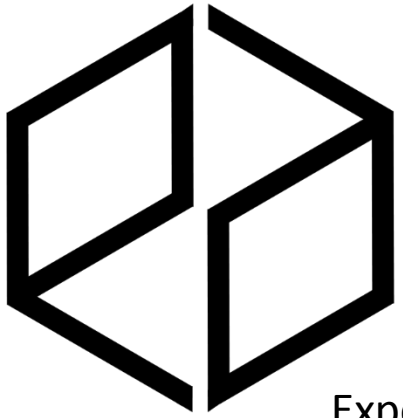
www.pandoralabs.net

PANDORA
SECURITY LABS

Expert Advice. Experience Advantage.
Proactive Security Solutions Through Cutting-Edge Research.

Web App Testing: Building Your Own Lab

By @isaacsabas



PANDORA
SECURITY LABS

Expert Advice. Experience Advantage.
Proactive Security Solutions Through Cutting-Edge Research.
www.pandoralabs.net

We are a Security-as-a-Service company

Providing businesses with on-demand threat detection & intelligence capabilities to secure their IT infrastructure, 24x7.

We Make IT Secure

WEB APPLICATION TESTING

Building Your Test Lab

/ˈprinsəpəl/

PRIN-CI-PLE

A rule or belief governing one's personal behavior.

Guiding Principles

Why build a lab?

Principles

- Do not test your skills on live sites; hence we build a lab.
- The lab should be segmented from your production environment.
- Once the lab is installed, always make a backup or take snapshot images of your lab (virtual environment).
- Document changes.

Virtual Box

Amazon AWS

VMware

The Environment

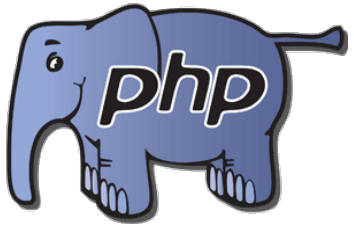
The sandbox environment where the target will be located.

Operating Systems

Web Servers

DB Servers

Platform

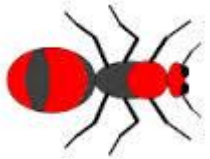


The Sandbox Stack

The sandbox and the needed applications and services.

Xtreme Vulnerable Web App

Damn Vulnerable Web App



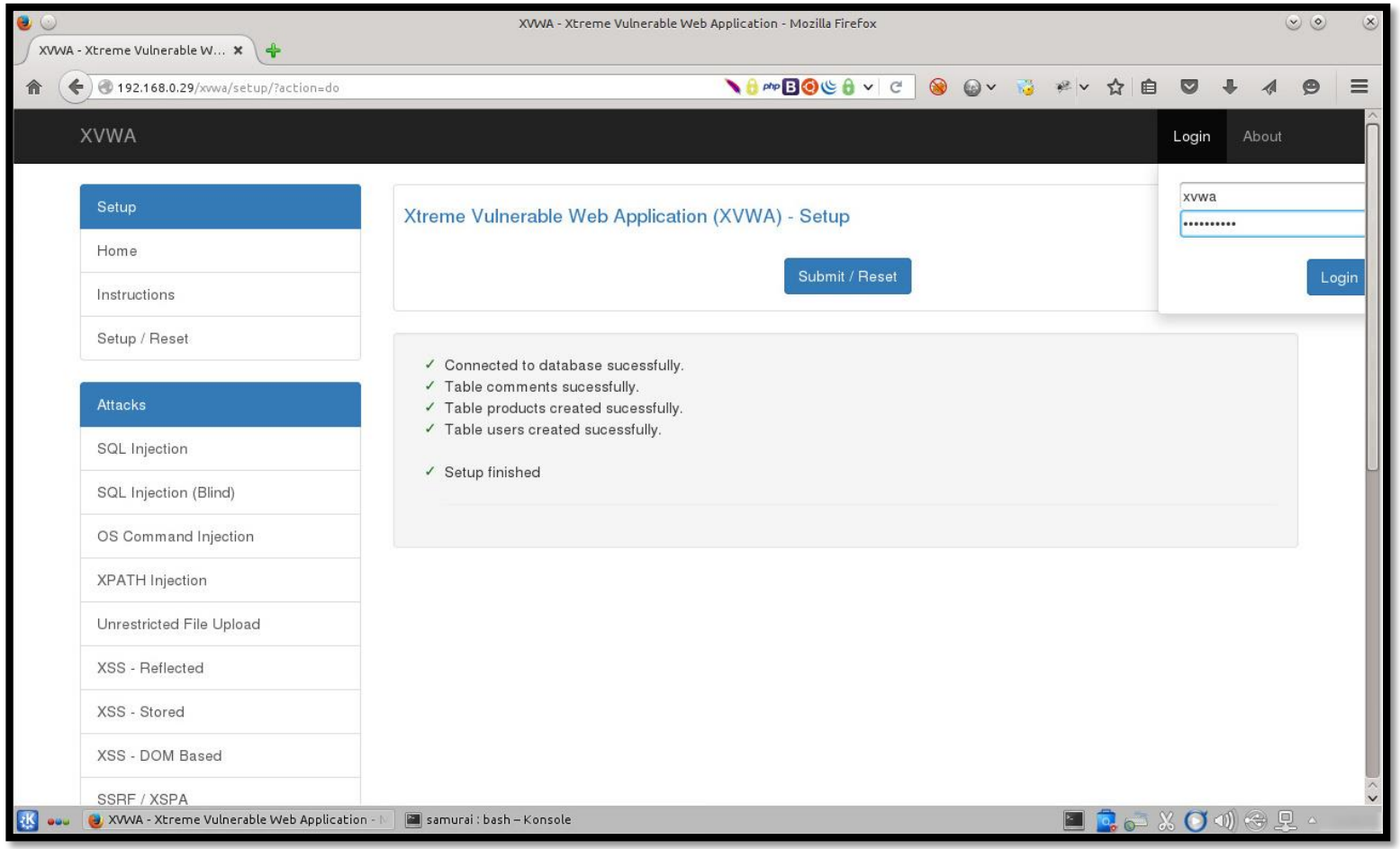
Mutillidae

WebGoat

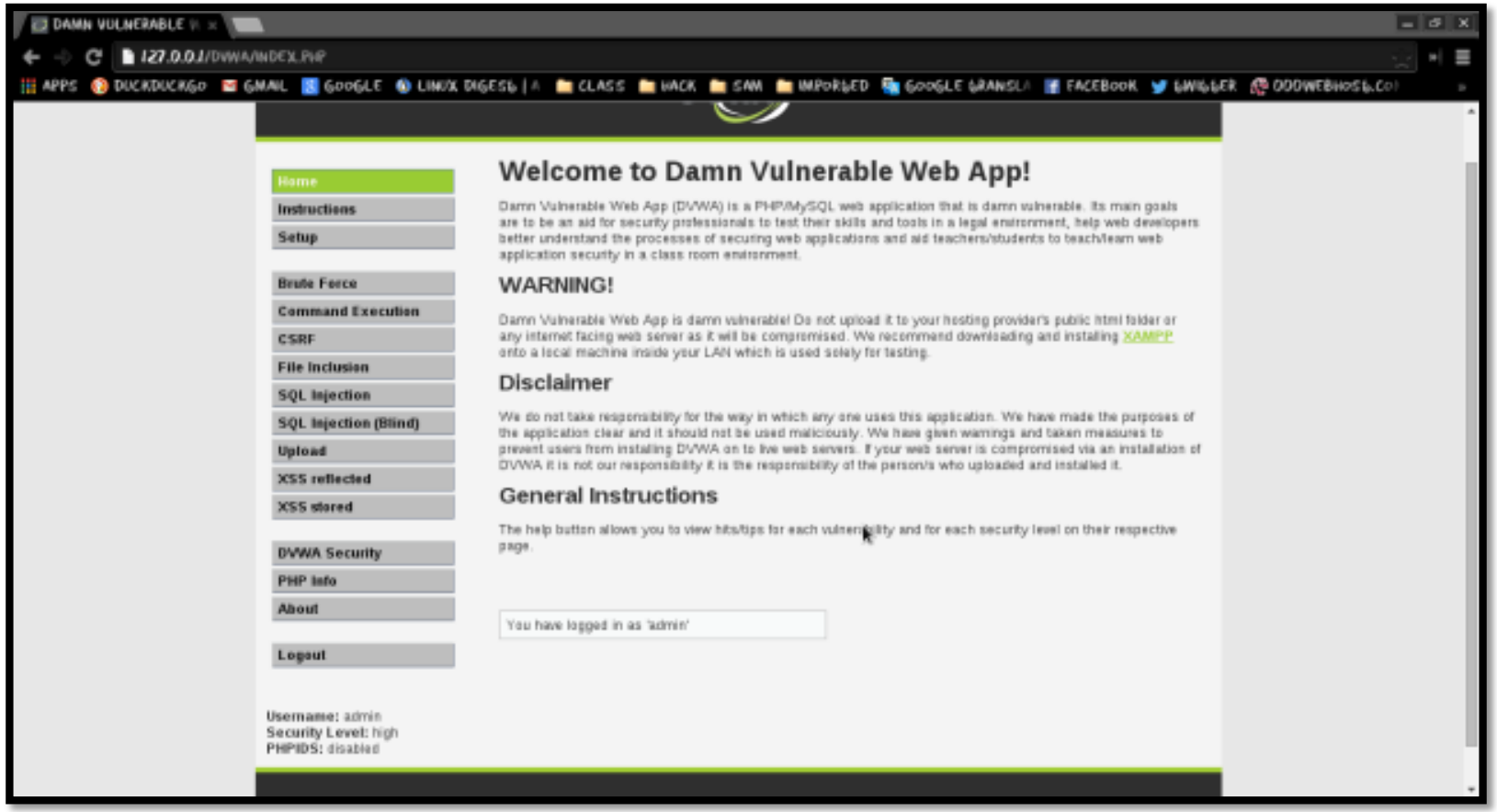
The Target Application

The target application that we can launch our attack against.

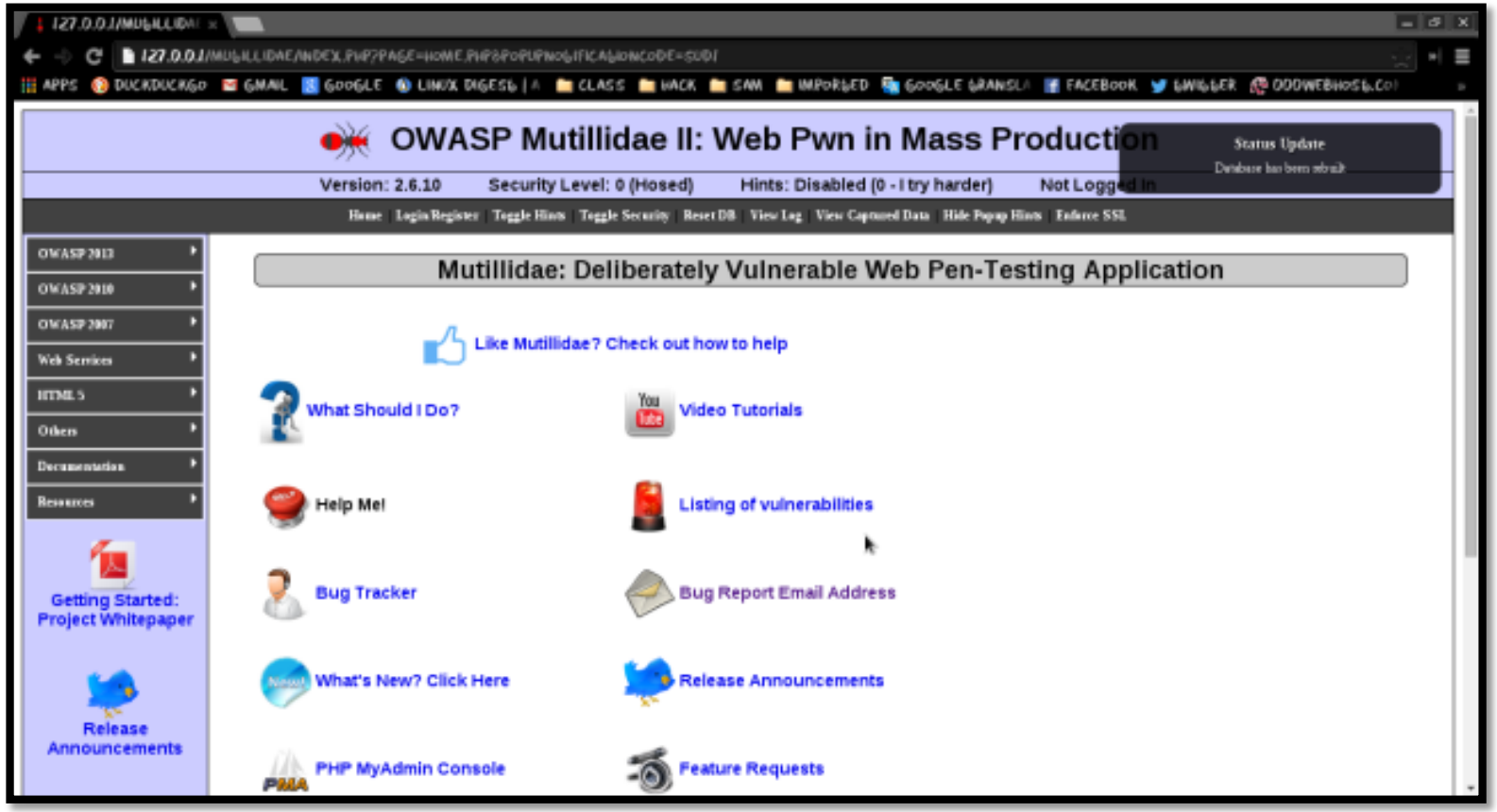
XVWA


XVWA - Xtreme Vulnerable Web Application - Mozilla Firefox
XVWA - Xtreme Vulnerable W...
192.168.0.29/xvwa/setup/?action=do
XVWA
Login
About
Setup
Home
Instructions
Setup / Reset
Attacks
SQL Injection
SQL Injection (Blind)
OS Command Injection
XPATH Injection
Unrestricted File Upload
XSS - Reflected
XSS - Stored
XSS - DOM Based
SSRF / XSPA
Xtreme Vulnerable Web Application (XVWA) - Setup
Submit / Reset
xvwa
Login
Connected to database sucessfully.
Table comments sucessfully.
Table products created sucessfully.
Table users created sucessfully.
Setup finished
XVWA - Xtreme Vulnerable Web Application
samurai : bash – Konsole

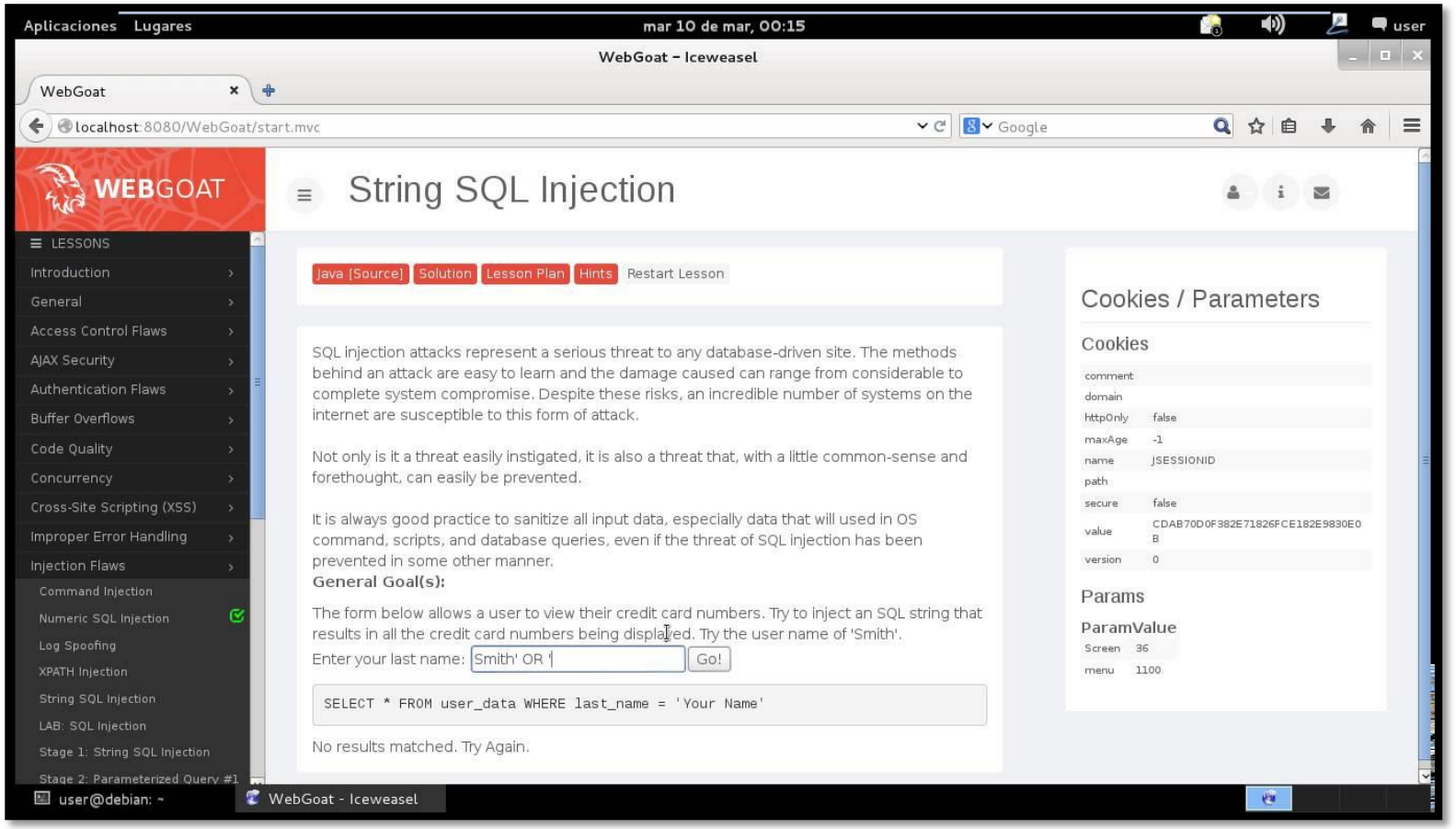
DVWA


DAMN VULNERABLE
127.0.0.1/DVWA/INDEX.PHP
Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored
DVWA Security
PHP Info
About
Logout
Welcome to Damn Vulnerable Web App!
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.
WARNING!
Damn Vulnerable Web App is damn vulnerable! Do not upload it to your hosting provider's public html folder or any internet facing web server as it will be compromised. We recommend downloading and installing XAMPP onto a local machine inside your LAN which is used solely for testing.
Disclaimer
We do not take responsibility for the way in which any one uses this application. We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the person/s who uploaded and installed it.
General Instructions
The help button allows you to view hits/tips for each vulnerability and for each security level on their respective page.
You have logged in as 'admin'
Username: admin
Security Level: high
PHPIDS: disabled

Mutillidae


OWASP Mutillidae II: Web Pwn in Mass Production
Version: 2.6.10
Security Level: 0 (Hosed)
Hints: Disabled (0 - I try harder)
Not Logged In
Home | Login/Register | Toggle Hints | Toggle Security | Reset DB | View Log | View Captured Data | Hide Popup Hints | Enforce SSL
Status Update
OWASP 2013
OWASP 2010
OWASP 2007
Web Services
HTML 5
Others
Documentation
Resources
Mutillidae: Deliberately Vulnerable Web Pen-Testing Application
Like Mutillidae? Check out how to help
What Should I Do?
Video Tutorials
Help Me!
Listing of vulnerabilities
Bug Tracker
Bug Report Email Address
What's New? Click Here
Release Announcements
PHP MyAdmin Console
Feature Requests
Getting Started: Project Whitepaper
Release Announcements

WebGoat


Aplicaciones Lugares
mar 10 de mar, 00:15
user
WebGoat – Iceweasel
WebGoat
localhost:8080/WebGoat/start.mvc
Google
WEBGOAT
String SQL Injection
LESSONS
Introduction
General
Access Control Flaws
AJAX Security
Authentication Flaws
Buffer Overflows
Code Quality
Concurrency
Cross-Site Scripting (XSS)
Improper Error Handling
Injection Flaws
Command Injection
Numeric SQL Injection
Log Spoofing
XPATH Injection
String SQL Injection
LAB: SQL Injection
Stage 1: String SQL Injection
Stage 2: Parameterized Query #1
Java [Source] Solution Lesson Plan Hints Restart Lesson
SQL injection attacks represent a serious threat to any database-driven site. The methods behind an attack are easy to learn and the damage caused can range from considerable to complete system compromise. Despite these risks, an incredible number of systems on the internet are susceptible to this form of attack.
Not only is it a threat easily instigated, it is also a threat that, with a little common-sense and forethought, can easily be prevented.
It is always good practice to sanitize all input data, especially data that will used in OS command, scripts, and database queries, even if the threat of SQL injection has been prevented in some other manner.
General Goal(s):
The form below allows a user to view their credit card numbers. Try to inject an SQL string that results in all the credit card numbers being displayed. Try the user name of 'Smith'.
Enter your last name: Smith' OR '
Go!
SELECT * FROM user_data WHERE last_name = 'Your Name'
No results matched. Try Again.
Cookies / Parameters
Cookies
comment
domain
httpOnly false
maxAge -1
name JSESSIONID
path
secure false
value CDAB70D0F382E71826FCE182E9830E0B
version 0
Params
ParamValue
Screen 36
menu 1100
user@debian: ~
WebGoat - Iceweasel

Arachni Web Scanner

OWASP Web Testing Framework

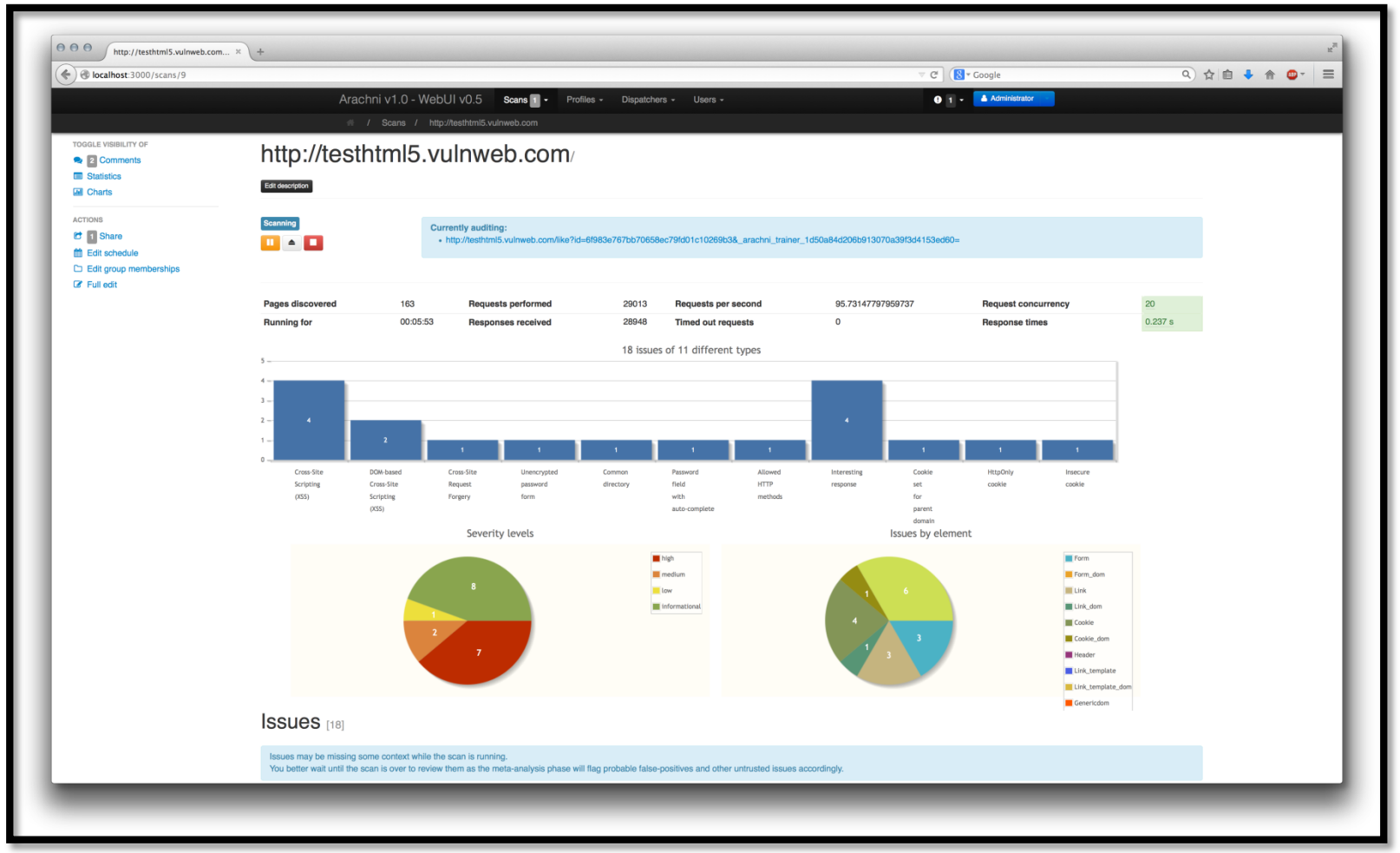
Mantra Browser Tool Suite

OWASP ZAP

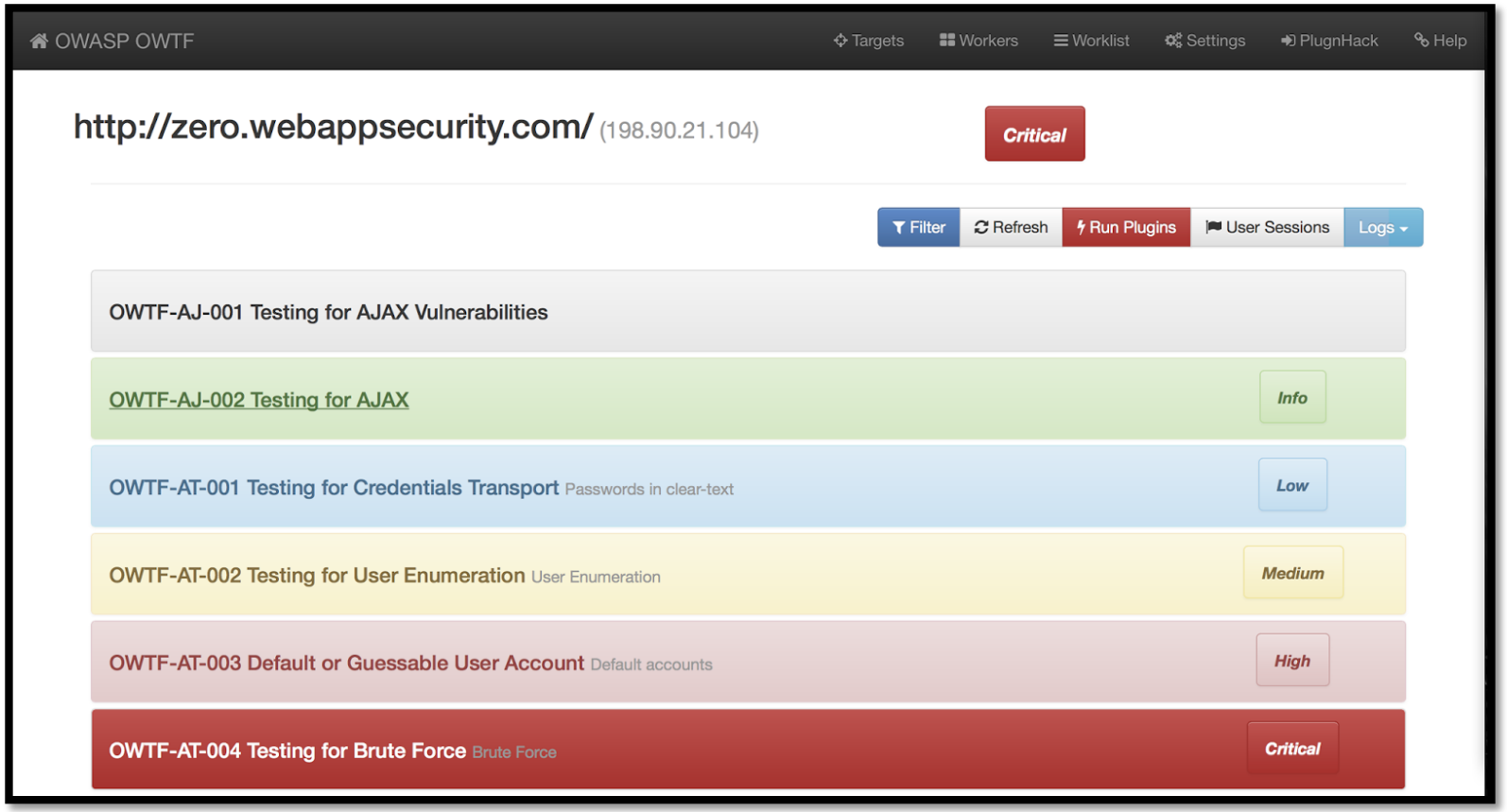
The Tool Suite

What are the web application tools that you need?

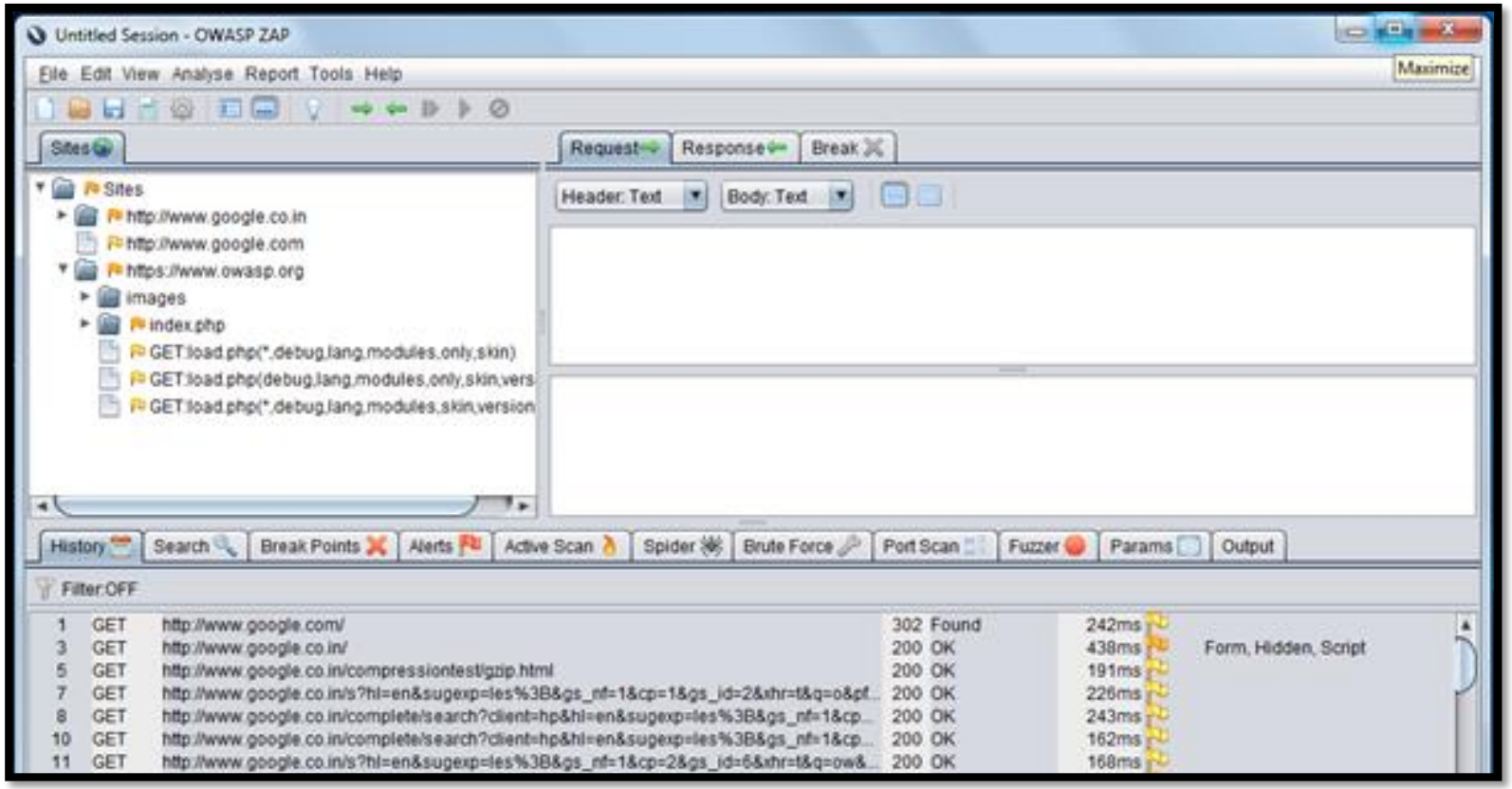
Arachni


http://testhtml5.vulnweb.com...
localhost:3000/scans/9
Arachni v1.0 - WebUI v0.5
Scans
Profiles
Dispatchers
Users
Administrator
Scans / http://testhtml5.vulnweb.com
TOGGLE VISIBILITY OF
Comments
Statistics
Charts
ACTIONS
Share
Edit schedule
Edit group memberships
Full edit
http://testhtml5.vulnweb.com/
Edit description
Scanning
Currently auditing:
http://testhtml5.vulnweb.com/like?id=6f983e767bb70658ec79fd01c10269b3&_arachni_trainer_1d50a84d206b913070a39f3d4153ed60=
Pages discovered 163
Requests performed 29013
Requests per second 95.73147797959737
Request concurrency 20
Running for 00:05:53
Responses received 28948
Timed out requests 0
Response times 0.237 s
18 issues of 11 different types
Cross-Site Scripting (XSS)
DOM-based Cross-Site Scripting (XSS)
Cross-Site Request Forgery
Unencrypted password form
Common directory
Password field with auto-complete
Allowed HTTP methods
Interesting response
Cookie set for parent domain
HttpOnly cookie
Insecure cookie
Severity levels
high
medium
low
Informational
Issues by element
Form
Form_dom
Link
Link_dom
Cookie
Cookie_dom
Header
Link_template
Link_template_dom
Genericdom
Issues [18]
Issues may be missing some context while the scan is running.
You better wait until the scan is over to review them as the meta-analysis phase will flag probable false-positives and other untrusted issues accordingly.

OWASP WTF


OWASP OWTF
Targets
Workers
Worklist
Settings
PlugnHack
Help
http://zero.webappsecurity.com/ (198.90.21.104)
Critical
Filter
Refresh
Run Plugins
User Sessions
Logs
OWTF-AJ-001 Testing for AJAX Vulnerabilities
OWTF-AJ-002 Testing for AJAX
Info
OWTF-AT-001 Testing for Credentials Transport Passwords in clear-text
Low
OWTF-AT-002 Testing for User Enumeration User Enumeration
Medium
OWTF-AT-003 Default or Guessable User Account Default accounts
High
OWTF-AT-004 Testing for Brute Force Brute Force
Critical

OWASP ZAP


Untitled Session - OWASP ZAP
File Edit View Analyse Report Tools Help
Maximize
Sites
Request
Response
Break
Header: Text
Body: Text
Sites
http://www.google.co.in
http://www.google.com
https://www.owasp.org
images
index.php
GET:load.php(*,debug,lang,modules.only,skin)
GET:load.php(debug,lang,modules.only,skin,vers
GET:load.php(*,debug,lang,modules,skin,version
History
Search
Break Points
Alerts
Active Scan
Spider
Brute Force
Port Scan
Fuzzer
Params
Output
Filter:OFF
1 GET http://www.google.com/ 302 Found 242ms
3 GET http://www.google.co.in/ 200 OK 438ms Form, Hidden, Script
5 GET http://www.google.co.in/compressiontest/gzip.html 200 OK 191ms
7 GET http://www.google.co.in/s?hl=en&sugexp=les%3B&gs_rf=1&cp=1&gs_id=2&xhr=t&q=o&pf... 200 OK 226ms
8 GET http://www.google.co.in/complete/search?client=hp&hl=en&sugexp=les%3B&gs_rf=1&cp... 200 OK 243ms
10 GET http://www.google.co.in/complete/search?client=hp&hl=en&sugexp=les%3B&gs_rf=1&cp... 200 OK 162ms
11 GET http://www.google.co.in/s?hl=en&sugexp=les%3B&gs_rf=1&cp=2&gs_id=6&xhr=t&q=ow&... 200 OK 168ms

TIME FOR

Q&A

Building Your Test Lab

Homework

- Build your own virtual environment lab
- Install XVWA and DVWA
- Install the following tools:
  - OWASP ZAP
  - OWTF
  - Arachni
  - Mantra
- Run the tools against your lab (XVWA and DVWA) and see the results
- Check out *https://pentesterlab.com/* for more interesting & advance lessons

WEB APPLICATION TESTING

Building Your Test Lab

www.pandoralabs.net

PANDORA
SECURITY LABS

Expert Advice. Experience Advantage.
Proactive Security Solutions Through Cutting-Edge Research.

Web App Testing: Building Your Own Lab

By @isaacsabas