

COMPUTER NETWORKS II PROJECT

GIOVANNI ZANIN

CONTENTS

1	Traffic analysis using Wireshark	2
2	MITM against smartphone browser	4

LIST OF FIGURES

Figure 1	DNS traffic	6
Figure 2	IP configuration of the PC	6
Figure 3	IP configuration of the smartphone	7
Figure 4	DNS round robin	7
Figure 5	TCP conversation	8
Figure 6	Sequence numbers' evolution from client to server	8
Figure 7	Sequence numbers' evolution from server to client	8
Figure 8	Segment length	9
Figure 9	ACK	9
Figure 10	WIN-SIZE	9
Figure 11	Varied WIN-SIZE	9
Figure 12	Segment length	10
Figure 13	In flight bytes	10
Figure 14	Encrypted payload	10
Figure 15	Proxy server configuration	11
Figure 16	Proxy configuration on client	11
Figure 17	Certificate pinning	12
Figure 18	Network configuration	12
Figure 19	Adding DNS zone	13
Figure 20	Web server configuration	13
Figure 21	HTTP connection to fake website	13
Figure 22	Fake website	14
Figure 23	HTTP warnings	15
Figure 24	HTTPS connection to fake website	15
Figure 25	HTTPS warnings	16
Figure 26	Real website	17
Figure 27	Correct DNS response	17
Figure 28	TTL of fake RR	18
Figure 29	TTL of real RR	18

1 TRAFFIC ANALYSIS USING WIRESHARK

In this first section I analyzed the traffic generated during the authentication on the SIAE+ smartphone app, in terms of TCP and other web protocols communications. SIAE+ is an app where authors and editors associated with SIAE can visualize, modify and remove their deposits, or the works that SIAE protects. Because of the economic consequences of those actions, the service is available only after authentication. At login time, the user can choose between biometric authentication (fingerprint reader) or by 2FA with password and OTP sent via SMS (the latter is the modality used during this analysis). All the analyzed traffic have been captured running Wireshark on the PC while it was acting as hotspot for the smartphone. The capture was active during all the authentication operation, for a total of about 30 seconds during which almost 1000 packages were captured. Here are some observations.

DNS TRAFFIC Using the "dns" filter on Wireshark is possible to visualize 18 packages, out of the 1000 overall captured, that carry DNS messages (see figure 1 on page 6). These messages are both:

- DNS requests sent by the smartphone to the PC, which would forward them and get the responses;
- DNS responses obtained by the PC and forwarded to the smartphone.

The requests and the responses can be easily distinguished by checking the IP addresses of the source and of the destination of the packages on Wireshark and confronting them with the ones that the PC (figure 2 on page 6) and the smartphone (figure 3 on page 7) have in the network. Many of these 18 packages are caused by processes running on the smartphone that has nothing to do with the authentication on SIAE+ (see for example packages 322-323 in figure 1 on page 6, those contain DNS requests produced by BeReal, another app on the smartphone). The interesting packages for our activity are number 63-64. These packages contain respectively the DNS request of type A and the response for the name "pae.servizi.siae.it" (the name of the server to which the smartphone should connect). The response contains two IP values, in this way the client can try to connect to the second one if the first isn't available (in the examined case the connection was immediately successful). Different DNS requests of type "pae.servizi.siae.it A ?" might have responses with these two addresses inverted (see figure 4 on page 7), that's because the so called "DNS round robin", a practice useful for distributing the requests that a specific server receives.

TCP CONNECTION After the IP address is obtained the TCP connection between the smartphone and the server can be initialized. In figure 5 on page 8 the packages containing the first segments of the TCP conversation are visualized. Frames 65-67 carry the segments of the 3-way handshake between the smartphone and the "pae.servizi.siae.it" server. In these first segments no application data is transmitted, as proved by the field Len=0 for each of them. The most relevant information in the TCP headers contained in these messages include:

- the ports between which the connection is set.
- the communication by both ends of the initial sequence numbers, or rather the identifier of the TX-buffer's byte from which each node would start transmitting application data. These communication takes place in the first two segments, the ones with the SYN flag. In the image only the sequence numbers relative to this specific connection are visible, not the raw ones (the connection is just began so the relative sequence numbers all equal 0).
- the acknowledge numbers, that the two ends use to communicate the other's TX-buffer's byte they expect to receive next. This happens in the second and

the third segment, which contain the ACK flag. No data has been transmitted yet so the relative acknowledge numbers equal 1.

- the window sizes, or rather the free space in the RX-buffers (at the beginning they are completely empty). These values might be communicated with scaling factors and are useful for the congestion control.

TCP CONVERSATION After frame 67, application data starts being transmitted. The size of the TCP segments never exceeds the MSS=1440 bytes accorded by the two nodes during the handshake. The values in the "length" field in figure 5 on page 8 are the lengths of the IP packets, so include IP headers, TCP headers and TCP payloads, that's why some of these values are larger than 1440, which is an upper bound for the only TCP payload. While the data are transmitted is possible to appreciate the evolution of the SEQ numbers, ACK numbers and window sizes that each of the two nodes communicates to the other.

Figures 6 on page 8 and 7 on page 8 represent in the time domain the SEQ number of the last TCP segment sent, respectively from the client to the server and vice versa. In the central part the SEQ numbers seem stable as no data is transmitted in neither direction (this is reasonably the period in which the SMS app was open to read the generated OTP). Neglecting this central section the shapes of the graphs are the typical ones of slow start transmission phase.

Figure 8 on page 9 shows the length of the TCP payload in package 69. This is the first non-empty TCP segment sent by the server to the client and is completely acknowledged by the latter in package 73 (figure 9 on page 9), that contains an empty TCP segment from the smartphone to acknowledge the server of the successful reception. This modality is used for many other segments further in this direction: one acknowledgment segment for each received segment.

Comparing figures 10 on page 9 and 11 on page 9 is possible to observe the variation of the window size communicated by the server to the smartphone, in two different packages. In package 80 the communicated window size is 342 bytes smaller than the one in package 72, the exact same dimension of the TCP payload in package 78 (figure 12 on page 10), the only package between 72 and 80 with TCP data transmitted from the client. This means that at the time package 80 was sent, the data transmitted in package 78 (sent approximately 24 ms earlier), were already received by the server but hadn't been processed yet (no "receive()" operation had removed them from the RX-buffer yet). The time difference between packages 78 and 80 (24 ms) can such be a good approximation of the RTT from the client to the server (at that specific time of the authentication operation).

In both directions the communicated window size is always larger than 30 kB, while the values of "in flight bytes" never exceed 4 kB (figure 13 on page 10). It is therefore possible to assume that congestion control is never active during the operation.

SIAE+ OVER HTTPS Application data is encrypted during the conversation (as shown in figure 14 on page 10 which takes as example package 84), this means the server uses HTTPS. Even looking at figure 5 on page 8 is possible to observe that all packages use TLSv1.2 as protocol, except for those that has the field Len=0 in the TCP header (there is nothing to encrypt so the protocol shown by Wireshark is TCP). Another proof of the usage of HTTPS could be obtained simply looking at the port used by the server to communicate with the client: 443.

CERTIFICATE PINNING SIAE+ uses certificate pinning. This was assumed after initializing a proxy server on BURP (figure 15 on page 11), configuring the smartphone connection to use that server (figure 16 on page 11), installing BURP certificate in smartphone's trust set and noticing that the app doesn't work (it only accepts SIAE+ pinned certificates, figure 17 on page 12).

2 MITM AGAINST SMARTPHONE BROWSER

In this section a form of MITM attack against a smartphone was implemented. The smartphone has been connected to a PC's hotspot network and during a web research it was directed to a wrong website, thanks to false DNS responses.

PC CONFIGURATION A PC with a Windows 10 operative system was configured as hotspot and a DNS server running on the PC was set to be the default DNS for this network (figure 18 on page 12 shows the IP address of the favourite DNS server set to be the IP address of the PC on the local network interface, the one shown in figure 2 on page 6). All the DNS requests from devices in the network would have been received by the server on the PC which would answer them by checking in its competence's zones and possibly forwarding them to other DNS servers. In this way some specific DNS requests could receive responses that directed the navigation to a "fraudulent" web server, which was running on the PC too.

DNS SERVER CONFIGURATION Technitium DNS Server v11.0.3 was used as server running on the PC. A zone called "unicredit.it" was added to the zones of the server, containing a RR type A with the name "www.unicredit.it" associated to the IP address of the PC on the hotspot network interface (see figure 19 on page 13). In this way any device of the network would try to connect to a web server on the PC after resolving that name.

WEB SERVER CONFIGURATION To have a web server running on the PC, Abyss Web Server X1 v2.16.4 was used. The server was configured to listen on all PC's internet interfaces both for HTTP and HTTPS connections (respectively on port 80 and 443, see figure 20 on page 13). For HTTPS connections an auto-signed certificate was created.

SMARTPHONE ACTIVITY Opera was used as browser for the smartphone. At first a connection was instantiated by inserting the URL "http://www.unicredit.it". As shown in figure 21 on page 13 the DNS response of package 3 causes two TCP connections to the web server on the PC (the three-way handshakes are contained in packages 4-9). The result shown on the screen is a simple HTML page prepared and indicated as content of the web server's service (figure 22 on page 14). The page is open but the user is warned by two messages, shown in figure 23 on page 15: the website isn't verified and the communication isn't encrypted (that's because the connection is not on HTTPS). Then a connection is instantiated inserting the URL "https://www.unicredit.it" (figure 24 on page 15 shows the beginning of the connection on Wireshark). The certificate for this connection is self signed by the server (it isn't in the smartphone trust set), so the browser asks the user if he wants to trust the connection or not. After accepting and "taking the risk" the page shown is the same of the HTTP connection, but this time the server, still unverified, offers and encrypted conversation (figure 25 on page 16). After disabling the zone "unicredit.it" the connections inserting the previous URLs would be instantiated with the real unicredit.it server, as shown in figure 26 on page 17 (figure 27 on page 17 shows the DNS responses that directed the smartphone to the true website). Connecting to the real web server after connecting to the fake one can be done immediately, but the opposite can't. That's because the RR added in the Technitium DNS server has TTL=0 (see figure 28 on page 18, this means that it doesn't remain in smartphone's cache memory), while the RR record with the address of the real web server has TTL>0 (the first A response has a TTL of nearly one hour, as shown in figure 29 on page 18). To connect to the fake website after connecting to the real one is then necessary to wait for the TTL to elapse or to clean the cache of the smartphone.

DIFFICULTIES As discussed during the demo, analyzing on Wireshark the traffic generated during the http connection to the fake website, two HTTP "GET" requests from the client received "404 Not Found" as response. Although the whole operation worked as desired (the right page was opened), this is certainly an unpleasant behaviour by the web server.

While setting up the scenario I've run into many other small difficulties. In many cases to solve them it was very helpful to try the MITM attack with the Wireshark capture active, in this way it was easy to understand whether the problem was in the configuration of the hotspot, of the DNS server or of the web server. Sometimes the problem was caused by the browser too. In the simplest case it was necessary to clean the cache memory, however, using Chrome I found out the operation worked correctly for DNS names that didn't correspond to real websites, while the URL "http://www.unicredit.it" led to the real website (although the Wireshark capture showed that the DNS server responded with the IP address of the PC). Using Opera this difficulty wasn't encountered, so it was probably caused by a specific Chrome functionality.

REFERENCES

- [1] T_EX Templates, <http://www.latextemplates.com>.
- [2] wireshark.org, <https://osqa-ask.wireshark.org/>.
- [3] Abyss Web Server For Windows User's Guide, <https://aprelum.com/data/doc/2/abyssws-win-doc-html/index.html>.
- [4] technitium.com, <https://technitium.com/dns/>

No.	Time	Source	Destination	Protocol	Length	Info
5	1.917994	192.168.137.135	192.168.137.1	DNS	93	Standard query 0x7a21 A firebaselogging-pa.googleapis.com
6	1.933789	192.168.137.1	192.168.137.135	DNS	189	Standard query response 0x7a21 A firebaselogging-pa.googleapis.com A 142.251.209.42 A 142.250.184.74 A 142.250.184...
8	1.976436	192.168.137.135	192.168.137.1	DNS	97	Standard query 0x7252 A firebaseinappmessaging.googleapis.com
13	1.991613	192.168.137.1	192.168.137.135	DNS	209	Standard query response 0x7252 A firebaseinappmessaging.googleapis.com A 142.251.209.42 A 216.58.209.42 A 142.250...
63	3.214674	192.168.137.135	192.168.137.1	DNS	79	Standard query 0xe085 A pae.servizi.siae.it
64	3.256251	192.168.137.1	192.168.137.135	DNS	111	Standard query response 0xe085 A pae.servizi.siae.it A 2.228.32.120 A 151.22.5.120
77	3.313842	192.168.137.135	192.168.137.1	DNS	75	Standard query 0x6a10 A api.ionicjs.com
79	3.332810	192.168.137.1	192.168.137.135	DNS	177	Standard query response 0x6a10 A api.ionicjs.com CNAME public-cloud-staging-1610158363-us-west-2.elb.amazonaws.com.
82	3.536804	192.168.137.135	192.168.137.1	DNS	95	Standard query 0xbcd0 A firebaseremoteconfig.googleapis.com
85	3.555818	192.168.137.1	192.168.137.135	DNS	207	Standard query response 0xbcd0 A firebaseremoteconfig.googleapis.com A 142.250.180.170 A 142.251.209.10 A 142.251...
320	19.244762	192.168.137.135	192.168.137.1	DNS	74	Standard query 0x5b81 A 0.pool.ntp.org
321	19.262856	192.168.137.1	192.168.137.135	DNS	138	Standard query response 0x5b81 A 0.pool.ntp.org A 217.61.62.224 A 212.6.50.243 A 95.110.254.234 A 85.199.214.99
322	19.271114	192.168.137.135	192.168.137.1	DNS	77	Standard query 0xbdd3 A mobile.bereal.com
323	19.271114	192.168.137.135	192.168.137.1	DNS	78	Standard query 0xed7a A cdn.bereal.network
325	19.287329	192.168.137.1	192.168.137.135	DNS	93	Standard query response 0xbdd3 A mobile.bereal.com A 35.239.22.149
326	19.287590	192.168.137.1	192.168.137.135	DNS	201	Standard query response 0xed7a A cdn.bereal.network CNAME 2-01-4c30-0003.cdx.cedexis.net CNAME fp2EE8.wpc.17A0E9.m...
861	30.456300	192.168.137.135	192.168.137.1	DNS	83	Standard query 0x4225 A android.apis.google.com
868	30.474766	192.168.137.1	192.168.137.135	DNS	123	Standard query response 0x4225 A android.apis.google.com CNAME clients.l.google.com A 142.251.209.14

Figure 1: Captured packages containing DNS messages

[Back to first reference](#)

```

Microsoft Windows [Versione 10.0.19044.2220]
(c) Microsoft Corporation. Tutti i diritti sono riservati.

C:\Users\Utente>ipconfig

ip* non è riconosciuto come comando interno o esterno,
un programma eseguibile o un file batch.

C:\Users\Utente>ipconfig

Configurazione IP di Windows

Scheda LAN wireless Connessione alla rete locale (LAN)* 8:

Stato supporto. . . . . : Supporto disconnesso
Suffisso DNS specifico per connessione:

Scheda LAN wireless Connessione alla rete locale (LAN)* 10:

Suffisso DNS specifico per connessione:
Indirizzo IPv6 locale rispetto al collegamento . : fe80::4014:6344:b39a:914eX38
Indirizzo IPv4. . . . . : 192.168.137.1
Subnet mask . . . . . : 255.255.255.0
Gateway predefinito . . . . . :

Scheda LAN wireless Wi-Fi:

Suffisso DNS specifico per connessione: homenet.telecomitalia.it
Indirizzo IPv6 locale rispetto al collegamento . : fe80::b1a0:36ef:4bc8:9452X6
Indirizzo IPv4. . . . . : 192.168.1.17
Subnet mask . . . . . : 255.255.255.0
Gateway predefinito . . . . . : 192.168.1.1

Scheda Ethernet Connessione di rete Bluetooth:

Stato supporto. . . . . : Supporto disconnesso
Suffisso DNS specifico per connessione:

C:\Users\Utente>

```

Figure 2: IP addresses of the PC on different interfaces

[Back to first reference](#)

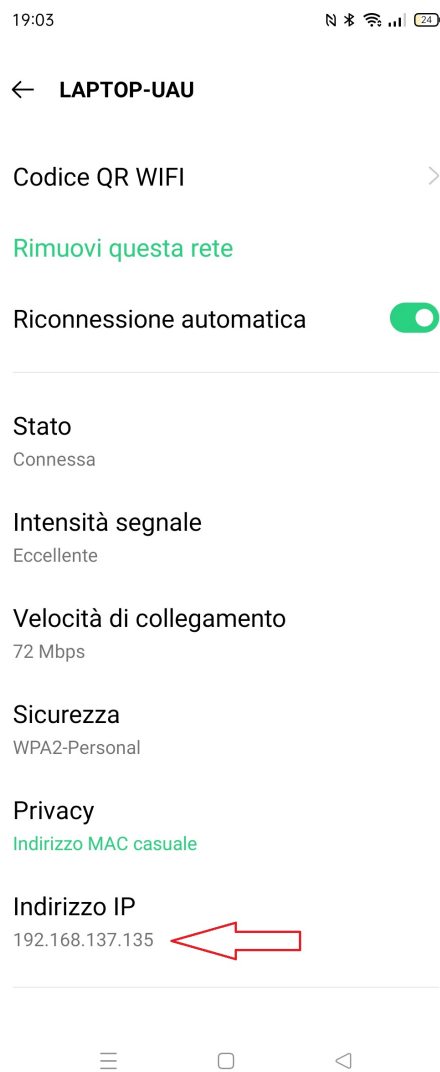


Figure 3: Information about smartphone’s connection to the network
[Back to text](#)

No.	Time	Source	Destination	Protocol	Length	Info
32	7.494624	192.168.137.60	192.168.137.1	DNS	79	Standard query 0x4fda A pae.servizi.siae.it
33	7.594793	192.168.137.1	192.168.137.60	DNS	111	Standard query response 0x4fda A pae.servizi.siae.it A 151.22.5.120 A 2.228.32.120
50	7.802211	192.168.137.60	192.168.137.1	DNS	75	Standard query 0xd2e7 A api.ionicjs.com
55	8.186948	192.168.137.1	192.168.137.60	DNS	177	Standard query response 0xd2e7 A api.ionicjs.com CNAME public.s3.amazonaws.com
114	11.202644	192.168.137.60	192.168.137.1	DNS	82	Standard query 0x35b2 A 0.datadog.pool.ntp.org
115	11.236187	192.168.137.60	192.168.137.1	DNS	74	Standard query 0x7c7c A 0.pool.ntp.org
116	11.957086	192.168.137.1	192.168.137.60	DNS	138	Standard query response 0x7c7c A 0.pool.ntp.org A 212.45.144.206 A 93.94.88.51 A 212.6.50.243 A 31.14.133.122
121	11.978247	192.168.137.1	192.168.137.60	DNS	146	Standard query response 0x35b2 A 0.datadog.pool.ntp.org A 93.94.88.51 A 212.6.50.243 A 212.45.144.206 A 31.14.133...
225	16.680413	192.168.137.60	192.168.137.1	DNS	86	Standard query 0xd871 A moa-upload-eu.allamos.com
226	16.969560	192.168.137.1	192.168.137.60	DNS	293	Standard query response 0xd871 A moa-upload-eu.allamos.com CNAME eu-lagrange-eu.allamos-pubgw-1850928845.eu-west...

Figure 4: DNS response in which the server’s IP addresses are inverted
[Back to text](#)

No.	Time	Source	Destination	Protocol	Length	Info
65	3.259953	192.168.137.135	2.228.32.120	TCP	74	36152 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM TSval=57396687 TSecr=0 WS=256
66	3.283688	2.228.32.120	192.168.137.135	TCP	58	443 → 36152 [SYN, ACK] Seq=0 Ack=1 Win=6190 Len=0 MSS=1440
67	3.285477	192.168.137.135	2.228.32.120	TCP	54	36152 → 443 [ACK] Seq=1 Ack=1 Win=65535 Len=0
68	3.286284	192.168.137.135	2.228.32.120	TLSv1.2	571	Client Hello
69	3.310333	2.228.32.120	192.168.137.135	TLSv1.2	150	Server Hello
70	3.310397	2.228.32.120	192.168.137.135	TLSv1.2	1404	
71	3.310451	2.228.32.120	192.168.137.135	TLSv1.2	1494	Ignored Unknown Record
72	3.310479	2.228.32.120	192.168.137.135	TLSv1.2	202	Ignored Unknown Record
73	3.312263	192.168.137.135	2.228.32.120	TCP	54	36152 → 443 [ACK] Seq=518 Ack=97 Win=65535 Len=0
74	3.312263	192.168.137.135	2.228.32.120	TCP	54	36152 → 443 [ACK] Seq=518 Ack=1337 Win=65535 Len=0
75	3.312353	192.168.137.135	2.228.32.120	TCP	54	36152 → 443 [ACK] Seq=518 Ack=2977 Win=65535 Len=0
76	3.312353	192.168.137.135	2.228.32.120	TCP	54	36152 → 443 [ACK] Seq=518 Ack=3125 Win=65535 Len=0
78	3.327386	192.168.137.135	2.228.32.120	TLSv1.2	396	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
80	3.351181	2.228.32.120	192.168.137.135	TCP	54	443 → 36152 [ACK] Seq=3125 Ack=868 Win=34312 Len=0
81	3.357335	2.228.32.120	192.168.137.135	TLSv1.2	129	Change Cipher Spec, Encrypted Handshake Message
84	3.540769	192.168.137.135	2.228.32.120	TLSv1.2	763	Application Data
91	3.598985	2.228.32.120	192.168.137.135	TLSv1.2	779	Application Data
93	3.606544	192.168.137.135	2.228.32.120	TLSv1.2	731	Application Data
102	3.638908	2.228.32.120	192.168.137.135	TLSv1.2	923	Application Data
109	3.852103	192.168.137.135	2.228.32.120	TCP	54	36152 → 443 [ACK] Seq=2246 Ack=4794 Win=65535 Len=0
188	5.870419	192.168.137.135	2.228.32.120	TLSv1.2	747	Application Data
189	5.900239	2.228.32.120	192.168.137.135	TLSv1.2	795	Application Data
190	5.916282	192.168.137.135	2.228.32.120	TCP	54	36152 → 443 [ACK] Seq=2939 Ack=5535 Win=65535 Len=0
191	5.937975	192.168.137.135	2.228.32.120	TLSv1.2	859	Application Data
194	6.062638	2.228.32.120	192.168.137.135	TCP	54	443 → 36152 [ACK] Seq=5535 Ack=3744 Win=40153 Len=0
195	6.176533	2.228.32.120	192.168.137.135	TLSv1.2	1387	Application Data
196	6.380226	192.168.137.135	2.228.32.120	TCP	54	36152 → 443 [ACK] Seq=3744 Ack=6868 Win=65535 Len=0
197	6.415635	192.168.137.135	2.228.32.120	TLSv1.2	763	Application Data
201	6.449529	2.228.32.120	192.168.137.135	TLSv1.2	779	Application Data
202	6.451132	192.168.137.135	2.228.32.120	TCP	54	36152 → 443 [ACK] Seq=4453 Ack=7593 Win=65535 Len=0
203	6.454997	192.168.137.135	2.228.32.120	TLSv1.2	763	Application Data
214	6.482939	2.228.32.120	192.168.137.135	TLSv1.2	779	Application Data

Figure 5: TCP conversation between smartphone and SIAE+
[Back to first reference](#)

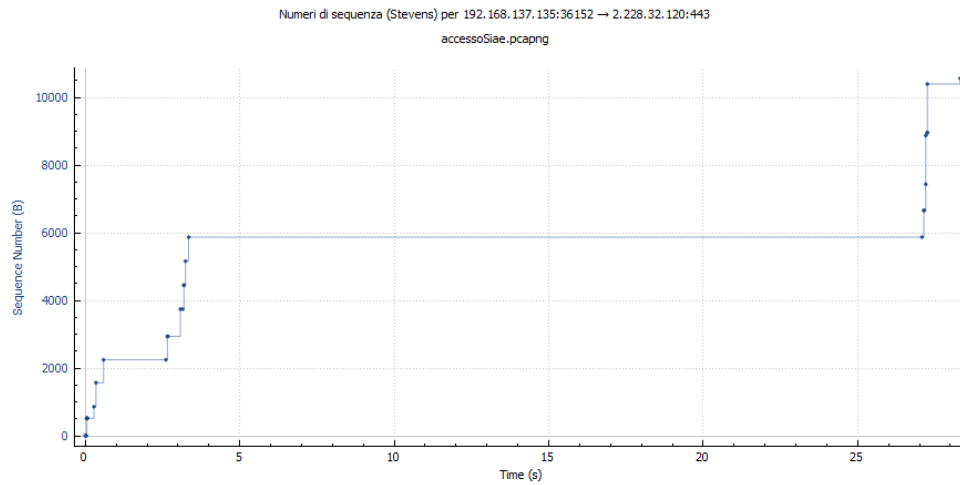


Figure 6: Sequence numbers' evolution from client to server
[Back to text](#)

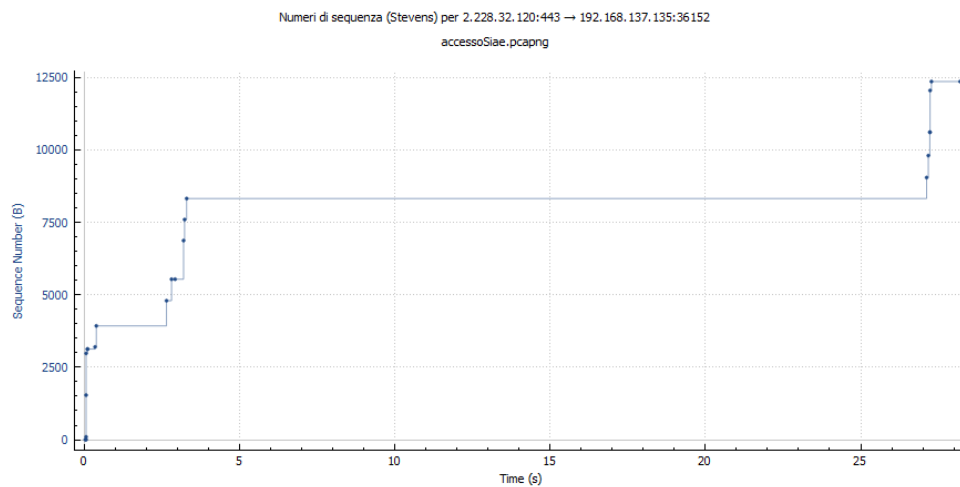
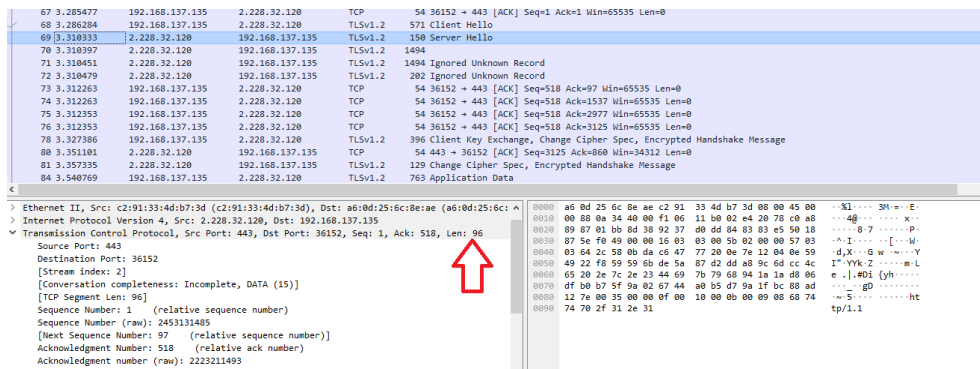
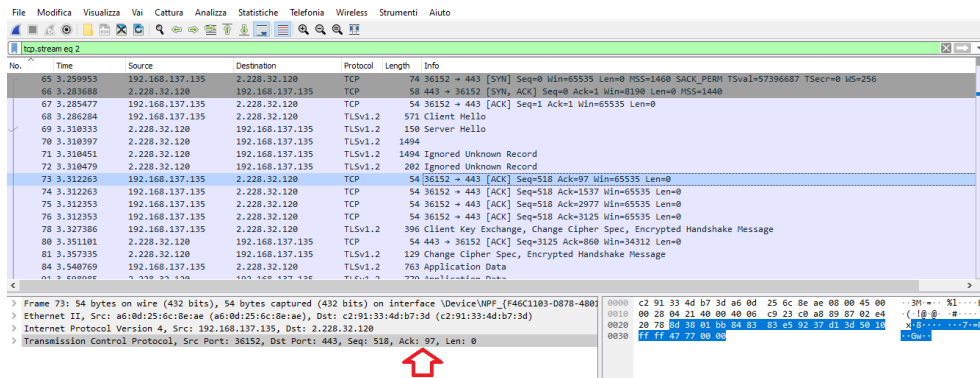


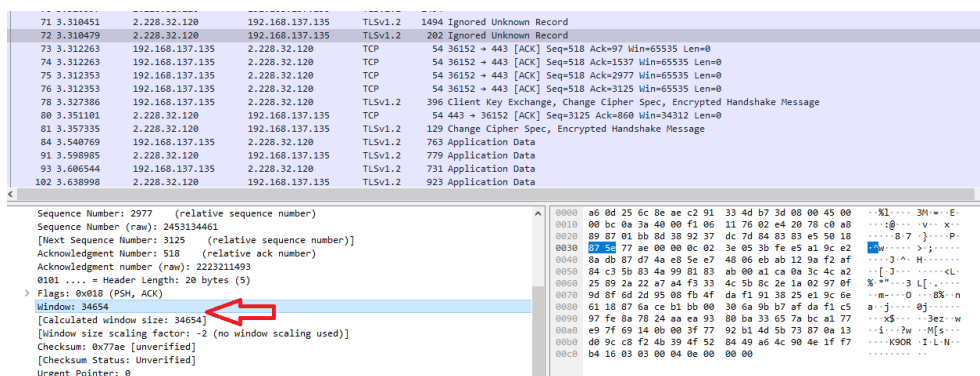
Figure 7: Sequence numbers' evolution from server to client
[Back to text](#)



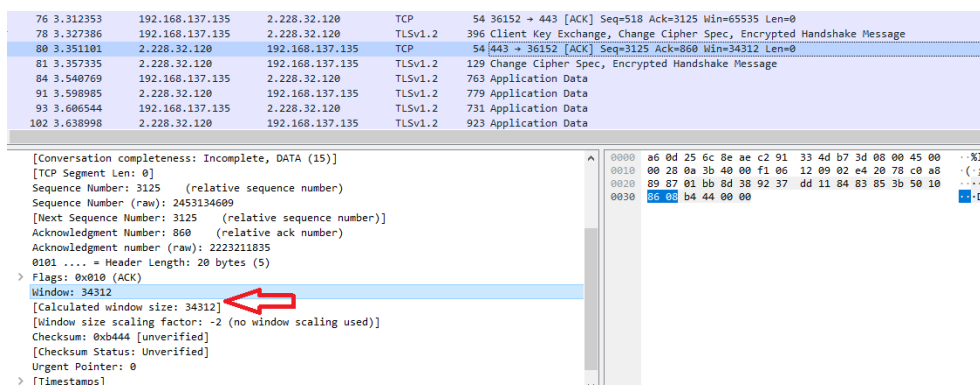
[Back to text](#)



[Back to text](#)



[Back to text](#)



[Back to text](#)

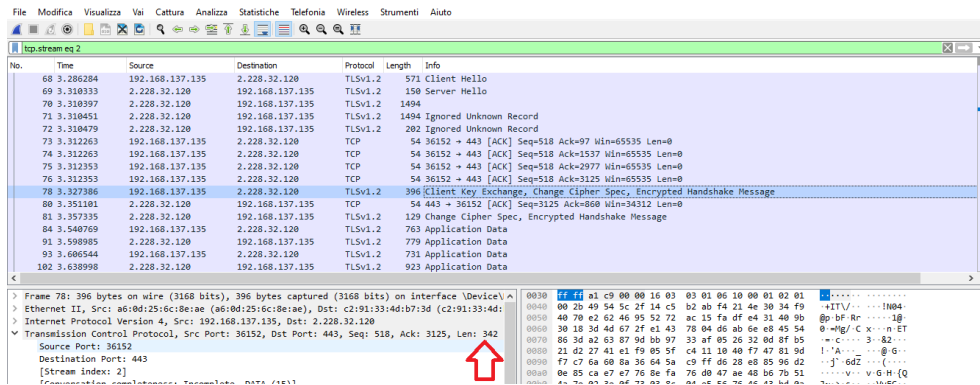


Figure 12: TCP payload length in package 78

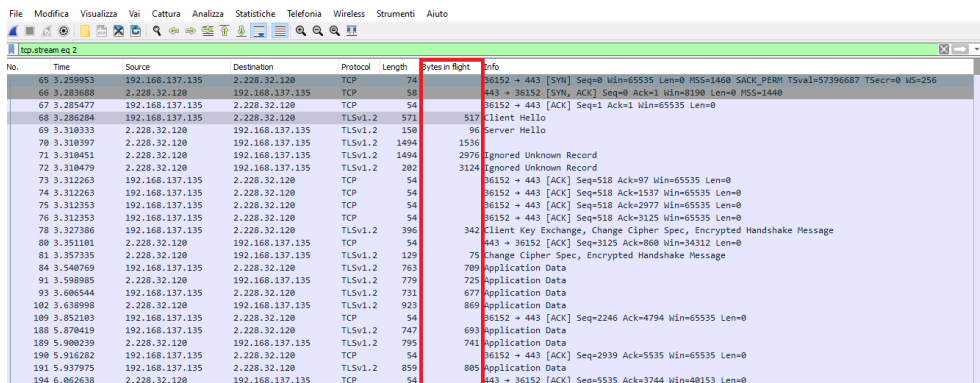
[Back to text](#)

Figure 13: In flight bytes

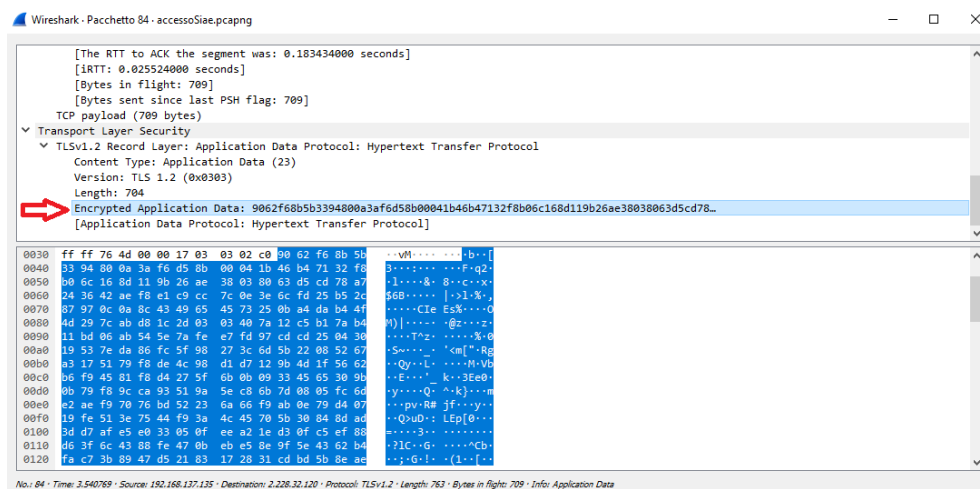
[Back to text](#)

Figure 14: Encrypted payload of package 84

[Back to text](#)

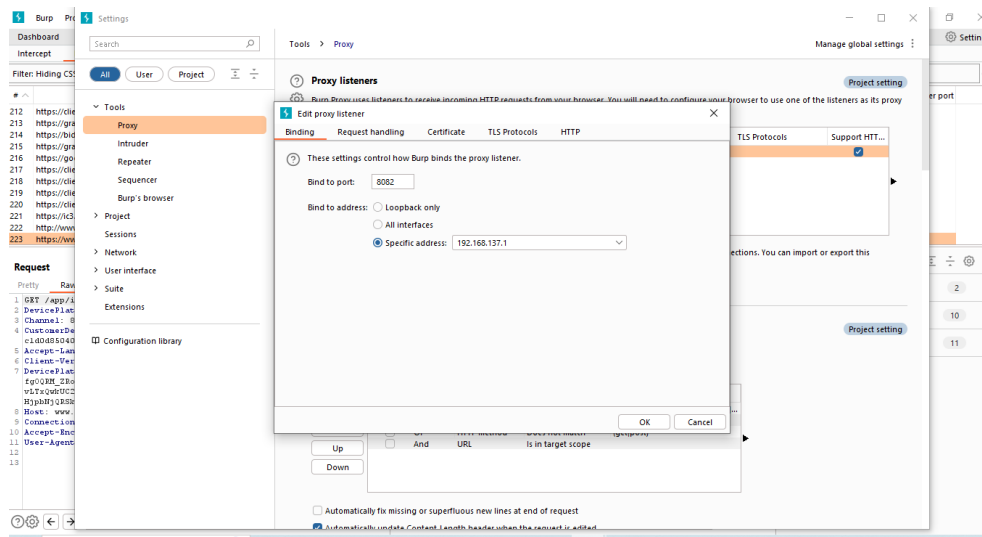


Figure 15: Burp proxy server configuration
[Back to text](#)

← LAPTOP-UAU

Sicurezza

WPA2-Personal

Privacy

Indirizzo MAC casuale

Indirizzo IP

192.168.137.60

Proxy

Manuale

Nome host

192.168.137.1

Porta

8082

Siti web inutilizzati

example.com, localhost

Impostazioni IP

DHCP



Figure 16: Proxy configuration on smartphone
[Back to text](#)

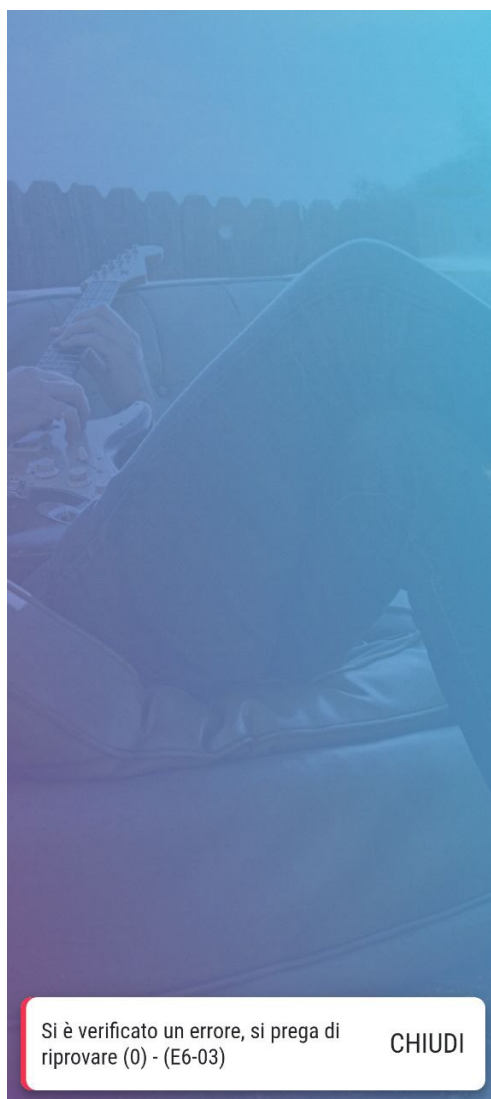


Figure 17: Proof that SIAE+ uses certificate pinning

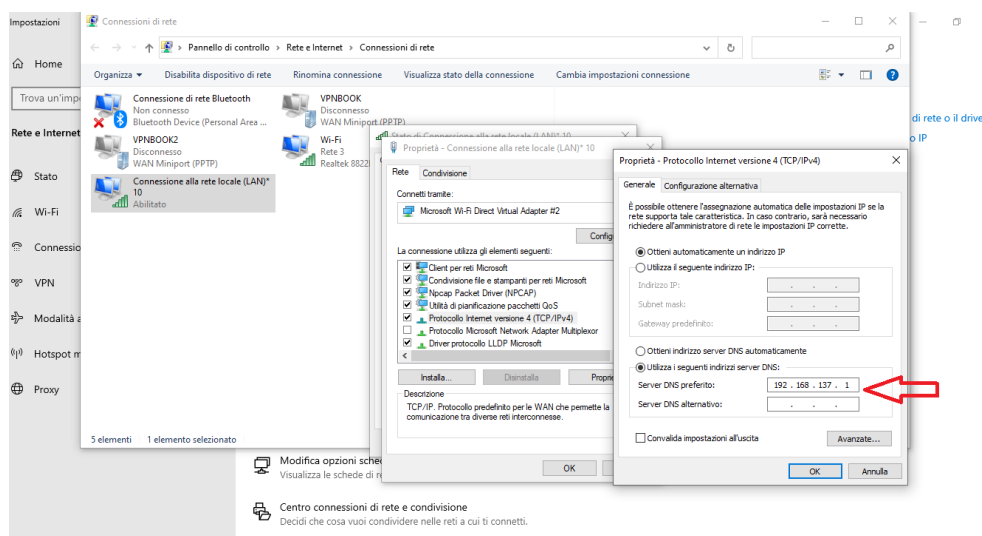

[Back to text](#)


Figure 18: Configuration of the local DNS server as favourite DNS server of the network

[Back to text](#)

unicredit.it 

Primary Enabled

Add Record Disable Zone Delete Zone Options Permissions DNSSEC

Page Number 1 Records Per Page 10 Go

1-3 (3) of 3 records (page 1 of 1)

#	Name	Type	TTL	Data
1	@	NS	3600	Name Server: laptop-uau7lrhu Last Used: 0001-01-01 00:00:00 (never)
2	@	SOA	900	Primary Name Server: laptop-uau7lrhu Responsible Person: hostadmin@unicredit.it Serial: 8 Refresh: 900 Retry: 300 Expire: 604800 Minimum: 900 Last Used: 2023-04-28 14:20:59 (4 hours ago)
3	www	A	0	192.168.137.1 Last Used: 2023-04-28 18:20:27 (a minute ago)

Figure 19: Zone uncredit.it on DNS server

[Back to text](#)

General

Abyss Web Server Console :: Hosts - Edit - Default Host On Port 80 + Default Secure Host On Port 443 :: General

Protocol : HTTP+HTTPS

HTTP Port : Default HTTP Port (80)

HTTPS Port : Default HTTPS Port (443)

Certificate Type : From the certificate store

Certificate : uncredit's certificate

Figure 20: Web server configuration

[Back to text](#)

Applica un filtro di visualizzazione: <Ctrl>-/

No.	Time	Source	Destination	Protocol	Length	Bytes in flight	Info
1	0.000000	192.168.137.60	192.168.137.1	DNS	76		Standard query 0x6e32 A www.unicredit.it
2	0.000994	192.168.137.60	192.168.137.1	DNS	72		Standard query response 0x6e32 A www.unicredit.it A 192.168.137.1
3	0.001002	192.168.137.1	192.168.137.60	DNS	92		
4	0.007342	192.168.137.60	192.168.137.1	TCP	74		46758 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM TSval=309914054 TSecr=0 WS=256
5	0.007673	192.168.137.1	192.168.137.60	TCP	66		80 → 46758 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
6	0.008550	192.168.137.60	192.168.137.1	TCP	74		46760 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM TSval=309914056 TSecr=0 WS=256
7	0.008882	192.168.137.1	192.168.137.60	TCP	66		80 → 46760 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
8	0.010168	192.168.137.60	192.168.137.1	TCP	54		46758 → 80 [ACK] Seq=1 Ack=1 Win=87888 Len=0
9	0.010636	192.168.137.60	192.168.137.1	TCP	54		46760 → 80 [ACK] Seq=1 Ack=1 Win=87888 Len=0
10	0.011869	192.168.137.60	192.168.137.1	HTTP	446	392	GET /css/bootstrap.min.css HTTP/1.1
11	0.011869	192.168.137.60	192.168.137.1	HTTP	439	385	GET /css/custom.css HTTP/1.1

Figure 21: HTTP connection to fake website

[Back to text](#)



Figure 22: Fake website
[Back to text](#)

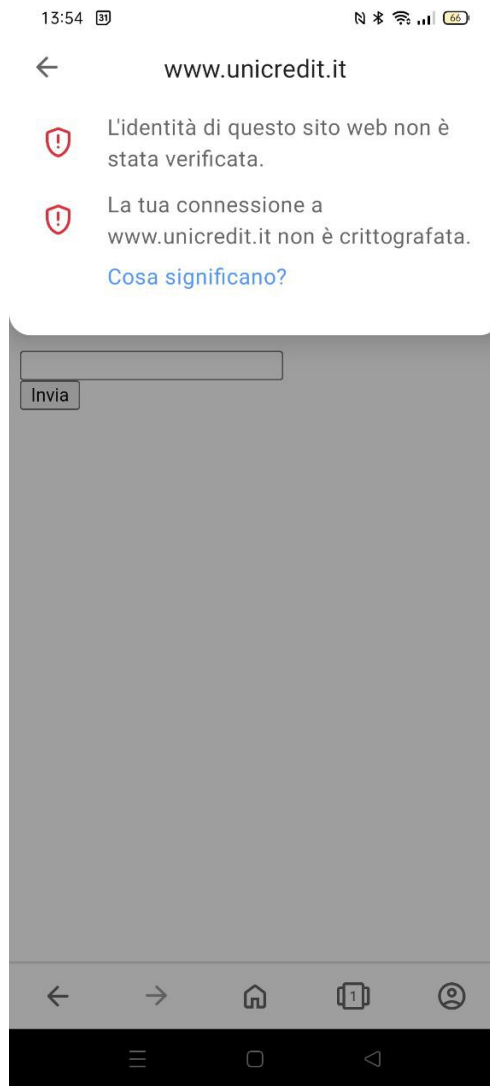


Figure 23: HTTP warnings

[Back to text](#)

io.	Time	Source	Destination	Protocol	Length	Bytes in flight	Info
1	0.000000	192.168.137.60	192.168.137.1	TCP	76	41234 → 443 [EST, ACK] Seq=1 Ack=1 Win=373 Len=0 TSval=309995611 TSecr=3277359224	
2	0.000000	192.168.137.60	192.168.137.1	DNS	76		Standard query 0x5101 A www.unicredit.it
3	0.001281	192.168.137.1	192.168.137.60	DNS	92		Standard query response 0x5101 A www.unicredit.it A 192.168.137.1
4	0.009078	192.168.137.60	192.168.137.1	TCP	74	41234 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM TSval=309995631 TSecr=0 WS=256	
5	0.009440	192.168.137.1	192.168.137.60	TCP	66	443 → 41234 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM	
6	0.014890	192.168.137.60	192.168.137.1	TCP	54	41234 → 443 [ACK] Seq=1 Ack=1 Win=87808 Len=0	
7	0.016498	192.168.137.60	192.168.137.1	TLV1.3	607	553 Client Hello	
8	0.018457	192.168.137.1	192.168.137.60	TLV1.3	288	234 Server Hello, Change Cipher Spec, Application Data, Application Data	
9	0.022814	192.168.137.60	192.168.137.1	TCP	54	41234 → 443 [ACK] Seq=554 Ack=235 Win=88832 Len=0	
10	0.023141	192.168.137.60	192.168.137.1	TLV1.3	84	30 Change Cipher Spec, Application Data	
11	0.023889	192.168.137.1	192.168.137.60	TCP	54	443 → 41234 [FIN, ACK] Seq=235 Ack=554 Win=1573120 Len=0	
12	0.025981	192.168.137.60	192.168.137.1	TCP	54	41234 → 443 [FIN, ACK] Seq=584 Ack=235 Win=88832 Len=0	
13	0.026166	192.168.137.1	192.168.137.60	TCP	54	443 → 41234 [ACK] Seq=236 Ack=585 Win=1573120 Len=0	
14	0.027028	192.168.137.60	192.168.137.1	TCP	54	41234 → 443 [ACK] Seq=585 Ack=236 Win=88832 Len=0	
15	0.032280	192.168.137.60	192.168.137.1	TCP	74	41236 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM TSval=309995654 TSecr=0 WS=256	
16	0.032649	192.168.137.1	192.168.137.60	TCP	66	443 → 41236 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM	
17	0.034834	192.168.137.60	192.168.137.1	TCP	54	41236 → 443 [ACK] Seq=1 Ack=1 Win=87808 Len=0	
18	0.035580	192.168.137.60	192.168.137.1	TLV1.3	607	553 Client Hello	
19	0.036861	192.168.137.1	192.168.137.60	TLV1.3	288	234 Server Hello, Change Cipher Spec, Application Data, Application Data	
20	0.039854	192.168.137.60	192.168.137.1	TCP	54	41236 → 443 [ACK] Seq=554 Ack=235 Win=88832 Len=0	
21	0.041436	192.168.137.60	192.168.137.1	TLV1.3	118	64 Change Cipher Spec, Application Data	
22	0.042280	192.168.137.1	192.168.137.60	TLV1.3	293	239 Application Data	
23	0.044470	192.168.137.60	192.168.137.1	TLV1.3	152	98 Application Data	
24	0.044470	192.168.137.60	192.168.137.1	TLV1.3	574	618 Application Data	
25	0.044710	192.168.137.1	192.168.137.60	TLV1.3	91	276 Application Data	
26	0.047031	192.168.137.60	192.168.137.1	TCP	54	41236 → 443 [ACK] Seq=1236 Ack=511 Win=89856 Len=0	
27	0.047031	192.168.137.60	192.168.137.1	TLV1.3	85	31 Application Data	
28	0.047270	192.168.137.1	192.168.137.60	TLV1.3	85	31 Application Data	
29	0.089893	192.168.137.60	192.168.137.1	TCP	54	41236 → 443 [ACK] Seq=1267 Ack=542 Win=89856 Len=0	
30	0.090849	192.168.137.1	192.168.137.60	TLV1.3	1200	1154 Application Data, Application Data, Application Data, Application Data	
31	0.092296	192.168.137.60	192.168.137.1	TCP	54	41236 → 443 [ACK] Seq=1267 Ack=1696 Win=92160 Len=0	
32	0.267460	192.168.137.60	192.168.137.1	TLV1.3	178	124 Application Data	
33	0.277692	192.168.137.1	192.168.137.60	TLV1.3	95	61 Application Data	

Figure 24: HTTPS connection with fake website

[Back to text](#)

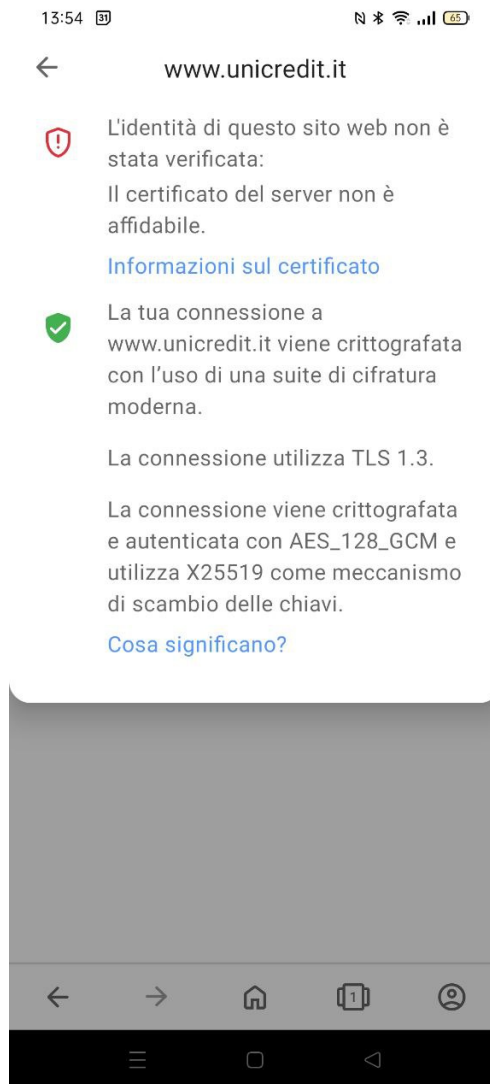


Figure 25: HTTPS warning
[Back to text](#)



Figure 26: Real website

[Back to text](#)

```

2842 3.451383 192.168.137.60 192.168.137.1 DNS 83 Standard query 0x1fff0 A sucmetrics.unicredit.it
2843 3.517551 192.168.137.1 192.168.137.60 DNS 178 Standard query response 0x1fff0 A sucmetrics.unicredit.it CNAME uncredit.it.ssl.d2.sc.omtrdc.net A 63.140.62.164 A 63.140.62.164
2844 3.522993 192.168.137.60 63.140.62.164 TCP 74 33882 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM TSval=310356886 TSecr=0 WS=256
2845 3.559184 63.140.62.164 192.168.137.60 TCP 74 443 → 33882 [SYN, ACK] Seq=0 Ack=1 Win=26847 Len=0 MSS=1386 SACK_PERM TSval=114778648 TSecr=310356886 WS=128
2846 3.560996 192.168.137.60 63.140.62.164 TCP 66 33882 → 443 [ACK] Seq=1 Ack=1 Win=87808 Len=0 TSval=310356845 TSecr=114778648
2847 3.563345 192.168.137.60 63.140.62.164 TLSv1.3 626 Client Hello
2848 3.568295 63.140.62.164 192.168.137.60 TCP 66 443 → 33882 [ACK] Seq=1 Ack=561 Win=1153152 Len=0 TSval=114778648 TSecr=310356846
2849 3.606873 63.140.62.164 192.168.137.60 TLSv1.3 1354 Server Hello, Change Cipher Spec
2850 3.607119 63.140.62.164 192.168.137.60 TLSv1.3 1354 Continuation Data
2851 3.607264 63.140.62.164 192.168.137.60 TLSv1.3 942 Continuation Data
2852 3.608012 192.168.137.60 63.140.62.164 TCP 66 33882 → 443 [ACK] Seq=561 Ack=1289 Win=90368 Len=0 TSval=310356893 TSecr=114778704
2853 3.609012 192.168.137.60 63.140.62.164 TCP 66 33882 → 443 [ACK] Seq=561 Ack=2577 Win=92928 Len=0 TSval=310356893 TSecr=114778704
2854 3.609503 192.168.137.60 63.140.62.164 TCP 66 33882 → 443 [ACK] Seq=561 Ack=3453 Win=95488 Len=0 TSval=310356894 TSecr=114778704
2855 3.610905 192.168.137.60 63.140.62.164 TLSv1.3 130 Change Cipher Spec, Application Data
2856 3.611490 192.168.137.60 63.140.62.164 TLSv1.3 164 Application Data
2857 3.613832 192.168.137.60 63.140.62.164 TLSv1.3 1440 Continuation Data
2858 3.613832 192.168.137.60 63.140.62.164 TLSv1.3 1303 Continuation Data
2859 3.613832 192.168.137.60 63.140.62.164 TLSv1.3 1303 Continuation Data
2860 3.646515 63.140.62.164 192.168.137.60 TCP 66 443 → 33882 [ACK] Seq=3453 Ack=723 Win=27136 Len=0 TSval=114778750 TSecr=310356895

Questions: 1
Answer RRs: 4
Authority RRs: 0
Additional RRs: 0
> Queries
> Answers
> sucmetrics.unicredit.it: type CNAME, class IN, cname uncredit.it.ssl.d2.sc.omtrdc.net
> uncredit.it.ssl.d2.sc.omtrdc.net: type A, class IN, addr 63.140.62.164
> uncredit.it.ssl.d2.sc.omtrdc.net: type A, class IN, addr 63.140.62.135
> uncredit.it.ssl.d2.sc.omtrdc.net: type A, class IN, addr 63.140.62.160
[Request in: 2842]
[Time: 0.066168000 seconds]

```

Figure 27: Correct DNS response

[Back to text](#)



Name	<input type="text" value="www"/>
	.unicredit.it
Type	<input type="text" value="A"/>
TTL	<input type="text" value="0"/> 
IPv4 Address	<input type="text" value="192.168.137.1"/>

Figure 28: TTL of added RR

[Back to text](#)

```

▼ Answers
  ▼ sucmetrics.unicredit.it: type CNAME, class IN, cname uncredit.it.ssl.d2.sc.omtrdc.net
    Name: sucmetrics.unicredit.it
    Type: CNAME (Canonical NAME for an alias) (5)
    Class: IN (0x0001)
    Time to live: 3398 (56 minutes, 38 seconds) 
    Data length: 35
    CNAME: uncredit.it.ssl.d2.sc.omtrdc.net
  ▼ uncredit.it.ssl.d2.sc.omtrdc.net: type A, class IN, addr 63.140.62.164
    Name: uncredit.it.ssl.d2.sc.omtrdc.net
    Type: A (Host Address) (1)
    Class: IN (0x0001)
    Time to live: 180 (3 minutes)

```

Figure 29: TTL of real RR

[Back to text](#)