Hack the hackers: Leaking data over SSL/TLS

Ionut Cernica

Table of contents

whoami

Security Researcher @Future Networks 5G Lab

PhD Student @Department of Computer Science

CTF Player

Penetration Tester

Entrepreneur

Former Bug Bounty Hunter

Special Thanks

Pentest Tools.com

Get free access to the best online penetration testing platform for a limited time!

Sign up for a free 2 weeks trial:

https://pentest-tools.com/DEFCON2021

Blind Injection - How it works?

- One byte from each position of the information querying the server for True or False

- Boolean
- Time Based

- SQL Injection (Blind SQL Injection)

Problem???

➡ We can distinguish true and false from the encrypted traffic just looking on the length of the responses or the delays between them

Real Problem -> BILOST

- BILOST -> Blind Injection Leak Over SSL/TLS
- Passive attack; no interaction with the victim
- Blind Injection exploits are written in a PREDICTABLE way

f(n) = x ---- where “n” is the leak

- ➡ f -> the method
- ➡ x -> 2 types of output; True or False
- ➡ n -> the result; or the leak!

Known methods of exploiting with blind technique

- ⇨ Charset – abcdef-z 0123-9
- ⇨ Binary search – sqlmap demo
- ⇨ Bit Shifting – not so common

Challenges with the extraction methods

- No challenges when we are dealing with optimization techniques like binary search or bit shifting
- Charset method was the biggest challenge.

Exploit over SSL/TLS

- ⇨ We have the length of the packets; Boolean
- ⇨ We have the time between packets; Time based
- ⇨ Even there is a padding involved, in some cases we still exploit this problem. Would work if the block size is smaller than the difference between True and False responses
- ⚠ Is not a problem with the SSL/TLS protocol

Scenarios of exploiting

- Penetration testing company
- ISP – Internet Service Provider
- Maybe, one more reason why large countries have a strategic interest to pass the traffic of another country through their infrastructure

Over Tor? Future work?

- ⇨ You have exit nodes?
- ⇨ Be aware of the cell padding!

Tools and exploit databases

- All tools we analyzed exploits blind injections in a predictable manner; no exceptions ☹
- All public blind injection exploits we analyzed were found vulnerable to this problem

KEEP
CALM
IT'S
DEMO
TIME

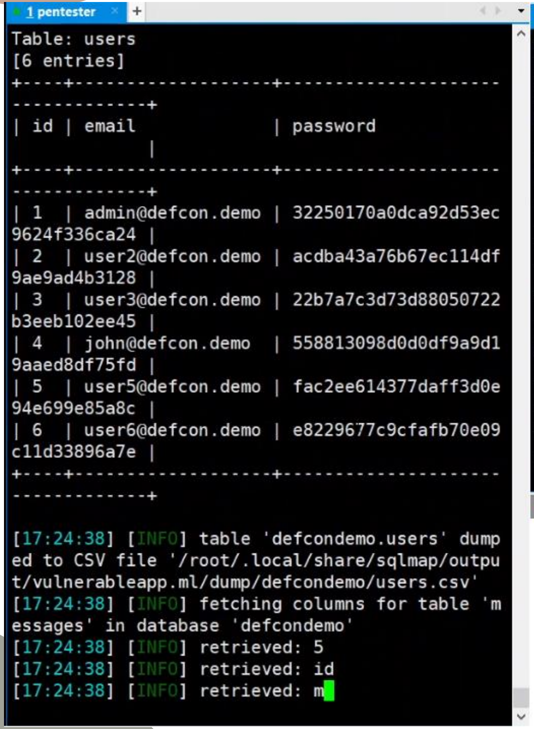
What we did in the demo?

- ⇨ **The pentester from company A exploits a vulnerable web app with sqlmap**

- ⇨ **The mitm took the encrypted traffic and did a passive attack; no interaction**

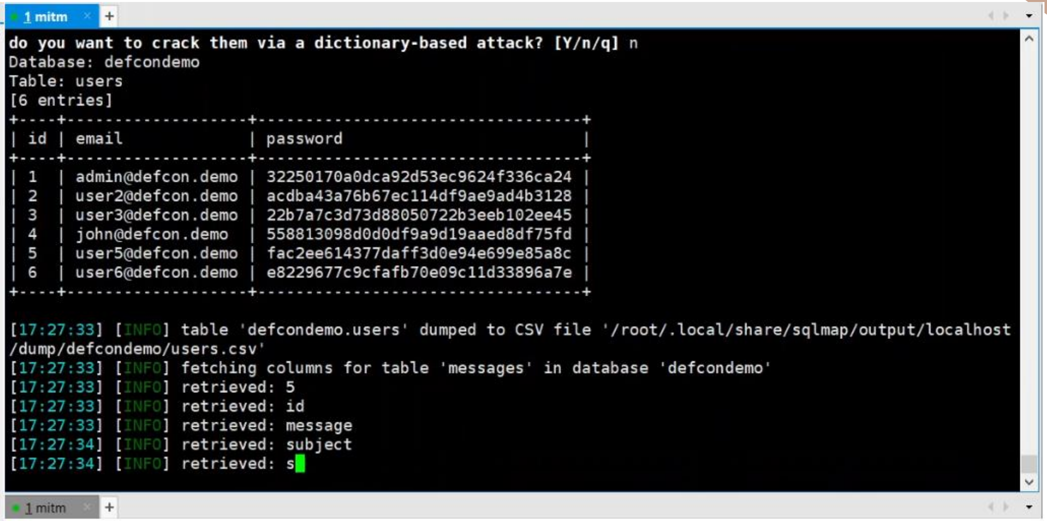
  extracted the True and False from packet lengths

  feed his local sqlmap with the same True and False to leak the data



```
Table: users
[6 entries]
+----+-------------------+---------------------
-------------+
| id | email             | password
            |
+----+-------------------+---------------------
-------------+
| 1  | admin@defcon.demo | 32250170a0dca92d53ec
9624f336ca24 |
| 2  | user2@defcon.demo | acdba43a76b67ec114df
9ae9ad4b3128 |
| 3  | user3@defcon.demo | 22b7a7c3d73d88050722
b3eeb102ee45 |
| 4  | john@defcon.demo  | 558813098d0d0df9a9d1
9aaed8df75fd |
| 5  | user5@defcon.demo | fac2ee614377daff3d0e
94e699e85a8c |
| 6  | user6@defcon.demo | e8229677c9cfafb70e09
c11d33896a7e |
+----+-------------------+---------------------
-------------+

[17:24:38] [INFO] table 'defcondemo.users' dump
ed to CSV file '/root/.local/share/sqlmap/outpu
t/vulnerableapp.ml/dump/defcondemo/users.csv'
[17:24:38] [INFO] fetching columns for table 'm
essages' in database 'defcondemo'
[17:24:38] [INFO] retrieved: 5
[17:24:38] [INFO] retrieved: id
[17:24:38] [INFO] retrieved: m
```



```
do you want to crack them via a dictionary-based attack? [Y/n/q] n
Database: defcondemo
Table: users
[6 entries]
+----+-------------------+----------------------------------+
| id | email             | password                         |
+----+-------------------+----------------------------------+
| 1  | admin@defcon.demo | 32250170a0dca92d53ec9624f336ca24 |
| 2  | user2@defcon.demo | acdba43a76b67ec114df9ae9ad4b3128 |
| 3  | user3@defcon.demo | 22b7a7c3d73d88050722b3eeb102ee45 |
| 4  | john@defcon.demo  | 558813098d0d0df9a9d19aaed8df75fd |
| 5  | user5@defcon.demo | fac2ee614377daff3d0e94e699e85a8c |
| 6  | user6@defcon.demo | e8229677c9cfafb70e09c11d33896a7e |
+----+-------------------+----------------------------------+

[17:27:33] [INFO] table 'defcondemo.users' dumped to CSV file '/root/.local/share/sqlmap/output/localhost
/dump/defcondemo/users.csv'
[17:27:33] [INFO] fetching columns for table 'messages' in database 'defcondemo'
[17:27:33] [INFO] retrieved: 5
[17:27:33] [INFO] retrieved: id
[17:27:33] [INFO] retrieved: message
[17:27:34] [INFO] retrieved: subject
[17:27:34] [INFO] retrieved: s
```

Fixing this problem

- For charset method just shuffle the charset

  ~~abcdefghijklmnopqrstuvwxyz0123456789~~

  xd0tc7ouysq53lek9inabrzw2j84mh16vgfp

- For the binary search add some extra steps which your exploit won’t take them into account

Conclusion

- The way we write Blind Injection exploits might be predictable
- When we want to optimize the Blind Injection attacks, we must consider inserting random steps through the optimization algorithm, be it binary search, bit shifting or any other type of optimization
- Defensive technique? Payload less detection! The payload complexity doesn’t matter!

Thanks

Do you have any questions?

ionut.cernica@gmail.com

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, infographics & images by **Freepik**