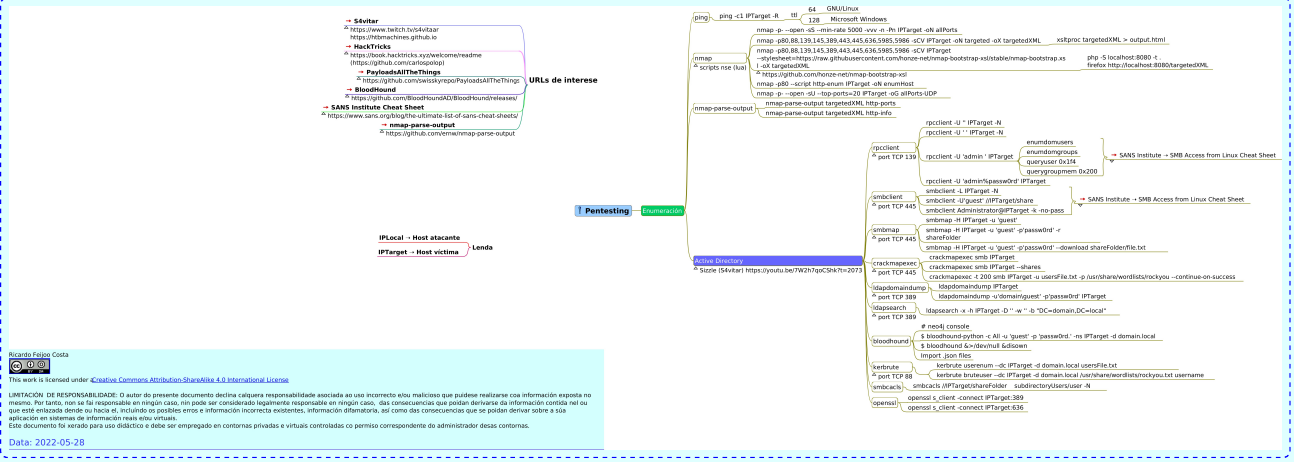
Práctica Seguridade Informática: Active Directory - Enumeración

LIMITACIÓN DE RESPONSABILIDADE O autor do presente documento declina calquera responsabilidade asociada ao uso incorrecto e/ou malicioso que puidese realizarse coa información exposta no mesmo. Por tanto, non se fai responsable en ningún caso, nin pode ser considerado legalmente responsable en ningún caso, das consecuencias que poidan derivarse da información contida nel ou que esté enlazada dende ou hacia el, incluíndo os posibles erros e información incorrecta existentes, información difamatoria, así como das consecuencias que se poidan derivar sobre a súa aplicación en sistemas de información reais e/ou virtuais. Este documento foi xerado para uso didáctico e debe ser empregado en contornas privadas e virtuais controladas co permiso correspondente do administrador desas contornas.

NOTA:

- Documentación de interese:

Enumeración AD-DC



Passwords cracking en Windows de hashes NTLM ntds.dit de Active Directory

- NTLM hashes → PasstheHass
- Net-NTLM hashes → NOT PasstheHass → Relay → SMB Relay → **responder + ntlmrelayx**
- **john** || **hashcat** → Posibility crack all hashes
  - **wordlist → rockyou**, **kaonashi**

    # john --wordlist=/usr/share/wordlists/rockyou.txt hashFile.txt
    # john --restore
    # hashcat -a 0 -m 5600 hashFile.txt /usr/share/wordlists/rockyou.txt -o cracked.txt #MODE: 5600, TYPE: NetNTLMv2
    # hashcat --restore
  - brute force

    # hashcat -a 3 -m 5600 hashFile.txt -1 ?l?d?u ?1?1?1?1?1?1?1 -o cracked.txt
    # hashcat --restore
  - rules

    # hashcat -m 1000 hashFile.txt.ntds /usr/share/wordlists/rockyou.txt -r /usr/share/hashcat/rules/InsidePro-PasswordsPro.rule --force #MODE: 1000, TYPE: NTLM
    # hashcat --restore
  - Información sobre hashes

    # hashcat --example
    # hashcat --example | grep -B2 -i kerberos
    # hashcat -m 7500 --example--hashes

1. **Comprobar conectividade e Enumerar sistema operativo mediante TTL (64→Linux, 128→Windows)**

   # ping -c1 192.168.120.100 -R

2. **Enumerar portos TCP open (nmap)**

   # nmap -p- --open -sS --min-rate 5000 -vvv -n -Pn 192.168.120.100 -oN allPorts

   ```
   # nmap -p- --open -T5 -vvv -Pn 192.168.120.100 -oN allPorts
   ```

3. **Enumerar servizos e versións**

   # nmap -sCV -p88,139,389,445,5985,5986 192.168.120.100 -oN targeted

   ```
   # nmap -sCV -p88,139,389,445,5985,5986 192.168.120.100 -oN targeted -oX targeted.xml
   # xsltproc targeted.xml -o targeted.html
   ```

4. **Enumerar recursos compartidos (port 445) do dominio**

   - **Sen saber credenciais (null session)**

     $ crackmapexec smb 192.168.120.100
     $ crackmapexec smb 192.168.120.100 --shares
     $ smbclient -L 192.168.120.100 -N

     ```
     $ smbclient //192.168.120.100/folderShare -N # Conexión null session ao recurso compartido folderShare
     ```

     $ smbmap -H 192.168.120.100
     $ smbmap -H 192.168.120.100 -u 'sdklfladkjf'

   - **Sabendo credenciais (user%contrasinal)**

     $ smbmap -H 192.168.120.100 -u 'guest' -p 'abc123.'
     ...
     Users # Recurso compartido
     ...
     $ smbmap -H 192.168.120.100 -u 'guest' -p 'abc123.' -r Users # Ver contido recurso compartido Users
     $ smbmap -H 192.168.120.100 -u 'guest' -p 'abc123.' --download Users/guest/Desktop/user.txt # Descargar ficheiro user.txt

5. **Enumerar usuarios do dominio**

   - **Sen saber credenciais (null session)**

     $ rpcclient -U '' 192.168.120.100 -N

     - rpc → port TCP 139

     $ kerbrute userenum --dc 192.168.120.100 -d empresa.local usersFile.txt # O ficheiro usersFile.txt contén soamente por liña un nome de posible usuario do dominio

     - kerberos → port TCP 88
       $ git clone https://github.com/ropnop/kerbrute.git && cd kerbrute
       $ sudo apt -y install golang-go && go build -ldflags '-s -w' kerbrute && upx –brute kerbrute # Minimizar tamaño executable kerbrute
       $ ntpdate -s 192.168.120.100 # Sincronizar reloxo do DC

   - **Sabendo credenciais (user%contrasinal)**

     A. **rpcclient**

     $ rpcclient -U 'guest%abc123.' 192.168.120.100
     >rpcclient> enundomusers
     >rpcclient> enundomgroups (ver 0x200 grupo Domain Admins)
     >rpcclient> querygroupnem 0x200
     >rid:[0xNNN] ... # user administrador
     >rpcclient> queryuser 0xNNN # Vemos info sobre este user administrador

     $ rpcclient -U 'guest%abc123.' 192.168.120.100 -c 'enumdomusers' # Lista usuarios do dominio
     $ rpcclient -U 'guest%abc123.' 192.168.120.100 -c 'enumdomusers' | grep -oP '\[.*?\]' | grep -v '0x' | tr -d '[]' | sort -u | sponge usersFile.txt # Gardar a lista usuarios do dominio no ficheiro usersFile.txt

$ for rid in $(rpcclient -U 'empresa.local\guest%abc123.' 192.168.120.100 -c 'enumdomusers' | grep -oP '\[.*?\]' | grep '0x' | tr -d '[]'); do
echo -e "\n[+] Para o $rid facer:\n"; rpcclient -U 'empresa.local\guest%abc123.' 192.168.120.100 -c "queryuser $rid" | grep -Ei 'user name|description';
done # Lista usuarios/descrición do dominio polo seu RID

B. **ldapdomaindump**

$ ldapdomaindump -u 'EMPRESA.LOCAL\guest' -p 'abc123' 192.168.120.100 # Todo a info vai ser visible a través de html

- ldap → port TCP 389
- ldapssl → port TCP 636
- globalcatLDAP → port TCP 3268
- globalcatLDAPssl → port TCP 3269

$ sudo php -S 0.0.0.0:80 -t . || sudo python -m http.server 80 || sudo /etc/init.d/apache2 start

$ sudo php -S 127.0.0.1:80 -t . || sudo python -m http.server --bind 127.0.0.1 80 # Por se queremos indicar a IP

$ firefox http://localhost/domain_users_by_group.html # De interese: Remote Managements Users → Acceso Remoto con eses usuarios → Conseguir hash/passwords → evil-winrm → consola remota con credenciais ou hash (PasstheHash)

C. **bloodhound || sharphound**

A. bloodhound → Execución no sistema local (atacante)

$ bloodhound-python -c All -u guest -p 'abc123.' -ns 192.168.120.100 -d empresa.local # Recopilar información en arquivos tipo json

$ mkdir bloodhound-local && cd bloodhound-local
$ git clone https://github.com/fox-it/BloodHound.py.git → alternativa a bloodhound impacket → bloodhound para empregar todo en local
$ cd BloodHound
$ sudo python setup.py install # Instala e coloca no PATH o executable bloodhound-python
$ which bloodhound-python
/usr/local/bin/bloodhound-python

B. sharphound → Execución no sistema remoto (víctima)

> .\SharpHound.exe -c All # Recopilar información nun arquivo tipo zip

$ mkdir sharphound-local && cd sharphound-local
$ wget https://github.com/BloodHoundAD/BloodHound/tree/master/Collectors/SharpHound.exe → SharpHound.exe → alternativa a bloodhound impacket → empregar no sistema víctima
$ wget https://github.com/BloodHoundAD/BloodHound/tree/master/Collectors/SharpHound.ps1 → SharpHound.ps1 → alternativa a bloodhound impacket → empregar no sistema víctima

GUI bloodhound

$ sudo apt -y install neo4j bloodhound
$ sudo update-alternatives --config java
...
Java11
...
$ sudo neo4j console
$ bloodhound &>/dev/null &
$ disown

Na GUI bloodhound http://localhost:7474 → neo4j/neo4j → cambiar credenciais → bloodhound → vías potenciais de buscar a forma de chegar a ser domain admin:

- → GUI → Upload data
  → subir json recolectados de bloodhound-python
  → descargar zip sharphound → descomprimir → subir json recolectados
- → GUI → Analysis
  → Find all Domain Admins
  → List all Kerberosastable Accounts
  → List AS-REP Roastable Users
- → GUI →Search → buscar polo user que temos as credenciais(user guest) → seleccionar ese usuario no mapa → botón dereito → Mark User as Owned → Ver caravela
  → Seleccionar de novo guest → Node info → Reachable High Value Targets

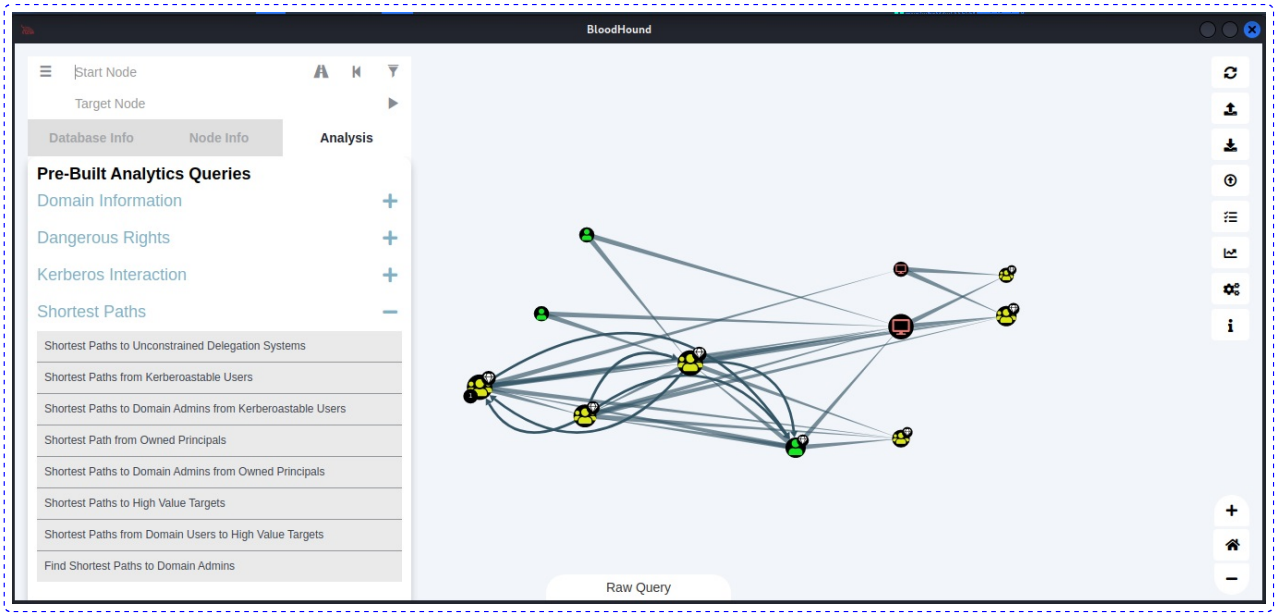


Fig.1 - Escenario bloodhound

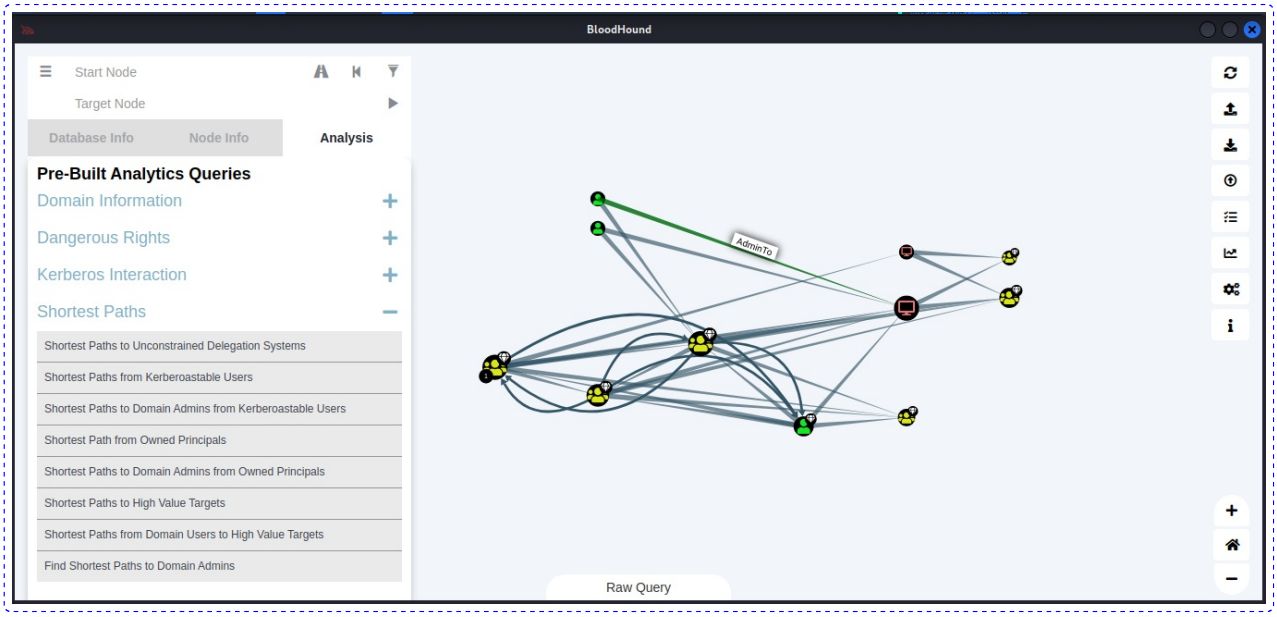


Fig.2 - bloodhound - AdminTo

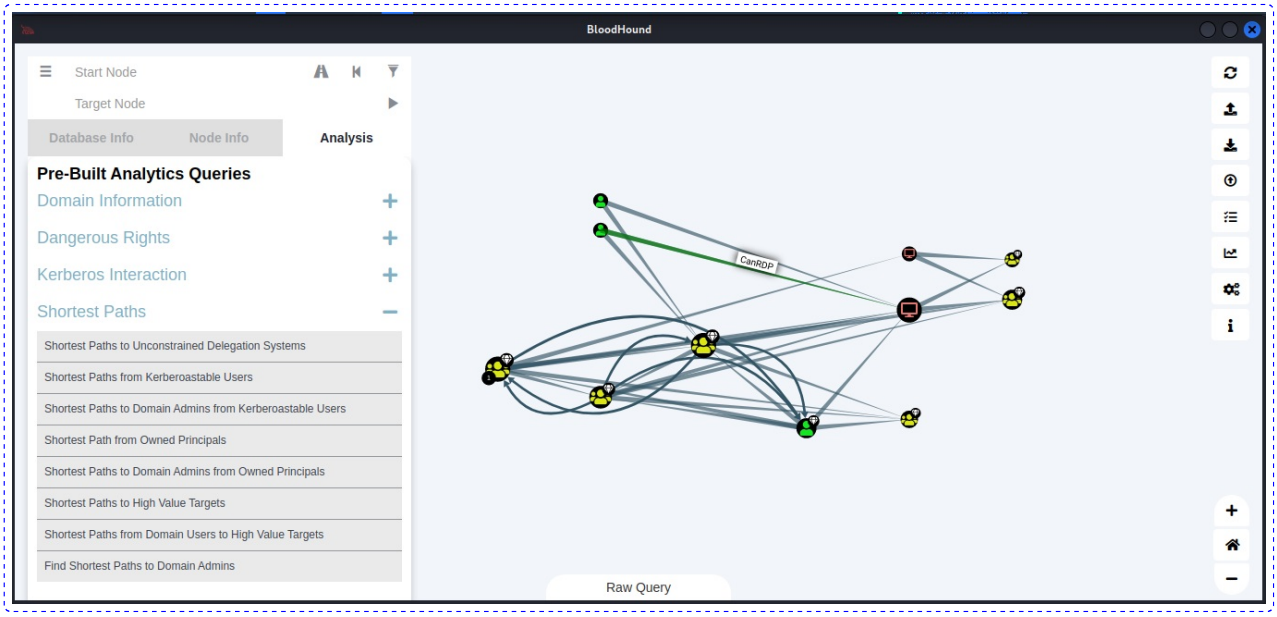


Fig.3 - bloodhound - CanRDP

Ricardo Feijoo Costa