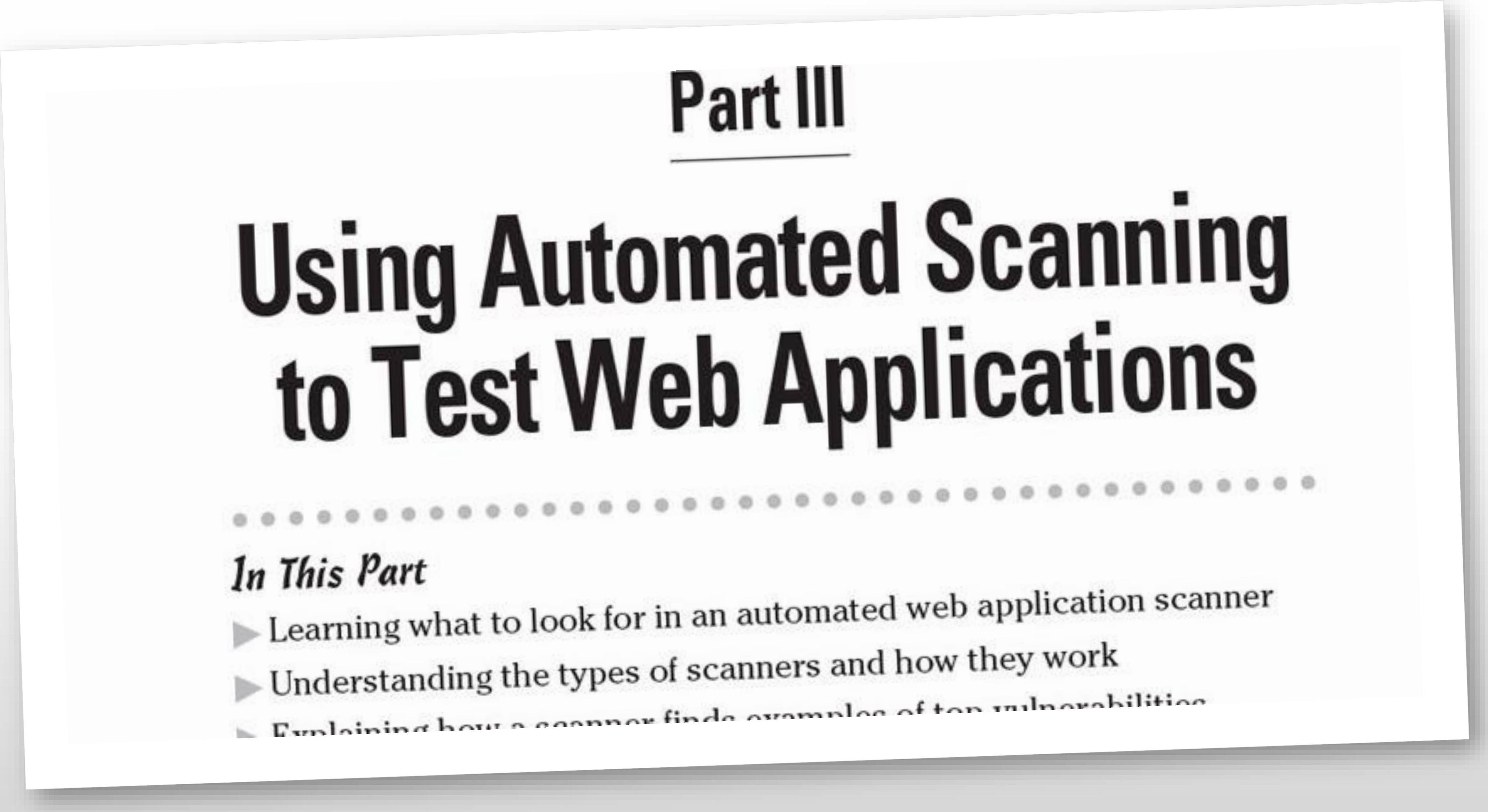


Part III

Using Automated Scanning to Test Web Applications

In This Part

- Learning what to look for in an automated web application scanner
- Understanding the types of scanners and how they work
- Explaining how a scanner finds examples of top vulnerabilities

Requirements

- Automated / scheduled / scriptable starts
- On premise
- Crawler supports HTML5 / DOM / JS
- Reliable fuzzer
- Cost efficient
- Reporting formats support (html, db, yaml, etc.)

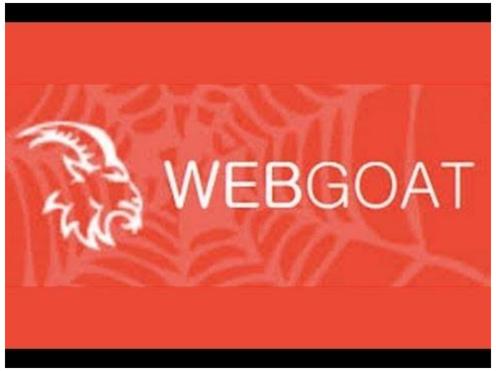
Scanners reviewed

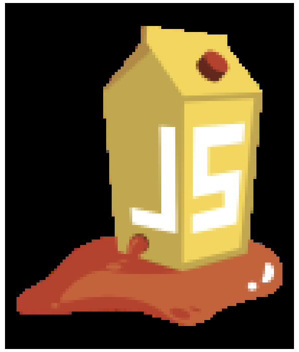
- Burp (manual vs automated)
- Acunetix
- Nessus
- ZAP
- Arachni
- Htdoc (crawler)

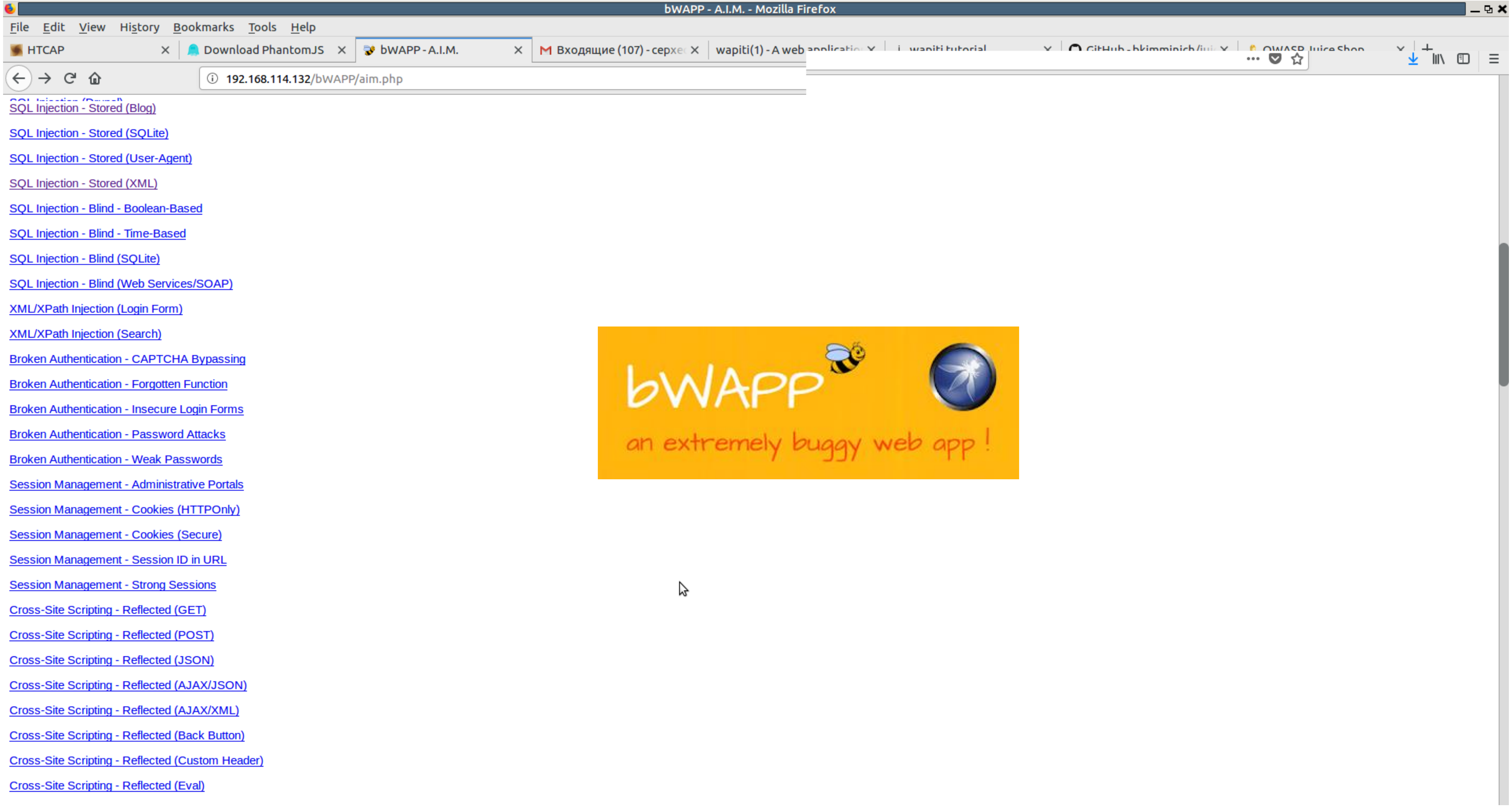
- Veracode
- Qualys
- Appcheck
- HP Fortify Webinspect
- IBM Appscan
- Wapiti

Apps tested

- bWAPP
- WebGoat
- Juice Shop







192.168.114.132/bWAPP/aim.php

SQL Injection - Stored (Blog)

SQL Injection - Stored (SQLite)

SQL Injection - Stored (User-Agent)

SQL Injection - Stored (XML)

SQL Injection - Blind - Boolean-Based

SQL Injection - Blind - Time-Based

SQL Injection - Blind (SQLite)

SQL Injection - Blind (Web Services/SOAP)

XML/XPath Injection (Login Form)

XML/XPath Injection (Search)

Broken Authentication - CAPTCHA Bypassing

Broken Authentication - Forgotten Function

Broken Authentication - Insecure Login Forms

Broken Authentication - Password Attacks

Broken Authentication - Weak Passwords

Session Management - Administrative Portals

Session Management - Cookies (HTTPOnly)

Session Management - Cookies (Secure)

Session Management - Session ID in URL

Session Management - Strong Sessions

Cross-Site Scripting - Reflected (GET)

Cross-Site Scripting - Reflected (POST)

Cross-Site Scripting - Reflected (JSON)

Cross-Site Scripting - Reflected (AJAX/JSON)

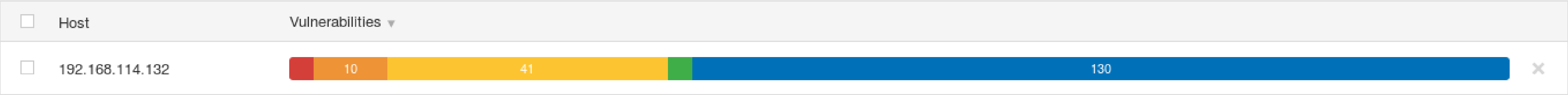
Cross-Site Scripting - Reflected (AJAX/XML)

Cross-Site Scripting - Reflected (Back Button)

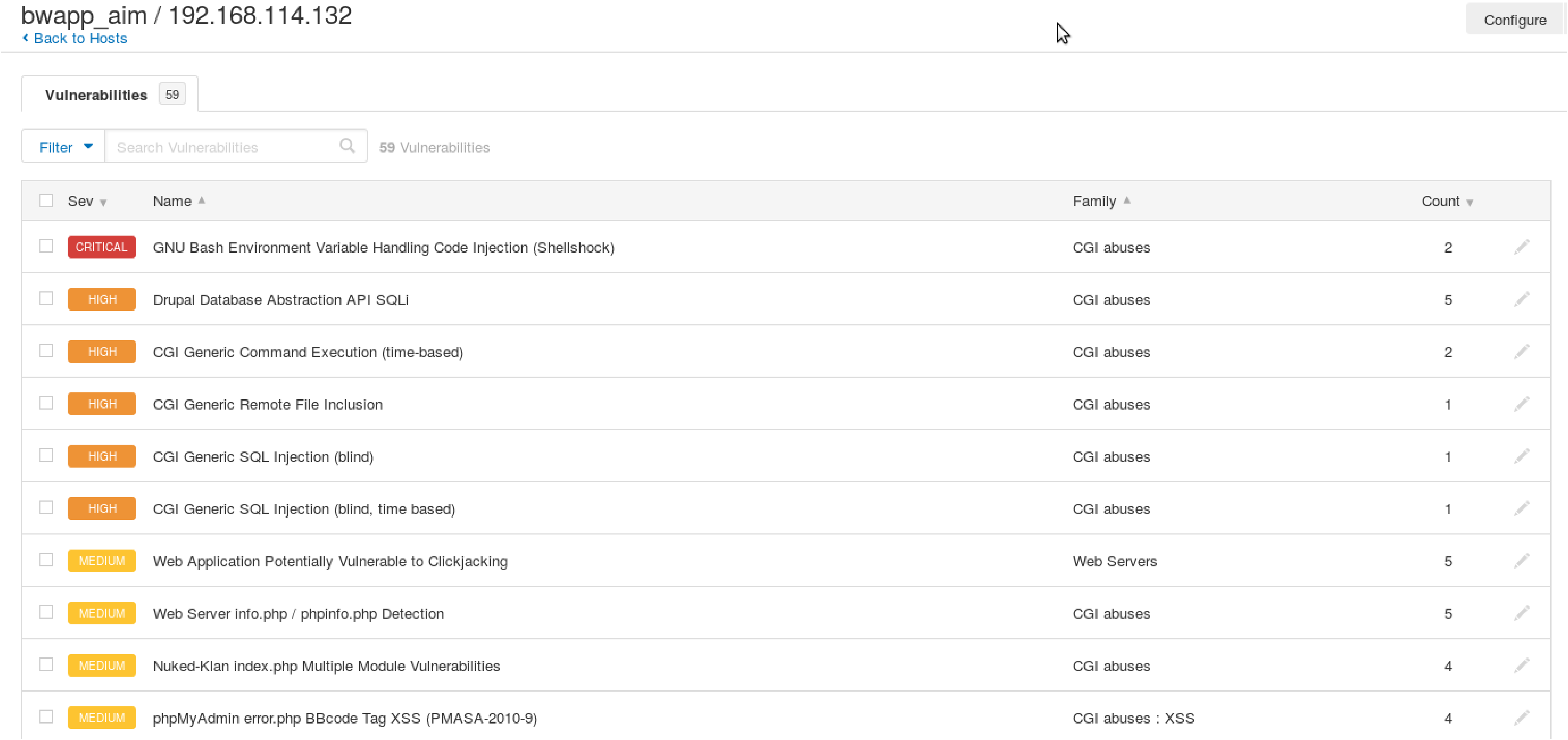
Cross-Site Scripting - Reflected (Custom Header)

Cross-Site Scripting - Reflected (Eval)

Nessus



□	Host	Vulnerabilities ▾
□	192.168.114.132	10 41 130

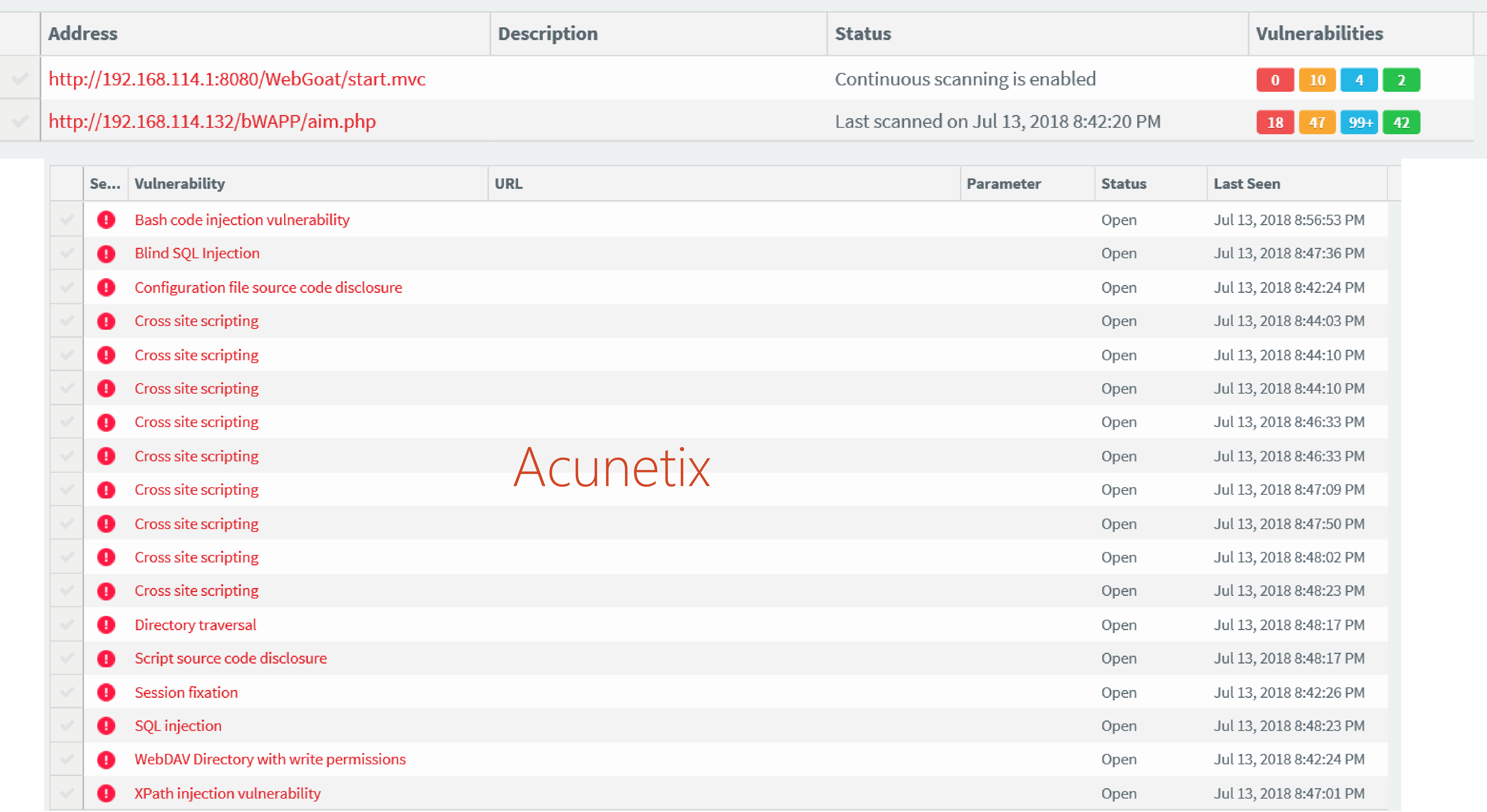


bwapp_aim / 192.168.114.132

‹ Back to Hosts

59 Vulnerabilities

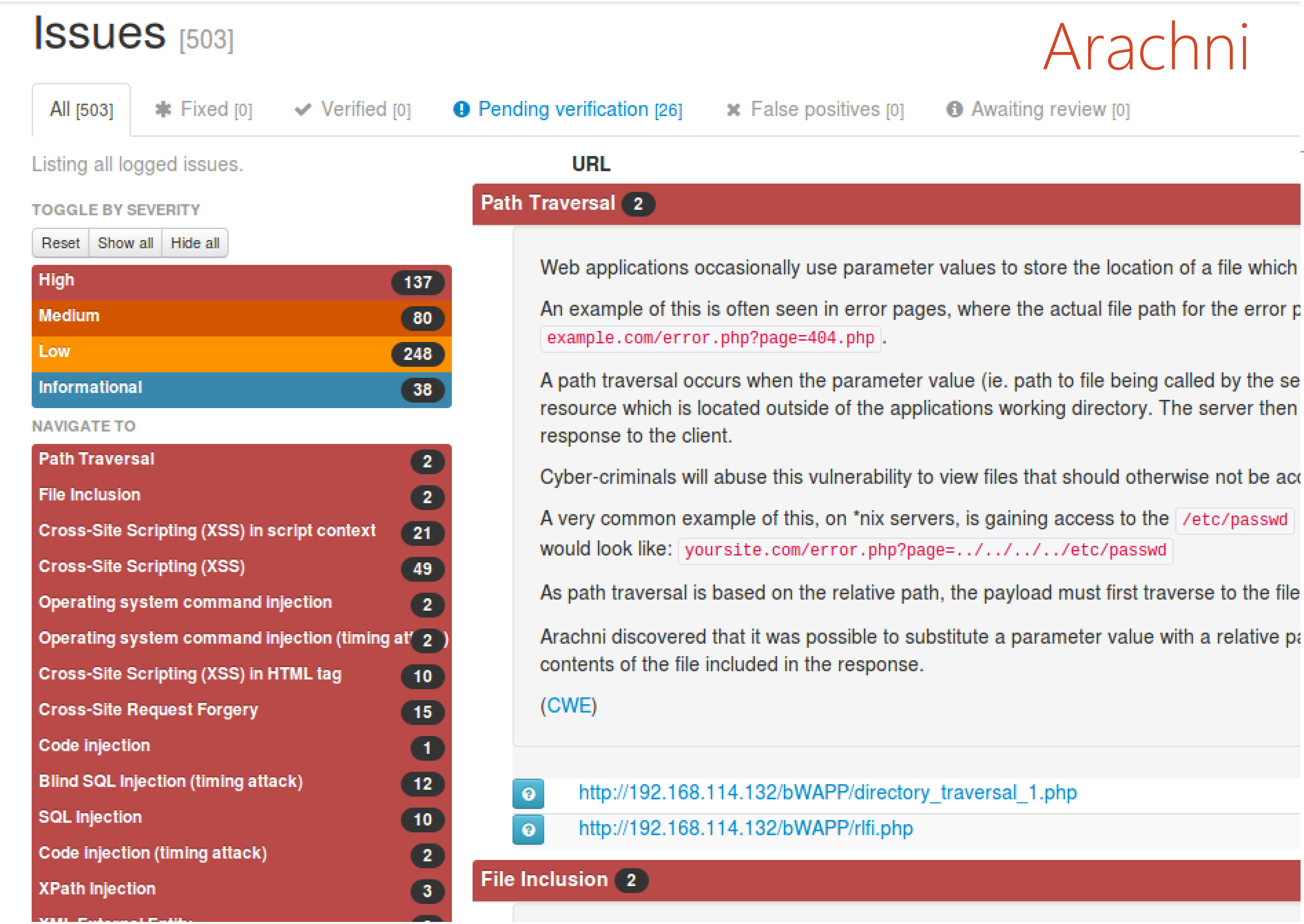
□	Sev ▾	Name ▴	Family ▴	Count ▾
□	CRITICAL	GNU Bash Environment Variable Handling Code Injection (Shellshock)	CGI abuses	2
□	HIGH	Drupal Database Abstraction API SQLi	CGI abuses	5
□	HIGH	CGI Generic Command Execution (time-based)	CGI abuses	2
□	HIGH	CGI Generic Remote File Inclusion	CGI abuses	1
□	HIGH	CGI Generic SQL Injection (blind)	CGI abuses	1
□	HIGH	CGI Generic SQL Injection (blind, time based)	CGI abuses	1
□	MEDIUM	Web Application Potentially Vulnerable to Clickjacking	Web Servers	5
□	MEDIUM	Web Server info.php / phpinfo.php Detection	CGI abuses	5
□	MEDIUM	Nuked-Klan index.php Multiple Module Vulnerabilities	CGI abuses	4
□	MEDIUM	phpMyAdmin error.php BBcode Tag XSS (PMASA-2010-9)	CGI abuses : XSS	4



	Address	Description	Status	Vulnerabilities
✓	http://192.168.114.1:8080/WebGoat/start.mvc		Continuous scanning is enabled	0 10 4 2
✓	http://192.168.114.132/bWAPP/aim.php		Last scanned on Jul 13, 2018 8:42:20 PM	18 47 99+ 42

	Se...	Vulnerability	URL	Parameter	Status	Last Seen
✓	!	Bash code injection vulnerability			Open	Jul 13, 2018 8:56:53 PM
✓	!	Blind SQL Injection			Open	Jul 13, 2018 8:47:36 PM
✓	!	Configuration file source code disclosure			Open	Jul 13, 2018 8:42:24 PM
✓	!	Cross site scripting			Open	Jul 13, 2018 8:44:03 PM
✓	!	Cross site scripting			Open	Jul 13, 2018 8:44:10 PM
✓	!	Cross site scripting			Open	Jul 13, 2018 8:44:10 PM
✓	!	Cross site scripting			Open	Jul 13, 2018 8:46:33 PM
✓	!	Cross site scripting			Open	Jul 13, 2018 8:46:33 PM
✓	!	Cross site scripting			Open	Jul 13, 2018 8:47:09 PM
✓	!	Cross site scripting			Open	Jul 13, 2018 8:47:50 PM
✓	!	Cross site scripting			Open	Jul 13, 2018 8:48:02 PM
✓	!	Cross site scripting			Open	Jul 13, 2018 8:48:23 PM
✓	!	Directory traversal			Open	Jul 13, 2018 8:48:17 PM
✓	!	Script source code disclosure			Open	Jul 13, 2018 8:48:17 PM
✓	!	Session fixation			Open	Jul 13, 2018 8:42:26 PM
✓	!	SQL injection			Open	Jul 13, 2018 8:48:23 PM
✓	!	WebDAV Directory with write permissions			Open	Jul 13, 2018 8:42:24 PM
✓	!	XPath injection vulnerability			Open	Jul 13, 2018 8:47:01 PM

Arachni



Issues [503]

All [503] | ✱ Fixed [0] | ✓ Verified [0] | Pending verification [26] | ✖ False positives [0] | Awaiting review [0]

Listing all logged issues.

TOGGLE BY SEVERITY

Reset | Show all | Hide all

High	137
Medium	80
Low	248
Informational	38

NAVIGATE TO

Path Traversal	2
File Inclusion	2
Cross-Site Scripting (XSS) in script context	21
Cross-Site Scripting (XSS)	49
Operating system command injection	2
Operating system command injection (timing at	2
Cross-Site Scripting (XSS) in HTML tag	10
Cross-Site Request Forgery	15
Code injection	1
Blind SQL Injection (timing attack)	12
SQL Injection	10
Code injection (timing attack)	2
XPath Injection	3
XML External Entity	

URL

Path Traversal 2

Web applications occasionally use parameter values to store the location of a file which

An example of this is often seen in error pages, where the actual file path for the error p `example.com/error.php?page=404.php`.

A path traversal occurs when the parameter value (ie. path to file being called by the se resource which is located outside of the applications working directory. The server then response to the client.

Cyber-criminals will abuse this vulnerability to view files that should otherwise not be acc

A very common example of this, on *nix servers, is gaining access to the `/etc/passwd` would look like: `yoursite.com/error.php?page=../../../../etc/passwd`

As path traversal is based on the relative path, the payload must first traverse to the file

Arachni discovered that it was possible to substitute a parameter value with a relative pa contents of the file included in the response.

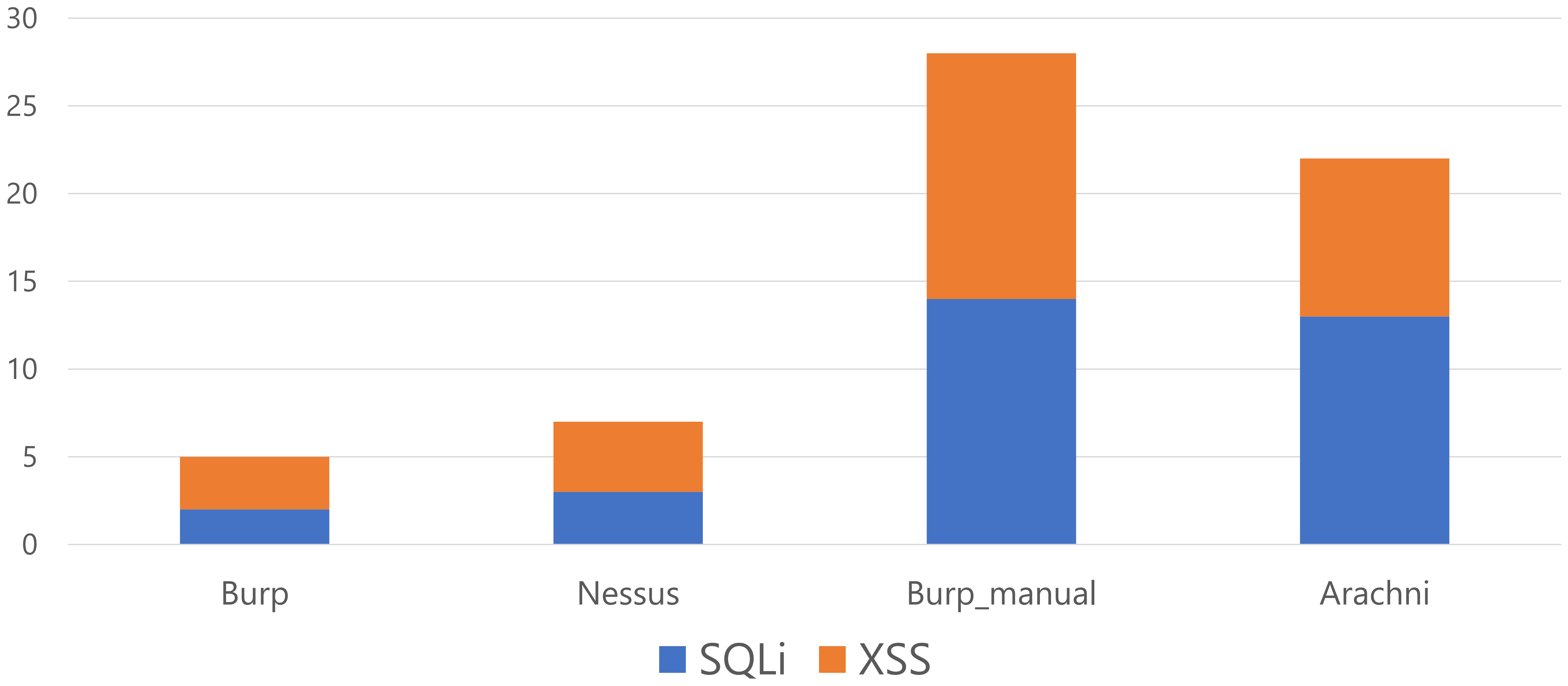
(CWE)

http://192.168.114.132/bWAPP/directory_traversal_1.php

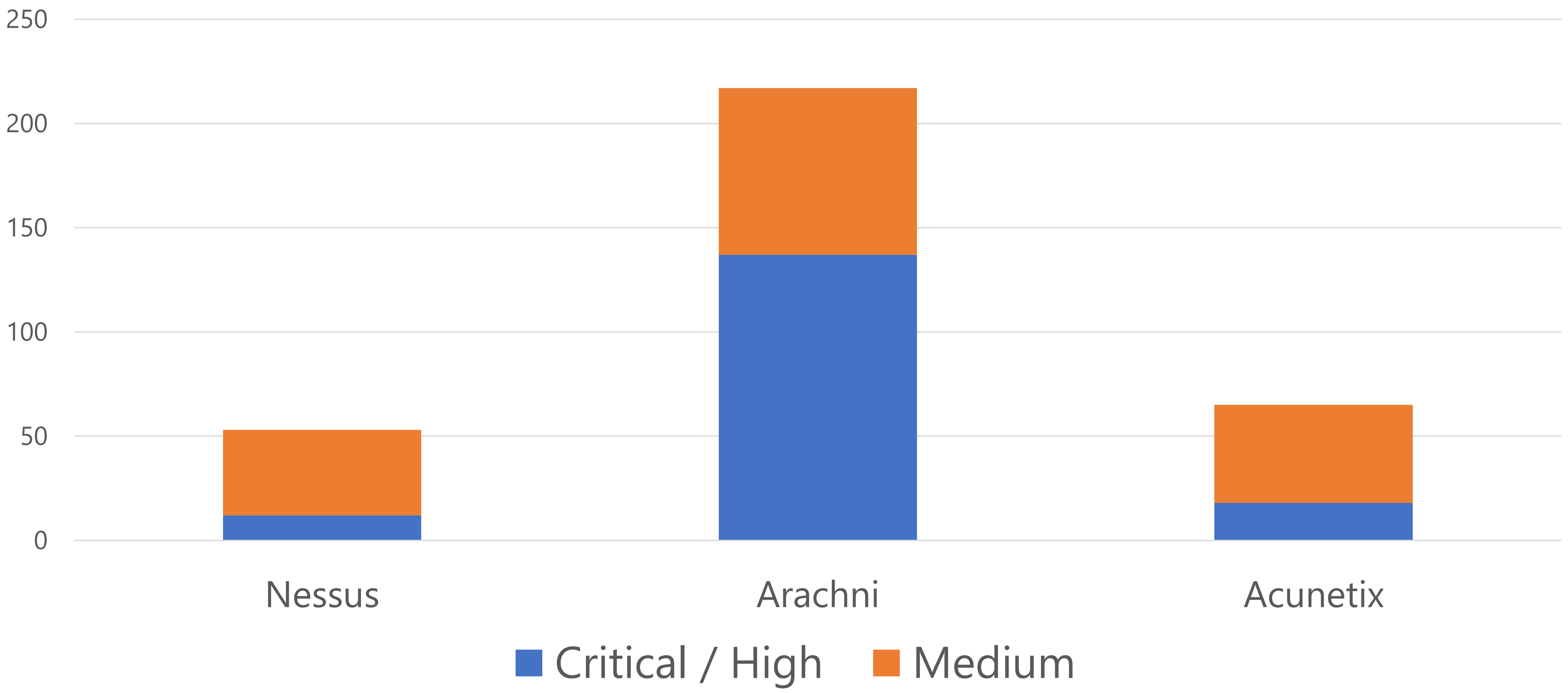
http://192.168.114.132/bWAPP/rlfi.php

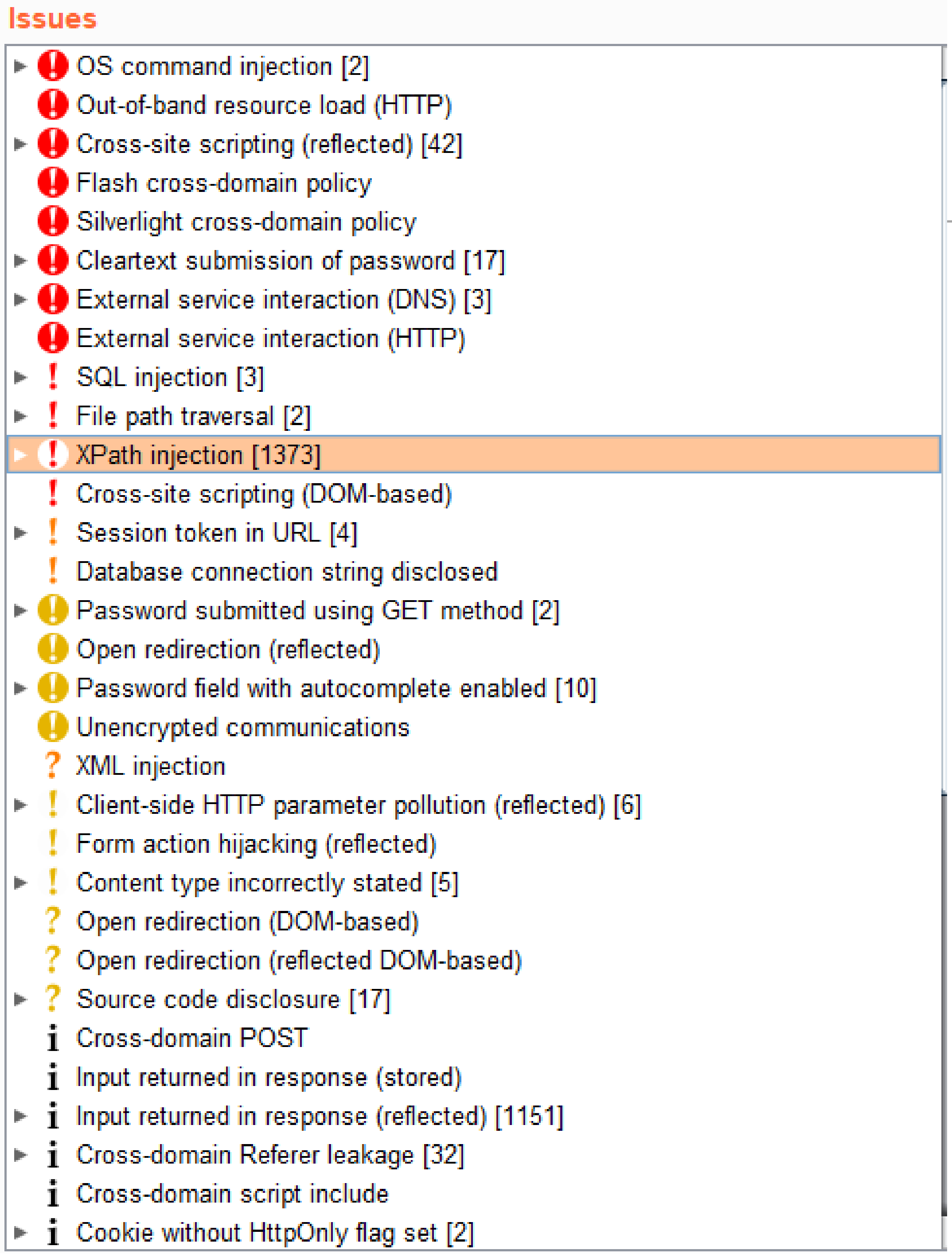
File Inclusion 2

SQLi (18) and XSS (19) found in bWAPP



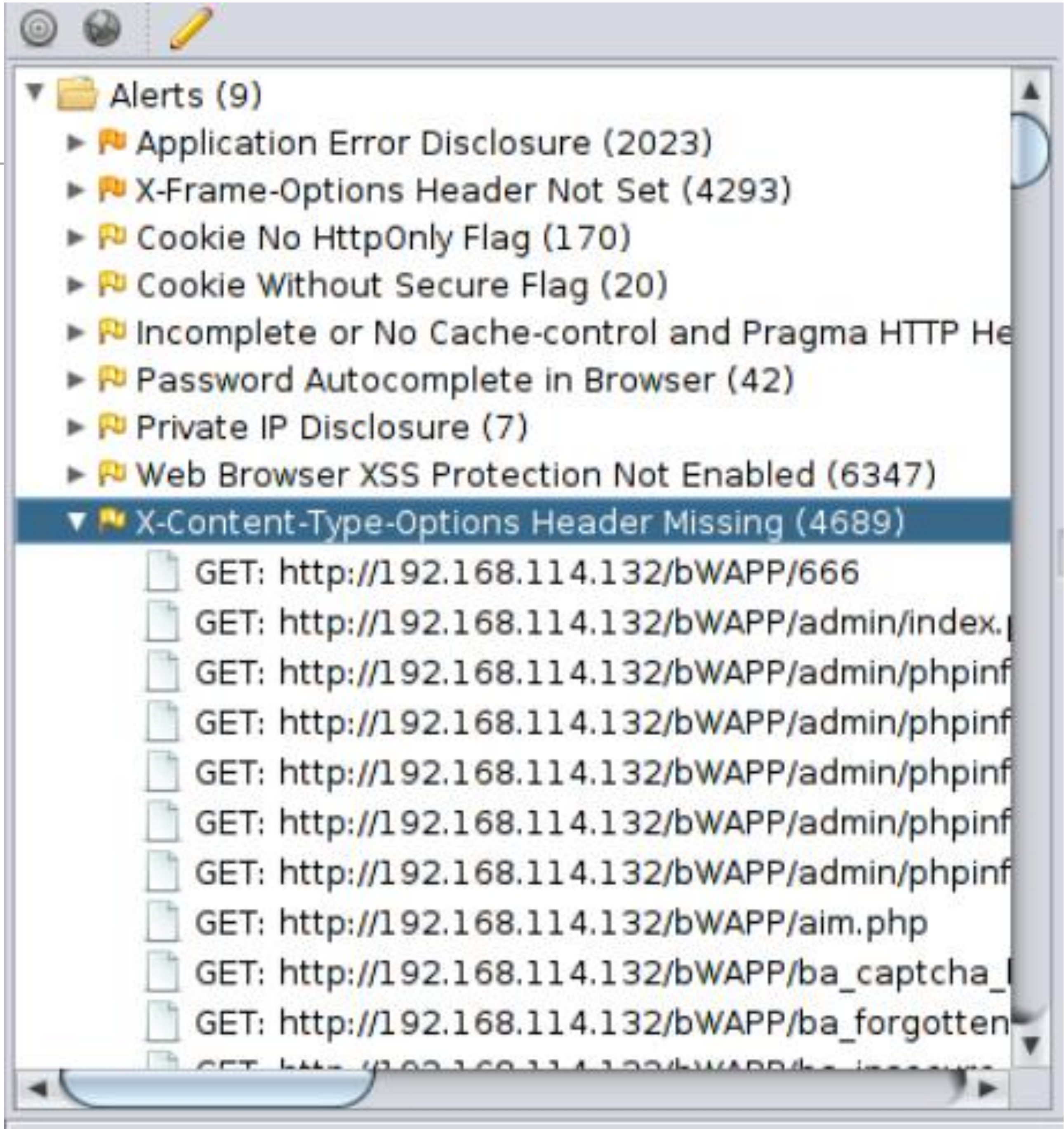
bWAPP findings



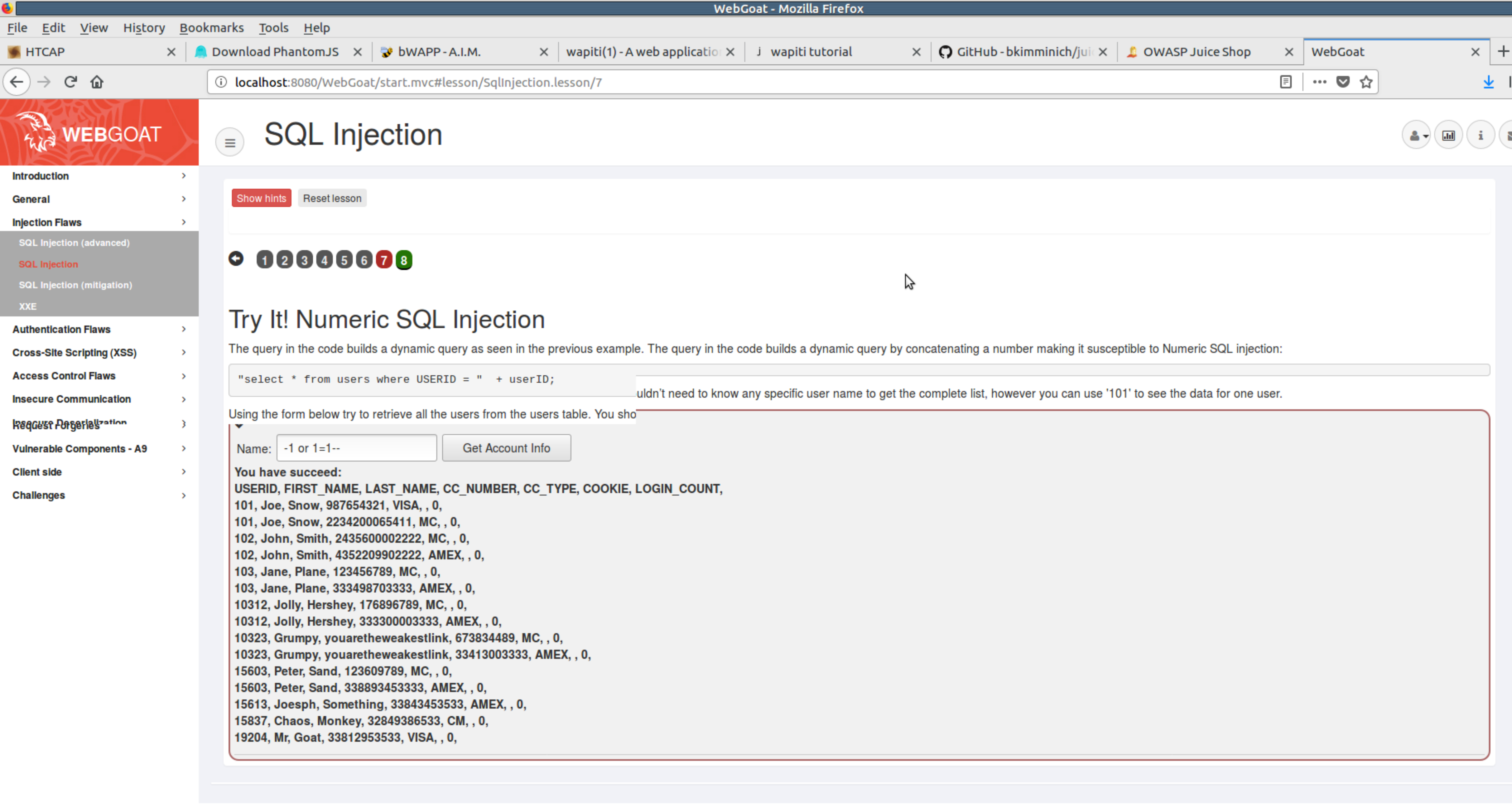


Issues

- OS command injection [2]
- Out-of-band resource load (HTTP)
- Cross-site scripting (reflected) [42]
- Flash cross-domain policy
- Silverlight cross-domain policy
- Cleartext submission of password [17]
- External service interaction (DNS) [3]
- External service interaction (HTTP)
- SQL injection [3]
- File path traversal [2]
- XPath injection [1373]
- Cross-site scripting (DOM-based)
- Session token in URL [4]
- Database connection string disclosed
- Password submitted using GET method [2]
- Open redirection (reflected)
- Password field with autocomplete enabled [10]
- Unencrypted communications
- XML injection
- Client-side HTTP parameter pollution (reflected) [6]
- Form action hijacking (reflected)
- Content type incorrectly stated [5]
- Open redirection (DOM-based)
- Open redirection (reflected DOM-based)
- Source code disclosure [17]
- Cross-domain POST
- Input returned in response (stored)
- Input returned in response (reflected) [1151]
- Cross-domain Referer leakage [32]
- Cross-domain script include
- Cookie without HttpOnly flag set [2]



Alerts (9)
- Application Error Disclosure (2023)
- X-Frame-Options Header Not Set (4293)
- Cookie No HttpOnly Flag (170)
- Cookie Without Secure Flag (20)
- Incomplete or No Cache-control and Pragma HTTP He
- Password Autocomplete in Browser (42)
- Private IP Disclosure (7)
- Web Browser XSS Protection Not Enabled (6347)
- X-Content-Type-Options Header Missing (4689)
  - GET: http://192.168.114.132/bWAPP/666
  - GET: http://192.168.114.132/bWAPP/admin/index.
  - GET: http://192.168.114.132/bWAPP/admin/phpinf
  - GET: http://192.168.114.132/bWAPP/admin/phpinf
  - GET: http://192.168.114.132/bWAPP/admin/phpinf
  - GET: http://192.168.114.132/bWAPP/admin/phpinf
  - GET: http://192.168.114.132/bWAPP/admin/phpinf
  - GET: http://192.168.114.132/bWAPP/aim.php
  - GET: http://192.168.114.132/bWAPP/ba_captcha_
  - GET: http://192.168.114.132/bWAPP/ba_forgotten



SQL Injection

Try It! Numeric SQL Injection

The query in the code builds a dynamic query as seen in the previous example. The query in the code builds a dynamic query by concatenating a number making it susceptible to Numeric SQL injection:

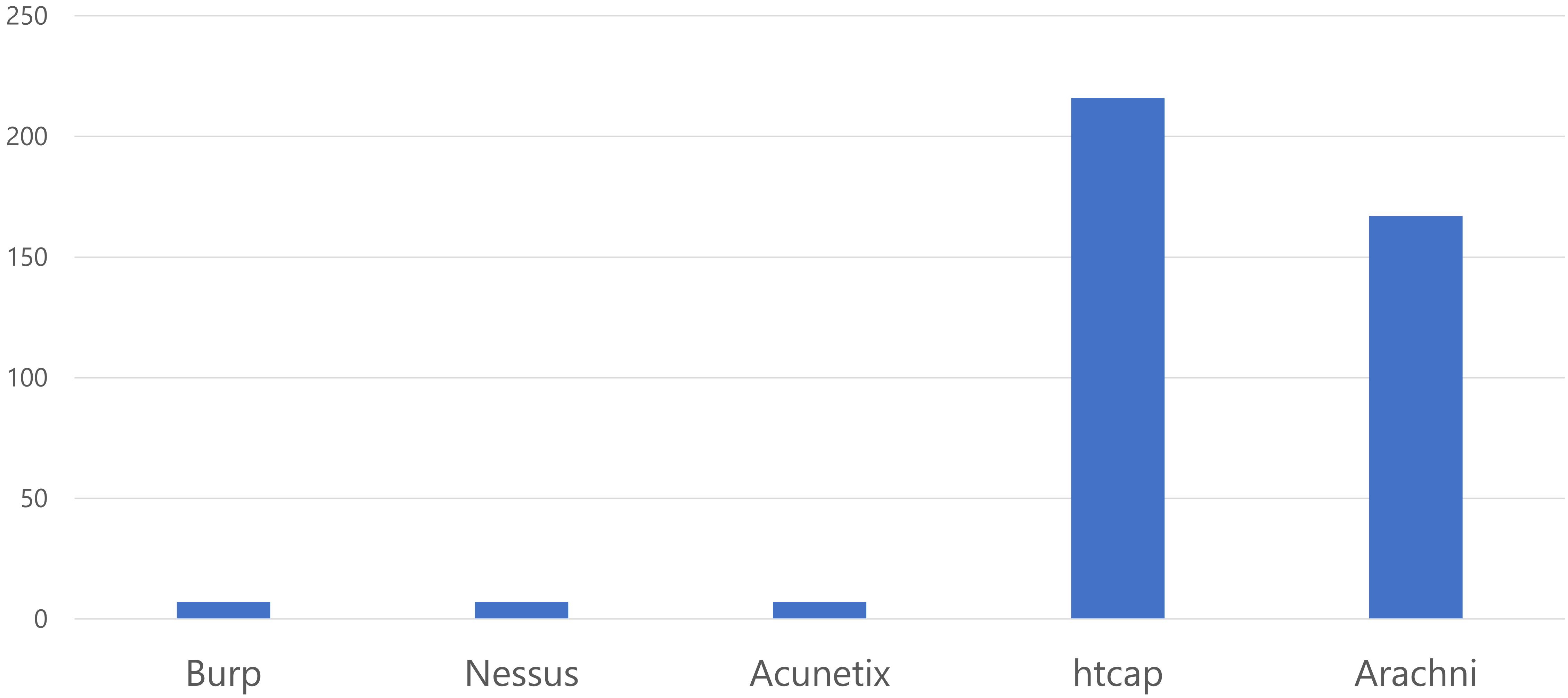
```
"select * from users where USERID = "  + userID;
```

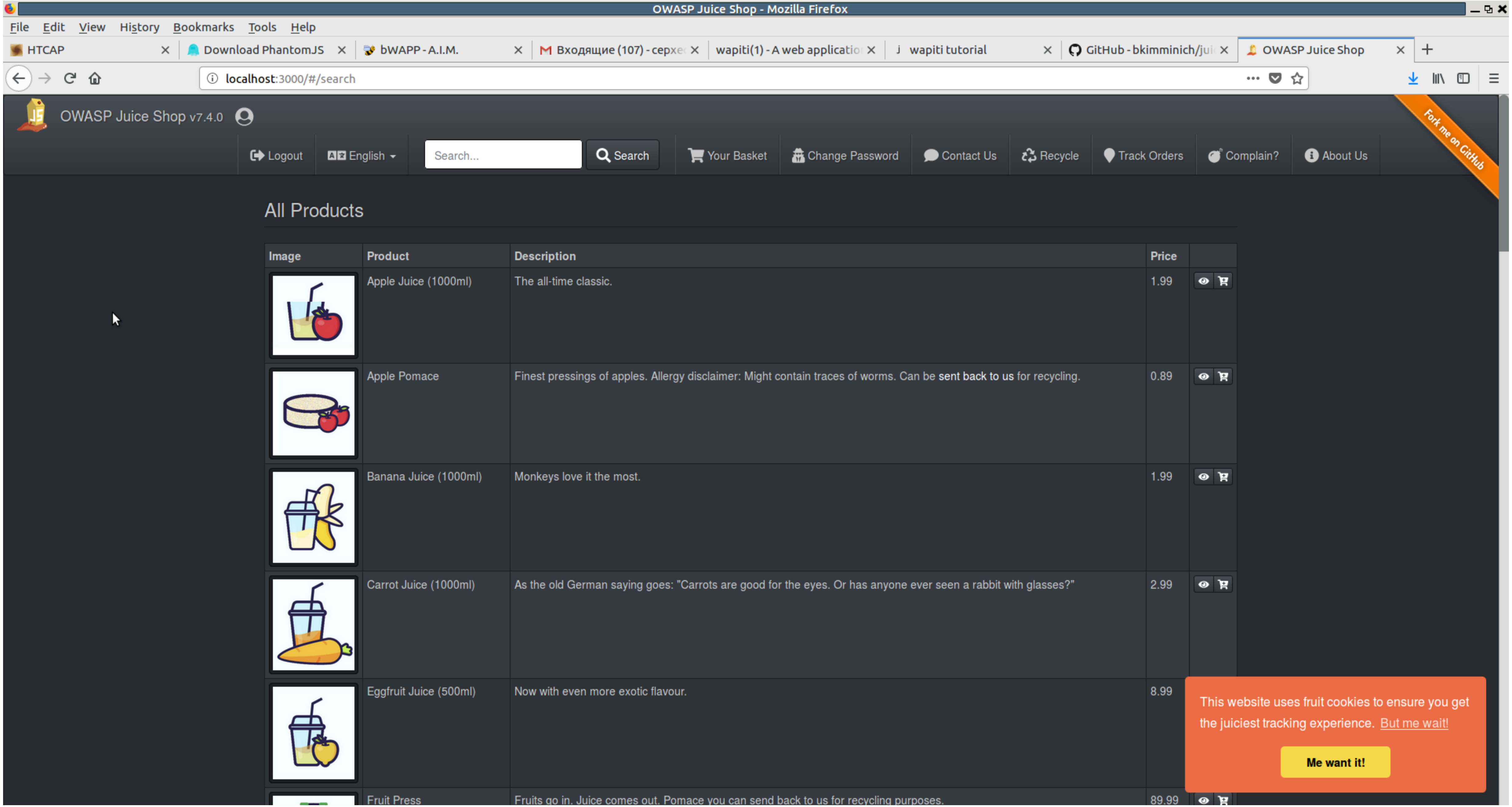
Using the form below try to retrieve all the users from the users table. You shouldn't need to know any specific user name to get the complete list, however you can use '101' to see the data for one user.

Name: -1 or 1=1--

```
You have succeed:
USERID, FIRST_NAME, LAST_NAME, CC_NUMBER, CC_TYPE, COOKIE, LOGIN_COUNT,
101, Joe, Snow, 987654321, VISA, , 0,
101, Joe, Snow, 2234200065411, MC, , 0,
102, John, Smith, 2435600002222, MC, , 0,
102, John, Smith, 4352209902222, AMEX, , 0,
103, Jane, Plane, 123456789, MC, , 0,
103, Jane, Plane, 333498703333, AMEX, , 0,
10312, Jolly, Hershey, 176896789, MC, , 0,
10312, Jolly, Hershey, 333300003333, AMEX, , 0,
10323, Grumpy, youaretheweakestlink, 673834489, MC, , 0,
10323, Grumpy, youaretheweakestlink, 33413003333, AMEX, , 0,
15603, Peter, Sand, 123609789, MC, , 0,
15603, Peter, Sand, 338893453333, AMEX, , 0,
15613, Joesph, Something, 33843453533, AMEX, , 0,
15837, Chaos, Monkey, 32849386533, CM, , 0,
19204, Mr, Goat, 33812953533, VISA, , 0,
```

WebGoat Unique URLs found

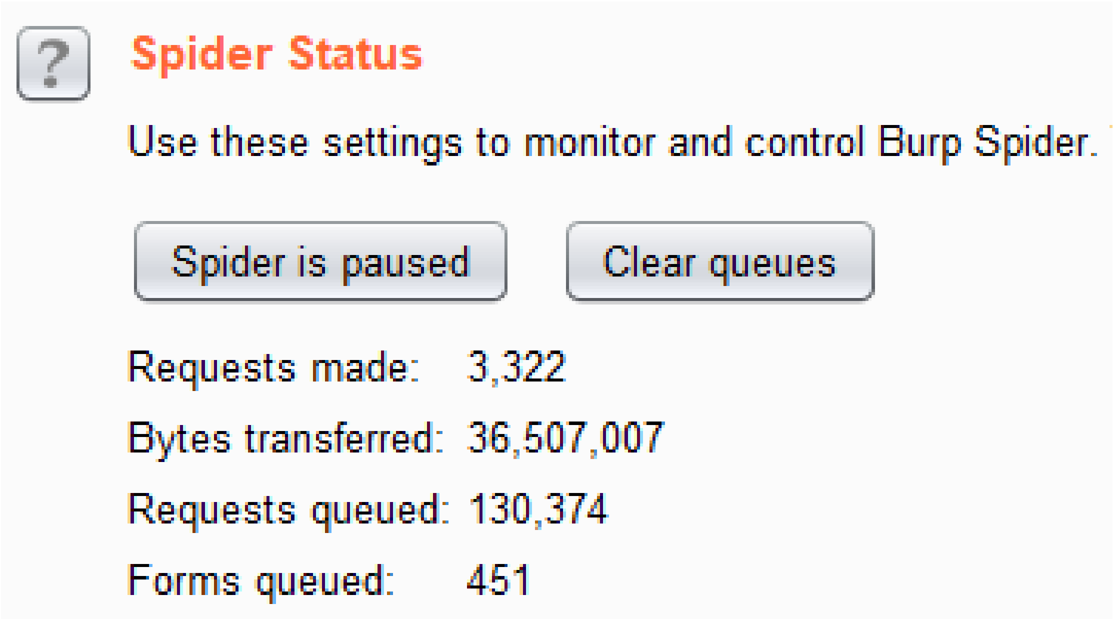




All Products

Image	Product	Description	Price	
	Apple Juice (1000ml)	The all-time classic.	1.99	
	Apple Pomace	Finest pressings of apples. Allergy disclaimer: Might contain traces of worms. Can be sent back to us for recycling.	0.89	
	Banana Juice (1000ml)	Monkeys love it the most.	1.99	
	Carrot Juice (1000ml)	As the old German saying goes: "Carrots are good for the eyes. Or has anyone ever seen a rabbit with glasses?"	2.99	
	Eggfruit Juice (500ml)	Now with even more exotic flavour.	8.99	
	Fruit Press	Fruits go in. Juice comes out. Pomace you can send back to us for recycling purposes.	89.99	

This website uses fruit cookies to ensure you get the juiciest tracking experience. But me wait!



Spider Status

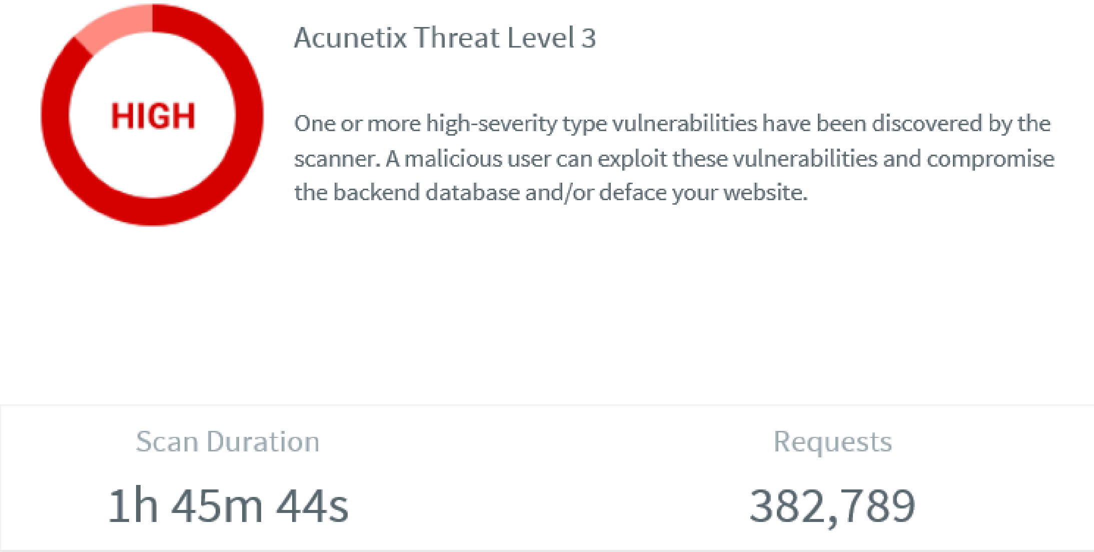
Use these settings to monitor and control Burp Spider.

Requests made: 3,322

Bytes transferred: 36,507,007

Requests queued: 130,374

Forms queued: 451



Acunetix Threat Level 3

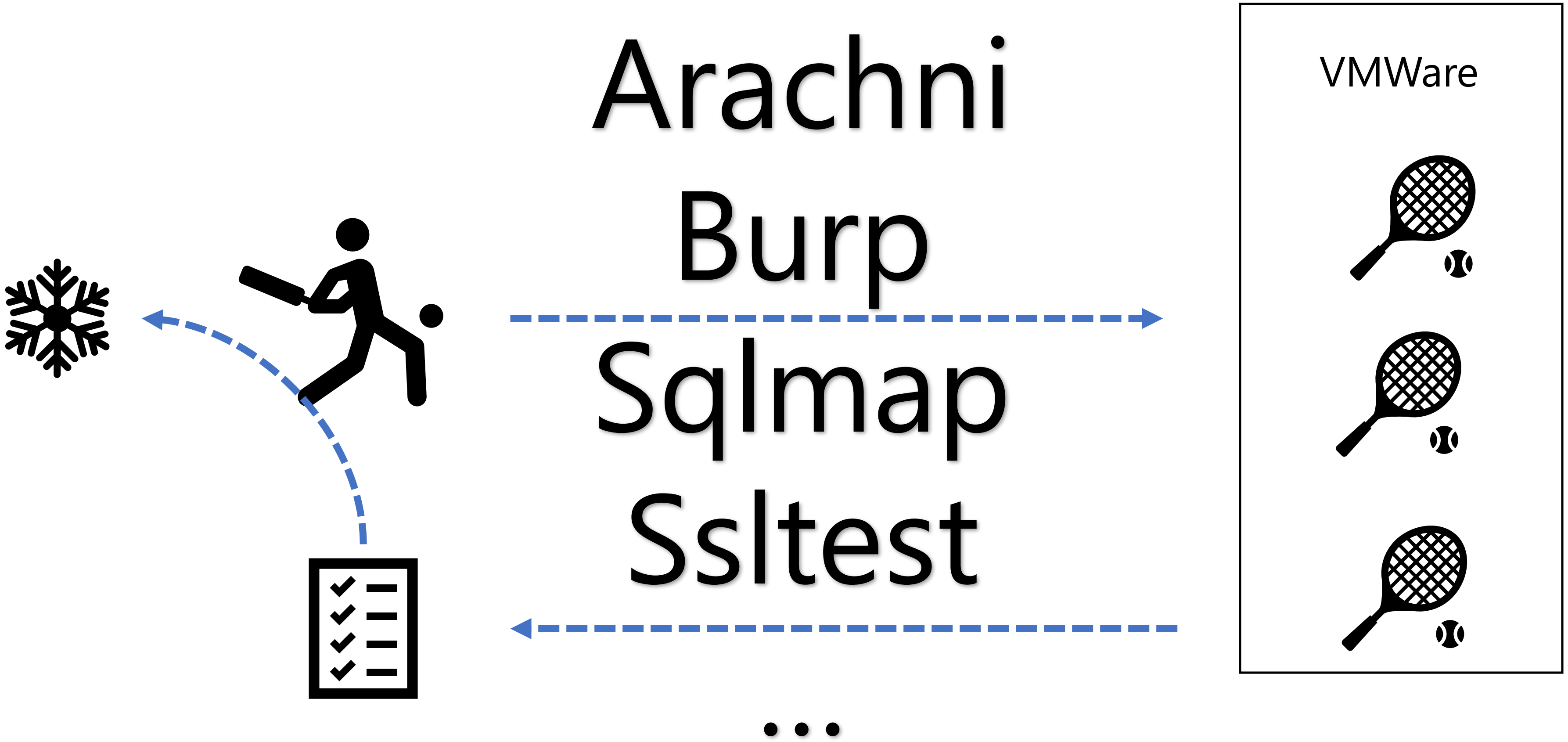
One or more high-severity type vulnerabilities have been discovered by the scanner. A malicious user can exploit these vulnerabilities and compromise the backend database and/or deface your website.

Scan Duration	Requests
1h 45m 44s	382,789

Outcome

- All scanners screw
- Some scanners screw less then others
- Knowledge of application logic is essential to set up the scan properly
- Solution for reporting automation is required

Webscan automation



Arachni

Burp

Sqlmap

Ssltest

…

VMWare