



ZAP Innovations

OWASP

Zed Attack Proxy

Simon Bennetts

OWASP ZAP Project Lead

Mozilla Security Team

`psiinon@gmail.com`

What is ZAP?

- An easy to use webapp pentest tool
- Completely free and open source
- An OWASP flagship project
- Ideal for beginners
- But also used by professionals
- Ideal for devs, esp. for automated security tests
- Becoming a framework for advanced testing
- Not a silver bullet!



ZAP Principles

- Free, Open source
- Involvement actively encouraged
- Cross platform
- Easy to use
- Easy to install
- Internationalized
- Fully documented
- Work well with other tools
- Reuse well regarded components



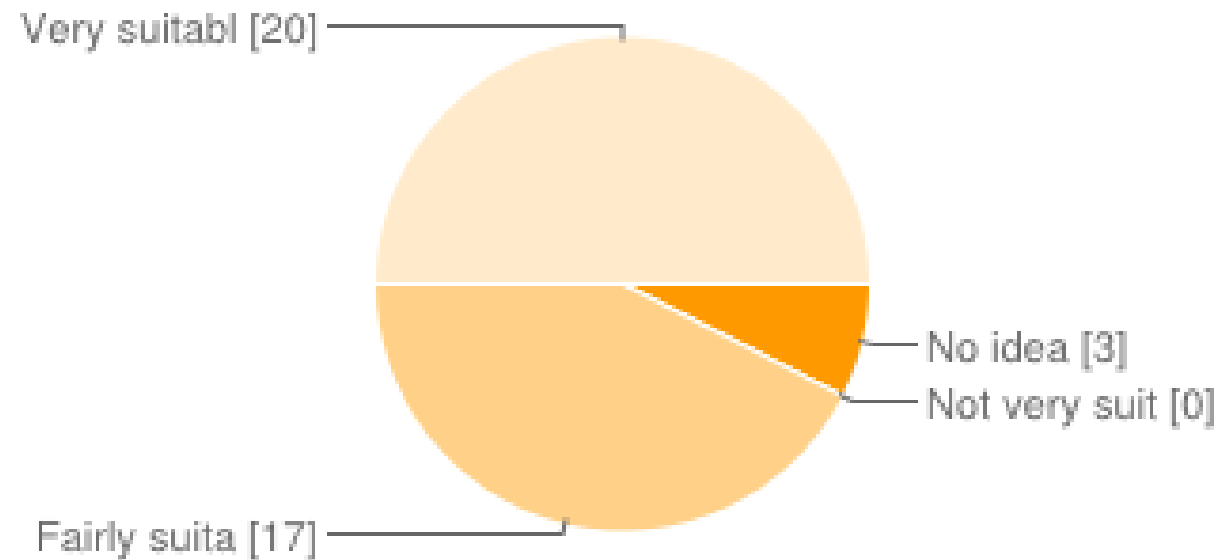
Statistics

- Released September 2010, fork of Paros
- V 2.1.0 released in April 2013
- V 2.1.0 downloaded > 20K times
- V 2.2.0 coming very soon!
- 16 active contributors (Ohloh)
- 120 Person years (Ohloh)
- Translated into 19 languages
- Mostly used by Professional Pentesters?
- Paros code: ~30% ZAP Code: ~70%



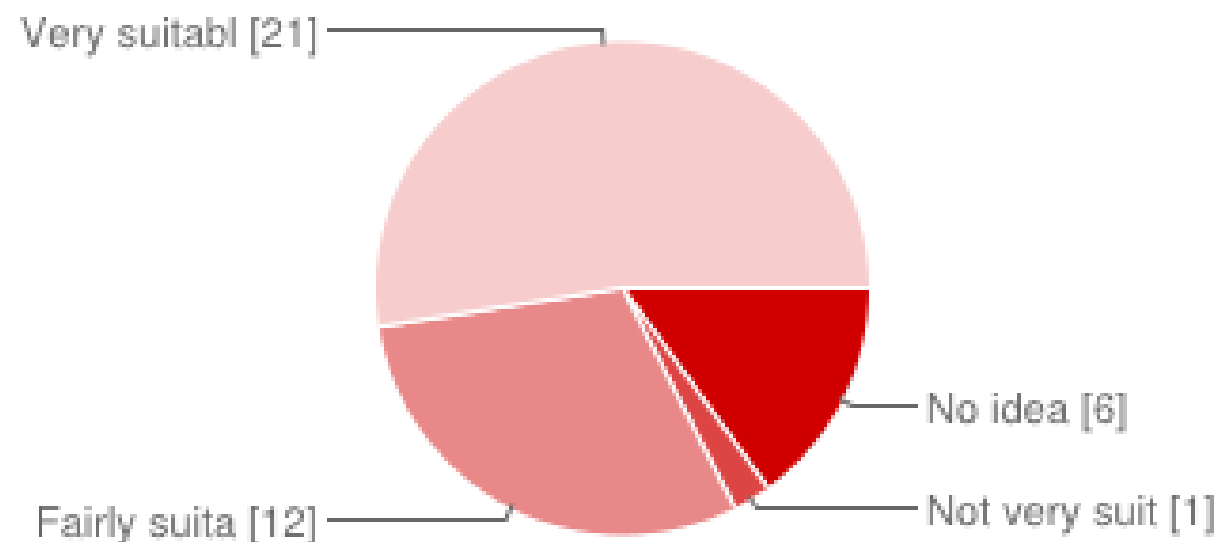
User Questionnaire

How well do you think ZAP is suited to people new to application security?



No idea	3	8%
Not very suitable	0	0%
Fairly suitable	17	43%
Very suitable	20	50%

How well do you think ZAP is suited to security professionals?

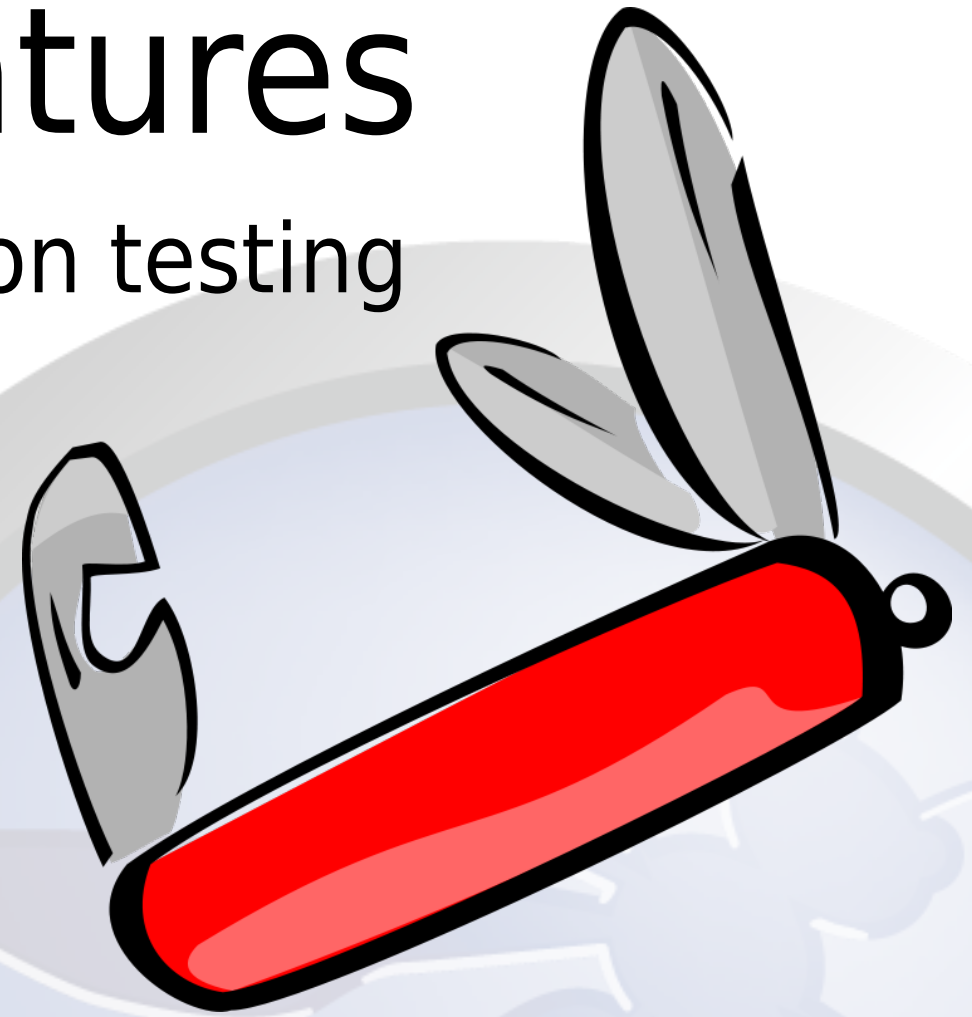


No idea	6	15%
Not very suitable	1	3%
Fairly suitable	12	30%
Very suitable	21	53%

The Main Features

All the essentials for web application testing

- Intercepting Proxy
- Active and Passive Scanners
- Traditional and Ajax Spiders
- WebSockets support
- Forced Browsing (using OWASP DirBuster code)
- Fuzzing (using fuzzdb & OWASP JBroFuzz)
- Online Add-ons Marketplace



Some Additional Features

- Auto tagging
- Port scanner
- Script Console
- Report generation
- Smart card support
- Contexts and scope
- Session management
- Invoke external apps
- Dynamic SSL Certificates



How can you use ZAP?

- Point and shoot – the Quick Start tab
- Proxying via ZAP, and then scanning
- Manual pentesting
- Automated security regression tests



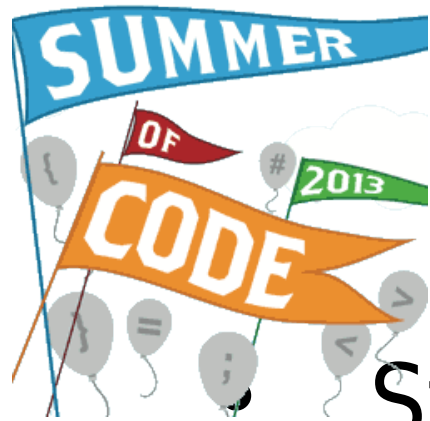


- New Spider plus Session awareness
Cosmin Stefan
- Ajax Spider via Crawljax
Guifre Ruiz
- WebSockets support
Robert Kock

All in current release (2.1.0)



- Enhanced HTTP Session Handling
Cosmin Stefan
- SAML 2.0
Pulasthi Mahawithana
- Advanced Reporting using BIRT
Rauf Butt
- CMS Scanner
Abdelhadi Azouni
- Dynamically Configurable Actions
Alessandro Secco



Enhanced Sessions

Student: Cosmin Stefan – Studying for MSc at University of Denmark

- Mentor: Guifre Ruiz (GSoC student 2012)
- Project: Plugable, fully integrated session and authentication, (semi) automation of access control testing, a platform to build on
- Status: Finalizing standardized session and authentication handling across ZAP



Enhanced Sessions

Session Properties

▼ Session

General

Exclude from proxy

Exclude from scanner

Exclude from spider

▼ Contexts

▼ 1

1: Include in context

1: Exclude from context

1: Technology

1: Users Authentication

1: Authentication

1: Session Management

1: Users

1: Users Authentication

This panel allows you to configure the authentication scheme used for this Context.

Currently selected Authentication method for the Context:

Form-based Authentication

Configure Authentication Method

Login URL:

http://www.example.com/login.jsp

Login POST Data (if any):

user={%username%}&pass={%password%}&rememberme=1

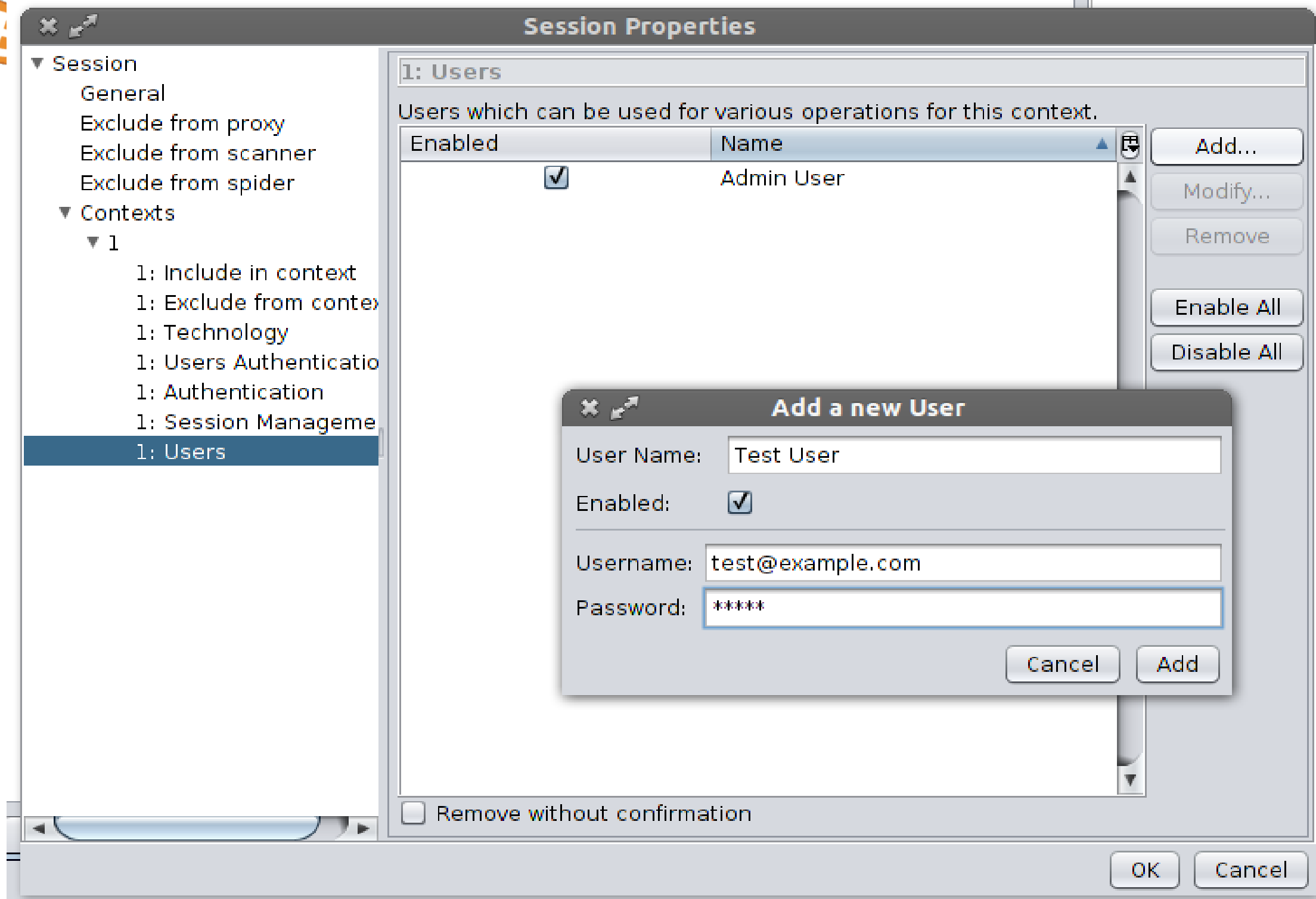
* The {%username%} and {%password%} substrings will be replaced, during authentication, with the username and password corresponding to application's users.

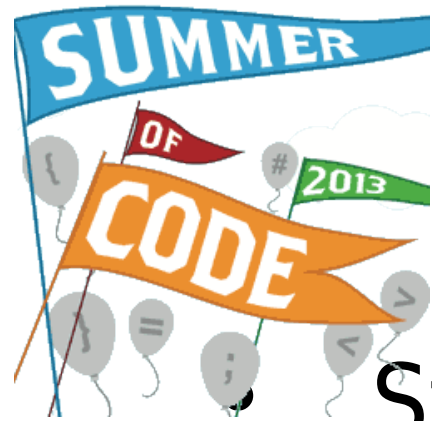
OK

Cancel



Enhanced Sessions

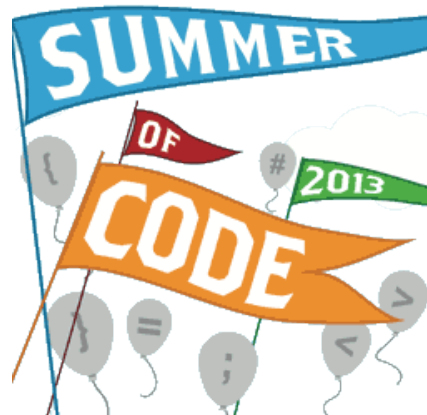




SAML 2.0

Student: Pulasthi Mahawithana – Studying at University of Moratuwa, Sri Lanka

- Mentors: Prasad Shenoy, Kevin Wall
- Project: Detect, decode and fuzz SAML messages, simulate XSW attacks
- Status: Replay and attack SAML requests



SAML 2.0

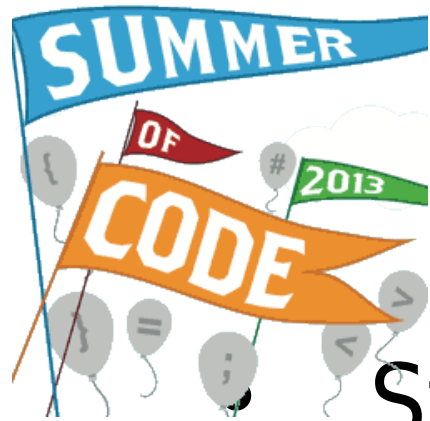
SAML Request editor

Request Response

```
<?xml version="1.0" encoding="UTF-8"?><samlp:AuthnRequest xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol" AssertionConsumerServiceURL="http://localhost:8080/SSOSampleApp/ssologin" Issuer="SSOSampleApp" NameIDPolicy="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent" RequestedAuthnContext="urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport" Comparison="exact" Format="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent" SPNameQualifier="urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport" Version="2.0" xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol" />
```

IssueInstant (AuthnRequest)	2013-08-13T16:16:43.138Z
NameIDPolicy SPNameQualifier	Issuer
ProtocolBinding	urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST
Version (AuthnRequest)	2.0
AuthnRequest ID	0
Attrib Consuming Serv. Index	1239245949
NameIDPolicy AllowCreate	true
NameIDPolicy Format	urn:oasis:names:tc:SAML:2.0:nameid-format:persistent
AuthnContextClassRef (Authn.)	urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
Comparison	exact
AssertionConsumerServiceURL	http://localhost:8080/SSOSampleApp/ssologin
Issuer (AuthnRequest)	SSOSampleApp

Resend Reset

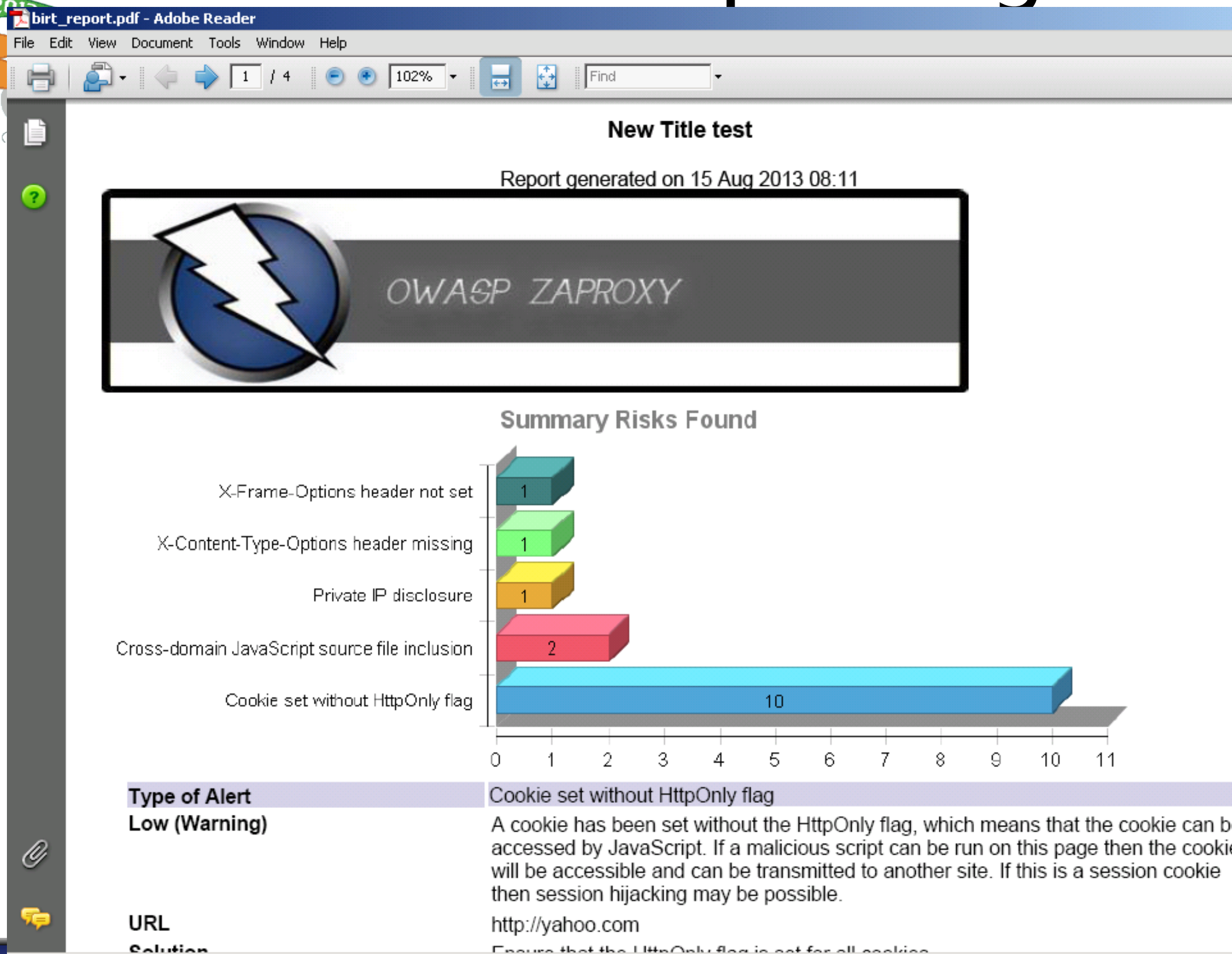


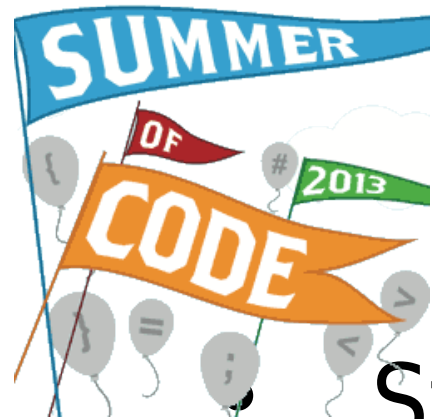
Advanced Reporting

Student: Rauf Butt

- Mentors: Johanna Curiel
- Project: Flexible, plugable and highly configurable BIRT generated reports
- Status: Prototype add-on available, generates PDFs inc charts

Advanced Reporting





CMS Scanner

Student: Abdelhadi Azouni

- Mentors: Mennouchi Islam Azedine
- Project: Fingerprint CMS software and versions, enumerate vulnerabilities in core, plugins or templates
- Status: Alpha fingerprinting extension available now



CMS Scanner

Untitled Session - OWASP ZAP

File Edit View Analyse Report Tools Online Help

Standard mode

Sites

Quick Start Request Response Break

Welcome to the OWASP Zed Attack Proxy (ZAP)

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

Given permission to test.

Regression tests while proxying through ZAP.

Fingerprinting tools

Fingerprint Details

Target:

☐ Get version
☒ Passive
☐ Agressive

App name:
Version:

What to fingerprint ?

☒ cms
☐ message-boards
☒ javascript-framew...
☐ web-frameworks
☒ web-servers
☐ databases

Fingerprinting time and occurance sett...

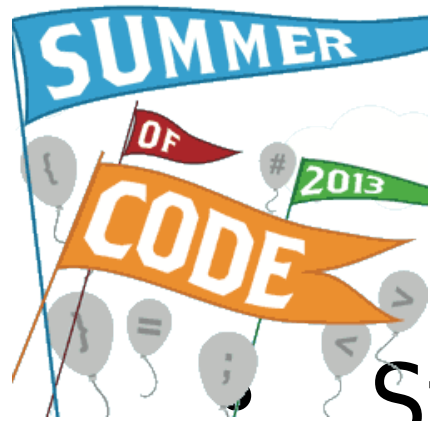
Show this tab on start up: ☒

History Search Break Points Alerts Active Scan Spider Forced Browse Fuzzer Params Http Sessions Fingerprint Output

Filter: OFF

Alerts 0 0 0 0

Current Scans 1 0 0 0 0 0



Dynamic actions


Student: Alessandro Secco, studying at University Padua, Italy

- Mentors: Simon Bennetts
- Project: Provide a very simple and flexible way to extend ZAP, replace old Paros Filters
- Status: <to be covered later;) >


New for
2.2.0


demo time....



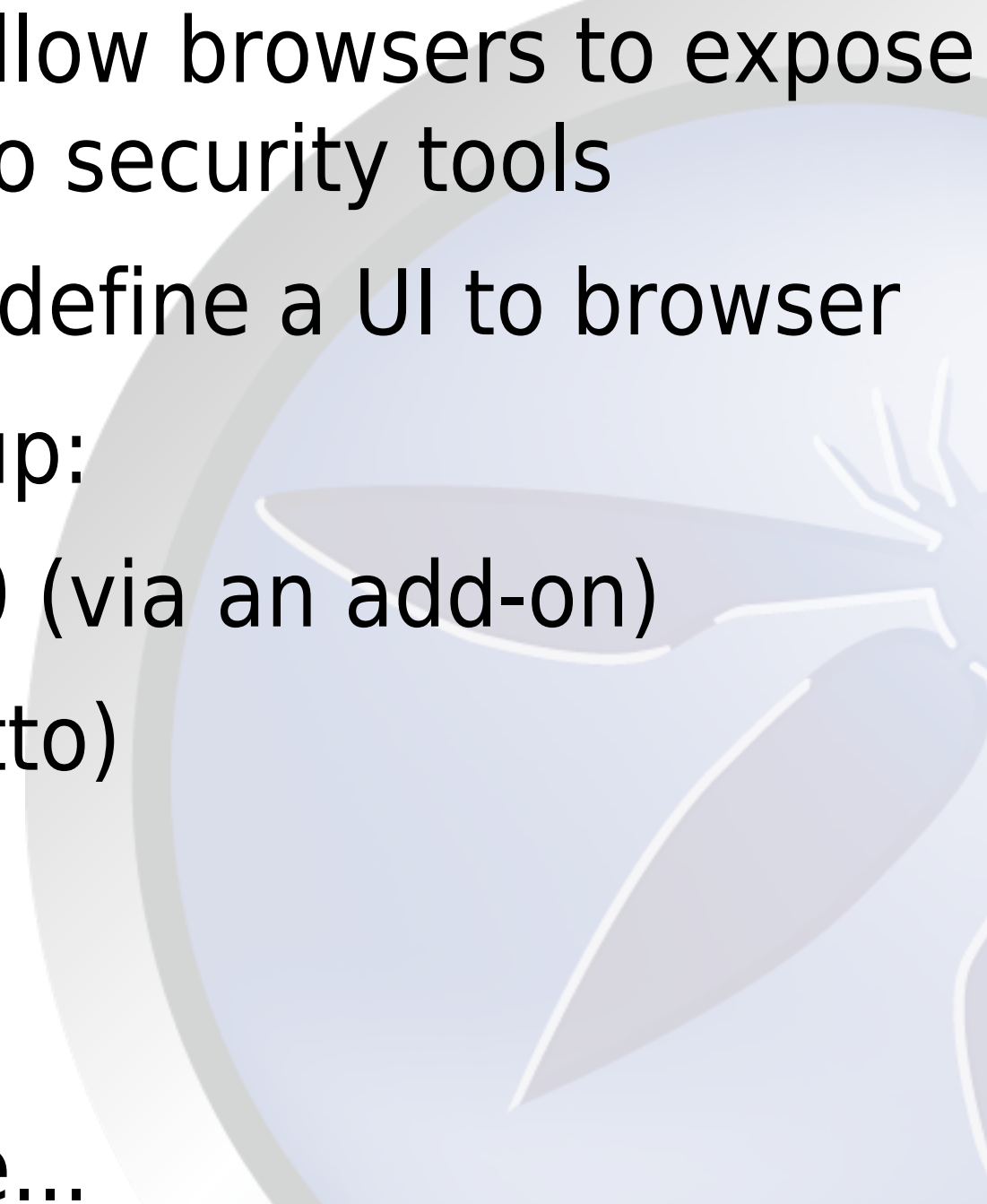


Plug-n-Hack

- Allow browsers and security tools to integrate more easily
 - Allows security tools to expose functionality to browsers
 - “Proposed standard”
 - Developed by Mozilla Security Team
 - Browser and security tool independent
- 



Plug-n-Hack

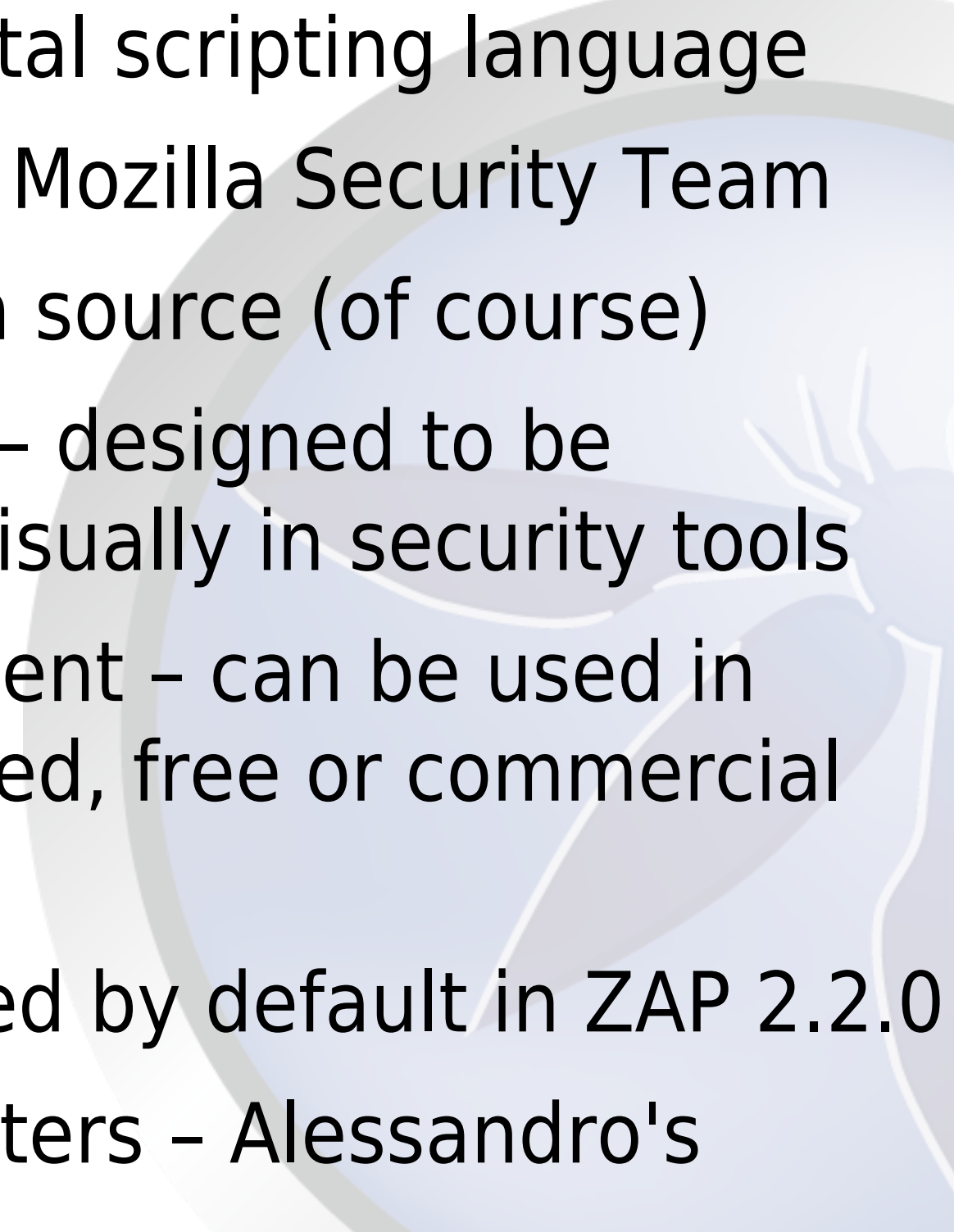
- Next phase: allow browsers to expose functionality to security tools
 - Allow tools to define a UI to browser
 - Tools signed up:
 - Firefox v 24.0 (via an add-on)
 - ZAP 2.2.0 (ditto)
 - Minion
 - Burp Suite
 - More to come...
- 

Script Console

- Current add-on just supports 'run now' scripts
- In 2.2.0 it will be embedded into ZAP.
- Different types of scripts
 - Stand alone As now
 - Targeted Specify URLs to run against
 - Active Run in Active scanner
 - Passive Run in Passive scanner
 - Proxy Run 'inline'
 - Library Use in other scripts



Zest - overview

- An experimental scripting language
 - Developed by Mozilla Security Team
 - Free and open source (of course)
 - Format: JSON – designed to be represented visually in security tools
 - Tool independent – can be used in open and closed, free or commercial software
 - Will be included by default in ZAP 2.2.0
 - Will replace filters – Alessandro's project
- 


Zest - statements

- HTTP(S) Requests
- Assertions
- Conditionals
- Assignments
- Actions
- Loops
- More to come ...



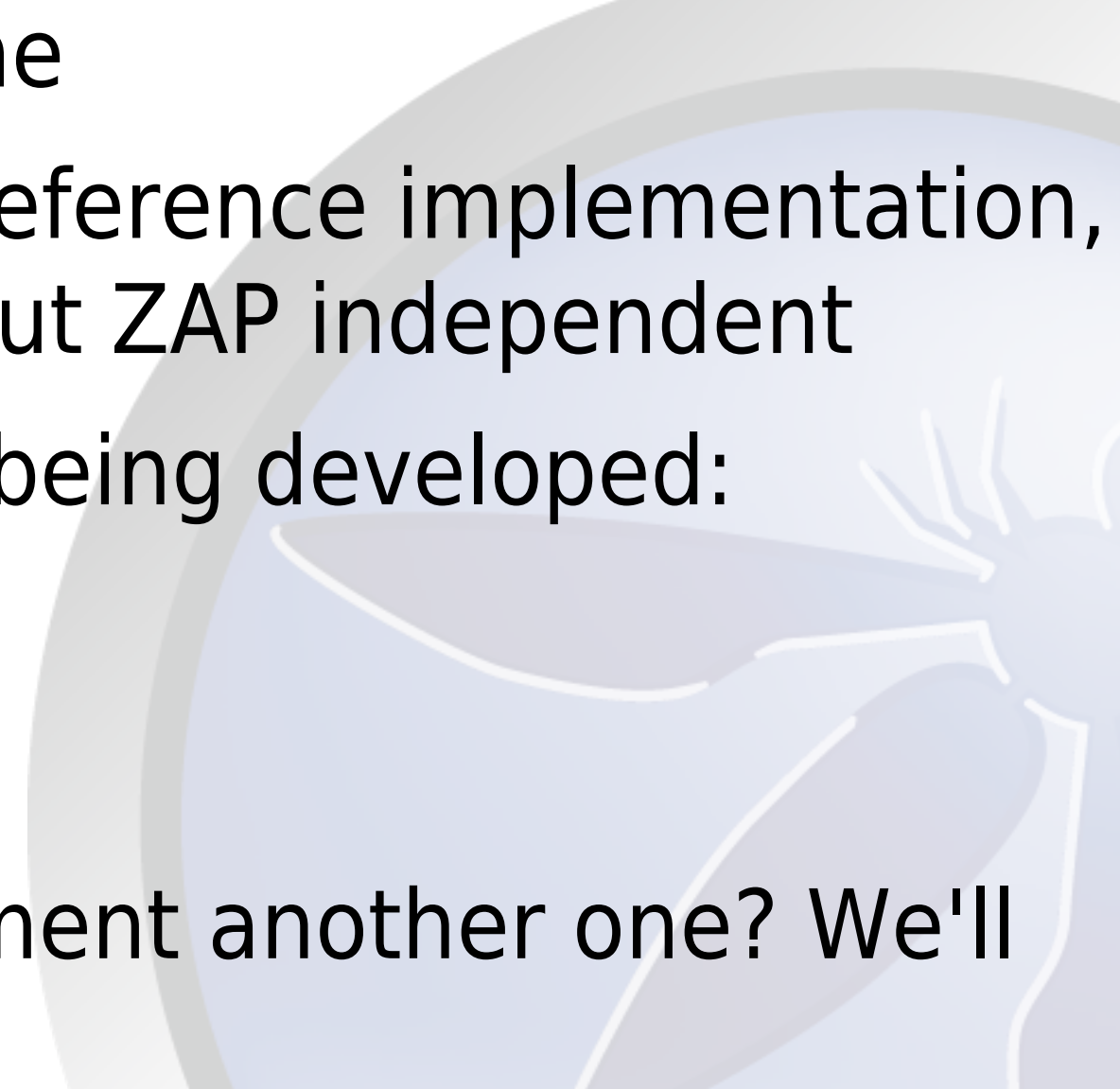


Zest - use cases

- Reporting vulnerabilities to companies
 - Reporting vulnerabilities to developers
 - Defining tool independent active and passive scan rules
 - Deep integration with security tools
- 



Zest - runtime

- Needs a runtime
 - Java runtime: reference implementation, used by ZAP, but ZAP independent
 - Runtimes also being developed:
 - Javascript
 - Python
 - Want to implement another one? We'll help you :)
- 

Conclusion

- ZAP is changing rapidly
- New features are being introduced which will exceed the capabilities of other tools
- We're implementing functionality so that it can be reused in other tools
- Its a community based tool - get involved!
- We want feedback - fill in the Questionnaire!
(linked of ZAP homepage)



Questions?

<http://www.owasp.org/index.php/ZAP>