

Downgrade Attacks by Example

How Compatibility breaks Security

Michael Rodler (@f0rki)

2012-01-21

about me

- ▶ @f0rki, <http://f0rki.at>
- ▶ Student “Sichere Informationssysteme Bachelor” at FH Hagenberg
 - ▶ 5th semester
- ▶ Member of Hagenberger Kreis and CTF Team
- ▶ Helps organize Security Forum
 - ▶ Annual security conference at Hagenberg
 - ▶ 18./19. April 2012
 - ▶ www.securityforum.at

What are “Downgrade Attacks”?

- ▶ In every application layer protocol there's some kind of Handshake
- ▶ Negotiation of common...
 - ▶ ... protocol version
 - ▶ ... protocol features
 - ▶ ... crypto algorithms
 - ▶ ... etc.

What are “Downgrade Attacks”?

- ▶ Man-in-the-middle (e.g. arp spoofing, fake ra, etc.)
- ▶ Attacker can alter traffic

What are “Downgrade Attacks”?

- ▶ Man-in-the-middle (e.g. arp spoofing, fake ra, etc.)
- ▶ Attacker can alter traffic

Downgrade Attack

The attacker acts as a proxy and alters the communication so that no or weaker security features are used by the client, the server or both.

- ▶ published 1994 – a long time ago
- ▶ had some serious security issues [7]
 - ▶ was fixed in SSL 3.0 in 1995
- ▶ Vulnerable to some kind of downgrade attack ¹
- ▶ No integrity protection of handshake messages

¹called Ciphersuite Rollback Attack back then

The Attack

- ▶ Replace Cipher Specs sent by client with weakest cipher suite

The Attack

- ▶ Replace Cipher Specs sent by client with weakest cipher suite

```
SSLv2 Record Layer: Client Hello
Length: 28
Handshake Message Type: Client Hello (1)
[...]
Cipher Specs (X specs)
  Cipher Spec: SSL2_DES_64_CBC_WITH_MD5 (0x060040)
  [...]
Challenge
```


- ▶ Integrity protection of handshake introduced
- ▶ Handshake ends with:
 - ▶ `change_cipher_spec` – change to negotiated parameters
 - ▶ `finished` – hash over handshake, key material
- ▶ need to check hash in finished message
 - ▶ detects tampering of handshake messages

Problem fixed!

Problem fixed!

... yeah right ...

- ▶ E-Mail is much older than SSL/TLS
 - ▶ First SMTP RFC in 1982
- ▶ Security introduced later
 - ▶ RFC for STARTTLS extension to SMTP in 2002
- ▶ Compatibility is essential

- ▶ explicit TLS
 - ▶ STARTTLS, STLS commandos
 - ▶ Client requests switching to TLS secured connection
- ▶ implicit TLS
 - ▶ imaps, pops
 - ▶ no attack vector here

```
* OK [CAPABILITY IMAP4rev1 LITERAL+ SASL-IR
  LOGIN-REFERRALS ID ENABLE IDLE STARTTLS
  LOGINDISABLED] Dovecot ready.
1 STARTTLS
1 OK Begin TLS negotiation now.
< TLS Handshake >
```

```
* OK [CAPABILITY IMAP4rev1 LITERAL+ SASL-IR
    LOGIN-REFERRALS ID ENABLE IDLE STARTTLS
    LOGINDISABLED] Dovecot ready.
1 STARTTLS
1 OK Begin TLS negotiation now.
< TLS Handshake >
```

The Attack

- ▶ Attacker strips out STARTTLS and LOGINDISABLED
- ▶ tricks client into thinking that the server does not support STARTTLS

```
S: +OK Dovecot ready.  
C: CAPA  
S: +OK  
S: CAPA  
S: [...]  
S: STLS  
S: .  
C: STLS  
S: +OK Begin TLS negotiation now.  
< TLS Handshake >
```



```
S: +OK Dovecot ready .  
C: CAPA  
S: +OK  
S: CAPA  
S: [...]  
S: STLS  
S: .  
C: STLS  
S: +OK Begin TLS negotiation now.  
< TLS Handshake >
```

The Attack

- ▶ Attacker strips out STLS
- ▶ tricks client into thinking that the server does not support STLS

```
S: 220 testmailer ESMTP Postfix (Ubuntu)
C: EHLO [10.42.42.2]
S: 250-testmailer
S: [...]
S: 250-STARTTLS
C: STARTTLS
S: 220 2.0.0 Ready to start TLS
< TLS Handshake >
```

```
S: 220 testmailer ESMTP Postfix (Ubuntu)
C: EHLO [10.42.42.2]
S: 250-testmailer
S: [...]
S: 250-STARTTLS
C: STARTTLS
S: 220 2.0.0 Ready to start TLS
< TLS Handshake >
```

The Attack

- ▶ Attacker strips out STARTTLS
- ▶ tricks client into thinking that the server does not support STARTTLS

- ▶ nothing new
- ▶ Attack is described in “Security Considerations” of RFCs
- ▶ Responsibility is at the client, to abort insecure connections

- ▶ nothing new
- ▶ Attack is described in “Security Considerations” of RFCs
- ▶ Responsibility is at the client, to abort insecure connections

Mail Clients

- ▶ Thunderbird > 3 – good
- ▶ Outlook 2007 – has “automatic” setting == bad
- ▶ Windos Live Mail – IMAP/POP: no support, SMTP: bad
- ▶ Apple Mail (v3.6) – no support
- ▶ Pegasus Mail – good, SMTP: bad

- ▶ don't use plaintext auth
- ▶ use PGP or S/MIME for end-to-end encryption
- ▶ use implicit TLS, e.g. imaps, pops

- ▶ most client software behaves correct anyway
- ▶ no real risk here

- ▶ Default is browsing over unsecured http:// connection
- ▶ Users get redirected to https:// via
 - ▶ links in html
 - ▶ 302 Redirects
 - ▶ Connection: Upgrade Header
- ▶ As with STARTTLS, this happens in unsecured traffic

Stripping https links

sslstrip by Moxie Marlinspike (presented at BlackHat DC 2009) [1] [2]

- ▶ http proxy
- ▶ strips out https links
- ▶ keeps track of https only resources

Stripping https links

sslstrip by Moxie Marlinspike (presented at BlackHat DC 2009) [1] [2]

- ▶ http proxy
- ▶ strips out https links
- ▶ keeps track of https only resources

Mitigation

- ▶ A smart user?
- ▶ https only website

Paper/presentations by László Tóth [5] [6], Steve Ocepek and Wendel G. Henrique [3]

Oracle protocols

- ▶ Proprietary protocols
 - ▶ Specifications only for \$\$\$
 - ▶ → hard to analyze
- ▶ Transparent Network Substrate (TNS)
 - ▶ simple/primitive protocol
 - ▶ Wireshark decoder exists
- ▶ Net8 or SQL*Net
 - ▶ complex and obscure
 - ▶ no wireshark decoder (only partial implementation)
- ▶ TNS transports Net8

Oracle Authentication I

- ▶ Challenge-Response
- ▶ Used crypto algorithms changed with every release

Oracle 8i

- ▶ Server sends session key encrypted with DES, Key is oraclehash of the user password
- ▶ Client sends user password encrypted with DES, Key is the session key

Oracle 9i

- ▶ Similar to 8i, but uses 3DES

Oracle Authentication II

Oracle 10g/11g

- ▶ Client/Server both send a session Key
→ MD5(XOR(ServerKey, ClientKey))
- ▶ uses AES-128/192 in 10g/11g

Problems

- ▶ DES is broken
- ▶ Bruteforce attack
- ▶ Java Thin Client till Version 10 supports only 8i

Several Downgrade Attacks published [5] [3] [6]

- ▶ Against old versions of Oracle 11 JDBC Driver
- ▶ “Downgrade through Replay”
 - ▶ Replace Handshake Packets with older Version
 - ▶ Combinations of versions and platforms behave differently
 - ▶ many *WTF?!?* moments...
- ▶ Attack against Oracle 10g Windows Client and Server
 - ▶ Downgrade to Oracle 8i level
 - ▶ metasploit module – release?

Attack!

No.	Source	Destination	Info
1	192.168.209.1	192.168.209.41	Request, Connect (1), Connect
2	192.168.209.41	192.168.209.1	Response, Resend (11)
3	192.168.209.1	192.168.209.41	Request, Connect (1), Connect
4	192.168.209.41	192.168.209.1	Response, Accept (2), Accept
5	192.168.209.1	192.168.209.41	Request, Data (6), SNS
6	192.168.209.41	192.168.209.1	Response, Data (6), SNS
7	192.168.209.1	192.168.209.41	Request, Data (6), Data
8	192.168.209.41	192.168.209.1	Response, Data (6), Data
9	192.168.209.1	192.168.209.41	Request, Data (6), Data
10	192.168.209.41	192.168.209.1	Response, Data (6), Data
11	192.168.209.1	192.168.209.41	Request, Data (6), Data
12	192.168.209.41	192.168.209.1	Response, Data (6), Data

```
0000 00 0c 29 7f 15 dc 00 0c 29 88 1b 7a 08 00 45 00 ..).....)..z..E.
0010 00 4d b8 c6 40 00 80 06 1e 68 c0 a8 d1 01 c0 a8 .M.@... .h.....
0020 d1 29 04 c3 05 f1 9d 09 c3 ce e4 ba ab 02 50 18 .)..... .P.
0030 ff 58 2a 00 00 00 00 25 00 00 06 00 00 00 00 00 .X*....% .....
0040 01 05 05 04 03 02 01 00 49 42 4d 50 43 2f 57 49 ..... IBMPC/WI
0050 4e 57 4e 54 2d 38 2e 31 2e 30 00 N_NT-8.1 .0.
```

first value was changed from 0x06 to 0x05

Attack!

No.	Source	Destination	Info
3	192.168.209.1	192.168.209.41	Request, Connect (1), Connect
4	192.168.209.41	192.168.209.1	Response, Accept (2), Accept
5	192.168.209.1	192.168.209.41	Request, Data (6), SNS
6	192.168.209.41	192.168.209.1	Response, Data (6), SNS
7	192.168.209.1	192.168.209.41	Request, Data (6), Data
8	192.168.209.41	192.168.209.1	Response, Data (6), Data
9	192.168.209.1	192.168.209.41	Request, Data (6), Data
10	192.168.209.41	192.168.209.1	Response, Data (6), Data
11	192.168.209.1	192.168.209.41	Request, Data (6), Data
12	192.168.209.41	192.168.209.1	Response, Data (6), Data
13	192.168.209.1	192.168.209.41	Request, Data (6), Data
14	192.168.209.41	192.168.209.1	Response, Data (6), Data

0000	00 50 56 c0 00 04 00 0c	29 88 1b 7a 08 00 45 00	.PV.....)..z..E.
0010	00 b3 1e 9e 40 00 80 06	b8 2a c0 a8 d1 29 c0 a8@...)*...)
0020	d1 01 05 f1 04 c3 e4 ba	ab 02 9d 09 c3 f3 50 18P.
0030	f9 48 39 50 00 00 00 8b	00 00 06 00 00 00 00 00	.H9P.....
0040	01 05 00 49 42 4d 50 43	2f 57 49 4e 5f 4e 54 2d	..IBMPC /WIN_NT-
0050	38 2e 31 2e 30 00 b2 00	01 00 00 00 64 00 00 00	8.1.0... ..d...
0060	60 01 24 0f 05 0b 0c 03	0c 0c 05 04 05 0d 06 09	.\$.....
0070	07 08 05 05 05 05 0f 05	05 05 05 05 05 0a 05 05
0080	05 05 05 04 05 06 07 08	08 23 47 23 23 08 11 23#G##..#
0090	08 11 41 b0 23 00 83 00	b2 07 d0 03 00 00 00 00	..A.#.....
00a0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00
00b0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00
00c0	00	

downgrade was accepted

Attack!

No.	Source	Destination	Info
2	192.168.209.41	192.168.209.1	Response, Resend (11)
3	192.168.209.1	192.168.209.41	Request, Connect (1), Connect
4	192.168.209.41	192.168.209.1	Response, Accept (2), Accept
5	192.168.209.1	192.168.209.41	Request, Data (6), SNS
6	192.168.209.41	192.168.209.1	Response, Data (6), SNS
7	192.168.209.1	192.168.209.41	Request, Data (6), Data
8	192.168.209.41	192.168.209.1	Response, Data (6), Data
9	192.168.209.1	192.168.209.41	Request, Data (6), Data
10	192.168.209.41	192.168.209.1	Response, Data (6), Data
11	192.168.209.1	192.168.209.41	Request, Data (6), Data
12	192.168.209.41	192.168.209.1	Response, Data (6), Data
13	192.168.209.1	192.168.209.41	Request, Data (6), Data

0040	03 73 03 90 80 a6 03 04 00 00 00 01 01 00 00 24	.s.....\$
0050	ef 12 00 0c 00 00 00 cc eb 12 00 b4 fb 12 00 04
0060	78 64 62 61 0d 00 00 0d 41 55 54 48 5f 50 41	xdba... .AUTH_PA
0070	53 53 57 4f 52 44 11 00 00 00 11 32 37 31 37 31	SSWORD... 27171
0080	36 46 46 46 35 37 44 31 31 34 39 34 00 00 00 00	6FFF57D1 1494...
0090	08 00 00 00 08 41 55 54 48 5f 52 54 54 05 00 00AUT H_RTT...
00a0	00 05 37 36 37 36 33 00 00 00 00 0d 00 00 00 0d	..76763.....
00b0	41 55 54 48 5f 43 4c 4e 54 5f 4d 45 4d 04 00 00	AUTH_CLN T_MEM...
00c0	00 04 34 30 39 36 00 00 00 00 0d 00 00 00 0d 41	..4096.....A
00d0	55 54 48 5f 54 45 52 4d 49 4e 41 4c 08 00 00 00	UTH_TERM INAL...
00e0	08 4d 43 48 30 36 39 35 43 00 00 00 0f 00 00	..MCH0695 C.....
00f0	00 0f 41 55 54 48 5f 50 52 4f 47 52 41 4d 5f 4e	..AUTH_P ROGRAM_N
0100	4d 0b 00 00 00 0b 73 71 6c 70 6c 75 73 2e 65 78	M.....sq lplus.ex
0110	65 00 00 00 00 0c 00 00 00 0c 41 55 54 48 5f 4d	e..... ..AUTH_M

- ▶ Strong passwords
- ▶ Keep Software up to date
 - ▶ especially JDBC driver
- ▶ Configure minimal accepted net8 version
`SQLNET.ALLOWED_LOGON_VERSION`
- ▶ (buy Oracle Advanced Security)
- ▶ (tunnel over SSH or SSL)

Questions?

- ▶ Tabular Data Stream Protocol (TDS)
 - ▶ Open Spezifikation [4]
 - not as painful as analyzing Oracle ;)
 - ▶ Wireshark Decoder exists
- ▶ Two types of authentication
 - ▶ Native authentication
 - ▶ Integrated/Windows authentication

Native Authentication

- ▶ Authentication with "Login7" packet
- ▶ No cryptographic Challenge-Response, no crypto at all???
- ▶ Password is obfuscated
 - ▶ no problem: obfuscation algorithm is in the standard

Native Authentication

- ▶ Authentication with "Login7" packet
- ▶ No cryptographic Challenge-Response, no crypto at all???
- ▶ Password is obfuscated
 - ▶ no problem: obfuscation algorithm is in the standard

but...

Wireshark – Normal Login Traffic

4	192.168.209.1	192.168.209.11	TDS7 pre-login message
5	192.168.209.11	192.168.209.1	Response
6	192.168.209.1	192.168.209.11	TDS7 pre-login message
7	192.168.209.11	192.168.209.1	TDS7 pre-login message
8	192.168.209.1	192.168.209.11	TDS7 pre-login message
9	192.168.209.11	192.168.209.1	TDS7 pre-login message
10	192.168.209.1	192.168.209.11	Unknown Packet Type: 23[Unreassembled Packet]
11	192.168.209.11	192.168.209.1	Response[Unreassembled Packet]
12	192.168.209.1	192.168.209.11	SQL batch
13	192.168.209.11	192.168.209.1	Response[Unreassembled Packet]

Wireshark – Decode as SSL

4	192.168.209.1	192.168.209.11	Ignored	Unknown	Record
5	192.168.209.11	192.168.209.1	Ignored	Unknown	Record
6	192.168.209.1	192.168.209.11	Ignored	Unknown	Record
7	192.168.209.11	192.168.209.1	Ignored	Unknown	Record
8	192.168.209.1	192.168.209.11	Ignored	Unknown	Record
9	192.168.209.11	192.168.209.1	Ignored	Unknown	Record
10	192.168.209.1	192.168.209.11	Application	Data	
11	192.168.209.11	192.168.209.1	Ignored	Unknown	Record
12	192.168.209.1	192.168.209.11	Ignored	Unknown	Record
13	192.168.209.11	192.168.209.1	Ignored	Unknown	Record

- ▶ SSL Handshake inside TDS Pre-Login packets
 - ▶ SSL Certificate is not checked

Native Authentication

- ▶ SSL Handshake inside TDS Pre-Login packets
 - ▶ SSL Certificate is not checked
- ▶ First Pre-Login packet
 - ▶ Sends protocol version, features, etc.
 - ▶ One field is called "Encryption" :)

Native Authentication

- ▶ SSL Handshake inside TDS Pre-Login packets
 - ▶ SSL Certificate is not checked
- ▶ First Pre-Login packet
 - ▶ Sends protocol version, features, etc.
 - ▶ One field is called "Encryption" :)

ENCRYPT_OFF	0x00	Encryption available but off.
ENCRYPT_ON	0x01	Encryption is available and on.
ENCRYPT_NOT_SUP	0x02	Encryption is not available.
ENCRYPT_REQ	0x03	Encryption is required.

Demo: Attack!!!

1. MITM Attack
2. Transparent “TDS-Proxy” as metasploit module
 - ▶ Sets “Encryption” field to “ENCRYPT_NOT_SUP”
3. ???
4. PROFIT!!!

Demo!

- ▶ use Windows Integrated Authentication
 - ▶ default during setup
 - ▶ Microsofts recommendation
- ▶ use “Force Encryption” option at server
- ▶ force encryption on client

- ▶ use Windows Integrated Authentication
 - ▶ default during setup
 - ▶ Microsofts recommendation
- ▶ use “Force Encryption” option at server
- ▶ force encryption on client

Responsible Disclosure → Answer

“Please note that SQL Server does not offer an option to enforce encryption of only the login packet (a.k.a. username & password), and at this point we have no plans to introduce such option.”

– Microsoft Incident Handler

Mitigation in general

Protocol Design

- ▶ Integrity protection of handshake messages
- ▶ Integrity more important than Confidentiality
 - ▶ no all or nothing
 - ▶ allow Integrity protection without Encryption
- ▶ use TLS from the beginning

Client/Server behaviour

- ▶ Abort connection on insufficient security
- ▶ alert user
- ▶ Ability to configure minimal version

Any Questions?






Moxie Marlinspike. *New Tricks For Defeating SSL In Practice*. URL: <https://www.blackhat.com/presentations/bh-dc-09/Marlinspike/BlackHat-DC-09-Marlinspike-Defeating-SSL.pdf> (visited on 01/05/2011).



Moxie Marlinspike. *sslstrip*. URL: <http://www.thoughtcrime.org/software/sslstrip/> (visited on 01/05/2011).



Steve Ocepek and Wendel G. Henrique. *Oracle, Interrupted: Stealing Sessions and Credentials*. Tech. rep. 2010. URL: <https://www.trustwave.com/downloads/spiderlabs/Trustwave-SpiderLabs-Oracle-Interrupted-Henrique-and-Ocepek.pdf> (visited on 11/16/2011).

-  . *Tabular Data Stream Protocol Specification*. 2011. URL: <http://msdn.microsoft.com/en-us/library/cc448435.aspx> (visited on 11/16/2011).
-  László Tóth. *Downgrading the Oracle native authentication*. Tech. rep. Price Waterhouse Coopers, Feb. 2007. URL: http://www.pwc.com/en_HU/hu/services/assets/oraauthdg-pub.pdf (visited on 11/16/2011).
-  László Tóth. *Oracle Authentication*. URL: http://soonerorlater.hu/download/hacktivity_lt_2009_en.pdf (visited on 12/19/2011).



David Wagner and Bruce Schneier. “Analysis of the SSL 3.0 protocol”. In: *Proceedings of the 2nd conference on Proceedings of the Second USENIX Workshop on Electronic Commerce - Volume 2*. Oakland, California: USENIX Association, 1996, pp. 4–4. URL: <https://www.schneier.com/paper-ssl-revised.pdf> (visited on 11/16/2011).