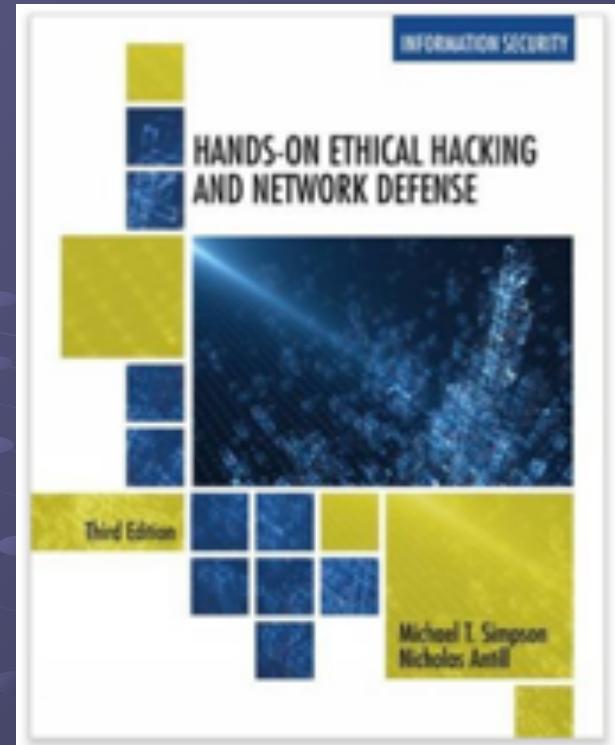


# Hands-On Ethical Hacking and Network Defense

## 3rd Edition

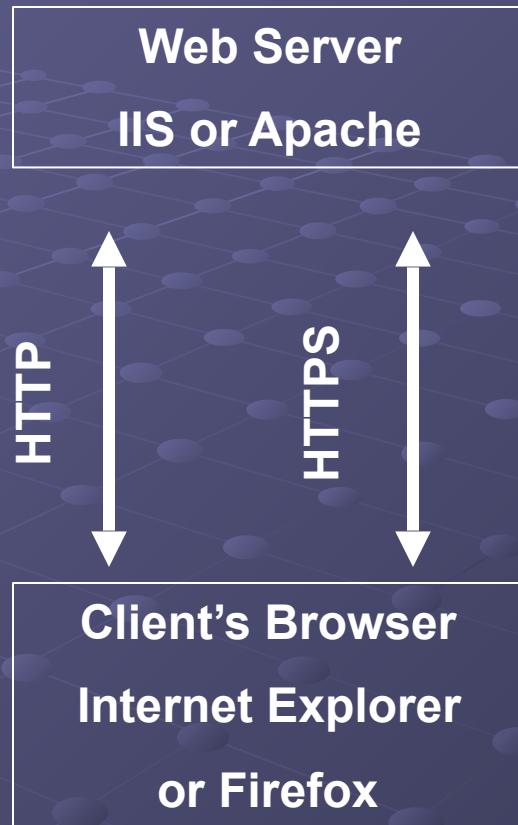
### *Chapter 10* *Hacking Web Servers*

Revised 11-8-17



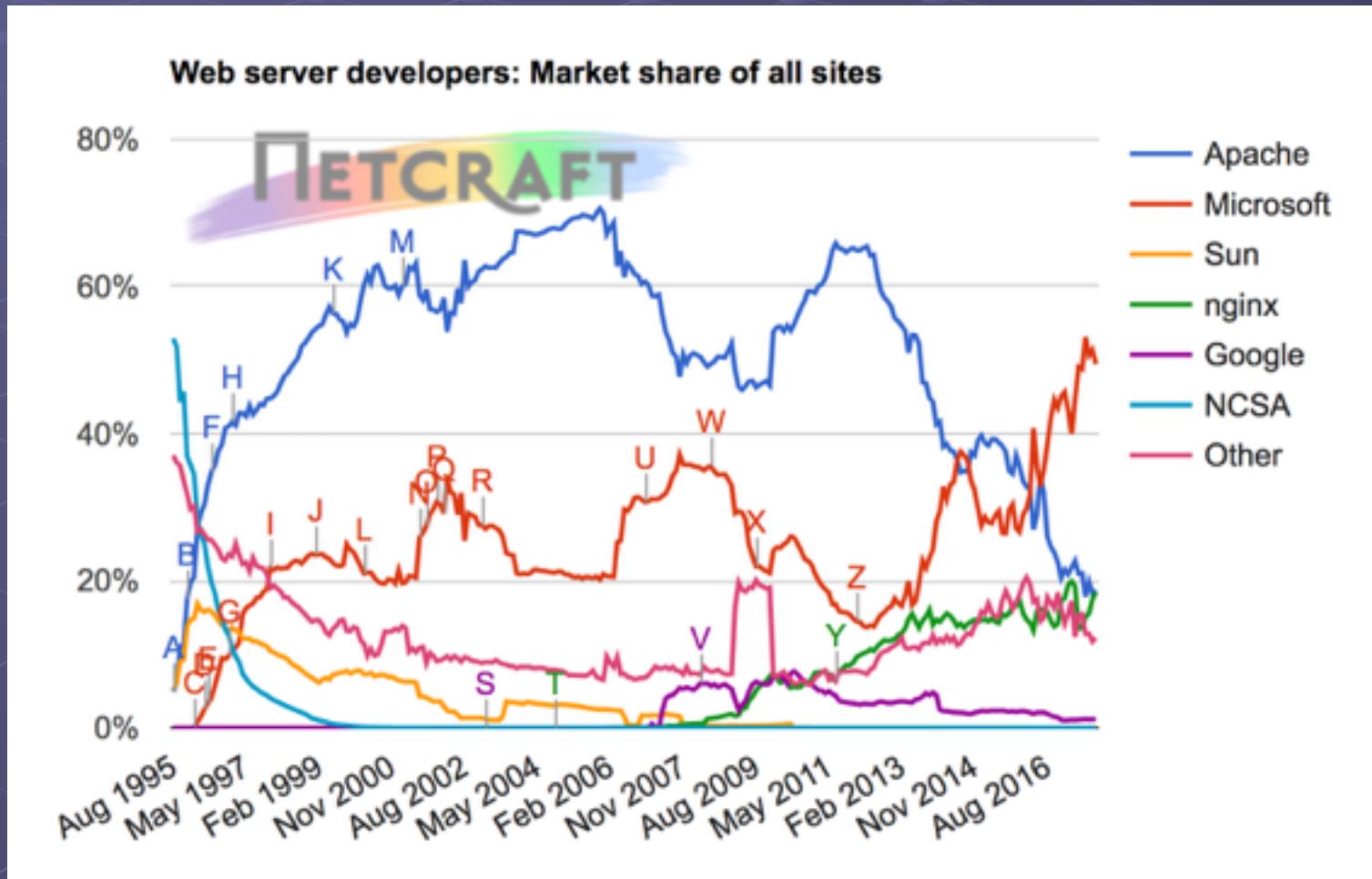
# Objectives

- Describe Web applications
- Explain Web application vulnerabilities
- Describe the tools used to attack Web servers



# Web Servers

- The three main Web servers are nginx and Apache (Open source) & IIS (Microsoft)
  - Link Ch 10c (image from 11-8-17)



# Understanding Web Applications

- It is nearly impossible to write a program without bugs
  - Some bugs create security vulnerabilities
  - Web applications also have bugs
    - Web applications have a larger user base than standalone applications
    - Bugs are a bigger problem for Web applications

# Web Application Components

- Static Web pages
  - Created using HTML
- Dynamic Web pages
  - Need special components
    - <form> tags
    - Common Gateway Interface (CGI) scripts
    - Active Server Pages (ASP)
    - PHP
    - ColdFusion
    - Scripting languages like JavaScript
    - ODBC (Open Database connector)

# Web Forms

- Use the <form> element or tag in an HTML document
  - Allows customer to submit information to the Web server
  - Web servers process information from a Web form by using a Web application
  - Easy way for attackers to intercept data that users submit to a Web server

Secure | https://fog.ccst.edu/~sbowne/feedback.html

# Anonymous Feedback Form

Which Class are You Taking?

None

What would you like to tell future students about this class?

To stop spam, please type the name of this animal below, using only lowercase letters:

SUBMIT   RESET   [Read Comments](#)



8

# HTML Form

```
<form action="/cgi-bin/cgiwrap/sbowne/feedback5.pl" method="post">

<font size=4>Which Class are You Taking? </font>
<select name="class">

<option value="none" SELECTED>None</option>

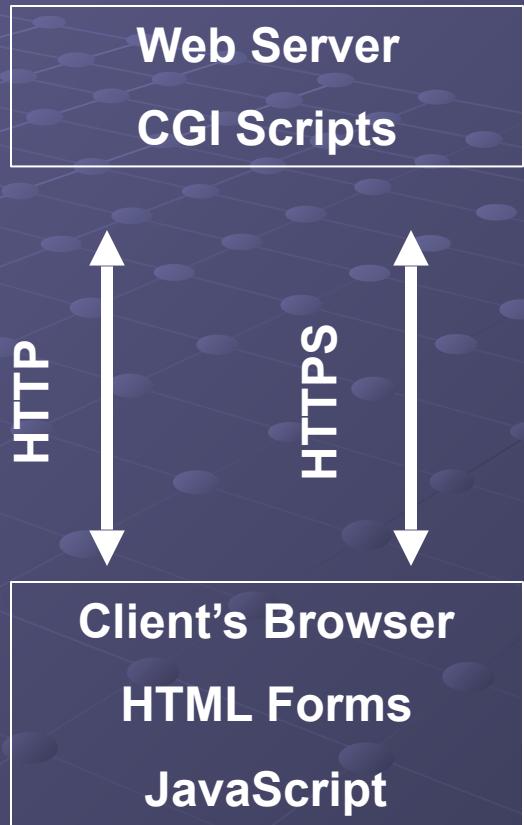
<option value="Fall 2010 - CNIT 106: Intro to Networking">
    Fall 2010 - CNIT 106: Intro to Networking</option>
<option value="Fall 2010 - CNIT 121: Computer Forensics">
    Fall 2010 - CNIT 121: Computer Forensics</option>

</select>

<p><font size=4>What would you like to tell future students about this class?</font><br>
<textarea name="comments" rows="12" cols="71"></textarea><br>

<p><font size=4>To stop spam, please type the name of this animal below, using only lowercase letters:</font><br>
<textarea name="animal" rows="1" cols="19"></textarea><br>
<input type="submit" value="SUBMIT">
<input type="reset" value="RESET">
&nbsp;&nbsp;&nbsp;
<a href="http://fog.ccsf.edu/~sbowne/comments2.html"><font size=4>Read Comments</font></a>
</p>

</form>
```



# Common Gateway Interface (CGI)

- Handles moving data from a Web server to a Web browser
- The majority of dynamic Web pages are created with CGI and scripting languages
- Describes how a Web server passes data to a Web browser
  - Relies on Perl or another scripting language to create dynamic Web pages

# CGI Languages

- CGI programs can be written in different programming and scripting languages
  - C or C++
  - Perl
  - Unix shell scripting
  - Visual Basic
  - FORTRAN

# Common Gateway Interface (CGI) (continued)

- CGI example
  - Written in Perl
  - Hello.pl
  - Should be placed in the *cgi-bin* directory on the Web server

```
#!/usr/bin/perl
print "Content-type: text/html\n\n";
print "Hello Security Testers!";
```



# Anonymous Feedback Form

Which Class are You Taking?

None

What would you like to tell future students about this class?

To stop spam, please type the name of this animal below, using only lowercase letters:

SUBMIT

RESET

[Read Comments](#)



```
#!/usr/bin/perl

use CGI qw(-debug :standard :html3);

$dtd =
"-//W3C//DTD XHTML 1.0 Transitional//EN\"  

"\n\"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd\";

$class = param("class");
$comment = param("comments");
$animal = param("animal");

# Remove spam
if ( ($comment =~ m/URL=/) || ($animal ne 'horse') ){
    print header();

    print( start_html( { dtd => $dtd,
                          title => "Error!" } ) );

    print ( h2({-ALIGN=>"center"}), "Your comments were not recorded, because they appeared to be spam." );
    print end_html();
}
else{
    print header();

    print( start_html( { dtd => $dtd,
                          title => "Thanks!" } ) );

    $resultsFile = param("resultsFile");

    # print("Results File: $resultsFile\n");

    print ( h2({-ALIGN=>"center"}), "Your comments have been recorded." );
    print ( h2({-ALIGN=>"center"}), "Thank You!" );
    print( "<h2 align=\"center\"><a href=\"http://fog.ccsf.org/~sbowne/comments2.html\">" );
    print( "Click here to read the comments</a></h2>" );
}
```

# Active Server Pages (ASP)

- Microsoft's server-side script engine
  - HTML pages are static—always the same
  - ASP creates HTML pages as needed. They are not static
- ASP uses scripting languages such as JScript or VBScript
- Not all Web servers support ASP
  - IIS supports ASP
  - Apache doesn't support ASP as well

# Active Server Pages (ASP)

- You can't see the source of an ASP page from a browser
- This makes it harder to hack into, although not impossible
- ASP examples at links  
Ch 10d, e, f



The screenshot shows a web browser window with the URL <http://home.wlv.ac.uk/~cm1901/clock.asp>. The title bar reads "My First ASP Example". Below the title, there is a digital-style clock display showing "19:27:21".

```
<%@language = vbscript %>
<% option explicit %>
<html><head><title>ASP Example</head>
<body><table border=6><tr><td bgcolor=black>
<font face=verdana color=green size=3>
<% = time() %>
</font></td></tr></table></body>
</html>
```

# Apache Web Server

- Apache is the most popular Web Server program
- Advantages
  - Stable and reliable
  - Works on just about any \*NIX and Windows platform
  - It is free and open source
    - See links Ch 10g, 10h

# Using Scripting Languages

- Dynamic Web pages can be developed using scripting languages
  - VBScript
  - JavaScript
  - PHP

# PHP: Hypertext Processor (PHP)

- Enables Web developers to create dynamic Web pages
  - Similar to ASP
- Open-source server-side scripting language
  - Can be embedded in an HTML Web page using PHP tags <?php and ?>
  - Users cannot see PHP code in their Web browser
  - Used primarily on UNIX systems
    - Also supported on Macintosh and Microsoft platforms

# PHP Example

```
<html><head><title>Example</title></head>
<body>
<?php
echo 'Hello, World!';
?>
</body></html>
```

- See links Ch 10k, 10l
- PHP has known vulnerabilities
  - See links Ch 10m, 10n
- PHP is often used with MySQL Databases

# ColdFusion

- Server-side scripting language used to develop dynamic Web pages
- Created by the Allaire Corporation
  - Purchased by Macromedia, now owned by Adobe -- Expensive
- Uses its own proprietary tags written in ColdFusion Markup Language (CFML)
- CFML Web applications can contain other technologies, such as HTML or JavaScript

# ColdFusion Example

```
<html><head><title>Ex</title></head>
<body>
<CFLOCATION URL="www.isecom.org/cf/
    index.htm" ADDTOKEN="NO">
</body>
</html>
```

- See links Ch 10o

# ColdFusion Vulnerabilities

## Macromedia ColdFusion Vulnerabilities :

- [Green] 14.02.2007 : Adobe ColdFusion MX Default Error Page Client-Side Cross Site Scripting Vulnerability
- [Yellow] 11.12.2006 : Adobe Macromedia ColdFusion Information Disclosure and Cross Site Scripting Issues
- [Yellow] 11.10.2006 : Adobe Macromedia ColdFusion Verity Library Privilege Escalation Vulnerabilities
- [Green] 12.09.2006 : Adobe Macromedia ColdFusion Error Page Cross Site Scripting Vulnerability
- [Yellow] 12.09.2006 : Adobe Macromedia ColdFusion Denial of Service and Security Bypass Vulnerabilities
- [Green] 09.08.2006 : Adobe Macromedia ColdFusion MX AdminAPI Local Authentication Bypass Vulnerability
- [Yellow] 16.12.2005 : Macromedia ColdFusion Multiple Security Bypass Vulnerabilities
- [Green] 15.07.2005 : Macromedia JRun Internal Authentication Token Vulnerability
- [Green] 10.05.2005 : Macromedia ColdFusion MX Error Page Cross Site Scripting Issue
- [Yellow] 08.04.2005 : Macromedia ColdFusion MX Updater File Disclosure Vulnerability

- See links Ch 10p, 10q

# VBScript

- Visual Basic Script is a scripting language developed by Microsoft
- You can insert VBScript commands into a static HTML page to make it dynamic
  - Provides the power of a full programming language
  - Executed by the client's browser

# VBScript Example

```
<html><body>
<script type="text/vbscript">
document.write("<h1>Hello!</h1>")
document.write("Date Activated: " &
date())
</script>
</body></html>
```

- See link Ch 10r – works in IE, but not in Firefox
- Firefox does not support VBScript (link Ch 10s)

# VBScript vulnerabilities

- See links Ch 10t, 10u

## Microsoft Security Bulletin MS02-009

Incorrect VBScript Handling in IE can Allow Web Pages to Read Local Files

**Originally posted:** February 21, 2002

**Updated:** May 09, 2003

# JavaScript

- Popular scripting language
- JavaScript also has the power of a programming language
  - Branching
  - Looping
  - Testing

# JavaScript Example

```
<html><head>
<script type="text/javascript">
function chastise_user() {
alert("So, you like breaking rules?")
document.getElementById("cmdButton") .focus (
)
</script></head>
<body><h3>Don't click the button!</h3>
<form>
<input type="button" value="Don't Click!"
      name="cmdButton"
      onClick="chastise_user()" />
</form></body></html>
```

- See link Ch 10v – works in IE and Firefox

# JavaScript Vulnerabilities

JavaScript vulnerabilities surface in multiple browsers

by [John McCormick](#) | More from John McCormick | 6/12/06

**Tags:** [Web browsers](#) | [Security threats](#) | [Internet Explorer \(IE\)](#) | [Patches](#)



See link Ch 10w

# Popularity of Programming Languages

See link Ch 10zs

Programming Language	Ratings
Java	12.431%
C	8.374%
C++	5.007%
C#	3.858%
Python	3.803%
JavaScript	3.010%
PHP	2.790%
Visual Basic .NET	2.735%
Assembly language	2.374%
Ruby	2.324%
Delphi/Object Pascal	2.180%
Perl	1.963%
MATLAB	1.880%
Scratch	1.819%
R	1.684%
Swift	1.668%
Objective-C	1.513%
Visual Basic	1.420%

The Kahoot! logo is displayed on a teal background. The word "Kahoot!" is written in a bold, white, sans-serif font. The exclamation mark is smaller than the other letters and has a white outline.

Kahoot!

1

Web Server  
*Apache or IIS*  
*HTML Forms*  
*CGI Scripts*

ODBC or  
OLE DB  
Or ADO

Database  
*SQL Server or*  
*Oracle or*  
*MySQL*

HTTP or HTTPS

Sign in to Gmail with your  
**Google Account**

Username:

Password:

Client's Browser

```
haldaemon:!:13548:0:99999:7:::  
hplip:!:13548:0:99999:7:::  
gdm:!:13548:0:99999:7:::  
yourname:$1$31N/PNc1$7IRVdaKE2vQ5Me/rYDLx70:13548:0:99999:7:::  
mysql:!:13548:0:99999:7:::
```

# Connecting to Databases

- Web pages can display information stored on databases
- There are several technologies used to connect databases with Web applications
  - Technology depends on the OS used
    - ODBC
    - OLE DB
    - ADO
  - Theory is the same

# Open Database Connectivity (ODBC)

- Standard database access method developed by the SQL Access Group
- ODBC interface allows an application to access
  - Data stored in a database management system (DBMS)
  - Can use Oracle, SQL, or any DBMS that understands and can issue ODBC commands
- Interoperability among back-end DBMS is a key feature of the ODBC interface

# Open Database Connectivity (ODBC) (continued)

- ODBC defines
  - Standardized representation of data types
  - A library of ODBC functions
  - Standard methods of connecting to and logging on to a DBMS

# OLE DB and ADO

- Object Linking and Embedding Database (OLE DB) and
- ActiveX Data Objects (ADO)
  - These two more modern, complex technologies replace ODBC and make up "Microsoft's Universal Data Access"
  - See link Ch 10x

# Understanding Web Application Vulnerabilities

- Many platforms and programming languages can be used to design a Web site
- Application security is as important as network security

# Attackers controlling a Web server can

- Deface the Web site
- Destroy or steal company's data
- Gain control of user accounts
- Perform secondary attacks from the Web site
- Gain root access to other applications or servers

# Open Web Application Security Project (OWASP)

- Open, not-for-profit organization dedicated to finding and fighting vulnerabilities in Web applications
- Publishes the Ten Most Critical Web Application Security Vulnerabilities

# OWASP Top 10 Application Security Risks – 2017

## A1:2017 Injection

Injection flaws, such as SQL, OS, and LDAP injection occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.

## A2:2017 Broken Authentication

Application functions related to authentication and session management are often implemented incorrectly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities (temporarily or permanently).

## A3:2017 Sensitive Data Exposure

Many web applications and APIs do not properly protect sensitive data, such as financial, healthcare, and PII. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data deserves extra protection such as encryption at rest or in transit, as well as special precautions when exchanged with the browser.

## A4:2017 XML External Entity (XXE)

Many older or poorly configured XML processors evaluate external entity references within XML documents. External entities can be used to disclose internal files using the file URI handler, internal SMB file shares on unpatched Windows servers, internal port scanning, remote code execution, and denial of service attacks, such as the Billion Laughs attack.

## A5:2017 Broken Access Control

Restrictions on what authenticated users are allowed to do are not properly enforced. Attackers can exploit these flaws to access unauthorized functionality and/or data, such as access other users' accounts, view sensitive files, modify other users' data, change access rights, etc.

## A6:2017 Security Misconfiguration

Security misconfiguration is the most common issue in the data, which is due in part to manual or ad hoc configuration (or not configuring at all), insecure default configurations, open S3 buckets, misconfigured HTTP headers, error messages containing sensitive information, not patching or upgrading systems, frameworks, dependencies, and components in a timely fashion (or at all).

## A7:2017 Cross-Site Scripting (XSS)

XSS flaws occur whenever an application includes untrusted data in a new web page without proper validation or escaping, or updates an existing web page with user supplied data using a browser API that can create JavaScript. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.

## A8:2017 Insecure Deserialization

Insecure deserialization flaws occur when an application receives hostile serialized objects. Insecure deserialization leads to remote code execution. Even if deserialization flaws do not result in remote code execution, serialized objects can be replayed, tampered or deleted to spoof users, conduct injection attacks, and elevate privileges.

## A9:2017 Using Components with Known Vulnerabilities

Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications and APIs using components with known vulnerabilities may undermine application defenses and enable various attacks and impacts.

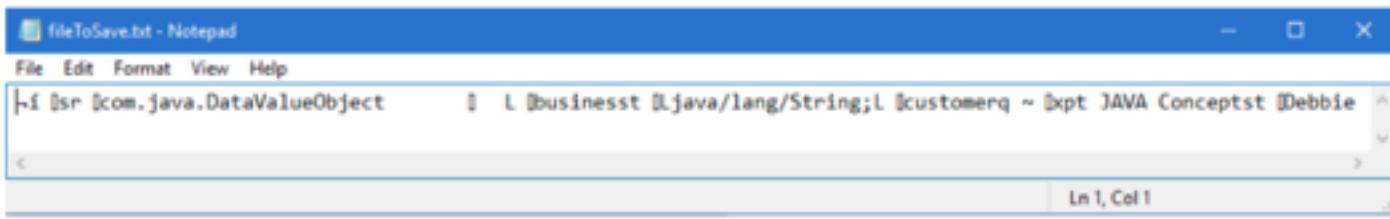
## A10:2017 Insufficient Logging & Monitoring

Insufficient logging and monitoring, coupled with missing or ineffective integration with incident response allows attackers to further attack systems, maintain persistence, pivot to more systems, and tamper, extract or destroy data. Most breach studies show time to detect a breach is over 200 days, typically detected by external parties rather than internal processes or monitoring.

# Java Serialization

```
public static void main(String args[]) {  
    DataValueObject dataValueObject = new DataValueObject();  
    dataValueObject.setCustomer("Debbie");  
    dataValueObject.setBusiness("JAVA Concepts");  
    dataValueObject.setContractID("ZZZZZZ");  
    dataValueObject.setPassKeys("!@wer#$");  
  
    try {  
        SerializationDemo.serialization("fileToSave.txt", dataValueObject);  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

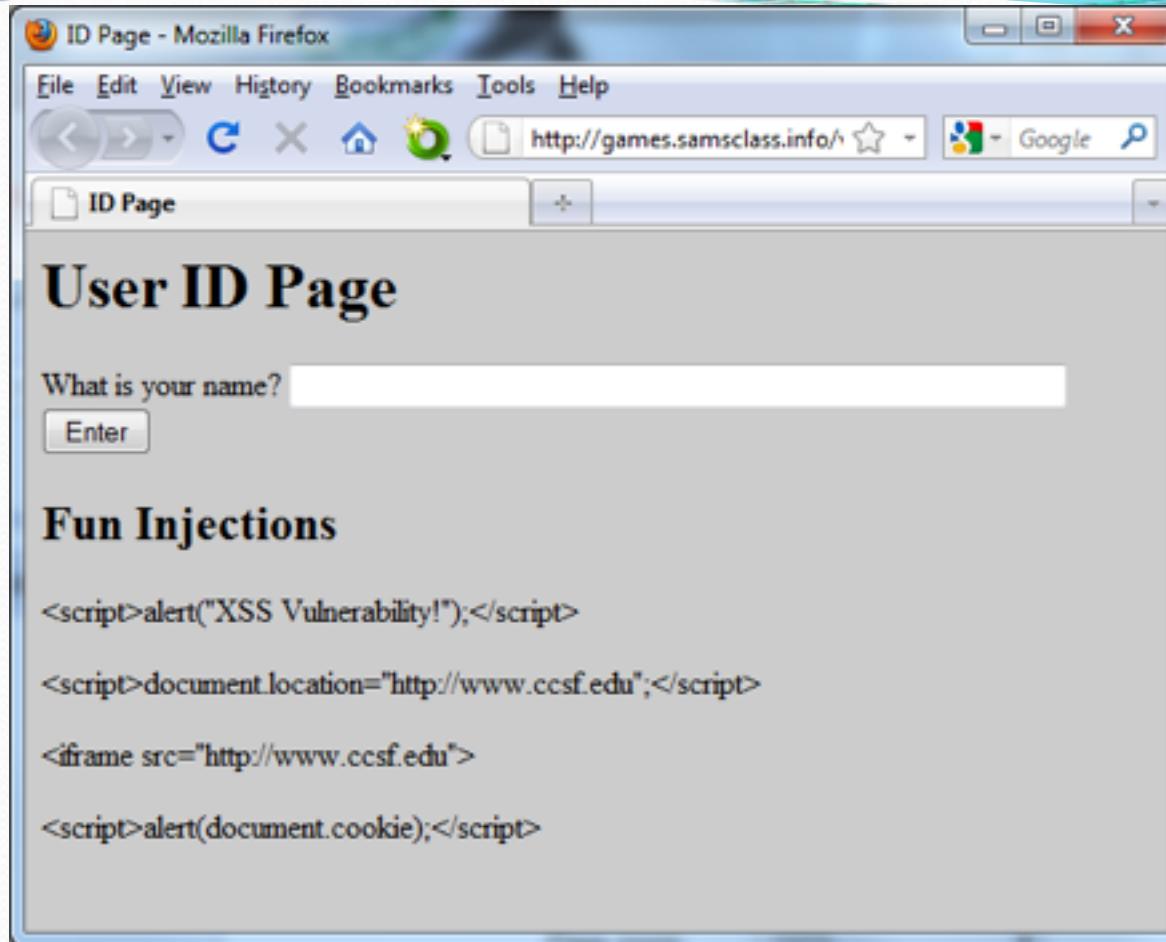
fileToSave.txt FILE content after serialization as byte stream



- Link Ch 10zt

# Cross-Site Scripting (XSS)

- One client posts active content, with <script> tags or other programming content
- When another client reads the messages, the scripts are executed in his or her browser
- One user attacks another user, using the vulnerable Web application as a weapon



- <script>alert("XSS vulnerability!")</script>
- <script>alert(document.cookie)</script>
- <script>window.location="http://www.ccsf.edu"</script>

# XSS Scripting Effects

- Steal another user's authentication cookie
  - Hijack session
- Harvest stored passwords from the target's browser
- Take over machine through browser vulnerability
- Redirect Webpage
- Many, many other evil things...

# Assessing Web Applications

- Issues to consider
  - Dynamic Web pages
  - Connection to a backend database server
  - User authentication
  - What platform was used?

# Does the Web Application Use Dynamic Web Pages?

- Static Web pages do not create a secure environment
- IIS attack example: Directory Traversal
  - Adding ..\ to a URL refers to a directory above the Web page directory
  - Early versions of IIS filtered out \, but not %c1%9c, which is a Unicode version of the same character
  - See link Ch 10 zh

# Connection to a Backend Database Server

- Security testers should check for the possibility of SQL injection being used to attack the system
- SQL injection involves the attacker supplying SQL commands on a Web application field

# SQL Injection Example

HTML form collects *name* and *pw*  
SQL then uses those fields:

```
SELECT * FROM customer
```

```
WHERE username = 'name' AND password = 'pw'
```

If a hacker enters a name of  
' OR 1=1 --

The SQL becomes:

```
SELECT * FROM customer
```

```
WHERE username = '' OR 1=1 --' AND password =  
'pw'
```

Which is always true, and returns all the records

① <https://attack.samsclass.info/sqlol-raw/123-p19.htm>

## Project 19 for CNIT 123: SQLol (20 pts. + 10 pts. extra)

### Introduction to SQL Injection: Hands-On

# Connection to a Backend Database Server

- Basic testing should look for
  - Whether you can enter text with punctuation marks
  - Whether you can enter a single quotation mark followed by any SQL keywords
  - Whether you can get any sort of database error when attempting to inject SQL

# User Authentication

- Many Web applications require another server to authenticate users
- Examine how information is passed between the two servers
  - Encrypted channels
- Verify that logon and password information is stored on secure places
- Authentication servers introduce a second target

# What Platform Was Used?

- Popular platforms include:
  - IIS with ASP and SQL Server (Microsoft)
  - Linux, Apache, MySQL, and PHP (LAMP)
- Footprinting is used to find out the platform
  - The more you know about a system the easier it is to gather information about its vulnerabilities

# SQLI on Pastebin

The screenshot shows a web browser window with the URL <https://pastebin.com/zdvQVRen>. The page is titled "PASTEBIN". The content of the paste is as follows:

```
1. 11/7/2017 1:08:18 PM
2. SQLi Dumper v.8.2
3. http://www.lila-weiss.de/index.php?include=spieldetails&spiel=999999.9 union all select 1,2,3,4,5,[t],7,8
4.
5. password      email
6. el [REDACTED] bartling@tobunga.de
7. wir [REDACTED] bartling@mail.uni-marburg.de
8. tax [REDACTED] kalla.wefel@t-online.de
9. tecumseh@meppen.de
10. nt [REDACTED] feschmann@debitel.net
11. kalla.wefel@t-online.de
12. kalla.wefel@t-online.de
13. au [REDACTED] bartling@mail.uni-marburg.de
```



## Sqli sites



ANONGSA



OCT 11TH, 2017 (EDITED)



52



NEVER

```
15. http://forum.bonzaisoftware.com/viewthread.php?tid=10
16. http://www.starushka.net/users.php?id=2
17. http://soundborb.com/users.php?id=15
18. http://www.sheratonresorts-ap.com/resort.php?id=18
19. http://www.lumensfactory.com/online_shop.php?cid=10
20. http://www.vortexcdi.com/product_generic.php?cid=1
21. http://www.aceronline.net/support.php?cid=52
22. http://www.allanschore.com/booksDetail.php?bookID=3
23. http://c2-europe.eu/news-full.php?id=999999.9 union all select 1,2,3,[t],5,6,7,8,9,10,11,12,13,14
24. http://www.lasamwesthill.co.uk/add_to_cart.php?id=4999999.9 union all select [t],2,3,4,5,6,7,8,9
25. http://www.dublinfoodchain.ie/producer-directory/prepared-foods/company.php?id=73' and [t] and '1'='1
26. http://www.royaljapanesemotors.com/details.php?id=999999.9 union all select 1,2,3,4,5,
[t],7,8,9,10,11,12,13,14,15,16,17,18,19,20
27. http://www.speedofsound.dk/page.php?id=999999.9 union all select 1,[t],3,4,5,6,7
28. http://www.toploop.com.tw/productinfo.php?id=999999.9 union all select 1,2,3,4,5,6,7,8,9,[t],11,12
29. http://www.romanianwriters.ro/s.php?id=1999999.9' union all select 1,[t],3 and '0'='0
30. http://bowievetsupply.com/listing.php?cid=999999.9 union all select [t],2
31. http://www.amspecinc.com/products/harnesses_and_belts/search_detail.php?id=999999.9 union all select 1,2,3,
[t],5,6,7,8,9,10
```

[www.vortexcdi.com/product\\_generic.php?cid=1%27](http://www.vortexcdi.com/product_generic.php?cid=1%27)

The screenshot shows a web browser window displaying the Vortex Ignitions website. The URL in the address bar is [www.vortexcdi.com/product\\_generic.php?cid=1%27](http://www.vortexcdi.com/product_generic.php?cid=1%27). The page features the Vortex Ignitions logo and slogan "When timing is Everything!" in yellow script. A navigation menu includes links for Home, Products, Technical Updates, Research, Race Teams, Sales Locations, and Distributor Login. Below the menu is a large image of a dirt bike in motion, kicking up dust, with a blurred crowd of spectators in the background. At the bottom of the page, a white banner contains the error message: "No go c: You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'l' at line 1".

← → ⌛ ⌂ 🔍 Secure | https://pastebin.com/mPaYKDTu

 PASTEBIN + new paste trends search...

 Burp Suite Workshop V2

JOEMCCRAY PRO 📧 SEP 7TH, 2017 (EDITED) 531 NEVER

← → ⌛ ⌂ 🔍 45.63.104.73/showfile.php?filename=about.txt

 ACME LAPTOP  Register

Categories Home Buy Career

- > Acer
- > Compaq
- > Dell
- > Gateway
- > Hewlett
- > Ibm

**about**

AcmeLaptop is a unique laptop store which believes in giving best different configuration at low and affordable prize.

Customer satisfaction is our business model.

← → C ⌂ ① 45.63.104.73/showfile.php?filename=/etc/passwd



ACMELAPTOP



Register

Categories

- > Acer
- > Compaq
- > Dell
- > Gateway
- > Hewlett
- > Ibm
- > Sony
- > Toshiba

Home Buy Career

/etc/pa

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/sbin/nologin
systemd-bus-proxy:x:999:997:systemd Bus Proxy:/sbin/nologin
systemd-network:x:192:192:systemd Network Management:/sbin/nologin
dbus:x:81:81:System message bus:/sbin/nologin
```

## 4. Web Application Security Testing

### 4.2 Information Gathering

- 4.2.1 Conduct Search Engine Discovery and Reconnaissance for Information Leakage (OTG-INFO-001)
- 4.2.2 Fingerprint Web Server (OTG-INFO-002)
- 4.2.3 Review Webserver Metafiles for Information Leakage (OTG-INFO-003)
- 4.2.4 Enumerate Applications on Webserver (OTG-INFO-004)
- 4.2.5 Review Webpage Comments and Metadata for Information Leakage (OTG-INFO-005)
- 4.2.6 Identify application entry points (OTG-INFO-006)
- 4.2.7 Map execution paths through application (OTG-INFO-007)
- 4.2.8 Fingerprint Web Application Framework (OTG-INFO-008)
- 4.2.9 Fingerprint Web Application (OTG-INFO-009)
- 4.2.10 Map Application Architecture (OTG-INFO-010)

## **4.3 Configuration and Deployment Management Testing**

- 4.3.1 Test Network/Infrastructure Configuration (OTG-CONFIG-001)
- 4.3.2 Test Application Platform Configuration (OTG-CONFIG-002)
- 4.3.3 Test File Extensions Handling for Sensitive Information (OTG-CONFIG-003)
- 4.3.4 Review Old, Backup and Unreferenced Files for Sensitive Information (OTG-CONFIG-004)
- 4.3.5 Enumerate Infrastructure and Application Admin Interfaces (OTG-CONFIG-005)
- 4.3.6 Test HTTP Methods (OTG-CONFIG-006)
- 4.3.7 Test HTTP Strict Transport Security (OTG-CONFIG-007)
- 4.3.8 Test RIA cross domain policy (OTG-CONFIG-008)

## **4.4 Identity Management Testing**

- 4.4.1 Test Role Definitions (OTG-IDENT-001)
- 4.4.2 Test User Registration Process (OTG-IDENT-002)
- 4.4.3 Test Account Provisioning Process (OTG-IDENT-003)
- 4.4.4 Testing for Account Enumeration and Guessable User Account (OTG-IDENT-004)
- 4.4.5 Testing for Weak or unenforced username policy (OTG-IDENT-005)

## **4.5 Authentication Testing**

- 4.5.1 Testing for Credentials Transported over an Encrypted Channel (OTG-AUTHN-001)
- 4.5.2 Testing for default credentials (OTG-AUTHN-002)
- 4.5.3 Testing for Weak lock out mechanism (OTG-AUTHN-003)
- 4.5.4 Testing for bypassing authentication schema (OTG-AUTHN-004)
- 4.5.5 Test remember password functionality (OTG-AUTHN-005)
- 4.5.6 Testing for Browser cache weakness (OTG-AUTHN-006)
- 4.5.7 Testing for Weak password policy (OTG-AUTHN-007)
- 4.5.8 Testing for Weak security question/answer (OTG-AUTHN-008)
- 4.5.9 Testing for weak password change or reset functionalities (OTG-AUTHN-009)
- 4.5.10 Testing for Weaker authentication in alternative channel (OTG-AUTHN-010)

## **4.6 Authorization Testing**

- 4.6.1 Testing Directory traversal/file include (OTG-AUTHZ-001)
- 4.6.2 Testing for bypassing authorization schema (OTG-AUTHZ-002)
- 4.6.3 Testing for Privilege Escalation (OTG-AUTHZ-003)
- 4.6.4 Testing for Insecure Direct Object References (OTG-AUTHZ-004)

## **4.7 Session Management Testing**

- 4.7.1 Testing for Bypassing Session Management Schema (OTG-SESS-001)**
- 4.7.2 Testing for Cookies attributes (OTG-SESS-002)**
- 4.7.3 Testing for Session Fixation (OTG-SESS-003)**
- 4.7.4 Testing for Exposed Session Variables (OTG-SESS-004)**
- 4.7.5 Testing for Cross Site Request Forgery (CSRF) (OTG-SESS-005)**
- 4.7.6 Testing for logout functionality (OTG-SESS-006)**
- 4.7.7 Test Session Timeout (OTG-SESS-007)**
- 4.7.8 Testing for Session puzzling (OTG-SESS-008)**

## **4.8 Input Validation Testing**

**4.8.1 Testing for Reflected Cross Site Scripting (OTG-INPVAL-001)**

**4.8.2 Testing for Stored Cross Site Scripting (OTG-INPVAL-002)**

**4.8.3 Testing for HTTP Verb Tampering (OTG-INPVAL-003)**

**4.8.4 Testing for HTTP Parameter pollution (OTG-INPVAL-004)**

**4.8.5 Testing for SQL Injection (OTG-INPVAL-005)**

**4.8.5.1 Oracle Testing**

**4.8.5.2 MySQL Testing**

**4.8.5.3 SQL Server Testing**

**4.8.5.4 Testing PostgreSQL (from OWASP BSP)**

**4.8.5.5 MS Access Testing**

**4.8.5.6 Testing for NoSQL injection**

## **4.9 Testing for Error Handling**

4.9.1 Analysis of Error Codes (OTG-ERR-001)

4.9.2 Analysis of Stack Traces (OTG-ERR-002)

## **4.10 Testing for weak Cryptography**

4.10.1 Testing for Weak SSL/TLS Ciphers, Insufficient Transport Layer Protection (OTG-CRYPST-001)

4.10.2 Testing for Padding Oracle (OTG-CRYPST-002)

4.10.3 Testing for Sensitive information sent via unencrypted channels (OTG-CRYPST-003)

## **4.11 Business Logic Testing**

4.11.1 Test Business Logic Data Validation (OTG-BUSLOGIC-001)

4.11.2 Test Ability to Forge Requests (OTG-BUSLOGIC-002)

4.11.3 Test Integrity Checks (OTG-BUSLOGIC-003)

4.11.4 Test for Process Timing (OTG-BUSLOGIC-004)

4.11.5 Test Number of Times a Function Can be Used Limits (OTG-BUSLOGIC-005)

4.11.6 Testing for the Circumvention of Work Flows (OTG-BUSLOGIC-006)

4.11.7 Test Defenses Against Application Mis-use (OTG-BUSLOGIC-007)

4.11.8 Test Upload of Unexpected File Types (OTG-BUSLOGIC-008)

4.11.9 Test Upload of Malicious Files (OTG-BUSLOGIC-009)

## **4.12 Client Side Testing**

- 4.12.1 Testing for DOM based Cross Site Scripting (OTG-CLIENT-001)**
- 4.12.2 Testing for JavaScript Execution (OTG-CLIENT-002)**
- 4.12.3 Testing for HTML Injection (OTG-CLIENT-003)**
- 4.12.4 Testing for Client Side URL Redirect (OTG-CLIENT-004)**
- 4.12.5 Testing for CSS Injection (OTG-CLIENT-005)**
- 4.12.6 Testing for Client Side Resource Manipulation (OTG-CLIENT-006)**
- 4.12.7 Test Cross Origin Resource Sharing (OTG-CLIENT-007)**
- 4.12.8 Testing for Cross Site Flashing (OTG-CLIENT-008)**
- 4.12.9 Testing for Clickjacking (OTG-CLIENT-009)**
- 4.12.10 Testing WebSockets (OTG-CLIENT-010)**
- 4.12.11 Test Web Messaging (OTG-CLIENT-011)**
- 4.12.12 Test Local Storage (OTG-CLIENT-012)**

# Tools of Web Attackers and Security Testers

- Choose the right tools for the job
- Attackers look for tools that enable them to attack the system
  - They choose their tools based on the vulnerabilities found on a target system or application

# Web Tools

- Firefox and Chrome Developer Tools
  - View parameters and cookies
  - Modify and resend requests
- BurpSuite
  - Powerful proxy used for Web App hacking
- Zed Attack Proxy
  - Can do simple vulnerability scans

# Nikto

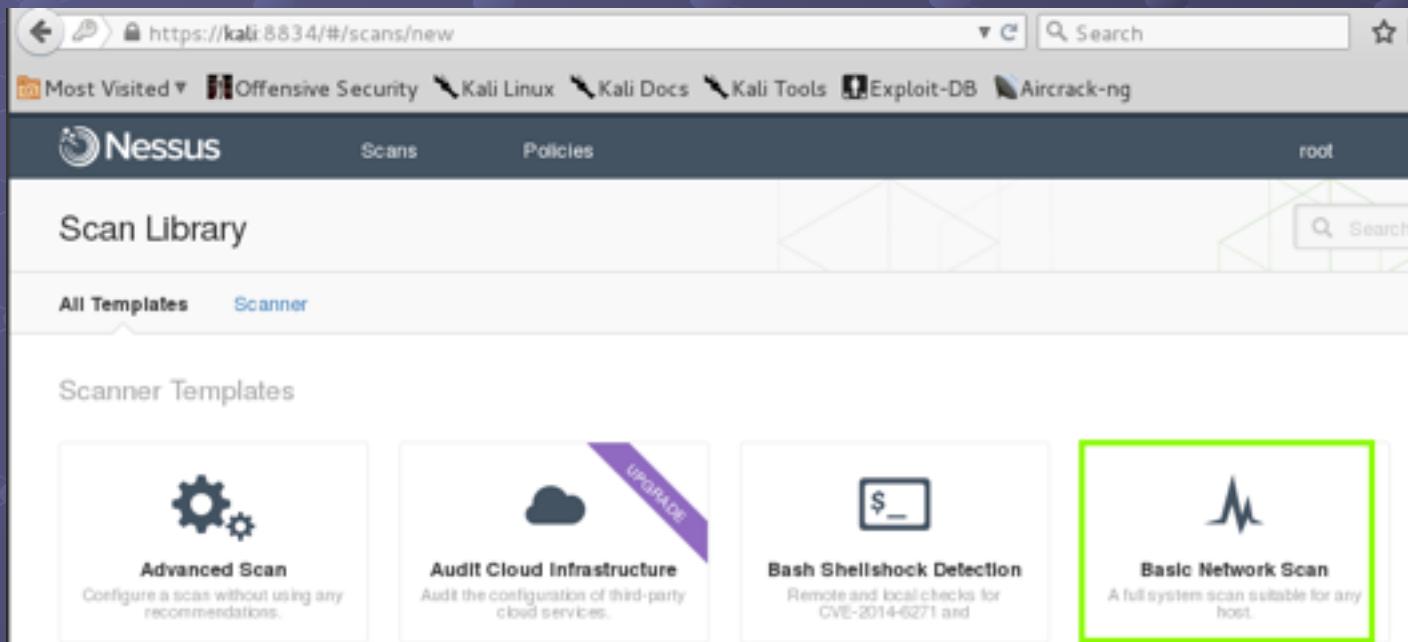
- Free, specialized for web apps

```
nikto -h 192.168.119.129
```

```
+ OSVDB-3092: /phpmyadmin/changelog.php: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.  
+ OSVDB-3092: /phpmyadmin/ChangeLog: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.  
+ OSVDB-3268: /icons/: Directory indexing found.  
+ OSVDB-562: /server-info: This gives a lot of Apache information. Comment out appropriate line in httpd.conf or restrict access to allowed hosts.  
+ OSVDB-3233: /icons/README: Apache default file found.  
+ /phpmyadmin/: phpMyAdmin directory found  
+ OSVDB-3092: /.svn/wc.db: Subversion SQLite DB file may contain directory listing information. See http://pen-testing.sans.org/blog/pen-testing/2012/12/06/all-your-svn-are-belong-to-us  
+ OSVDB-3092: /phpmyadmin/Documentation.html: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.  
+ /server-status: Apache server-status interface found (pass protected)  
+ /server-info: Apache server-info interface found (pass protected)  
+ 8345 requests: 0 error(s) and 29 item(s) reported on remote host  
+ End Time: 2015-09-23 18:45:52 (GMT-4) (41 seconds)  
-----  
+ 1 host(s) tested  
root@kali:~# █
```

# Nessus

- Commercial, thorough and popular
- Open-source fork is OpenVAS



# Nessus

**Nessus** Scans Policies root ▾ 🔍 Filter Hosts ▾

Win2008-YOURNAME  
CURRENT RESULTS: TODAY AT 7:36 PM

Configure Audit Trail Launch ▾ Export ▾ Filter Hosts ▾

Scans > Hosts [!] Vulnerabilities [!] Remediations [!] History [!]

Host Vulnerabilities ▾

Host	Vulnerabilities
192.168.119.129	39 Critical 74 High 241 Medium

Scan Details

Name:	Win2008-YOURNAME
Status:	Completed
Policy:	Basic Network Scan
Scanner:	Local Scanner
Folder:	My Scans
Start:	Today at 7:30 PM
End:	Today at 7:36 PM
Elapsed:	6 minutes
Targets:	192.168.119.129

Vulnerabilities



- Critical
- High
- Medium
- Low
- Info

# Nessus

Screenshot of the Nessus web interface showing a scan results page for host Win2008-YOURNAME / 172.16.1.191.

The left sidebar includes FOLDERS (My Scans, All Scans, Trash) and RESOURCES (Policies, Plugin Rules, Scanners).

The main content area shows the host details: IP: 172.16.1.191, MAC: 00:0c:29:e4:f5:81, OS: Microsoft Windows Server 2008 Standard Service Pack 1, Start: Today at 12:45 PM, End: Today at 12:52 PM, Elapsed: 7 minutes, KB: Download.

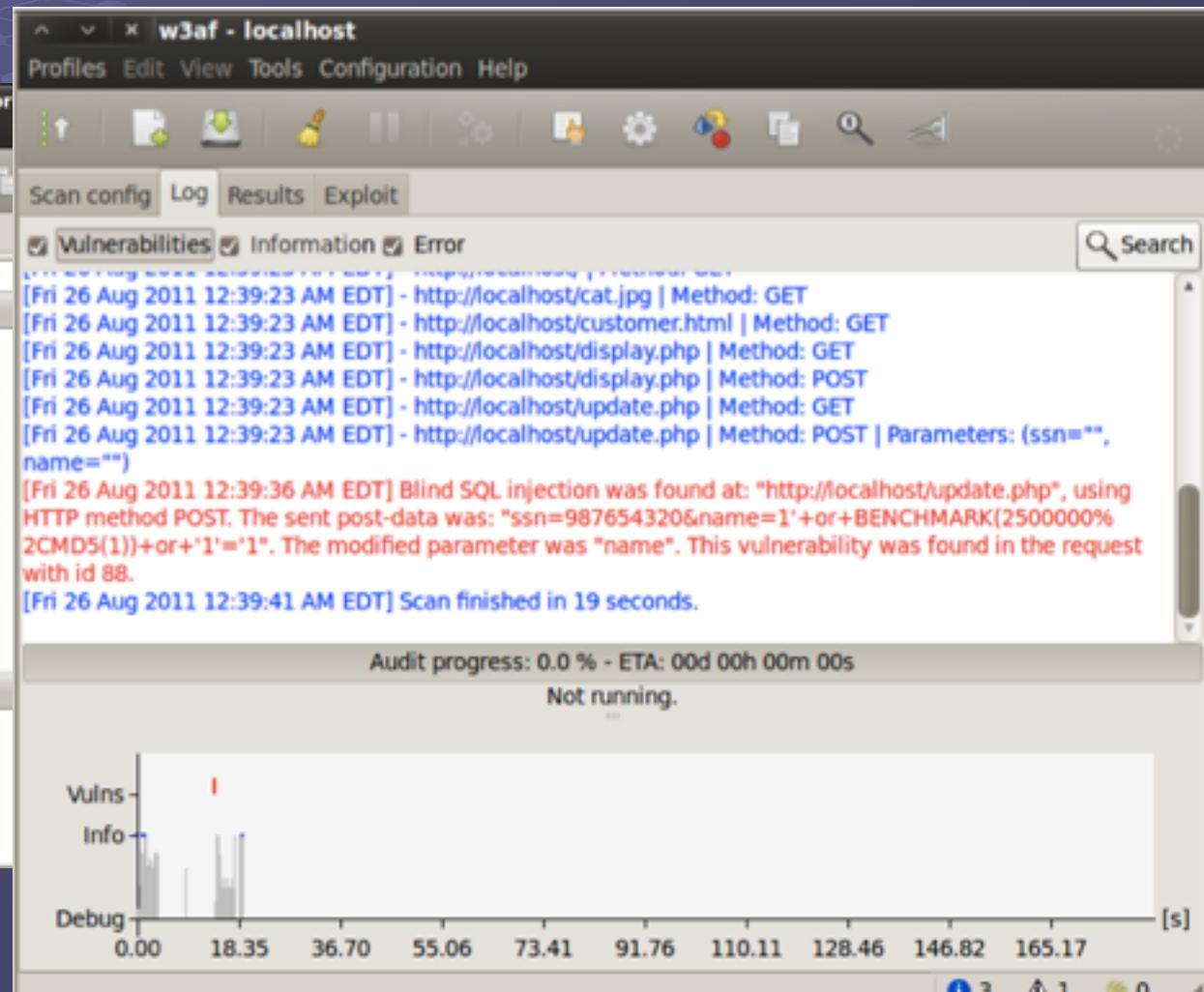
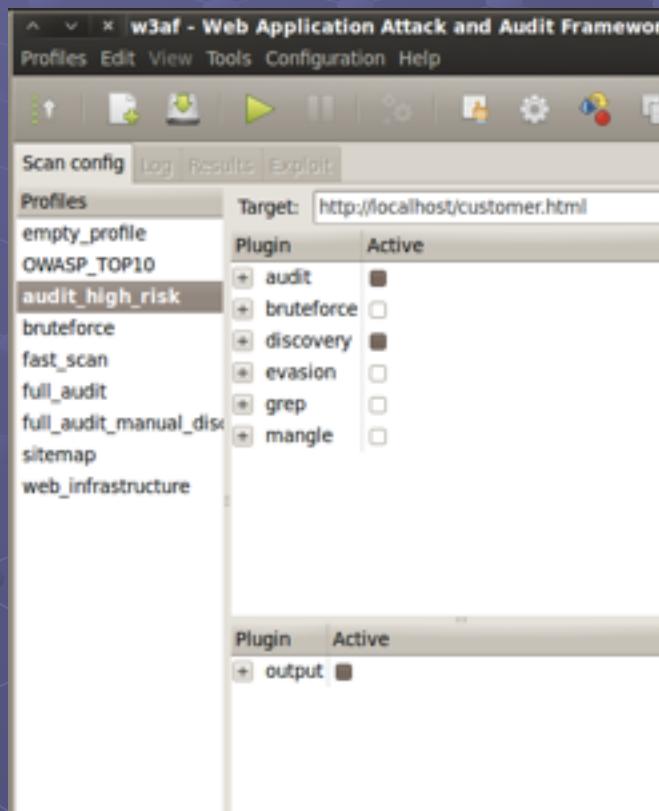
The Vulnerabilities section displays 118 vulnerabilities:

Sev	Name	Family	Count
Critical	Apache 2.2.x < 2.2.13 APR apr_palloc ...	Web Servers	2
Critical	Apache 2.2.x < 2.2.15 Multiple Vulnerab...	Web Servers	2
Critical	OpenSSL Unsupported	Web Servers	2
Critical	PHP 5.3.x < 5.3.15 Multiple Vulnerabilities	CGI abuses	2
Critical	PHP Unsupported Version Detection	CGI abuses	2
Critical	MS09-050: Microsoft Windows SMB2 ...	Windows	1
Critical	MS11-030: Vulnerability In DNS Resoluti...	Windows	1

A green box highlights the "CRITICAL" severity column header. A red box highlights the first critical vulnerability entry.

A pie chart in the Vulnerabilities section indicates the distribution of severity levels: Critical (red), High (orange), Medium (yellow), Low (green), and Info (blue).

# W3af (Free, in Kali)



# Skipfish from Google (Free)

Welcome to skipfish. Here are some useful tips:

- 1) To abort the scan at any time, press **Ctrl-C**. A partial report will be written to the specified location. To view a list of currently scanned URLs, you can press **space** at any time during the scan.
- 2) Watch the number requests per second shown on the main screen. If this figure drops below 100-200, the scan will likely take a very long time.
- 3) The scanner does not auto-limit the scope of the scan; on complex sites, you may need to specify locations to exclude, or limit brute-force steps.
- 4) There are several new releases of the scanner every month. If you run into trouble, check for a newer version first, let the author know next.

More info: <http://code.google.com/p/skipfish/wiki/KnownIssues>

Press any key to continue (or wait 60 seconds)... ■

Skipfish - scan results browser - Mozilla Firefox

File Edit View History Bookmarks Tools Help

BackTrack Linux Offensive-Security Tiger Security Exploit Database Air

file:///tmp/skip3/index.html

skipfish WEB APP SCANNER

Scanner version: 2.02b Scan date:  
Random seed: ox02a391f9 Total time:  
Problems w:

Crawl results - click to expand:

**http://192.168.5.93/** 49 3 47 7 23 58  
Code: 200, length: 8932, declared: text/html, detected: application/javascript, char

The Kahoot! logo is displayed on a teal background. The word "Kahoot!" is written in a bold, white, sans-serif font. The exclamation mark is smaller than the other letters and has a white outline.

Kahoot!

2