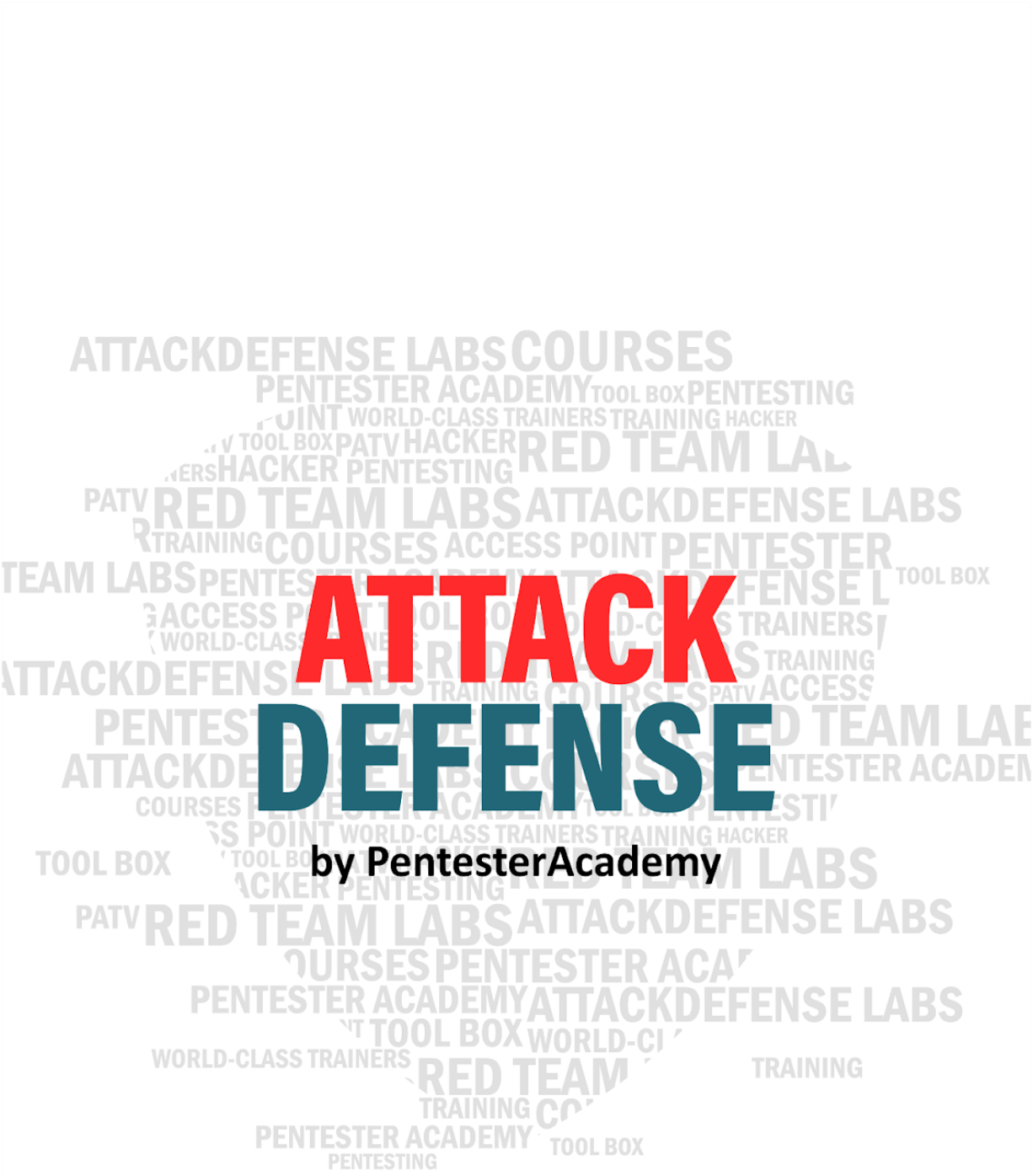
ATTACK
DEFENSE
by PentesterAcademy

Name	OSSAudit: Auditing Python Packages
URL	https://www.attackdefense.com/challengedetails?cid=2060
Type	DevSecOps Basics: Software Component Analysis

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Challenge Description

OSSAudit tool is used to audit installed Python packages and packages mentioned in the dependency files for known vulnerabilities. It uses the Sonatype OSS Index for reference.

A Kali CLI machine (kali-cli) is provided to the user with OSSAudit installed on it. The source code for three sample Python web applications is provided in the home directory of the root user.

Objective: Use OSSAudit utility to find issues in the web applications!

Instructions:

- The source code of web applications is provided at /root/github-repos

Solution

Step 1: Check the provided web applications.

Command: ls -l github-repos

```
root@attackdefense:~# ls -l github-repos/
total 12
drwxr-xr-x 5 root root 4096 Sep 14 07:03 django-rosetta
drwxr-xr-x 6 root root 4096 Sep 14 07:03 django-todolist
drwxr-xr-x 4 root root 4096 Sep 14 07:03 starter-python-bot
root@attackdefense:~#
```

Step 2: Check the available options for ossaudit tool

Command: ossaudit --help

```
root@attackdefense:~# ossaudit  --help
Usage: ossaudit [OPTIONS]

Options:
  -c, --config TEXT    Configuration file.
  -i, --installed      Audit installed packages.
  -f, --file FILENAME  Audit packages in file (can be specified multiple
                       times).
  --username TEXT      Username for authentication.
  --token TEXT         Token for authentication.
  --column TEXT        Column to show (can be specified multiple times).
                       [default: name, version, title]
  --ignore-id TEXT     Ignore a vulnerability by ID (can be specified multiple
                       times).
  --help               Show this message and exit.
root@attackdefense:~#
```

We will take one example at a time and run the tool on that.

Example 1: Django Rosetta

Step 1: Change to the django-rosetta directory and check its contents.

Commands:
cd github-repos/django-rosetta
ls

```
root@attackdefense:~# cd github-repos/django-rosetta/
root@attackdefense:~/github-repos/django-rosetta#
root@attackdefense:~/github-repos/django-rosetta# ls
CHANGES  docs  LICENSE  MANIFEST.in  README.rst  rosetta  setup.py  testproject  tox.ini
root@attackdefense:~/github-repos/django-rosetta#
```

Step 2: Run the ossaudit tool to find possible flaws in the python packages. Scan the tox.ini of the Django application. Tox is a generic test command-line tool.

Command: ossaudit -f tox.ini

```
root@attackdefense:~/github-repos/django-rosetta# ossaudit -f tox.ini
+----------+---------+-------------------------------------------------------------------------------------+
|   name   | version |                                        title                                        |
+==========+=========+=====================================================================================+
| requests | 0       | [CVE-2014-1830]  Information Exposure                                               |
+----------+---------+-------------------------------------------------------------------------------------+
| requests | 0       | [CVE-2014-1829]  Information Exposure                                               |
+----------+---------+-------------------------------------------------------------------------------------+
| requests | 0       | [CVE-2018-18074]  Credentials Management                                            |
+----------+---------+-------------------------------------------------------------------------------------+
| lxml     | 3.3.4   | [CVE-2018-19787]  Improper Neutralization of Input During Web Page Generation       |
|          |         | ("Cross-site Scripting")                                                            |
+----------+---------+-------------------------------------------------------------------------------------+
| lxml     | 3.3.4   | [CVE-2014-3146] Incomplete blacklist vulnerability in the lxml.html.clean module in |
|          |         | lxml before ...                                                                     |
+----------+---------+-------------------------------------------------------------------------------------+
Found 5 vulnerabilities in 10 packages
root@attackdefense:~/github-repos/django-rosetta#
```

Issues Detected

- Information Exposure
- Credentials Management
- Improper Neutralization of Input During Web Page Generation
- Incomplete blacklist vulnerability in the lxml.html.clean mobile in lxml

Example 2: Starter Python Bot

Step 1: Change to the starter-python-bot directory and check its contents.

Commands:

cd ~/github-repos/starter-python-bot
ls

```
root@attackdefense:~/github-repos# cd starter-python-bot/
root@attackdefense:~/github-repos/starter-python-bot#
root@attackdefense:~/github-repos/starter-python-bot# ls
bot  bot.yml  Dockerfile  LICENSE.txt  README.md  requirements.txt  resources
root@attackdefense:~/github-repos/starter-python-bot#
```

Step 2: Run the ossaudit tool to find possible flaws in the python packages. Requirement.txt contains the names of the packages which will be used by the python project.

Command: ossaudit -f requirements.txt

```
root@attackdefense:~/github-repos/starter-python-bot# ossaudit -f requirements.txt
+--------------+---------+-----------------------------------------------+
|     name     | version |                     title                     |
+==============+=========+===============================================+
| PyYAML       | 3.11    | [CVE-2017-18342]  Improper Input Validation |
+--------------+---------+-----------------------------------------------+
| requests     | 2.9.1   | [CVE-2018-18074]  Credentials Management    |
+--------------+---------+-----------------------------------------------+
| cryptography | 0.9.1   | [CVE-2016-9243]  Improper Input Validation  |
+--------------+---------+-----------------------------------------------+
Found 3 vulnerabilities in 11 packages
root@attackdefense:~/github-repos/starter-python-bot#
```

Issues Detected

- Improper Input Validation
- Credentials Management

Example 3: Django Todolist

Step 1: Change to the django-todolist directory and check its contents.

Commands:

cd ~/github-repos/django-todolist
ls

```
root@attackdefense:~/github-repos# cd django-todolist/
root@attackdefense:~/github-repos/django-todolist#
root@attackdefense:~/github-repos/django-todolist# ls
accounts  api  db.sqlite3  LICENSE  lists  manage.py  README.md  requirements.txt  todolist
root@attackdefense:~/github-repos/django-todolist#
```

Step 2: Run the ossaudit tool to find possible flaws in the python packages.

Command: ossaudit -f requirements.txt

```
root@attackdefense:~/github-repos/django-todolist# ossaudit -f requirements.txt
+---------+---------+-----------------------------------+
|  name   | version |               title               |
+=========+=========+===================================+
| aiohttp | 0.16.2  | Directory traversal vulnerability |
+---------+---------+-----------------------------------+
Found 1 vulnerabilities in 3 packages
root@attackdefense:~/github-repos/django-todolist#
```

Issues Detected

- Directory traversal vulnerability

Learnings

Perform Software Component Analysis using the ossaudit tool.