

Linnéuniversitetet

INTRODUCTION TO PRACTICAL WORK

2DV702 – Internet Security

Paolo Molinaro

pm222py@student.lnu.se



FACTS AND LOGISTICS

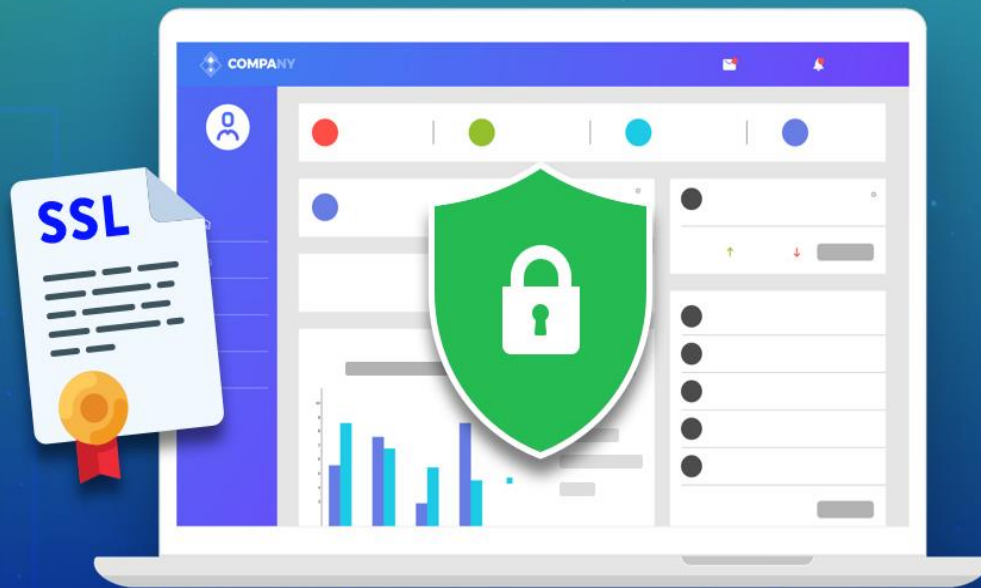
- Only one TA (it's me).
- Four assignments including theoretical and practical tasks.
- Grading scheme: Pass (A-E) || Fail (F).
- MyMoodle as e-learning platform.
- **Slack** as a main channel of communication.
- Plagiarism is forbidden! Reports will be checked!

WHAT IS EXPECTED FROM YOU?

- A report in PDF - use the “Report Template” as basis.
- Follow our naming convention, “Firstname_Lastname_Assign#”
E.g., “Harald_Gormsson_Assign1.pdf”.
- Well structured report, easy to follow, good language.
- Referencing is a must.
- Meet the deadlines.
- Refer to “Introduction to Practical Work”, section “IV. Assignment Rules”.

WHAT IS EXPECTED FROM US?

- Tutoring sessions: (check TimeEdit to be sure)
 - Tuesday 15:15 – 16:00, 16:15 – 17:00
 - Thursday 13:15-14:00, 14:15-15:00
- There will be on campus tutoring sessions.
- Help with the tasks is given at tutoring sessions.
- Grades with feedback comments.



ASSIGNMENT 1

Certificates, Authentication,
Confidentiality and SSL/TLS

Paolo Molinaro
pm222py@student.lnu.se



TOOLS AND REQUIREMENTS

- A GNU/Linux system, e.g., Ubuntu (recommended).
- How to use? Hypervisor (VirtualBox, VMware), Live CD, local installation (Use WSL if possible).
- OpenSSL: Command-line tool.
- s_client: SSL/TLS client program (part of OpenSSL).
- Wireshark: Network protocol analyzer.
- Firefox web browser.
- Chromium web browser.
- 10-pages limit!

NAME

openssl - OpenSSL command line tool

SYNOPSIS

openssl command [command opts] [command args]

openssl [list-standard-commands | list-message-digest-commands | list-cipher-commands]

openssl no-XXX [arbitrary options]

DESCRIPTION

OpenSSL is a cryptography toolkit implementing the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1) network protocols and related cryptography standards required by them.

The **openssl** program is a command line tool for using the various cryptography functions of OpenSSL's **crypto** library from the shell. It can be used for

- o Creation of RSA, DH and DSA key parameters
- o Creation of X.509 certificates, CSRs and CRLs
- o Calculation of Message Digests
- o Encryption and Decryption with Ciphers
- o SSL/TLS Client and Server Tests
- o Handling of S/MIME signed or encrypted mail

COMMAND SUMMARY

The **openssl** program provides a rich variety of commands (command in the SYNOPSIS above), each of which often has a wealth of options and arguments (command opts and command args in the SYNOPSIS).

The pseudo-commands list-standard-commands, list-message-digest-commands, and list-cipher-commands output a list (one entry per line) of the names of all standard commands, message digest commands, or cipher commands, respectively, that are available in the present **openssl** utility.

The pseudo-command no-XXX tests whether a command of the specified name is available. If no command named XXX exists, it returns 0 (success) and prints no-XXX; otherwise it returns 1 and prints XXX. In both cases, the output goes to **stdout** and nothing is printed to **stderr**. Additional command line arguments are always ignored. Since for each cipher there is a command of the same name, this provides an easy way for shell scripts to test for the availability of ciphers in the **openssl** program. (no-XXX is

STRATEGY

- Work with terminal => Bash
- Use embedded help functionality
 - man <command>
 - <command> --help
 - <command> -h
- Search the web.
- Extra material, online tutorials, etc ...
- 'tldr' tool is excellent for quick references

POST-MORTEM TRACE ANALYSIS

- Wireshark for network traffic traces
- Very capable tool => Learn how it works!
- Trace collection workflow:
- Capable of decrypting SSL/TLS traffic
- Capable of decoding compressed content

[Wireshark Masterclass by Chris Greer](#)

The screenshot shows the Wireshark interface with a network traffic trace. The main pane displays a list of packets with columns for No., Time, Source, Destination, Protocol, Length, and Info. Packet 348 is highlighted, showing a DNS Standard query for cdn-0.nflximg.com. The packet details pane below shows the structure of the DNS response, including the transaction ID (0x2188), flags, and the query details. The packet bytes pane at the bottom shows the raw data of the DNS response, with the transaction ID 2188 highlighted in blue.

No.	Time	Source	Destination	Protocol	Length	Info
343	65.142415	192.168.0.21	174.129.249.228	TCP	66	40555 → 80 [ACK] Seq=1 Ack=1 Win=5888 Len=0 TSva
344	65.142715	192.168.0.21	174.129.249.228	HTTP	253	GET /clients/netflix/flash/application.swf?flash
345	65.230738	174.129.249.228	192.168.0.21	TCP	66	80 → 40555 [ACK] Seq=1 Ack=188 Win=6864 Len=0 TS
346	65.240742	174.129.249.228	192.168.0.21	HTTP	828	HTTP/1.1 302 Moved Temporarily
347	65.241592	192.168.0.21	174.129.249.228	TCP	66	40555 → 80 [ACK] Seq=188 Ack=763 Win=7424 Len=0
348	65.242532	192.168.0.21	192.168.0.1	DNS	77	Standard query 0x2188 A cdn-0.nflximg.com
349	65.276870	192.168.0.1	192.168.0.21	DNS	489	Standard query response 0x2188 A cdn-0.nflximg.c
350	65.277992	192.168.0.21	63.80.242.48	TCP	74	37063 → 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 S
351	65.297757	63.80.242.48	192.168.0.21	TCP	74	80 → 37063 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0
352	65.298396	192.168.0.21	63.80.242.48	TCP	66	37063 → 80 [ACK] Seq=1 Ack=1 Win=5888 Len=0 TSva
353	65.298687	192.168.0.21	63.80.242.48	HTTP	153	GET /us/nrd/clients/flash/814540.bun HTTP/1.1
354	65.318730	63.80.242.48	192.168.0.21	TCP	66	80 → 37063 [ACK] Seq=1 Ack=88 Win=5792 Len=0 TSv
355	65.321733	63.80.242.48	192.168.0.21	TCP	1514	[TCP segment of a reassembled PDU]

> Frame 349: 489 bytes on wire (3912 bits), 489 bytes captured (3912 bits)
Ethernet II, Src: Globalsc_00:3b:0a (f0:ad:4e:00:3b:0a), Dst: Vizio_14:8a:e1 (00:19:9d:14:8a:e1)
Internet Protocol Version 4, Src: 192.168.0.1, Dst: 192.168.0.21
> User Datagram Protocol, Src Port: 53 (53), Dst Port: 34036 (34036)
▼ Domain Name System (response)
 [Request In: 348]
 [Time: 0.034338000 seconds]
 Transaction ID: 0x2188
 > Flags: 0x8180 Standard query response, No error
 Questions: 1
 Answer RRs: 4
 Authority RRs: 9
 Additional RRs: 9
 ▼ Queries
 > cdn-0.nflximg.com: type A, class IN
 > Answers
 > Authoritative nameservers

```
0020 00 15 00 35 84 f4 01 c7 83 3f 21 88 81 80 00 01 ...5.... ?!....
0030 00 04 00 09 00 09 05 63 64 6e 2d 30 07 6e 66 6c .....c dn-0.nfl
0040 78 69 6d 67 03 63 6f 6d 00 00 01 00 01 c0 0c 00 ximg.com .....
0050 05 00 01 00 00 05 29 00 22 06 69 6d 61 67 65 73 .....). ".images
0060 07 6e 65 74 66 6c 69 78 03 63 6f 6d 09 65 64 67 .netflix .com.edg
0070 65 73 75 69 74 65 03 6e 65 74 00 c0 2f 00 05 00 esuite.n et.../...
```

Identification of transaction (dns.id), 2 bytes | Packets: 10299 · Displayed: 10299 (100.0%)

USEFUL LINKS



- [OpenSSL](#)
- [s_client](#)
- [Wireshark](#)
- [Wireshark Masterclass by Chris Greer](#)
- [VirtualBox](#), [VMWare](#), [UTM \(Mac\)](#)
- [Firefox](#), [Chromium](#)
- [WSL](#)