TTM4536 - Ethical Hacking - Information Security, Specialization Course

- Plan for 04 Oct 2015

Follow the material of Chapter 5 of the textbook

- The goal is to perform penetration testing on web servers

Follow the material of Chapter 5 of the textbook

- The goal is to perform penetration testing on web servers

Do not perform the Python scripts from the textbook directly on some real web server.

We will install vulnerable test servers in some other virtual machine!

Download “Web Security Dojo” VirtualBox machine from http://sourceforge.net/projects/websecuritydojo/?source=typ_redirect

Web Security Dojo

Brought to you by: gi0cann, mavensecurity, spinkham

★ 5.0 Stars (3)
180 Downloads (This Week)
Last Update: 2017-09-25

Download
Dojo-3.1.ova

Browse All Files

Description

Web Security Dojo is a preconfigured, stand-alone training environment for Web Application Security. Virtualbox and VMware versions for download. See "View all files" for VMware version.

Web Security Dojo Web Site ›

Looking for the latest version? **Download Dojo-3.1.ova (2.8 GB)**

Home

Name	Modified	Size	Downloads / Week
Version_3.1	2017-09-25		118 
Build_Files	2017-08-28		4 
Version_3.0	2017-05-08		22 
Version_3.0-beta	2017-05-08		0 
Version_2.1	2015-08-25		2 
Docs	2015-02-19		3 
Version_2.1-beta	2014-09-25		0 

Web Security Dojo

- An open source self-contained training environment for Web Application Security penetration testing. Tools + Targets = Dojo
- Various web application security testing tools and vulnerable web applications were added to a clean install of xubuntu 12.04 (now it offers Ubuntu 16.04 LTS). Build scripts are available in git at Sourceforge.
- For learning and practicing web app security testing techniques. It does not need a network connection since it contains both tools and targets. Therefore, it is ideal for self-study, training classes, and conferences. Also, this removes the possibility of remote attack on the targets, which are insecure by design.
- Sponsored by Maven Security Consulting. Open source.

- 6 minutes video about Web Security Dojo
- https://www.youtube.com/watch?v=lum6bSsyJ38
-

In VirtualBox
Import Appliance …

Install Dojo in the local folder:
courses/ttm4536/
(you need: Dojo root password: dojo)

If you have disk space – it would be useful to install Dojo on your machine.

Before starting Kali and Dojo do the following in VirtualBox

1. In Preferences go to Network
2. In NAT Networks Add a new NatNetwork
3. Make sure that it Supports DHCP

4. Change Network adapters for both Kali and Dojo to be NAT Network

1. Start Dojo and find out its IP address
2. Start Firefox (it will authomatically open targets.localh)
3. Start the vulnerable web server Insecure Web App.
4. See that the port that the server is listening is 8080
4. In a terminal run another web server that listens the port 81 with the following command:

```
python -m SimpleHTTPServer 81
```

Follow the material of Chapter 5 of the textbook

- Open a new project Ch05 in PyCharm
- Before you import and start the three scripts named: web_app_mapper.py, content_bruter.py and joomla_killer.py produce a small script 01.py and put the following commands:

01.py

Check the real IP address of Dojo machine

```
__author__ = 'root'
import urllib2

body = urllib2.urlopen("http://10.0.2.15:81")
print body.read()
```

01.py

First set Kali machine to allow all incoming web traffic with the command:

```
sudo iptables -A INPUT -p tcp --dport 81 -j ACCEPT
```

Then start the scrypt in a terminal with

```
python 01.py
```

01.py

Start the scrypt in a terminal with

```
python 01.py
```

You will get a response from the web page

02.py
How to disguise your browsing as “Googlebot”

```
import urllib2

url = "http://10.0.2.15:81"

headers = {}
headers['User-Agent'] = "Googlebot”

request = urllib2.Request(url,headers=headers)
response = urllib2.urlopen(request)

print response.read()
response.close()
```

02.py
How to disguise your browsing as “Googlebot”

```
import urllib2

url = "http://pi-cipher.org"

headers = {}
headers['User-Agent'] = "Googlebot”

request = urllib2.Request(url,headers=headers)
response = urllib2.urlopen(request)

print response.read()
response.close()
```

Learn about the Python’s power to work with indexed sets where the indexing is achieved not just by numbers (as in classical languages) but with arbitrary strings as indexes.

Adapt
web_app_mapper.py,
content_bruter.py
and
joomla_killer.py
such that they analyze the
web servers running on Dojo
machine

web_app_mapper.py

For this script to show you some results, you have to have same folder structure (some folder names) between the client and remote server.

For example to examine the started web server on the port 81 on Dojo machine, make some folder in your PyCharm and put some files that have the same names as in the Dojo machine.

After running the scripts you will get responses of the type
[200] => ….

content_bruter.py

For this script to show you some results, you have to do the following:

1. Download from internet a zip file SVNDigger
2. Unzip it, take the file all.txt and put it in the same folder where is your content_bruter.py script.
3. Run the script (attacking the web server in the Dojo machine with the port 81).
4. The analysis of the script will allocate some possible vulnerable files and folders in the server machine

joomla_killer.py

For this script to show you some results, you have to do the following:

1. Find the file cain.txt (in the zip file SVNDigger or somewhere from internet)
2. Put it in the same folder where is your joomla_killer.py script.
3. We will attack the Dojo server of the “Insecure Web App” that listens the port 8080
4. In the script put correct path to the web app scripts:

```
# target specific settings
target_url = "http://10.0.2.15:8080/insecure/public/Login.jsp"
target_post = "http://10.0.2.15:8080/insecure/public/Login.jsp"
```

5. By right clicking on web app boxes “Name” and “Password” and choosing “Inspect element” find the names of the variables that you have to crack (and put them in the fields)

```
username_field= "???????"
password_field= "?????"
```

6. In the field

```
success_check = "???????"
```

put some word that you expect it will be printed once you successfully log inn (if you are trying to crack administrator password then the word Administrator is reasonable to expect to be printed)

Play with the tasks and vulnerabilities in Dojo as described in its documentation