



Abgabedokument Lab2

Introduction to Security

183.594 – WS 2020

11.01.2020

Team 15

Name	MatrNr.
Maximilian Hagn	11808237
Pascal Poremba	11809911
Hannes Rokitte	11806914
Paul Schmitt	11714338

Inhaltsverzeichnis

1	Bruteforce	3
1.1	Boss Hunt	3
1.2	RingHim	3
2	Intranet	4
2.1	Network Dump	4
2.2	VoIP	6
2.3	E-Mail	7
2.4	IRC	9
2.5	FTP	10
2.6	FTP 2	12
2.7	WLAN Passwort	13
2.8	E-Mail 2	14
2.9	HTTPS	15
3	Passwoerter	18
3.1	Passwort 1	18
3.2	Passwort 2	19
3.3	Passwort 3	20
4	Web-Challenges 2.0	21
4.1	Same Old, Same Old	21
4.2	Bad Timing	22
4.3	Sorcerer	22

1 Bruteforce

1.1 Boss Hunt

In diesem Beispiel war ein Passwortgeschütztes .zip-File gegeben. Natürlich war der erste Ansatz, das Archiv mittels einer Wordlist zu knacken. Hierfür wurde wieder die `rockyou.txt` aus der ersten Abgabe verwendet.

```
fcrackzip --dictionary b3JnYW5pemF0aW9u_15.
cf0025116490f615f019df13c8ff79a2.zip -up rockyou.txt
```

Listing 1: Brute-force mithilfe einer Wordlist

Das gefundene Passwort lautet `dullard`. Das Archiv beinhaltete eine .txt-Datei, in der eine Team-Liste geführt war. Jedoch war vor jedem Namen ein base64-Verschlüsselter Text. Wenn man diesen für jeden Team-Member entschlüsselt, findet man bei Kain Guy den vorgestellten Text: Boss.

Lösung: Kain Guy

1.2 RingHim

Der Tipp des verschütteten Wassers gibt hier den Anschein, als wären nicht alle Zeichen in dem gewählten Passwort möglich. Die Vermutung liegt nahe, dass hier nur Kleinbuchstaben - genauer genommen nur die Buchstaben: q,w,e,a,s,d - im Passwort enthalten sind. Mit einer eingeschränkten Brut-force Methode konnte das Passwort dann gefunden werden.

```
fcrackzip -b -v -c 'a' -l 4-10 -u c2Vjc mV0_15.2
aa605c7c9cae34ff3d864824fb4289f.zip
```

Listing 2: Brute-force mit Character-Einschränkung

Dieser Befehl liefert das Passwort `dqeqs`. In der Datei `number.txt` kann dann die Nummer des Bosses einfach gefunden werden.

Lösung: 84647908

2 Intranet

2.1 Network Dump

Zu Beginn von Lab2 wurde ein Network-Dump durchgeführt. Dabei wurden zwei Netzwerk-Clients angegriffen.

1. Im ersten Schritt wurde die Konfiguration des Netzwerks erfasst und verändert. Genauer gesagt, wurde der Modus von Managed auf Monitor gestellt. Hier für wurde der Befehl **airmon-ng start wlp3s0** asugeführt.

```
wlp3s0mon IEEE 802.11 Mode:Monitor Frequency:2.457 GHz Tx-Power=14 dBm
Retry short limit:7 RTS thr:off Fragment thr:off
Power Management:off
```

Abbildung 1: Einstellungen für Mode in den Konfigurationen

2. Im zweiten Schritt wurde der Dump gestartet. Mit dem Befehl **airodump-ng wlp3s0mon** konnten alle Accesspoints identifiziert werden. Für uns war vor allem die BSSID interessant. Durch diese Mac-Adresse konnten einzelne Zugriffspunkte gezielt angegriffen werden.

BSSID	PWR	RXQ	Beacons	#Data,	#/s	CH	MB	ENC	CIPHER	AUTH	E
58:6D:8F:A9:25:53	-28	7	347	0	0	11	54	WPA	TKIP	PSK	t
5A:6D:8F:A9:25:50	-26	87	356	3729	204	11	54	OPN			f
BSSID	STATION		PWR	Rate		Lost		Frames	Probe		
58:6D:8F:A9:25:53	B8:27:EB:41:2C:D4		-1	1 - 0		0		1			
58:6D:8F:A9:25:53	B8:27:EB:B9:CE:3A		-22	1 -11		0		3			
5A:6D:8F:A9:25:50	B8:27:EB:93:94:99		-29	54 -48		3		1839			
5A:6D:8F:A9:25:50	B8:27:EB:50:D9:1E		-30	54 -48		31		1779			
5A:6D:8F:A9:25:50	B8:27:EB:D6:8F:F5		-34	54 -54		0		35			
5A:6D:8F:A9:25:50	74:DA:38:F0:41:1C		-47	54 -54		0		87			

Abbildung 2: Informationen über gefundene Accesspoints und deren Clients

3. Im dritten Schritt wurde eine BSSID ausgewählt und mit dem Befehl **airodump-ng -d 58:6D:8F:A9:25:53 wlp3s0mon** ein Zugriffspunkt zum Abhören spezifiziert.
4. Um nun den Handshake abzugreifen, musste ein aktuell in dem WLAN eingeloggter User, zum erneuten Einloggen gezwungen werden. Hier für wurde mit dem Befehl **aireplay-ng -deauth 0 -a 58:6D:8F:A9:25:53 -c B8:27:EB:B9:CE:3A wlp3s0mon** immer wieder Deauthentifizierungsanfragen geschickt, bis der Client sich erneut angemeldet hatte und der Handshake abgegriffen wurde.

CH 11][Elapsed: 8 mins][2020-12-06 18:59][WPA handshake: 58:6D:8F:A9:25:5A:6D:8F:A9:25:50												
BSSID	PWR	RXQ	Beacons	#Data,	#/s	CH	MB	ENC	CIPHER	AUTH	E	
58:6D:8F:A9:25:53	0	100	4632	190	0	11	54	WPA	TKIP	PSK	t	
5A:6D:8F:A9:25:50	-23	37	4646	5482	0	11	54	OPN			f	

Abbildung 3: Information über erfolgreich abgegriffenem Handshake

Sobald der Handshake abgegriffen wurde, konnte der Network-Dump beendet und mit der Auswertung begonnen werden. Da bei unserem ersten Network-Dump dringend benötigte FTP files nicht gespeichert werden konnten, wurde auf oben beschriebene Art und Weise ein zweiter Network-Dump durchgeführt. So konnten auch die fehlenden FTP files erbeutet werden.

2.2 VoIP

Um das Gespräch zu finden, wurde das Dump File in Wireshark geöffnet. Anschließend können über Telephony -> VoIP Calls die Gespräche angezeigt werden (Quelle: <https://www.innosoft.at/news/169/voip-grundlagen-wireshark-analyse-von-sip-telefonie>). Über den Button Play Streams wird ein Audio Player geöffnet. Da bei der Wiedergabe zwei Audiospuren übereinander liegen wurde eine davon stumm geschaltet, um das Gespräch besser zu hören. Das Passwort für das Online Banking ist ab Sekunde 20 zu hören. Die Audio Streams sind in **Abbildung 4** ersichtlich.

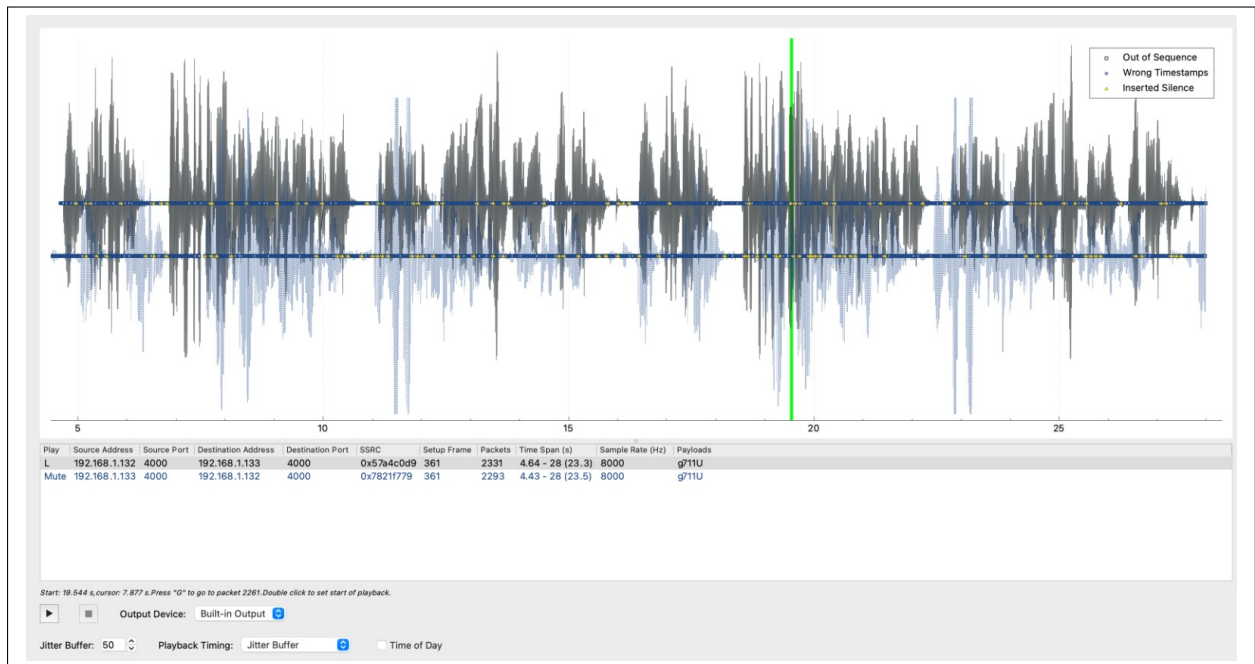


Abbildung 4: Audio Streams in Wireshark

Lösung: Laura0523

2.3 E-Mail

Mit Rechtsklick auf ein TCP Packet kann dem TCP Stream gefolgt werden (Follow -> TCP Stream). **Abbildung 5** zeigt den gefundenen Nachrichtenverlauf.

The image shows a Wireshark packet capture window. The selected packet is a TCP segment containing an email message. The packet details pane on the left shows the packet structure: Ethernet II, Internet Protocol Version 4, and Transmission Control Protocol. The packet bytes pane on the right displays the raw data of the email message, including headers, MIME delimiters, and the body text. The email is from 'yoshiko@aic.vog.invalid' to 'franzm@aic.vog.invalid' with the subject 'Date: Sun, 6 Dec 2020 17:55:02 +0000'. The body text is in German and mentions 'Halle Anna' and 'Wie vereinbart'. The packet number is 220, and the sequence number is 333931FF54.

```
220 aic.vog.invalid ESMTP Postfix (Raspbian)
EHLO wien.localdomain
250-aic.vog.invalid
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-STARTTLS
250-ENHANCEDSTATUSCODES
250-8BITIME
250-DSN
250-SMTPUTF8
250-CHUNKING
250 2.1.0 Ok
[37 bytes missing in capture file].RCPT TO:<franzm@aic.vog.invalid>
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
Message-ID: <663836.651339324-sendEmail@wien>
From: "yoshiko@aic.vog.invalid" <yoshiko@aic.vog.invalid>
To: "franzm@aic.vog.invalid" <franzm@aic.vog.invalid>
Subject:
Date: Sun, 6 Dec 2020 17:55:02 +0000
X-Mailer: sendEmail-1.56
MIME-Version: 1.0
Content-Type: multipart/related; boundary="-----MIME delimiter for sendEmail-77995.9276123954"

This is a multi-part message in MIME format. To properly display this message you need a MIME-Version 1.0 compliant Email program.

-----MIME delimiter for sendEmail-77995.9276123954
Content-Type: text/plain;
 charset="iso-8859-1"
Content-Transfer-Encoding: 7bit

Halle Anna,
Wie vereinbart:
SGludGuVIGltIEKhZZVYgZpbmRlc3QgZHUgZmVlZS8rbGVpbmluZS2ldGUGbWl0IGRlc1B8dWZz Y2hyaWZ0ICJGw7xyIEFybms9sZCiuIEljaCBiaXR0ZS8kaWNoIGRpZWNlIEtpc3RlIDNpY2hlc1B6 dSBlnRzb3JnZW4uCG== Danke und viel Erfolg!

-----MIME delimiter for sendEmail-77995.9276123954---
.
250 2.0.0 Ok: queued as 333931FF54
QUIT
221 2.0.0 Bye
```

Abbildung 5: Follow TCP Stream in Wireshark

Dort wird Anna offenbar eine Base64 verschlüsselte Nachricht übermittelt. Diese wurde über einen Base64 Decoder (Quelle: <https://www.base64decode.org>) entschlüsselt, wie in **Abbildung 6** ersichtlich. Die Kiste "Für Arnoldßoll entsorgt werden.

Decode from Base64 format

Simply enter your data then push the decode button.

SGludGVuIGltExhZ2VyIGZpbmRic3QgZHUgZWluZSBrbGVpbmUgS2lzdGUgbWl0IGRlciBBdWZz Y2hyaWZ0ICJGw7xylEFybm9sZC1uIEljaCBiaXR0ZSBkaWNolGRpZXNlIEtpc3RIIHNPY2hldiB6 dSBibnRzb3JnZW4uCg==

UTF-8

Source character set.

☐ Decode each line separately (useful for multiple entries).

Live mode OFF

Decodes in real-time when you type or paste (supports only UTF-8 character set).

< DECODE >

Decodes your data into the textarea below.

Hinten im Lager findest du eine kleine Kiste mit der Aufschrift "Für Arnold". Ich bitte dich diese Kiste sicher zu entsorgen.

Abbildung 6: Ergebnis Base 64 Decode

Lösung: Arnold

Team 15

Seite 8 von 22

183.594, WS 2020, Lab2

2.4 IRC

Genau wie bei der E-Mail Challenge konnte auch der IRC Chatverlauf über Follow -> TCP Stream verfolgt werden. **Abbildung 7** zeigt den gefundenen Chatverlauf. 4u_only teilt hier seinem Gesprächspartner den Hash „53e72caa5b4ccb3a97d78f2ed033e5c2af25f643“ für das Firmen-Wlan mit.

```
:gangster3000!~gangster3@192.168.1.133 MODE gangster3000 :+i
JOIN #4u_only
:gangster3000!~gangster3@192.168.1.133 JOIN :#4u_only
:hybrid8.debian.local 353 gangster3000 = #4u_only :gangster3000 @gangster3001
:hybrid8.debian.local 366 gangster3000 #4u_only :End of /NAMES list.
PRIVMSG #4u_only :hallo
:gangster3001!~gangster3@192.168.1.131 PRIVMSG #4u_only :bist du endlich da!
PRIVMSG #4u_only :Was wollten Sie mit mir besprechen?
:gangster3001!~gangster3@192.168.1.131 PRIVMSG #4u_only :es geht um die Infos, von denen ich letztes mal berichtet habe. immer noch interessiert?
PRIVMSG #4u_only :Wenn der Preis stimmt, dann erhalten Sie von mir die Daten s..mtlicher Kunden. Aber ... Details nicht hier! Wir unterhalten uns weiter auf einer sicheren Verbindung. Das verschl..sselte
Firmen-WLAN ist daf..r top geeignet, wird garantiert nicht abgeh..rt. Der Chef hat n..mlich keine Ahnung. Der hat so eine unf..hige IT-Firma im Sommer beauftragt. Seither kann man im Firmennetz machen, was
man will.
PRIVMSG #4u_only :hier noch der hash 53e72caa5b4ccb3a97d78f2ed033e5c2af25f643
variable zu der Verbindung
:hybrid8.debian.local 421 gangster3000 variable :Unknown command
:gangster3001!~gangster3@192.168.1.131 PRIVMSG #4u_only :find ich eine gute idee.
PRIVMSG #4u_only :passt, byebye
QUIT #4u_only
ERROR :Closing Link: 192.168.1.133 (Quit: )
```

Abbildung 7: Chatverlauf IRC

Lösung: 53e72caa5b4ccb3a97d78f2ed033e5c2af25f643

2.5 FTP

Um die FTP Pakete zu erhalten, wurde das zuvor entschlüsselte WLAN-Passwort (unter 2.7 detailliert erklärt) in den Decryption Keys des 802.11 Protokolls hinzugefügt (Preferences->Protocols->IEEE 802.11->Edit), wie in **Abbildung 8** zu erkennen.

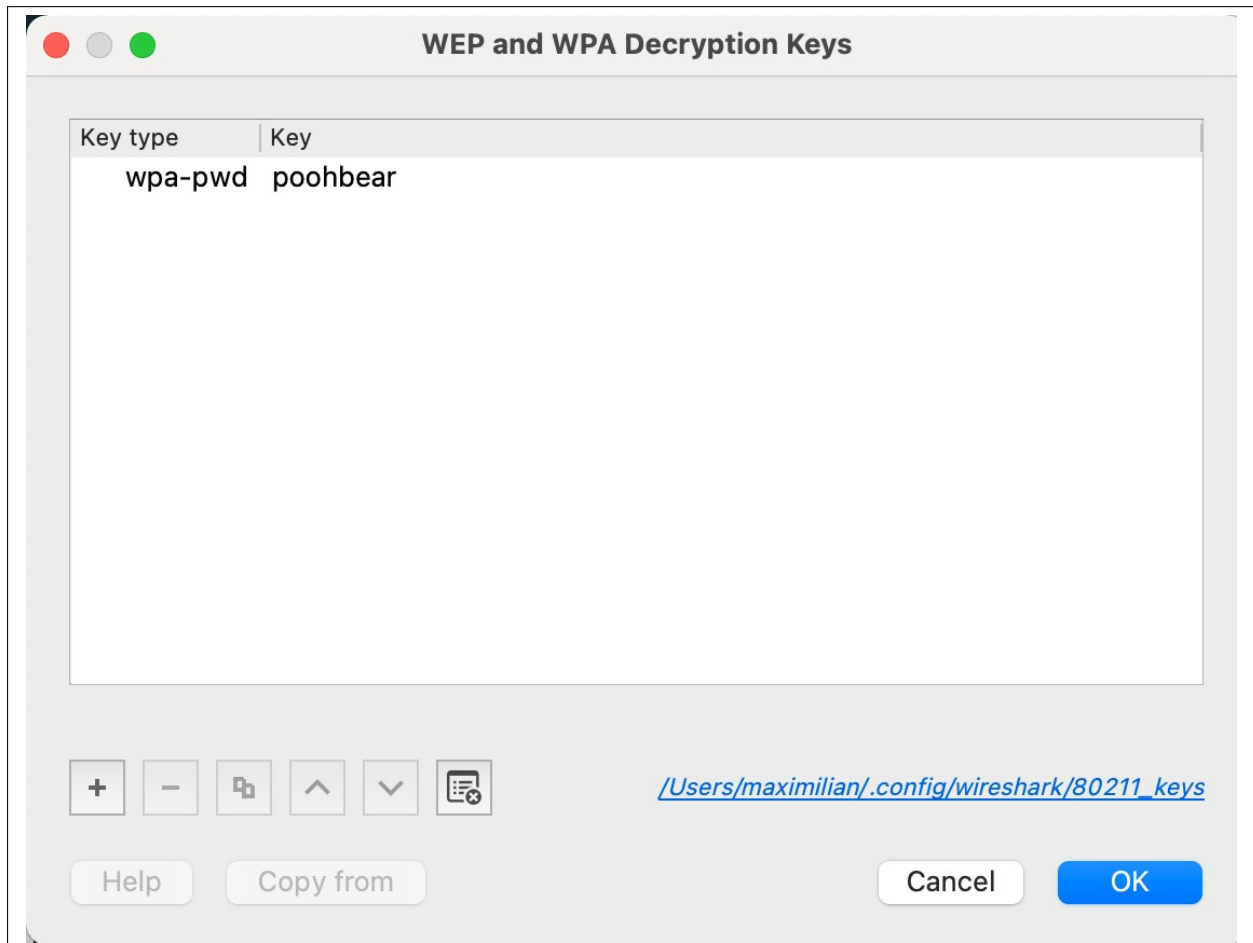


Abbildung 8: WPA Entschlüsselung

Anschließend konnte das Capture File in Wireshark nach FTP Paketen gefiltert werden. Des Weiteren konnte der FTP Verlauf über Follow -> TCP Stream angezeigt werden, dieser ist in **Abbildung 9** ersichtlich. Hier ist zu erkennen, dass sich ein Nutzer mit dem Befehl **USER** als „oberboss“ angemeldet hat. Weiteres wurde mit dem Befehl **PASS** das Passwort „unh4ck4bl3_5up3r_53cr37“ angegeben.

```
220 ProFTPD Server (Debian) [192.168.0.130]
USER oberboss
331 Password required for oberboss
PASS unh4ck4bl3_5up3r_53cr37
230 User oberboss logged in
PWD
257 "/" is the current directory
PASV
227 Entering Passive Mode (192,168,0,130,152,95).
TYPE I
200 Type set to I
SIZE db_admin.key
[10 bytes missing in capture file].150 Opening BINARY mode data connection for db_admin.key (3268 bytes)
226 Transfer complete
[19 bytes missing in capture file].QUIT
221 Goodbye.
```

Abbildung 9: FTP Nachrichtenverlauf Server Credentials

Lösung: oberboss:unh4ck4bl3_5up3r_53cr37

2.6 FTP 2

Nachdem die FTP Pakete, wie in 2.5 beschrieben entschlüsselt wurden, konnte noch ein zweiter Nachrichtenverlauf gefunden werden, dieser ist in **Abbildung 10** ersichtlich. Ein Nutzer hat mit dem Befehl RETR, der eine Kopie der Datei zurück gibt (Quelle: https://en.wikipedia.org/wiki/List_of_FTP_commands), die Datei „db_admin.key“ angefragt.

```
220 ProFTPD Server (Debian) [192.168.0.130]
USER oberboss
331 Password required for oberboss
230 User oberboss logged in
[30 bytes missing in capture file].PWD
257 "/" is the current directory
PASV
TYPE A
[52 bytes missing in capture file].200 Type set to A
SIZE db_admin.key
550 SIZE not allowed in ASCII mode
RETR db_admin.key
150 Opening ASCII mode data connection for db_admin.key (3268 bytes)
226 Transfer complete
QUIT
221 Goodbye.
```

Abbildung 10: FTP Nachrichtenverlauf Datei

Lösung: db_admin.key

2.7 WLAN Passwort

Um das Wlan Passwort zu hacken haben wir im Network Dump die Verbindung zu den Clients getrennt, damit wir bei erneutem Anmelden ein Handshake bekommen. Die genaue Vorgehensweise wurde unter 2.1 erklärt. Die Informationen für sowohl den Network Dump, als auch für das cracken des Passworts haben wir von https://www.youtube.com/watch?v=zAWcu3NQLME&ab_channel=PranshuBajpai. Um nun das Passwort zu hacken, haben wir vorerst in Wireshark geschaut, in welchem Dump File sich der Handshake befindet. Dafür wurden die Protokolle nach dem Protokoll EAPOL gefiltert, die Pakete des Handshakes sind in **Abbildung 11** ersichtlich.

No.	Time	Source	Destination	Protocol	Length	Info
	46195	495.079870	Cisco-Li_a9:25:53	Raspberr_41:2c:d4	EAPOL	131 Key (Message 1 of 4)
	46197	495.084992	Raspberr_41:2c:d4	Cisco-Li_a9:25:53	EAPOL	155 Key (Message 2 of 4)
	46199	495.088573	Cisco-Li_a9:25:53	Raspberr_41:2c:d4	EAPOL	157 Key (Message 3 of 4)
	46201	495.089599	Raspberr_41:2c:d4	Cisco-Li_a9:25:53	EAPOL	131 Key (Message 4 of 4)

Abbildung 11: Handshake Pakete

Anschließend wurde der Befehl in **Listing 3** verwendet, um das Passwort herauszufinden. Die verwendete Wordlist heißt 'rockyou' und wurde von <https://github.com/mishrasunny174/WordLists> heruntergeladen. Um den Befehl auch auf Mac OS verwenden zu können wurde die Anleitung unter https://www.aircrack-ng.org/doku.php?id=install_aircrack#installing_on_mac_osx befolgt. Um den Befehl besser zu verstehen haben wir uns die Dokumentation unter <https://www.aircrack-ng.org/~v/doku.php?id=aircrack-ng> angeschaut. Mit aircrack-ng können WPA Passwörter gehackt werden, die Option -w steht dabei für die Auswahl der Wordlist und -b für die MAC Adresse des Access Points. *.cap wählt alle Capture Files im aktuellen Ordner aus. Die genaue Ausgabe ist in **Abbildung 12** ersichtlich.

```
aircrack-ng -w rockyou.txt -b 58:6D:8F:A9:25:53 *.cap
```

Listing 3: Befehl um das Handshake im Capture File mit der rockyou Wordlist zu cracken.

```
Max-MacBook-Pro:is_team15 maximilian$ aircrack-ng -w rockyou.txt -b 58:6D:8F:A9:25:53 *.cap
Reading packets, please wait...
Opening dumpfile_20201206_184041-01.cap
Opening dumpfile_20201206_184947-01.cap
Opening dumpfile_20201206_185059-01.cap
Opening dumpfile_20201206_191446-01.cap
Opening dumpfile_20201206_191834-01.cap
Read 124937 packets.

1 potential targets

Aircrack-ng 1.6

[00:00:04] 23967/10303727 keys tested (5356.48 k/s)

Time left: 31 minutes, 59 seconds                                0.23%

KEY FOUND! [ poohbear ]

Master Key   : C8 EE 48 09 EC A3 86 5C BD CE 55 BA 6F C8 12 BA
              FD B3 5A DD E0 6B 50 52 5D 42 14 4F 47 B1 E4 B2

Transient Key : F2 3D 5B 3A E1 32 06 29 AA 28 E4 82 67 5A 25 BB
              AF F3 5C D3 D2 B8 C6 52 18 19 E0 5F A1 98 0D 9C
              C9 37 7B FC E0 BE 81 16 7E AD C2 84 54 18 8C 74
              0A 6D D0 36 C3 31 01 B6 B3 BE 1D 42 42 79 57 18

EAPOL HMAC   : B4 A5 E7 7E D4 A3 04 17 1E 94 CE 50 34 87 9F CA
```

Abbildung 12: Passwort Hacking mit aircrack-ng

Lösung: poohbear

2.8 E-Mail 2

Genau wie bei der vorhergehenden E-Mail Challenge konnte auch dieser Nachrichtenverlauf über Follow -> TCP Stream verfolgt werden. **Abbildung 13** zeigt den gefundenen Nachrichtenverlauf. Hier schickt jemand eine Nachricht an Max. Die Nachricht soll umgehend nach dem lesen gelöscht werden, da sie ein geheimes Passwort „cl3v3r_p30pl3_u53_600d_p455w0rd5“ enthält.



```
220 aic.vog.invalid ESMTP Postfix (Raspbian)
EHLO london.localdomain
250 aic.vog.invalid
250 PIPELINING
250 SIZE 10240000
250 VRFY
250 ETRN
250 STARTTLS
250 ENHANCEDSTATUSCODES
250 8BITMIME
250 DSN
250 SMTPUTF8
250 CHUNKING
MAIL FROM:<anna@aic.vog.invalid>
250 2.1.0 Ok
RCPT TO:<maxk@aic.vog.invalid>
250 2.1.5 Ok
DATA
354 End data with <Cr><Lf>.<Cr><Lf>
Message-Id: <534582.829981165-sendEmail@london>
From: "anna@aic.vog.invalid" <anna@aic.vog.invalid>
To: "maxk@aic.vog.invalid" <maxk@aic.vog.invalid>
Subject:
Date: Sun, 6 Dec 2020 17:55:01 +0000
X-Mailer: sendEmail-1.56
MIME-Version: 1.0
Content-Type: multipart/related; boundary="-----MIME delimiter for sendEmail-18473.8853286071"

This is a multi-part message in MIME format. To properly display this message you need a MIME-Version 1.0 compliant Email program.

-----MIME delimiter for sendEmail-18473.8853286071
Content-Type: text/plain;
charset="iso-8859-1"
Content-Transfer-Encoding: 7bit

Hallo Max,
Den Key fuer die HighSecArea habe ich auf den Server geladen.
L..sche diese Mail unbedingt nachdem du sie gelesen hast!
PW: cl3v3r_p30pl3_u53_600d_p455w0rd5
Viel Erfolg!

-----MIME delimiter for sendEmail-18473.8853286071-----

250 2.0.0 Ok: queued as 14E7D1FF2D
QUIT
221 2.0.0 Bye
```

Abbildung 13: Nachrichtenverlauf Passwort HighSecArea

Lösung: cl3v3r_p30pl3_u53_600d_p455w0rd5

2.9 HTTPS

Nachdem die FTP Pakete, wie in 2.5 entschlüsselt wurden, konnte noch eine weitere FTP Nachricht gefunden werden. Diese enthält einen Private Key, dieser wird in [Abbildung 14](#) dargestellt.

```
-----BEGIN PRIVATE KEY-----
MIIEQIBADANBgkqhkiG9w0BAQEFAASCSSwggknAgEAAoICAQDHUxzgWeEBL0d0
CwkGcH8eAqQSB7PdLbIBLdRwrqoaJLAERbcYig+7MXFuX9YVD9qJGxVoc1oiYkMg
EeXdIBZfD4D/rGZArJWt0J36Me5MV14fwDGGit2hxp6CmdIpKHL9BT05YdvDbkf
k6fEJVvP0uUNN00FnTNjspj rCDXF9xFWHiJDKKWGDFSuQEP3iu19wh/Gc0LJ2ON
HPUpuB2L5HD5XuYDs2KG8PKInhVJG0hyeFNrYyWDKjvYldx9iXGFCAdGrgRB+91Y
3adIiJmKPhvHUFr8nvTePz1I/cGsVwrJTYUCZ907Eto00CIqHh0I36qzbTWxPx+
+ec8E0fNuzQtQJEEUd8Stkl0nbl/rVpTGMc7wPxdxi18LLYWKvX11xwjxR00dc44
AtuFfjame/sMXZ1sHNBItJrxazSXEzGrFCj78HqteKakxEhEEf+55GV8oLb9oukL
2abxFCAXWSjAnrWYBb+Zq/qdcMn0dX6m7ims08w0/KYcdA5+UgGnu0v00mW3jh7N
+8qYkjh3/F4dILaDGVa50crr8wjKn5LiV8JG+nB1Y9uzWvalBq3Q04r4hgfvVYRX
E4N06mK/djq5tHs9bKsFBN4tCoSx3TAMF21zH02w00HI00MRFU9nQQA8guEBkwK
VrHt7XjCkpgkchAdYbvs+0unbIQnyQIDAQABaoICAAiRUP9vvZUwWe6J/uVvEHe4
w1anEBskuk8GSkhqZwkiFhZgzJB1PVF5ctJwUj8endE08JWmwv5XwtE2a2MkImb
f/79aRPEL1Z2TiQP+186rign0d0/qLEmx43TKM4GGE2jezabjxu96kQm57hv/Id
dWE0eBkYkgK37YkLP/Ae7XEHu36glwIAN7ahjz0gXp6etLXJ5SCAte293efiKxr1
QcnmQ1rLXiPQvd0Z/+45hV6sn2FRtK57pUDLC050zDtRgkVniDm/sS4/rhLDLQ
b4VPK5250YtYev8Kgba+7uhH3DA2Tau1ynP/+GknXK4ZTHuJBgnpQ7KvTzos1fw1
br9zo1/VFHeZi8Rn2eAP9wCc9sRo6Lkt4WwkiwTznZH095NqpuRaT6gSYUec5Pwz
nz2cn5x/P3y/uml3ECrn5NzdrYXmkTDiWU9VUJULCsGqACvzwzLBicKnvNyaBAPZ
P/Ta+wgYgZiS0EF7yV79oCVazU8paUoazrnzWFr42/E2CpuHQjm+b8kBOuA7GH/6
k+IrdYLVlLkF3h2L23cyG7r1hxx4ccZ0x5q6Fm2bujr0RuRcVQhdEPcuFwtiCMD
i4aXg+kRMAJCARSASj70teZMVD3zUIVxujRBKdbdILwo5NHHYvbdMup1rcpsjMgB
aQCiyIumbZ8NxEQeWdCVAoIBAQPdX0C5PTLZZuYSIBXFPnn5a81ga5rFJ3LDUvEq
T8FVGhaEhkiYY7vX2fgiVFQ1TLH2e+kXVD+M91nSJPNH/A0grjolaN80W/d5a3hh
LLI0I8F95SrQZcD84n4KiQtMw8oVYJ01cu0wqpze68uwoJBic3hEn86n+0omxIpI
hPvtzIM8S42L25fE0jSzB/ZCC6Gy0DiAuJ00JExD3wG1gwwYNacGrHPZcofG+v17
QC73UpTZ6E0/ZkV26CSDU+5mKYUd+wotsli2qh8oFKL/EJu9s5+HGLU2leVCAiHG
6t402KR2gvfCszVWmYf2cJuXZcAyoza/bsmTqyb22vRyjnAoIBAQDaqPZKof66
pV6TC/bQGCf6WvoLqEt/gaFYKYMYaa85uu5VQavESbAH3jaf488T+4jL2FyVz+/
7DMFzCBZPvlnz54j4Mj+7cc25UaaLiQQLjQg54895FHBfUZ66vmuzSETFvTBSVPx
LoRQcqRSE8s54Zc9b1XcTB47fPD600zL/THKVYZNuF5u+IVku+NmbKorCQd4qMN
q6MDRJbdl3XdtDHXk/YLxuEcv+bdoZZ0n0Y0Cr3VzkJfedE8Sdf0SqbTpyv05TGS
uie/0CLF0cWE7mIAVUL+Nn+7yh1uFI80FwmJHZPLNr0IDQTMqyCuPkGTTtYv9yA
1X+H4zMMuaPPAoIBAQDEBHD7B/NJJkaxC0RghwcbvDKKADjDzEw0nmHwoaGgejBZ
5xXJc1TQjdWAuIb3M69jNfpcanj9c0r2D/FwmklqdNLoxhhzLxEUEKQLsET7MjV
wA5ZT1lY0cL/hHff5dAMPTAzxKp+tgEiutWgqm49VDK8qe0DcnD0WxiyS4SAUx4V
In1l8fE8stjffY1D6i7zLYhAja0naRqENDaA44DfjigWfdHH9Fsh4IMghfts0gdho
ZVmLnHLS1NH+whxpEg0vzr0RCNK9tI5xM0xqpX+8S8n/R4DHTgcvtTciaBx4FVWn
+oC0uaVevToRH0/VLqWfU0s5YEHMITWntk1SwpoXAOH/Uem6KnNRFTejfLdM4KP
K148cgtZrXdVagpt7h3tG2+vHl4aFi42DFWCCrjL3l16F5d+MH3Y6p/QHFDXj+2P
po6VgpcHNpQYkg+d3P8UPwLoISzla9RqMKHPP1MIWyb2/gtv04j2lph9/SBbQP1M
FPemXnvdCj6FjeHROB3m+uSg/IAxHREJualDWqiZTWUinM0AdfjB5kmfmBOU7siu
frmrk3zA5R4himzswHT8u+G/5VwMWHVj6arRKVHz+b9tKNWggFen5cR008HJAtOd
rMvbMEQeVTmtRUAjAh2Sd4398Lrp6bQsN8iP+pbbaR9y9Mm0j5gu02r5jmGhSWL
AoIBAQC/1rLFbgC2QKctzhE+2icudSpe0MGbTOUDvUaq1YBTZV/WQnCRwCUxaLhd
yqUTfaZ3uoaVTiS5gQfzh7V5g1++3YL7SoSRZKwFRY3otHEXlkeQ8LvaJ2RFWX2E
tFGuge79+bhTKLYhChjzXhCYh8p614co43UCeC0CsgrtnyLu63imbjbc90av5Zpn
NJYeBHOG3500J32LLW1kg36WCF1VStBUaHjCI8gjdKRSNaZupeuVB8KK9DyahFA
vXW5d0Cnkl8kl8sqx8nzuqLrQq4xG0zzAMPKNCYgepsCz5P9y6G4K4sHkgwXVU
HRsmtE2JqnVGrNcqtK2VV1RI3D+
-----END PRIVATE KEY-----
```

Abbildung 14: FTP Paket enthält RSA Private Key

Anschließend wurde ein .txt File erstellt und der Private Key hinein kopiert. Die Dateiendung wurde weiteres auf .pem geändert, damit die Datei von Wireshark als Zertifikat akzeptiert wird. Dieses Zertifikat wurde unter Preferences->Protocols->TLS->Edit hinzugefügt, wie in **Abbildung 15** abgebildet (Quelle: <https://wiki.wireshark.org/TLS>).

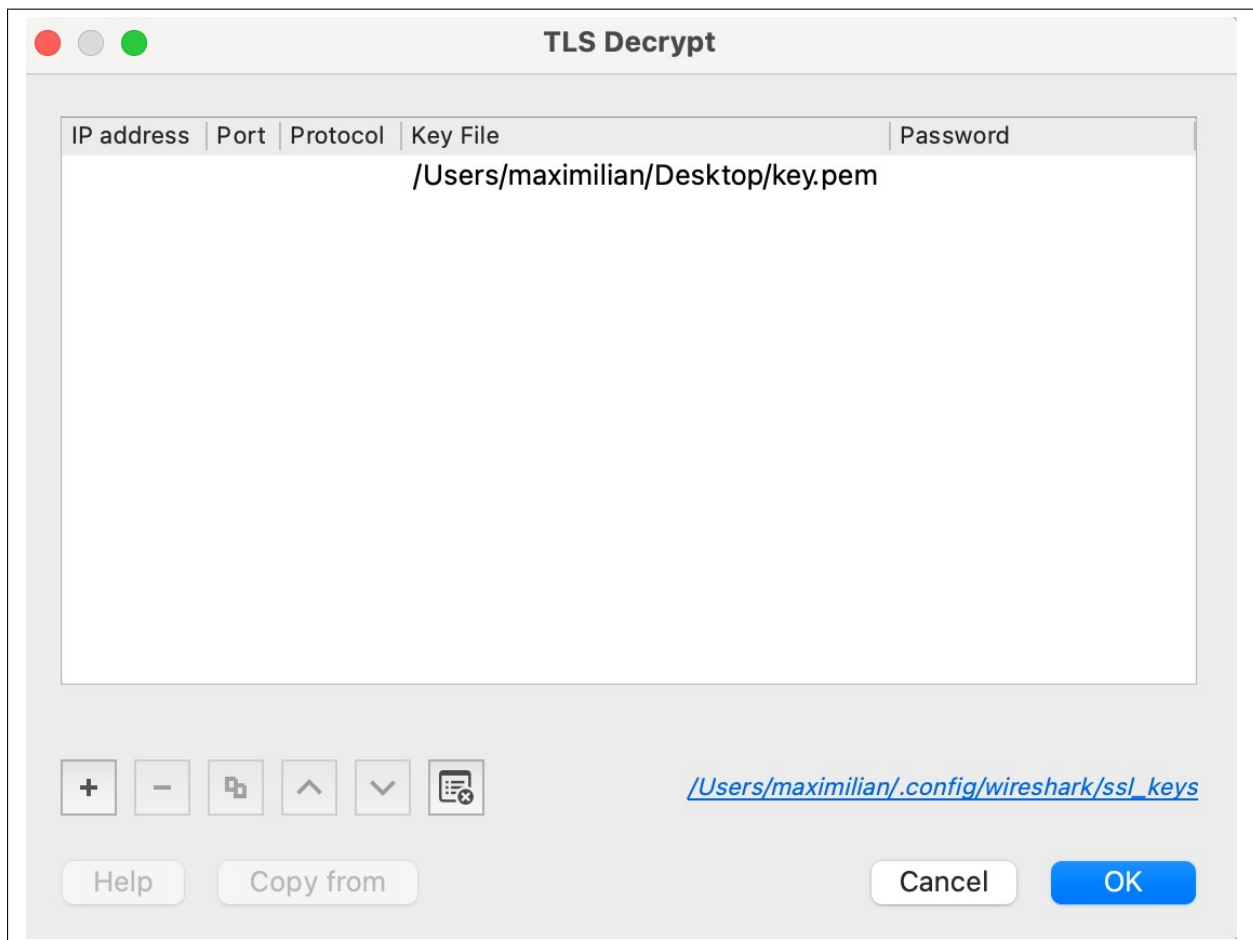


Abbildung 15: TLS Entschlüsselung

Schlussendlich konnten wir weitere HTTP Pakete im Capture File finden, der Nachrichtenverlauf konnte wieder über Follow dargestellt werden. **Abbildung 16** zeigt, dass zuerst ein GET Request auf eine .log Datei gestartet wurde. Anschließend antwortet der Server mit 200 OK und überträgt die gesuchte Datei. Der um 19:18:17 vermerkte Eintrag bezieht sich auf die „whqlprov.mof“ Datei.

```
GET /RXD7tUFD5.log HTTP/1.1
User-Agent: Wget/1.20.1 (linux-gnueabi)
Accept: */*
Accept-Encoding: identity
Host: secret.server.invalid
Connection: Keep-Alive

HTTP/1.1 200 OK
Server: nginx/1.14.2
Date: Tue, 12 Jan 2021 16:30:05 GMT
Content-Type: application/octet-stream
Content-Length: 1733
Last-Modified: Mon, 11 Jan 2021 13:19:27 GMT
Connection: keep-alive
ETag: "5ffc505f-6c5"
Accept-Ranges: bytes

(Fri Nov 08 19:18:05 2019): Beginning Wbemupgd.dll Registration
(Fri Nov 08 19:18:05 2019): Current build of Wbemupgd.dll is 5.1.2600.2180 (xpsp_sp2_rtm.040803-2158)
(Fri Nov 08 19:18:05 2019): Beginning Core Upgrade
(Fri Nov 08 19:18:05 2019): Beginning MOF load
(Fri Nov 08 19:18:05 2019): Processing C:\WINDOWS\system32\WBEM\cimwin32.mof
(Fri Nov 08 19:18:09 2019): Processing C:\WINDOWS\system32\WBEM\cimwin32.mfl
(Fri Nov 08 19:18:12 2019): Processing C:\WINDOWS\system32\WBEM\system.mof
(Fri Nov 08 19:18:16 2019): Processing C:\WINDOWS\system32\WBEM\eventprv.mof
(Fri Nov 08 19:18:16 2019): Processing C:\WINDOWS\system32\WBEM\inetcfg.mof
(Fri Nov 08 19:18:16 2019): Processing C:\WINDOWS\system32\WBEM\sr.mof
(Fri Nov 08 19:18:16 2019): Processing C:\WINDOWS\system32\WBEM\dgnet.mof
(Fri Nov 08 19:18:17 2019): Processing C:\WINDOWS\system32\WBEM\whqlprov.mof
(Fri Nov 08 19:18:18 2019): Processing C:\WINDOWS\system32\WBEM\leinfo.mof
(Fri Nov 08 19:18:18 2019): MOF load completed.
(Fri Nov 08 19:18:18 2019): Beginning MOF load
(Fri Nov 08 19:18:18 2019): MOF load completed.
(Fri Nov 08 19:18:18 2019): Core Upgrade completed.
(Fri Nov 08 19:18:18 2019): Wbemupgd.dll Service Security upgrade succeeded.
(Fri Nov 08 19:18:18 2019): Beginning WMI(WDM) Namespace Init
(Fri Nov 08 19:18:20 2019): WMI(WDM) Namespace Init Completed
(Fri Nov 08 19:18:20 2019): ESS enabled
(Fri Nov 08 19:18:20 2019): ODBC Driver <system32>\wbemdr32.dll not present
(Fri Nov 08 19:18:20 2019): Successfully verified Wbem ODBC adapter (incompatible version removed if it was detected).
(Fri Nov 08 19:18:20 2019): Wbemupgd.dll Registration completed.
(Fri Nov 08 19:18:20 2019): Source for file: https://de.wikipedia.org/wiki/Logdatei, adapted
```

Abbildung 16: Log Datei

Lösung: whqlprov.mof

3 Passwoerter

3.1 Passwort 1

Der erste Ansatz war, die beiden gegebenen Files mit dem Befehl `unshadow` zu kombinieren, und dann mit John the Ripper (Befehl `john`) und einer Wordlist bzw. mit Brute-force die Passwörter zu knacken.

Die drei Passwörter `pw1`, `pw2` und `pw3` waren aber als SHA-256 Hashwert gegeben (der Hash in der `shadow`-Datei beginnt mit `6`). So wurde das `hashcat` Commandlinetool verwendet. Dazu wurde der Hash von `pw1` in eine eigene Datei gespeichert. Danach wurde `hashcat` wie folgt aufgerufen:

```
hashcat -m 1800 -a 0 pw1.txt rockyou.txt
```

Listing 4: Einfache Entschlüsselung mit `hashcat`

So konnte aus der `rockyou.txt` ein passender Hash errechnet werden, der mit dem gegebenen Passwort 1 übereinstimmt. `6SZCdA0Ak/xMhgw.F$FN2dBw12Z9wV.fbWWNZ6zkp9StPtza31TAc7cthoVRK0dEbCHu4uRDA68Xo9wqTBZ3FVXacHMAbQpBm7240EP/:test123`

Lösung: test123

3.2 Passwort 2

Bei Passwort zwei wurde zuerst der gleiche Ansatz wie bei Passwort 1 versucht. Das hat jedoch ewig gedauert und im Endeffekt keine Wirkung gezeigt. Der gegebene Tipp war die ESSE-Website. Dabei sind wir auf das Commandlinetool

CeWL(<https://github.com/digininja/CeWL>) gestoßen, mit welchem man eine 'custom wordlist' erstellen kann. So wurde von der ESSE-Starseite mit der tiefe 2 eine Wordlist generiert. Danach wurde das Passwort 2 wie bei Passwort 1 mit hashcat entschlüsselt (nur mit unserer eigenen Wordlist):

```
2 ./cewl.rb -w esseWordlist.txt -d 2 -m 4 https://security.inso
  .tuwien.ac.at
hashcat -m 1800 -a 0 pw2.txt esseWordlist.txt
```

Listing 5: CeWL ESSE-Wordlist Generierung und pw2 Entschlüsselung

```
$6$/Xr/lS.M66CYxEkw$TjWD/53VRF11/6
c52Wo40ktMSAoJSspjFcJBsStcTs66Pwjtg1C9A0tTvAva0IVWL01TZX5/y0oGQ5Y2RYTvp.:
fuzzolution
```

Lösung: fuzzolution

3.3 Passwort 3

Beim dritten gegebenen Hash, den es zu knacken gilt, gibt es keine großartigen Hinweise, außer, dass es anscheinend unknackbar ist, oder auch nicht. Zuerst wurde wie bei pw1 und pw2 versucht mit `hashcat` und einer passenden Wordlist das Passwort zu knacken.

Nach `rockyou.txt` und

`1mm.txt`(<https://github.com/G0uth4m/Zip-File-Cracker/blob/master/1mm.txt>) wurde auch die selbst erzeugte Wordlist der ESSE-Website ausprobiert. Keine lieferte brauchbare Ergebnisse.

Danach wurde mit `hashcat` und verschiedensten Charsets herumexperimentiert (aber eigentlich nur auf gut Glück) und auch kein passendes Passwort gefunden.

Lösung: Passwort ist nicht mit den uns zu Verfügung stehenden Ressourcen hackbar.

4 Web-Challenges 2.0

4.1 Same Old, Same Old

Für den 1. Login benötigt man folgende Zugangsdaten

Username: is_team15

Password: Passwort des Teams für den Tese Server

Danach lädt man sich eine alte Backup Version der Index.php Datei herunter, welche in der Subdirectory **/index.bak** zu finden ist. Hier findet man weitere Informationen zu einer log-Datei, welche wiederum in der Subdirectory **/w2onrogaj1vgod3kdk47.log** liegt und verschiedene Zugangsdaten von Nutzern enthält. Hast man nun das Datum des Logins und den Nutzernamen (**2021/01/11johnny_atkins**) so erhält man einen funktionierenden 2FA Code (986c376dcd64eaab6331f825182ba0cf).

Um auch in das Profil der anderen Nutzerin (Eve) zu gelangen, encodet man zunächst ihren Nutzernamen mit Base64, um ihm im Anschluss daran URL zu encoden.

Den so erhaltenen String **ZXZlX3N1dGFubw%3D%3D** schreibt man nun in das „loggedInUser“-Cookie.

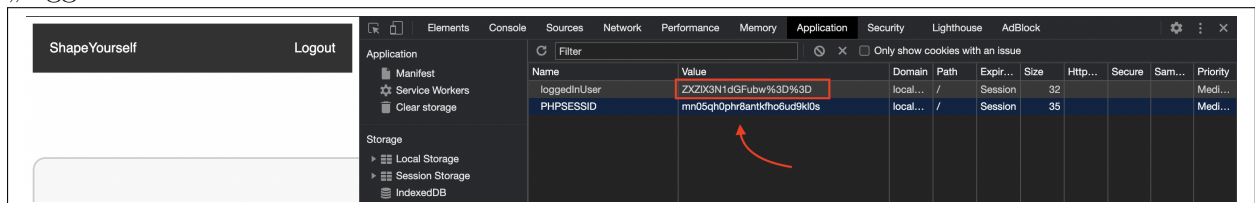


Abbildung 17: Cookie verändern

Lösung: FLAG_f87131300ad964cfUn54F3_2f4_15_l1k3_NO_2f4

4.2 Bad Timing

Diese Aufgabe wurde nicht bearbeitet.

4.3 Sorcerer

Diese Aufgabe wurde nicht bearbeitet.