



WATOBO

The Web Application Toolbox

Andreas Schmidt

SIBERAS

<http://www.siberas.de>

OWASP

20.10.2010

Copyright © The OWASP Foundation
Permission is granted to copy, distribute and/or modify this document
under the terms of the OWASP License.

The OWASP Foundation
<http://www.owasp.org>

Bio

■ Andreas Schmidt

- ▶ Seit 1998 im Security-Bereich tätig
- ▶ Seit 2001 spezialisiert auf Audits/Penetrationstests
- ▶ Mitgründer von siberas (2009)
 - <http://www.siberas.de>

Agenda

- (Markt-)Überblick
- Motivation
- Hauptkomponenten
- Highlights
- RoadMap
- Demo: WATOBO in action

Überblick

■ Kommerzielle Tools

- ▶ WebInspect, AppScan, NTOSpider, Acunetix,
- ▶ Primär für automatisierte Audits

■ Freie Tools

- ▶ WebScarab, Paros, BurpSuite(+\$\$), ...
- ▶ Primär für manuelle Penetrationstests

■ 1001+ Script-Tools

- ▶ Nikto, sqlmap, ...

Motivation

- Warum noch ein Tool?

Motivation

- Kosten/Nutzen-Verhältnis von (kommerziellen) automatisierten Tools zu hoch!
 - ▶ Typische Nachteile vollautomatisierter Tools, z.B. Logik-Fehler, ...
 - ▶ manuelle „Begehung“ der Applikation trotzdem notwendig
- Daseinsberechtigung dennoch gegeben!
 - ▶ Einfache Bedienung, Reporting, zentrales Management, QA-Schnittstellen, ...

```
pay() if pentester.needsFeature?(feature)
```

Motivation

- Fehlende Transparenz bei kommerziellen Scannern
 - ▶ Check-Methoden werden meist „geheim“ gehalten
 - ▶ Zuviel „Voodoo“



Motivation

- Manuelle Tools besitzen meist kein Session-Management
 - ▶ Erneutes Einloggen notwendig
 - ▶ Mühsames kopieren der SessionID

- Anpassen von (kommerziellen) Tools meist nur schwer möglich
 - ▶ Fehlender Source-Code
 - ▶ Entwicklungsumgebung/Compiler notwendig
 - ▶ Oftmals umständlich und unflexibel, z.B. XML,

Motivation

- Manuelle Tools haben oft nur begrenzte automatisierte Funktionen
 - ▶ Ausnahme: BurpSuite Pro (\$\$)
- Vorteile quell-offener Tools
 - ▶ Leistungsfähigkeit und Grenzen können eingeschätzt werden
 - ▶ Können schnell an neue Anforderungen angepasst werden
 - ▶ Skript-Sprachen



Ansatz: Vorteile beider „Welten“

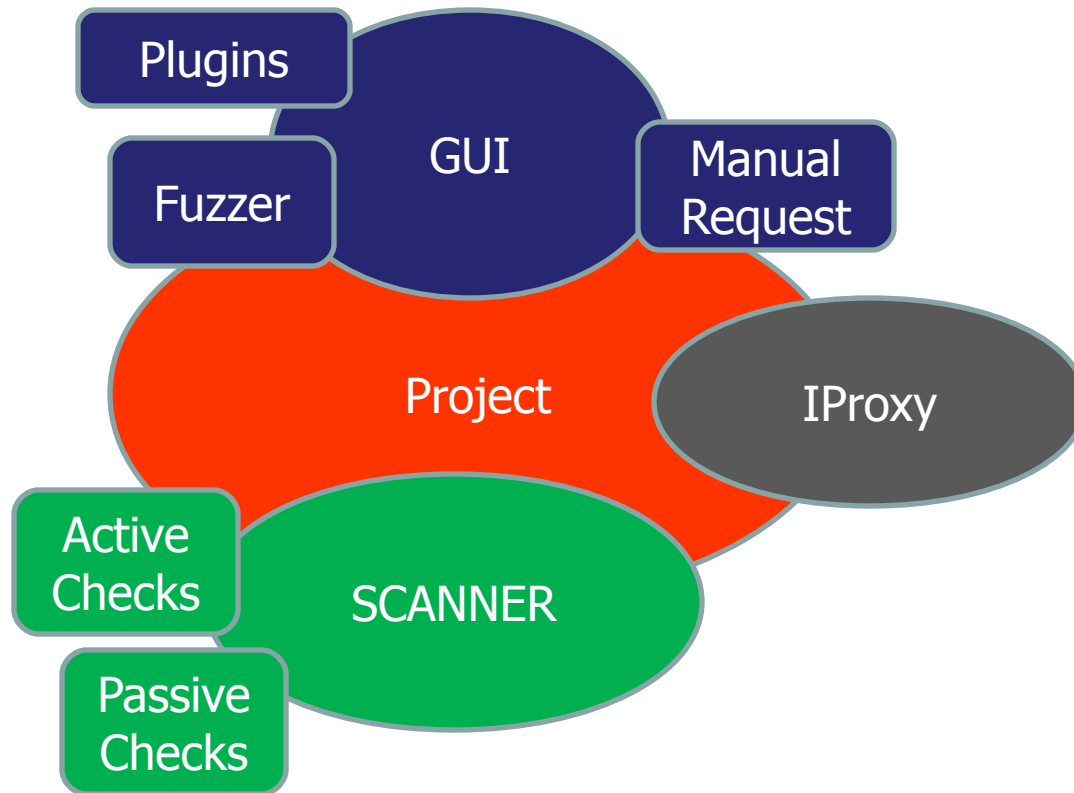
- Fokus: semi-automatisierte Penetrationstests
- Session-Management
- Proxy-Basiertes Tool
- Web-Testing-Framework
 - ▶ typische Funktionen, wie Parser, Shaper, ...
 - ▶ einfach zu erweitern!
- Kein Angriffswerkzeug!
 - ▶ Keine Exploitmodule in Open-Source-Version

Zielgruppe

- Primär für professionelle Pentester!
 - ▶ Idealerweise mit Ruby-Kenntnissen

- Aber auch für Entwickler, Admins,...
 - Basis-Checks einfach durchzuführen
 - Kurze Beschreibung der Schwachstellen sowie Maßnahmenempfehlung

Komponentenüberblick



Komponente: GUI

- GUI ist ein Muss!

- ▶ Web-App-Analyse ohne GUI nicht möglich
- ▶ CLI nicht für alle Bereiche sinnvoll ;)

- Für manuelle Tests optimiert

- ▶ One-Click En-/Decoder
- ▶ Filter Funktionen
- ▶ Schnelle Analyse der Funktionsweise

Komponente: GUI

The screenshot displays the WATOBO (Version: 0.9.5rev202) interface. On the left, a tree view shows various vulnerability categories like SQL-Injection, Reflected XSS, Local File Inclusion, and Security Options. The main area contains a table of findings with columns for Method, Host, Path, Parameters, Status, Set-Cookie, and Comment. A detailed view of a 'Local File Inclusion' finding (Chat-ID: 24) is shown on the right, including the request and response details.

Apply	Method	Host	Path	Parameters	Status	Set-Cookie	Comment
	1	GET	192.168.72.130	dwva	301	Mov	
	2	GET	192.168.72.130	dwva/	302	Foun	PHPSESSID=
	3	GET	192.168.72.130	dwva/login.php	200	OK	
	6	POST	192.168.72.130	dwva/login.php	302	Foun	username=admin&passwo
	7	GET	192.168.72.130	dwva/index.php	200	OK	
	13	GET	192.168.72.130	dwva/vulnerabilities/xss_r/	200	OK	
	14	GET	192.168.72.130	dwva/vulnerabilities/xss_r/	200	OK	
	15	GET	192.168.72.130	dwva/vulnerabilities/xss_r/	200	OK	name=
	16	GET	192.168.72.130	dwva/vulnerabilities/xsqli/	200	OK	
	17	GET	192.168.72.130	dwva/vulnerabilities/xsqli/	200	OK	
	18	GET	192.168.72.130	dwva/vulnerabilities/xsqli/	200	OK	id=&Submit=Submit
	19	GET	192.168.72.130	dwva/security.php	200	OK	
	20	GET	192.168.72.130	dwva/security.php	200	OK	
	22	POST	192.168.72.130	dwva/security.php	302	Foun	security=low&secl
	23	GET	192.168.72.130	dwva/security.php	200	OK	
	24	GET	192.168.72.130	dwva/vulnerabilities/ff/	200	OK	page=include.php
	25	GET	192.168.72.130	dwva/vulnerabilities/ff/	200	OK	page=include.php
	26	GET	192.168.72.130	dwva/vulnerabilities/xsqli/	200	OK	
	27	GET	192.168.72.130	dwva/vulnerabilities/xsqli/	200	OK	
	28	GET	192.168.72.130	dwva/vulnerabilities/xsqli/	200	OK	id=&Submit=Submit
	29	GET	192.168.72.130	dwva/vulnerabilities/xss_r/	200	OK	
	30	GET	192.168.72.130	dwva/vulnerabilities/xss_r/	200	OK	
	31	GET	192.168.72.130	dwva/vulnerabilities/xss_r/	200	OK	name=
	32	GET	192.168.72.130	dwva/vulnerabilities/upload/	200	OK	
	33	POST	192.168.72.130	dwva/vulnerabilities/upload/	200	OK	
	34	GET	192.168.72.130	dwva/vulnerabilities/exec/	200	OK	
	35	GET	192.168.72.130	dwva/vulnerabilities/exec/	200	OK	ip=&submit=submit
	36	POST	192.168.72.130	dwva/vulnerabilities/exec/	200	OK	
	37	GET	192.168.72.130	dwva/logout.php	302	Foun	
	38	GET	192.168.72.130	dwva/vulnerabilities/upload/	302	Foun	
	39	GET	192.168.72.130	dwva/logout.php	302	Foun	
	40	GET	192.168.72.130	dwva/vulnerabilities/upload/	302	Foun	
	41	GET	192.168.72.130	dwva/login.php	200	OK	
	42	POST	192.168.72.130	dwva/login.php	302	Foun	username=admin&passwo
	43	GET	192.168.72.130	dwva/index.php	200	OK	
	44	GET	192.168.72.130	dwva/vulnerabilities/xsqli/	302	Foun	PHPSESSID=
	45	GET	192.168.72.130	dwva/login.php	200	OK	security=higi
	46	GET	192.168.72.130	dwva/vulnerabilities/xsqli/	302	Foun	PHPSESSID=
	48	POST	192.168.72.130	dwva/login.php	302	Foun	username=admin&passwo
	49	GET	192.168.72.130	dwva/index.php	200	OK	

Finding "Local File Inclusion" [Chat-ID: 24]

Request:

```
Text | Hex | Grep | Highlight | Reset
```

```
GET http://192.168.72.130/dwva/vulnerabilities/ff/?page=../../../../etc/passwd HTTP/1.1
Host: 192.168.72.130
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.0; de; rv:1.9.2.8) Gecko/20100722 Firefox/3.6.8 (.NET CLR 3.5.30729)
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
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 115
Referer: http://192.168.72.130/dwva/security.php
Cookie: security=low; Milano=0012AA9B12goodandy; Brussels=0029A9B91crisp5; Geneva=92BEF34Apecan635; PHPSESSID=b9474c1d4707b54646ed6c
Connection: Close
Proxy-Connection: Close
Accept-Encoding: None
```

Response:

```
Text | Tagless | Hex | Grep | Highlight | Reset
```

```
HTTP/1.1 200 OK
Date: Fri, 15 Oct 2010 16:08:38 GMT
Server: Apache/2.2.12 (Ubuntu)
X-Powered-By: PHP/5.2.10-2ubuntu6.1
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 5324
Connection: close
Content-Type: text/html

root:x0:root:/root:/bin/bash
daemon:x1:daemon:/usr/sbin:/bin/sh
bin:x2:bin:/bin:/bin/sh
sync:x3:sync:/dev:/bin/sh
syncx4:65534:sync:/bin:/bin/sync
games:x5:games:/usr/games:/bin/sh
man:x6:12:man:/var/cache/man:/bin/sh
lpx:7:7lp:/var/spool/lpd:/bin/sh
mail:x8:8:mail:/var/mail:/bin/sh
news:x9:9:news:/var/spool/news:/bin/sh
uucpx:10:10:uucp:/var/spool/uucp:/bin/sh
```

Komponente: I(nterceptor/)Proxy

- Klassische Proxy-Funktion
- Interceptor
 - ▶ Abfangen und Manipulieren von Requests/Responses
- Pass-Through
 - ▶ Server-Antwort wird direkt an Browser durchgereicht
 - ▶ Einstellbar: Content-Type/Content-Length
 - ▶ Applikation lässt sich flüssig bedienen!
- Pseudo-Server
 - ▶ Z.b für HTML-Preview

Komponente: Scanner

- Multi-Threaded
- Smart-Scan-Funktion
 - ▶ Reduziert Anzahl von Requests
 - ▶ Ähnliche URLs werden zusammengefasst
 - ▶ Berücksichtigt „Non-Unique-Parameter“
 - Z.B. `action=addUser` oder `function=showFile`
- Steuert Active-Checks

Komponente: Scanner

■ Feingranulare Definition des Target-Scopes

- ▶ Site (host:port)
- ▶ Root-Path
- ▶ Exclude-Patterns

■ Session-Management

- ▶ Erkennt Logout
- ▶ Kann (Re-)Login automatisiert durchführen

Komponente: Fuzzer

- Multi-Tag
- Multi-Generator
- Multi-Action
- Multi-Filter
- ...



USE THE FORCE, ...

Komponente: Fuzzer

The screenshot displays the Fuzzer application window, which is divided into several sections:

- Request:** Shows the current HTTP request being fuzzed. The request is: `GET http://192.168.72.130/watobo/%%TAG2%%.php HTTP/1.1`. Other headers include `Host: 192.168.72.130`, `User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.0; de; rv:1.9.2.8) G`, `Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=`, `Accept-Language: de-de,de;q=0.8,en-us;q=0.5,en;q=0.3`, `Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7`, `Keep-Alive: 115`, `Cookie: Milano=%%FUZZ%%; Brussels=0029A9B91crisp5; Geneva=92E`, `Connection: Close`, `Proxy-Connection: Close`, and `Accept-Encoding: None`. A `Reset` button is located to the right of the request text.
- Start:** A button to initiate the fuzzing process.
- Request Options:** A section with three checkboxes: Update Content-Length, Update Session Information, and Run Login.
- Settings:** A tree view showing the configuration of the fuzzer components:
 - Tags:**
 - Tag: FUZZ:** Contains a **Counter** component with settings `start=0/stop=100/step=1` and a **Ruby: Proc** component with the code `proc { |input|# place your code betweenhere`.
 - Tag: TAG2:** Contains a **List-Input** component with `4 values`.
 - Filters:**
 - Filter: Regex:** Contains the regular expression `(HTTP[^\n]*OK)`.
- Results:** A section for displaying the results of the fuzzing process, currently empty.
- Logs:** A section for displaying the application logs, currently empty.

Komponente: Manual Request Editor

- Automatisierter Login
- Update der Session-Informationen
- Request-History
- Differ
- QuickScan
 - ▶ Gezieltes Scannen einer URL

Komponente: Manual Request Editor

The screenshot displays the Manual Request Toolkit interface. The main window is titled "Manual Request Toolkit" and contains several sections:

- History:** Shows a list of requests, with the current one selected. The selected request is: `GET http://192.168.72.130/dvwa/security.php HTTP/1.1`. The response is: `HTTP/1.1 200 OK`.
- Request Options:** A panel with checkboxes for Update Content-Length, Update Session Information, Run Login, Use Original Request (QuickScan), and Log Chat. A **SEND** button is highlighted with a red box.
- Request/Response Editor:** A large text area showing the request and response. The request is: `GET http://192.168.72.130/dvwa/security.php HTTP/1.1`. The response is: `HTTP/1.1 200 OK`. The response body contains HTML code for a Damn Vulnerable Web App (DVWA) page.
- Logs:** A section at the bottom showing a log of events: `10/15/2010 @ 18:17:54: got answer`, `10/15/2010 @ 18:17:54: sending request`, `10/15/2010 @ 18:17:54: got answer`, and `10/15/2010 @ 18:17:54: sending request`.

Komponente: Active Checks

- Werden über Scanner gesteuert
- Dienen zum aktiven Testen
 - ▶ SQL-Injection
 - ▶ XSS
 - ▶ ...
- Gute Balance zwischen Einfachheit/Flexibilität
 - ▶ Nur mit Skript-Sprachen möglich!
 - ▶ Einige Hersteller haben eigene (Skript-)Sprachen, oder nutzen JavaScript

Komponente: Active Checks

Aktuelle Checkliste (13):

- + Dirwalker
- + Fileextensions
- + Http_methods
- + Domino_db
- + Lfi_simple
- + Jboss_basic
- + Its_commands
- + Its_services
- + Its_service_parameters
- + Its_xss
- + Sqli_simple
- + Sql_boolean
- + Xss_simple

**IN STÄNDIGER
ENTWICKLUNG**

Komponente: Passive Checks

- Grep-Style-Checks

- ▶ Pattern-Matching

- Identifiziert Schwachstellen

- ▶ Z.B. Cookie-Security, unverschlüsselte Anmeldung, ...

- Extrahiert hilfreiche Informationen

- ▶ Z.B. HotSpots, Email, IP's...

Komponente: Passive Checks

Aktuelle Checkliste (14):

- + Cookie_options
- + Cookie_xss
- + Detect_code
- + Detect_fileupload
- + Detect_infrastructure
- + Dirindexing
- + Disclosure_emails
- + Disclosure_ipadd
- + Filename_as_paramete
- + Hotspots
- + Multiple_server_headers
- + Possible_login
- + Redirectionz
- + Redirect_url

**IN STÄNDIGER
ENTWICKLUNG**

Komponente: Plugins

■ Für individuelle Tests

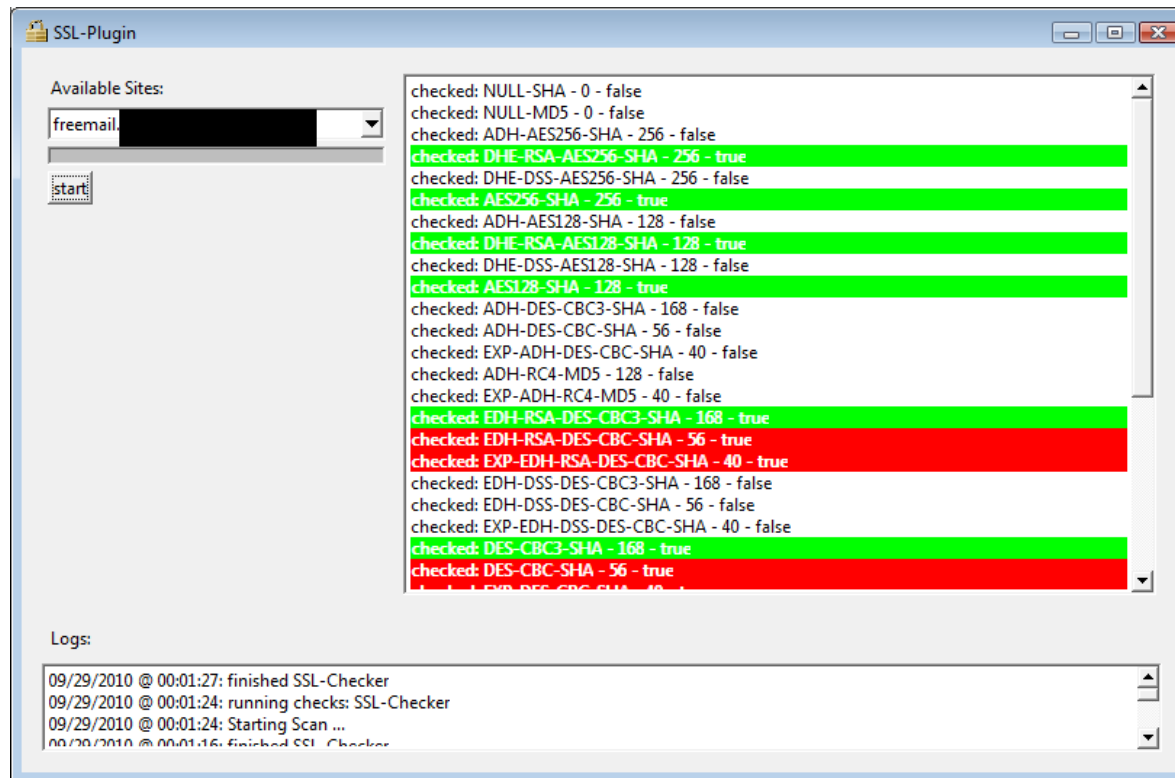
- ▶ Nicht Scanner-kompatibel
- ▶ Z.B. site-spezifische Checks, wie beispielsweise SSL-Cipher

■ Framework-Funktionen und Schnittstellen

- ▶ listSites, listDirs, ...
- ▶ SessionManagement
- ▶ Scanner

Plugin: SSL-Checker

- Prüft unterstützte SSL-Ciphers
 - ▶ Mittels vollständigen HTTP-Requests



Umsetzung

■ Ruby, Ruby, Ruby, ...

- ▶ <http://www.ruby-lang.org>

■ FXRuby für GUI

- ▶ Ruby-Port von Fox-Toolkit
<http://www.fxruby.org>

■ Plattformunabhängig

- ▶ (FX)Ruby für Windows, Linux, MacOS, ...

■ Entwicklungsplattform Windows

- ▶ Wird auch unter Linux (Backtrack) getestet



WATOBO Highlights

- Session Management
- Ruby-In-Ruby
- HTML-Preview

Highlight: Session Management

- Pattern-basiert

- ▶ Regular Expressions

- ▶ Hash[\$1]=\$2

- Header und Body wird analysiert

- ▶ Nur text/*-Content-Types => Geschwindigkeit

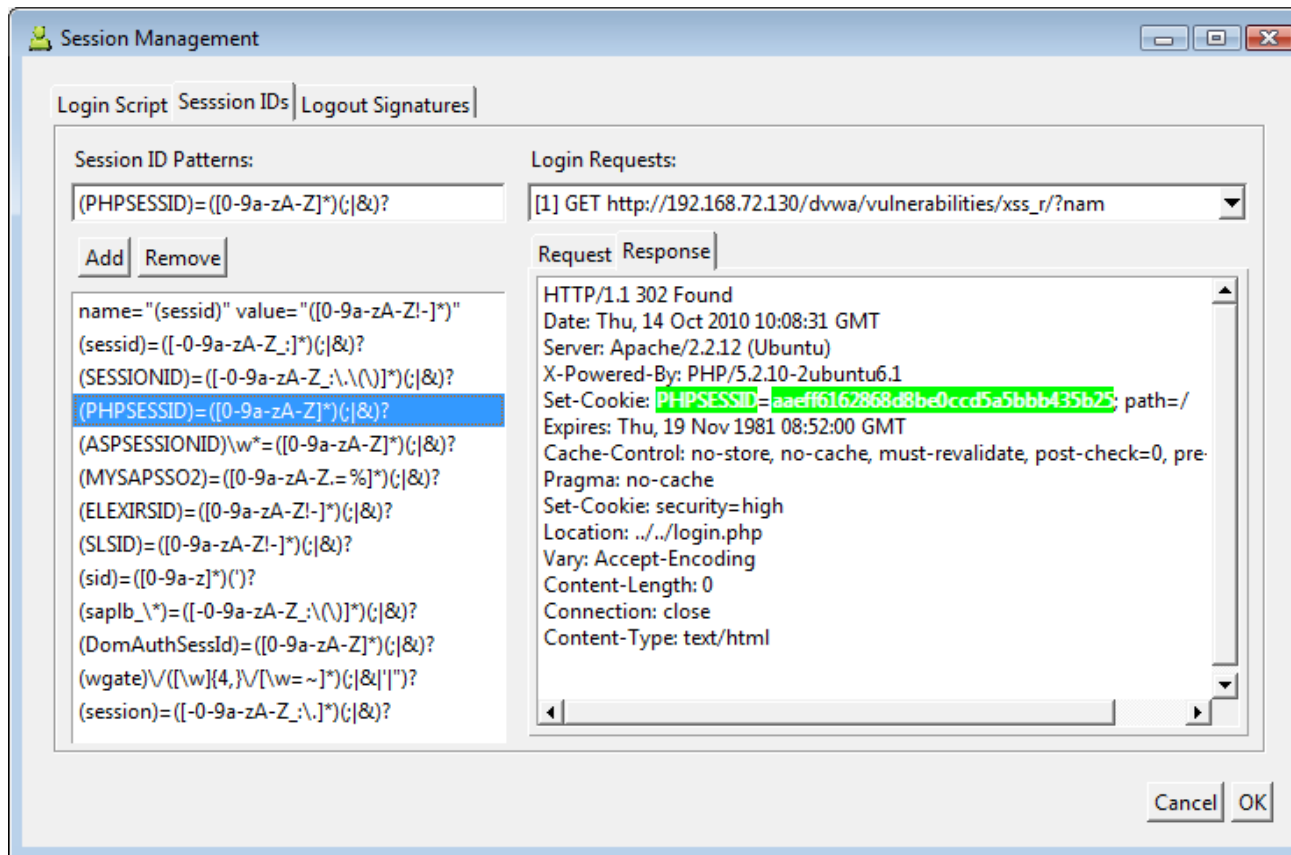
- Session-IDs in Cookie und URLs

- Ca. 15 vordefinierte Patterns

- Regex-Validator

Highlight: Session Management

Beispiel: (PHPSESSID)=([0-9a-zA-Z]*)(;|&)?



Highlight: Ruby-in-Ruby

- Mittels spezieller Tags (`,%%'`) lässt sich direkt Ruby-Code integrieren
- Nützlich für die Erzeugung von
 - ▶ vielen Zeichen, Headern, ...
 - ▶ Binaerzeichen, Konvertierung, Berechnungen, ...
 - ▶ Daten aus verschiedenen Quellen, z.B. Dateien
- Fuzzer nutzt Ruby (procs) für „Actions“

Highlight: Ruby-in-Ruby

Manual Request Editor: Including Binary-Files

The screenshot displays the Manual Request Toolkit interface. The main window shows a request editor with the following content:

```
Upload
-----41184676334
Content-Disposition: form-data; name="upload_registr
Content-Type: text/html

%%File.open("C:\UploadTest\php04.pdf", "rb")%%
41184676334
```

Below the request editor, there are "Request Options" and a "SEND" button:

- Update Content-Length
- Update Session Information
- Run Login
- Use Original Request (QuickScan)
- Log Chat

A "SEND" button is highlighted with a red box. Below it is a "preview >>" button.

The "Response" pane on the right shows the following content:

```
Upload
-----41184676334
Content-Disposition: form-data; name="upload_registr
Content-Type: text/html

Upload
-----41184676334
Content-Disposition: form-data; name="upload_registr
Content-Type: text/html

Upload
-----41184676334
Content-Disposition: form-data; name="upload_registr
Content-Type: text/html

Upload
-----41184676334
Content-Disposition: form-data; name="upload_registr
Content-Type: text/html

%PDF-1.5
%....
20 0 obj
<?php
print("PWNED!!!");
phpinfo();
exit();
?>
```

The response content is highlighted with a green box.

At the bottom, the "Logs" section shows the following entries:

```
09/28/2010 @ 11:18:28: got answer
09/28/2010 @ 11:18:25: sending request
09/28/2010 @ 11:18:09: got answer
09/28/2010 @ 11:18:00: sending request
```

Highlight: HTML-Preview

- HTML-Preview sehr hilfreich
 - ▶ Doku-Screenshots, schnelle visuelle Analyse
- FXRuby besitzt kein HTML(WebKit)-Widget ☹
- ..., aber Browser gibt's auf jedem System
 - ▶ IE, Firefox
- Browser-Steuerung mittels JSSH (Firefox) und Win32OLE (IE)
 - ▶ http://www.croczilla.com/bits_and_pieces/jssh/

Road-Map

- CSRF-Token Handling!
- Recheck-Funktion
 - ▶ KB-Diffing
- Neue Module, Plugins, Parser, En-/Decoder
 - ▶ SOAP/XML
- Source-Code-Unterstützung
 - ▶ zum Abgleich der Angriffsfläche

Road-Map

- Dokumentation
 - ▶ Videos, rdoc
- Installer
- Schulungen/Trainings/Workshops!

WATOBO - Demo