**Penetration Testing Report**

Black Box Penetration Testing

For IDE Tryhackme room

V1.0

November 20th, 2022

By: Stefan Codreanu

**1 Executive Summary**



This document details the security assessment (external penetration testing) of IDE TRYHACKME room. The purpose of the assessment was to provide a review of the security posture of IDE Internet infrastructure, as well, as to identify potential weaknesses in its Internet infrastructure.

**1.1 Scope of work**

This security assessment covers the remote penetration testing of an accessible server hosted on 10.10.99.12 address. The assessment was carried out from a black box perspective, with the only supplied information being the tested server IP address. No other information was assumed at the start of the assessment.

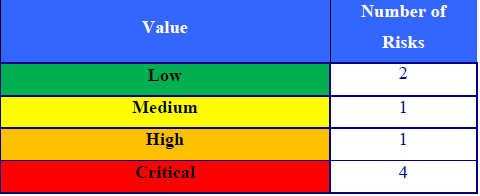
**1.2 Project Objectives**

This security assessment is carried out to gauge the security posture of IDE’s Internet-facing host. The result of the assessment is then analyzed for vulnerabilities. Given the limited time that is given to perform the assessment, only immediately exploitable services have been tested. The vulnerabilities are assigned a risk rating based on threat, vulnerability, and impact.

**1.3 Assumption**

While writing the report, we assume that the IP address is considered to be public IP address, the NDA and rules of engagement have been signed, and based on the information-gathering phase the company name is IDE.

**1.4 Summary of Findings**



IDE needs to pay more attention to information security and needs to invest in implementing a defence-in-depth approach to have multiple layers of security to protect their information

asset. Systems and networks hardening and secure configurations, for instance, should be implemented to strengthen the different layers of security within IDE.

Below are the high level findings from the external penetration test:

 It was obvious that IDE`S patch management policy and procedure is either

not existing or not implemented correctly. The web server was running Apache 2.4.29 and Codiad 2.8.4 which are not the latest versions. This opened a very high security risk for the organization.

 Services installed were running with default configurations such as FTP.

 The server was running OpenSSH 8.9 and sudo version 1.8.12 and were not updated to their latest versions which led to further compromise of the environment.

 Weak password policy, one of the users had a default password used for login and the other one had it stored on the server in the bash history.

**1.5 Summary of Recommendation**

Adopt defence-in-depth approach where IDE utilizes a variety of security

tools/systems and processes to protect its assets and information. Among these:

 Perform security hardening on servers in the production environment

especially those on the Internet and/or external DMZs.

 Implement Patch management system(s) to provide centralized control over

fixes, updates, and patches to all systems, devices, and equipment. This will

minimize overhead on the operations team and will elevate security resistance.

 Conduct vulnerability assessment at least twice a year and penetration testing

at least once a year or if there is a major change in the information assets.

 Develop and implement a training path for the current IT staff.

**2 Detail findings**

**2.1 Detailed System Information**

IP Address : 10.10.99.12

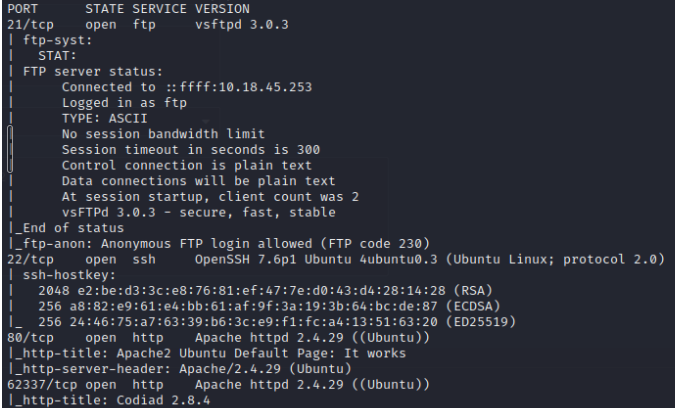
OS Information: Ubuntu 4unbuntu0.3

Open Ports: 21 FTP

22 SSH

80 HTTP

62337 HTTP



**2.2 Linux server 10.10.99.12**

**Anonymous logins are allowed on the remote FTP server:**

**Threat Level**

Medium

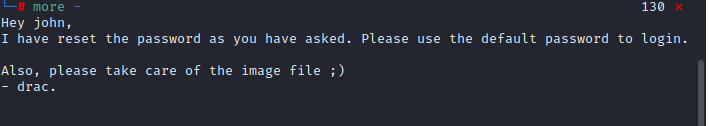
**Vulnerability**

Medium

**Analysis**

FTP server running on the remote host allows anonymous logins. Therefore, any remote user may connect and authenticate to the server without providing a password or unique credentials. This allows the user to access any files made available by the FTP server.

Login to FTP we discovered a file that had sensitive data in it such as a user and a password which allowed us to exploit the server further.



**Impact**

High

**Risk rating**

Medium

**Recommendation**

Disable anonymous FTP if it is not required. Routinely check the FTP server to ensure that sensitive content is not being made available.

[**Codiad 2.8.4 Remote Code Execution**](https://packetstormsecurity.com/files/download/162753/codiad284authed-exec.txt):

**Threat Level**

High

**Vulnerability**

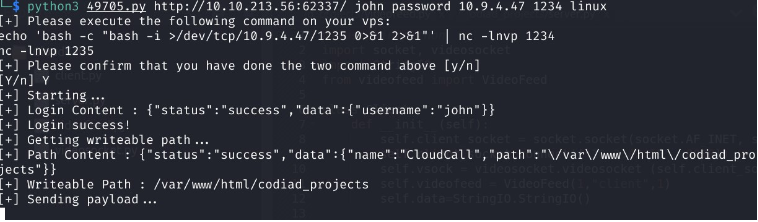
Critical

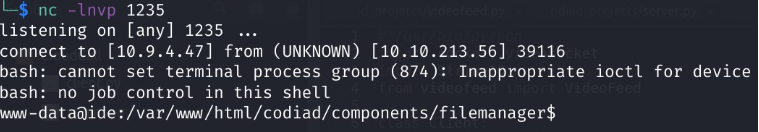
**Analysis**

The http web server running at port 62337 is using a version of Codiad that  suffers from a remote code execution vulnerability.

We exploit this vulnerability utilizing a ready exploit available on the internet and the credentials we found earlier on the FTP server. After exploiting this vulnerability we got a reverse shell on the system.

<https://www.exploit-db.com/exploits/49705>





**Impact**

Critical

**Risk rating**

Critical

**Recommendation**

Patch the web server running Codiad with the latest version.

**User drac member of sudo group**

**Threat Level**

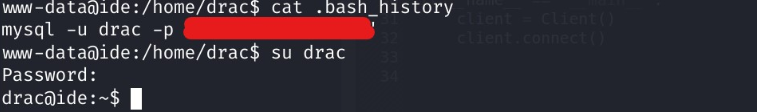
High

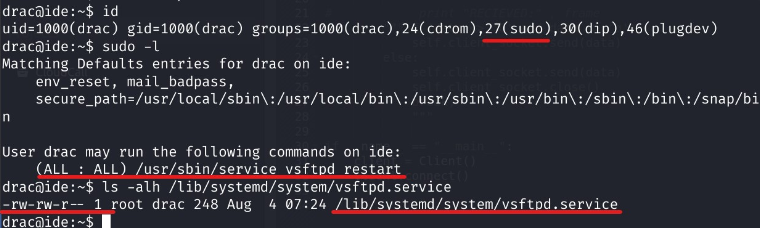
**Vulnerability**

Critical

**Analysis**

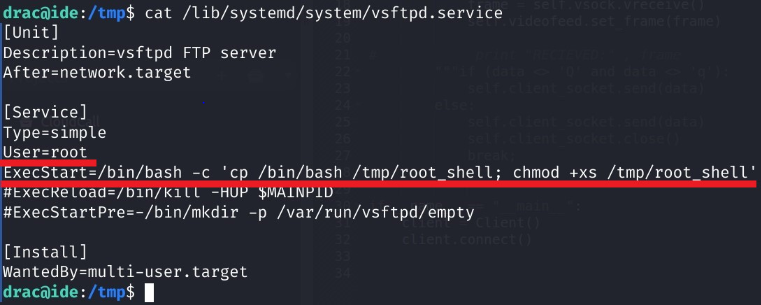
Upon further enumeration we found the password of user drac stored on the system in the .bash\_history file. We managed to login as user drac and we checked if he belongs to any groups and what commands can he run as root.



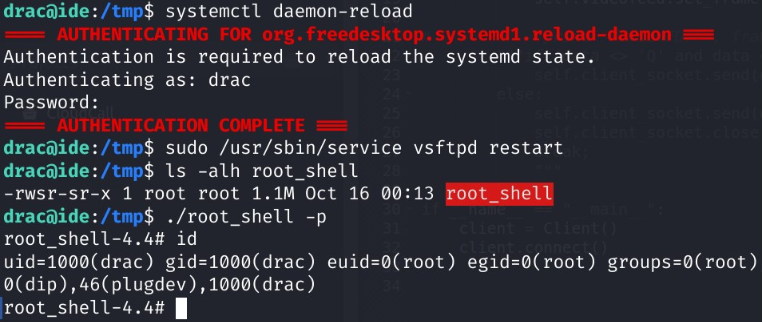


Since we could edit the vsftpd service file we modified the service file to run the following commands:

* cp /bin/bash /tmp/root\_shell
* chmod +xs /tmp/root\_shell



Once the file was modified we reloaded the daemon using systemctl daemon-reload and this allowed us to run our modified service using sudo /usr/sbin/service vsftpd restart that created a file called root\_shell. With that file we managed to elevate our privileges to root and fully compromise the environment.



**Impact**

Critical

**Risk Rating**

Critical

**Recommendation**

It is a good idea to hunt proactively across your Linux estate for insecure and weak passwords contained within the bash history file. If you are able to surface these issues at the earliest opportunity, it will strengthen your security posture and hamper the adversary. Remove any users from the sudo group and do not allow them to run any commands as root.

**Sudo version 1.8.21**

**Threat Level**

High

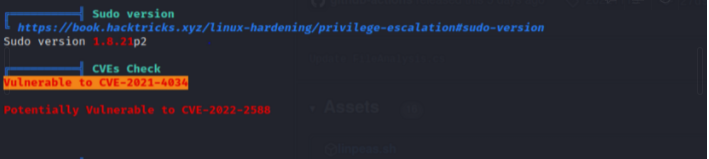
**Vulnerability**

High

**Analysis**

This version of sudo is vulnerable to **CVE-2021-4034**. By exploiting this vulnerability, attackers on a vulnerable host could easily gain full root privileges from any unprivileged user.

<https://www.exploit-db.com/exploits/50689>



**Impact**

High

**Risk rating**

Critical

**Recommendation**

The recommended fix is to update your systems according to the security advisories of your Linux distribution type.

RadHat: <https://access.redhat.com/security/vulnerabilities/RHSB-2022-001>

Ubuntu: <https://ubuntu.com/security/notices/USN-5252-1>

Debian: <https://security-tracker.debian.org/tracker/CVE-2021-4034>

NIST Advisory: <https://nvd.nist.gov/vuln/detail/CVE-2021-4034>

In order to mitigate it without updating, remove the setuid permission from pkexec:

chmod 0755 $(which pkexec)