# Cybermissionen 2024 Lærervejledning

# Victor Hjelmberg Feddersen og Andreas Alstrup

May 2024

# Indhold

1	Ind	ledning 2
	1.1	Læringsmål
	1.2	Opgaver og løsninger
<b>2</b>	Cha	allenges 3
	2.1	Certified Secrets
		2.1.1 Beskrivelse på Haaukins
		2.1.2 Løsning
	2.2	Meta
		2.2.1 Beskrivelse på Haaukins
		2.2.2 Løsning
	2.3	Peacock: En lang rejse
		2.3.1 Beskrivelse på Haaukins
		2.3.2 Løsning
	2.4	Peacock: OSINT hack
		2.4.1 Beskrivelse på Haaukins
		2.4.2 Løsning 7
	2.5	Bad librarian
		2.5.1 Beskrivelse på Haaukins
		2.5.2 Løsning
	2.6	dotsuffix $\ldots$ $\ldots$ $\ldots$ $\ldots$ $\ldots$ $\ldots$ $\ldots$ $\ldots$ $\ldots$ 10
		2.6.1 Beskrivelse på Haaukins
		2.6.2 Løsning
	2.7	Dictionary attacks
		2.7.1 Beskrivelse på Haaukins
		2.7.2 Løsning
	2.8	USBestiålet
		2.8.1 Beskrivelse på Haaukins
		2.8.2 Løsning
	2.9	Browserauth
	-	2.9.1 Beskrivelse på Haaukins
		2.9.2 Løsning

# 1 Indledning

# 1.1 Læringsmål

- 1. Awareness omkring de informationer du deler eller andre og om dig
- 2. Basalt kenskab til HTTP/HTTPS netværkspakker
- 3. Basale færdigheder inden for cybersikkerhed.
- 4. Kompetencer til at navigere Haaukins på egen hånd.

# 1.2 Opgaver og løsninger

Opgaverne som er inkluderet i denne cybermission vil herefter blive refereret til som challenges. Der er i alt 9 challenges som er inkluderet i opgavesættet. Rækkefølgen, de er beskrevet i herunder, vil også være den anbefalede, når eleverne skal løse opgaverne. Det er en god ide at lægge små pauser ind undervejs og også at gennemgå nogle af opgaverne løbende så elever ikke sidder fast i en opgave i længere tid. De første 5 opgaver er tiltænkt, som de vigtigste for at nå læringsmålene, som er beskrevet herover:

- 1. Certified Secrets
- 2. Meta
- 3. Peacock: En lang rejse
- 4. Peacock: OSINT hack
- 5. Bad librarian

# Extra opgaver

- 6. dotsuffix
- 7. Dictionary attacks
- 8. USBestjålet
- 9. Browserauth

# 2 Challenges

# 2.1 Certified Secrets

Denne opgave skal give forståelse for hvad et certifikat er og dets rolle i at skabe en sikker udvæksling af information for brugeren. Det er nemlig vigtigt at tjekke om den side man bruger er krypteret (HTTPS), for ellers kan adgangskoder og anden information blive opsnappet af andre der holder øje med ens aktiviteter. På en siden som ikke er krypteret (HTTP) vil ens browser advare som "Ikke sikker". Men også HTTPS sider med et ikke verificeret certifikat advares som "Ikke sikker". Opgaven går ud på at se hvad et certifikat indholder, og et spørgsmål til eleverne kunne være *Hvorfor vises et ikke verificeret certifikat som* "*Ikke sikkert*"?

#### 2.1.1 Beskrivelse på Haaukins

Look at this site!

It uses HTTPS! But what does an SSL certificate tell us about the site https://certified-secrets.hkn

#### 2.1.2 Løsning

- 1. Gå til https://certified-secrets.hkn. Her vises et hint til en URL sti: https://certified-secrets.hkn/C/ST/L/O/email/DNS
- 2. Klik på låseikonet i venstre side af URL'en, for at se sidens certifikat (se figure 1, 2, 3).
- 3. Brug informationen til at udfylde URL stien.
  - c: dk
  - ST: nordjylland
  - L: aalborg
  - 0: plainsight-corp
  - email: hidden@pc.com
  - DNS: secure.gov

#### 4. Besøg nu det korrekte URL for at få flaget:

https://certified-secrets.hkn /dk/nordjylland/aalborg/plainsight-corp/hidden@pc.com/secure.gov



Figur 1: Klik på låseikonet.



Figur 2: Klik på "More information".



Figur 3: Klik på "View Certificate".

# 2.2 Meta

En fil indeholder også information om filen selv. Denne metadata kan blandt andet afsløre hvor og hvornår et billede er blevet taget. Det er godt at vide at denne information følger med i de filer man deler og hvilket digitalt fodspor man efterlader.

#### 2.2.1 Beskrivelse på Haaukins

```
*Category: Forensics*
- Download<sup>1</sup>
- Vi har modtaget dette billede fra vores agent. Han siger, at der
er en hemmelighed gemt inde i den. Kan du finde det?
```

#### 2.2.2 Løsning

1. Download meta.zip og udpak den.

- 2. Læs billedets (meta.png) metadata
  - Gå til https://exif.tools/ (Ikke i Haaukins) og upload billedet
- 3. Flaget kan ses i Comment feltet, DDC{h1dd3n\_1n\_m3tad4t4}.

 $<sup>^{1}</sup> https://nextcloud.ntp-event.dk:8443/s/53ePbkx23MDPBey/download/meta.zip$ 

# 2.3 Peacock: En lang rejse

Denne challenge handler om hvordan man kan bruge detaljer i billeder til at finde lokationen hvor billedet er blevet taget. Peacock er et socialt medie hvor folk kan dele billeder og kommentare. Freja Hackmann har delt 3 billeder og ud fra billederne skal man finde ud af hvilke 3 byer hun er i.

#### 2.3.1 Beskrivelse på Haaukins

Vi er igang med at undersøge Freja Hackmann som er en del af en stor international kriminel organisation. Vi kan se at hun har postet feriebilleder fra 3 rejsedestinationer og vi har grund til at tro at hun brugte sin rejse til at kontakte medlemmer fra organisationen. Vi har brug for din hjælp til at finde ud af hvilke byer hun rejste til. Du kan finde hendes profil på http://peacock.hkn

Formen på flaget er: HKN{By1\_By2\_By3} hvor By1 er navnet på den første by hun rejste til, By2 er den anden og By3 er den sidste. Hvis hendes rejse f.eks. var fra Oslo til Berlin til København ville flaget være HKN{Oslo\_Berlin\_København} Vær opmærksom på store/små bogstaver.

#### 2.3.2 Løsning

- 1. Gå til Freja Hackermanns profil her kan man højreklikke på billederne og klikke åben i ny fane for at få en større version af dem.
- 2. På billedet af det første rejsemål, kan man se at bilerne kører i venstre side af vejbanen, derudover kan man se teksten PARLIMENT SQUARE SW1 som når man googler viser sig at være relateret til London. Bygningen man kan se er Palace of Westminster som ligger i London.
- 3. På billedet af den andede rejsemål, kan man se en cafe som hedder **Crêperie Parisienne**. Googler man den kan man se at den ligger på addressen 18 Quai du Louvre, 75001 Paris, Frankrig. Helt til venstre kan man også se det franske flag, og et ord man ikke kan se det hele af, dog ligner det at der kunne stå Paris.
- 4. På billedet af det tredje rejsemål kan man længst til højre se det grøndlanske flag. Byen ser relativt befolket ud, hvilket kunne tyde på at det var en af grøndlans største byer, da grøndlands befolkningstal er relativt lavt. Grøndlands største by er Nuuk, og googler man billeder af bygninger i Nuuk kan man finde bygningerne til venstre på billedet ved navn Tuapannguit.
- 5. Samlet set får man flaget HKN{London\_Paris\_Nuuk}

# 2.4 Peacock: OSINT hack

Denne opgave handler om de informationer som folk deler om hinanden. Emilie Netgaard har ikke selv postet noget på det sociale media Peacock andet end et billede af hendes hund, men det har hendes venner og familie. Ud fra oplysninger fra de posts kan man resette hendes password ved at svare på 3 sikkerhedsspsørgsmål. 1. Hvad er hendes fødselsdag, 2. Hvad hedder hendes ynlingskæledyr og 3. Hvad er navnet på hendes folkeskole.

#### 2.4.1 Beskrivelse på Haaukins

Emilie Netgaard har brug for din hjælp. Hun kan ikke huske hendes password, så hun kan ikke længere logge ind. Kan du hjælpe hende? Du kan finde hendes og andre profiler på http://peacock.hkn

#### 2.4.2 Løsning

- På loginsiden på http://peacock.hkn kan man se en knap der hedder Har du glemt et password. Klikker man på den kan man se at man kan resette sit password hvis man kender fødselsdag, navnet på ynlingskæledyret og hvilken folkeskole Emilie Netgaard gik i.
- 2. Man kan klikke "Sign up" og oprette sin egen bruger for at kunne se posts.
- 3. Scroller man gennem postene kan man se et post fra Magnus Cybersen som ønsker Emilie tillyke med fødselsdagen. I kommentarene kan man se at hun er født i 1986 og hun er en måned fra at have fødselsdag d. 1. April, hvilket betyder at hendes fødselsdag er enten 1/3/1986 eller 1/5/1986.
- 4. I et andet post fra Magnus Cybersen kan man se at hun passer Fido, hvilket kunne være hendes kæledyr, måske også det hun selv har lagt et billede op af.
- 5. Viktor Kodegaard har lagt et billede op hvor han rykker for betailinger til en genforeningsfest for hans folkeskole hvor Emilie er inviteret. Navnet på skolen kan læses i kommentarene: Engelbjergskolen.
- 6. Man kan nu logge ud og gå til password reset siden som i step 1. Her indtaster man Emilie Netgaard som navn, et tilfældigt kodeord, fødselsdagen 1/3/1986, ynlingskæledyret Fido og folkeskolen Engelbjergskolen så får man derefter flaget HKN{Hashtag\_AltDetViDeler}. Hvis man indtaster 1/5/1986 som fødselsdag får man en fejlbesked der siger at fødselsdagen er forkert.

# 2.5 Bad librarian

Det er ikke altid sikkert at begår sig online selv om ens forbindelse er krypteret med HTTPS. For hvis man benytter f.eks en offentlig computer på biblioteket kan en ondsindet person få den computer man bruger til at gemme de TLS keys som bruges til at krypteres de netværkspakker der senders over HTTPS. En person der har konfiguret en sådan computer på forhånd, vil senere kunne se al den aktivitet man har lavet fra computeren, og ens adgangskoder, bankoplysning ovs. vil f.eks kunne læses af hackeren. I denne opgave ses hvordan dette scenarie kunne udspille sig i praksis. Her for eleverne nemlig adgang til de TLS keys der bruges til at dekryptere de netværkspakker som en computer har sendt over HTTPS og eleverne skal læse HTTPS krypterede chat beskeder.

#### 2.5.1 Beskrivelse på Haaukins

I'm sure library computers are safe to use. But beware of the librarian i have heard she likes to log a lot.

• Download<sup>2</sup> the librarian's files.'

## 2.5.2 Løsning

- 1. Åben dump.pcapng.gz i Wireshark<sup>3</sup>
- 2. Wireshark viser nu en liste over netværkspakker hvor mange af dem er krypteret
- 3. Brug tlskey.log til at dekrypter netværkspakker
  - Gå til Edit->Preferences->Protocols->TLS og vælg tlskey.log (se figur 4)
- 4. Skriv http i Wireshark's display filter for kun at se HTTP netværkspakker (text input felt øverst i Wireshark Apply a display filter)
- 5. Undersøg dem hvis protocol er HTTP/JSON. En udveksling af skjulte meddelelse kan nu ses
- 6. Få fat i den skjulte meddelse som er Base64 encoded (se figur 5)
- 7. Decode SEtOe2wwMGstMHVOLWYwci10aDMtbDFicjNyMTNufQ== meddelelsen
  - Brug CyberChef (https://gchq.github.io/CyberChef) til at decode meddelelsen eller Google "decode Base64"
- 8. Få flaget HKN{100k-0ut-f0r-th3-11br3r13n} ved at decode Base64 meddelelsen

<sup>&</sup>lt;sup>2</sup>https://nextcloud.haaukins.com/s/wWDrdZFbjoXPb5Y/download <sup>3</sup>https://www.wireshark.org/

	dump.pcapng.gz		_ 0 X
File Edit View Go Capture Analyze Statistics	Telephony Wireless Tools Help		
Apply a display filter <ctri-></ctri->			
No. Time Source	Destination Protocol Length Info		
No.         Time         Source           12827         153.378015992         1283           12838         153.378015932         12839           12839         153.378024928         TACACS+           12841         153.378242645         TAL           12841         153.378242645         TAL           12844         153.37838974         TCP           12844         153.378389969         TCPL           12844         153.378389919         TCPENCAP           12844         153.378389919         TCPENCAP           12844         153.378389919         TCPENCAP           12845         153.378389919         TCPENCAP           12846         153.378389919         TCPENCAP           12847         153.378389919         TCPENCAP           12848         153.378389919         TCPENCAP           12849         153.378389919         TCPENCAP           12849         153.378389919         TCPENCAP           12849         153.374389919         TCPENCAP           12850         153.374389919         TCPENCAP           12851         153.374389919         TCPENCAP           12851         153.37458997         TEECMP	Destination     Protocol     Length     Info       Wireshark - Preferences       Transport Layer Security       RSA keys list     Edit       TLS debug file     Browse       ✓     Reassemble TLS records spanning multiple TCP segments       ✓     Reassemble TLS Application Data spanning multiple TLS records       ✓     Massemble TLS Application Data spanning multiple TLS records       ✓     Message Authentication Code (MAC), ignore "mac failed"       Pre-Shared Key     Browse       ✓     Øffendem Browse       ✓     10.0.2.15       TCP     62.4433 – 44686	Ack=959 Win=65535 Len=0           3259 Ack=3534 Win=625780 Lt           3259 Ack=3534 Win=65535 Len=0           14099 Ack=3534 Win=6535 Len=0           Ack=379 Win=6535 Len=0           Ack=379 Win=6535 Len=0           Ack=379 Win=6535 Len=0           Ack=359 Win=6535 Len=0           Ack=1238 Win=6535 Len=0           Ack=1398 Win=6535 Len=0           Ack=1398 Win=6535 Len=0           Ack=1398 Win=65535 Len=0           Ack=1398 Win=65535 Len=0           Ack=1398 Win=65535 Len=0           Ack=2139 Win=65535 Len=0           Ack=2139 Win=65535 Len=0           Ack=2138 Win=65535 Len=0           Ack=2138 Win=65535 Len=0           Ack=2131 Win=65535 Len=0           Ack=2131 Win=65535 Len=0           Ack=2138 Win=65535 Len=0           Ack=2510 Win=655535 L	.en=2920 [TCP
12074 153.746432519 10.0.2.15	34.120.208.123 TCP 56 46080 → 443 [A	ACK] Seq=25195 Ack=4645 Win=62780 Len=0	
<ul> <li>Frame 60: 357 bytes on wire (2856 bits)</li> <li>Linux cooked capture v1</li> <li>Internet Protocol Version 4, Src: 10.0</li> <li>Transmission Control Protocol, Src Port</li> <li>Hypertext Transfer Protocol</li> </ul>	a, 357 bytes captured (2856 bits) o 2.15, Dst: 34.107.221.82 518088, Dst Port: 80, Seq: 1, Ack 0000 67 40 67 60 60 0000 67 47 6 60 0000 67 40 60 0000 60 0000 60 60 0000 60 0000000000	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	E.U.f@.@_p. "k.R.'P.M@Q.( PGET /ca onical.h tml HTT /1.1.Ho st: det ctportal .firefo
O dump.pcapng.gz		Packets: 12074 · Displayed: 12074 (100.0%)	Profile: Default

Figur 4: Dekrypter netværkspakker ved at klik "Browse" og vælg tlskey.log.

				dump.pcapng.gz								- •	2 3	×
File	<u>E</u> dit <u>V</u> iew <u>G</u> o <u>C</u> apt	ture <u>A</u> nalyze <u>S</u> tatistics	s Telephon <u>y W</u> ireless <u>T</u>	ools <u>H</u> elp										
		🖹 🚺 🔍 🗧 -	→ · · · · · · · · ·											
ht	tp										Þ		•+	
No	Time	Source	+ Destination	Protocol	Length I	nfo								P
140.	603 11 89696/981	172 217 16 195	10 0 2 15	OCSP	757 6	Des	nonse							
	568 11.705548419	172.217.16.195	10.0.2.15	OCSP	758 6	Res	sponse							
	516 11.386934711	172.217.16.195	10.0.2.15	OCSP	758 F	Res	ponse							
	1772 20.927415711	157.240.200.35	10.0.2.15	TLSv1.3	1810 H	ITT	P/1.1 200 OK	(text/html	)					F
1	3651 38.365991694	142.250.185.238	10.0.2.15	HTTP	3528 H	ITT	P/1.1 200 OK	(text/html	5					1
	6117 67.296493125	13.33.141.109	10.0.2.15	HTTP	487 H	ITT	P/1.1 200 OK	(application	, on/javasc	ript	)			
	10193 94.440017111	127.0.0.1	127.0.0.1	HTTP/JSON	325 F	ITT	P/1.1 200 OK ,	JavaScrip	: Object	Nota	tion	(app		
1	10192 94.439778245	127.0.0.1	127.0.0.1	HTTP	254 F	209	ST /chat/thread.	/12/4 HTTP.	1.1					
+	9717 83.733916157	127.0.0.1	127.0.0.1	HTTP/JSON	322 H	ITT	P/1.1 200 OK ,	JavaScrip	: Object	Nota	tion	(app		
+	9716 83.733022462	127.0.0.1	127.0.0.1	HTTP	254 F	209	T /chat/thread	/12/3 HTTP.	1.1					
	4377 51.442819659	127.0.0.1	127.0.0.1	HTTP/JSON	308 H	ITT	P/1.1 200 OK ,	JavaScrip	: Object	Nota	tion	(app		
	4376 51.442309175	127.0.0.1	127.0.0.1	HTTP	254 F	209	T /chat/thread	/12/2 HTTP	1.1					
4	3718 41.440672370	127.0.0.1	127.0.0.1	HTTP/JSON	310 H	ITT	P/1.1 200 OK .	JavaScrip	Object	Nota	tion	(ann		
> Tr	ansmission Control	Protocol, Src Por	t: 45023, Dst Port:	49038, Seg: 2436,	Ack: 13	•][	0000 48 54 54	50 2f 31	2e 31 20	32	30 30	20	4f 4	4b
→ Tr	ansport Layer Secu	rity					0010 0a 43 6f	6e 74 65	5e 74 20	54	79 70	65	3a 2	20
→ Hy	pertext Transfer F	rotocol					0020 70 70 6c	69 63 61	74 69 6f	6e	2f 6a	73	6f 6	зe
- Ja	avaScript Object No	tation: applicatio	n/json				0030 0a 44 61	74 65 3a	20 53 75	6e	2c 20	31	35 2	20
*	Object						0040 63 74 20	32 30 32	33 20 31	. 35	3a 35	34	3a 3	32
	- Member: date						0050 20 47 4d	54 0d 0a	43 6f 6e	74	65 6e	74	2d 4	+c
	[Path with va	lue: /date:2023-04	-01T12:00:03Z]				0000 be 67 74	68 3a 20	31 32 36	000	⊎a ⊎d	⊎a	70 2	22
	[Member with	value: date:2023-0	4-01T12:00:03Z]				0070 61 74 65	22 38 22	32 30 32	53	20 30 22 20	34	20 3	30 70
	String value:	2023-04-01T12:00:	03Z				6d 22 3a	22 31 33	33 37 68	34	78 30	72	22 2	20
	Key: date						00a0 6d 65 73	73 61 67	65 22 3a	22	53 45	74	4f 6	ñ5
	[Path: /date]						00b0 77 77 4d	47 73 74	4d 48 56	30	4c 57	59	77 6	ô3
	<ul> <li>Member: from</li> </ul>		_				00c0 31 30 61	44 4d 74	62 44 46	69	63 6a	4e	79 4	4d
	[Path with va	lue: /from:1337h4x	0r]				00d0 4e 75 66	51 3d 3d	5c 6e 22	2c	22 74	6f	22 3	Зa
	[Member with	value: from:133/h4	xerj				00e0 <mark>68 61 63</mark>	6b 65 72	6d 61 6e	22	7d			
	String value:	1337h4x0r												
	Key: from													
	[Path: /Trom]													
	<ul> <li>Member: message</li> </ul>	1												
	[Mombor with	value: message.SEt	Opposite Contract MHV/01 hV/word 1	ADM TODE TO JAY MINUT	F0\n1									
	String value:	SEtOe2wwMGstMHVAL	WYwci10aDMthDEiciNvM	TNuf0==\n	(ii)									
	Key: message	occoocia noscianos		indi y m			4							Ð
	EDath: /macca	aol				•	Frame (325 bytes)	Decrypted	TLS (235 by	tes)				
0 5	Z JSON string value (is	on.value.string). 48 byte	s		,		Packets: 12074 · Dist	plaved: 142 (1.	2%)	-	Profile	: Def	ault	

Figur 5: Undersøg de dekrypteret netværkspakker med  $\tt http$ som filter og få fat i den sjkulte <code>Base64</code> meddelelse.

# 2.6 dotsuffix

Der findes et utal af file extension f.eks .txt, .pdf, .doc. En file extension er den måde vi navngiver filer på, således at syresystemet ved hvilket program det skal bruge til at håndterer filen. Når vi ændre denne file extension kan syresystemet havde svært ved at forså filen og man kan let tro at filen er beskadiget. Men ved at undersøge filen nærmere kan man finde frem til files rigtige format og se hvad filen indholder.

## 2.6.1 Beskrivelse på Haaukins

```
Things are never as they seem. It's always a matter of perspective. Look carefully at the FILE through the eyes of the terminal. pokemon.jpeg<sup>4</sup>
```

#### 2.6.2 Løsning

- 1. Download pokemon.jpg og opdag at billedet ikke kan åbnes.
- 2. Åben programmet Terminal i Haaukins.
- 3. Skriv file pokemon.jpg lær at filen er af et Zip archive date format.
- 4. Ændre file extension fra .txt til .zip og uppak nu filen.
- 5. Et billede af et pokemonkort (porygon39.jpg) kan nu åbnes.
- 6. Flaget er står nederst på pokemonkortet, HKN{p0ry60n\_15\_7h3\_rul3r\_0f\_cyb3r\_5p4c3}.

 $<sup>{}^{4}</sup>https://nextcloud.ntp-event.dk:8443/s/6aWBjPSSbepcBN4/download/pokemon.jpg$ 

# 2.7 Dictionary attacks

Når man logger på WIFI sender ens router (access point) en krypteret besked til den enhed man logger in fra og ens enhed svare routeren tilbage. Denne udveksling kaldes et WPA2-PSK handshake og er her routeren og enheden generer nøgler for at oprette en sikker forbindelse fremover. Ved at opfange dette WPA2-PSK handshake, kan en hacker finde frem til WIFI adgangskoden ved at prøve en masse forskellige adgangskoder. En lange list med adgangskoder er den kendte liste rockyou.txt, som nemt kan findes i Haaukins eller ved at google "rockyou"på google. Det er derfor vigtigt ikke har have en WIFI adgangkode som let kan gættes ud fra en kendt wordlist så som rockyou.txt.

#### 2.7.1 Beskrivelse på Haaukins

Antallet af usikre wifi-adgangskoder vil ROCK YOU! Se, om du kan bryde wifi-adgangskoden til dette TP-Link-adgangspunkt. Flaget er adgangskoden. Download<sup>5</sup> (Måske du kan nå at drikke en kop kaffe imens din pc arbejder! Eller en hel kande!)

#### 2.7.2 Løsning

- 1. Åben handshake.cap i Wireshark
- 2. Wireshark viser nu en liste over netværkspakker hvor vi skal finde MAC addressen på et TP-Link access point.
- 3. Vi kan se at MAC adressen er 14:eb:b6:86:3c:97 da dette device sandsynligvis er et access point fordi det broadcast'er til andre enheder (Kan ses under kolonne "Destination"i WireShark).
- 4. Brug værktøjet Aircrack-ng med rockyou.txt ordlisten til at crack det WPA2-PSK handshake som er WireShark
  - Åben programmet Terminal i haaukins
  - Skriv aircrack-ng -w /usr/share/wordlists/rockyou.txt -b 14:EB:B6:86:3C:97 handshake.cap
  - Vent på at Aircrack-ng er færdig. Men vær tålmodig, det kan tage op til 20 minutter.
  - Herefter får vi WIFI adgangskoden: dontstealmynameagainyoufreak
- 5. Flaget er DDC{dontstealmynameagainyoufreak}

 $<sup>^{5} \</sup>rm https://nextcloud.haaukins.com/s/Y6tMWkyZkoHZnHs/download/cryptodictattack.zip$ 

## 2.8 USBestjålet

Det USB image der downloads til opgaven svare til et USB stik der kunne sidde i computeren. Men der er slettet noget fra USB stikket og ved hjælp af et forensic værktøjet kan slettede file gendannes fra dets hukommelse. Dette er teknikker der bruge indenfor politiet og et spørgsmål til eleverne kunne være *Hvordan sletter man noget så der er ingen spor fra det?*.

Hvis eleverne synes det kunne være sjovt at se hvad der ligger på USB.001 uden først at finde de slettede filer kan de køre udisksctl loop-setup --file usb.001 i Haaukins og bruge det som var det et normalt USB stik.

#### 2.8.1 Beskrivelse på Haaukins

Politiet anholdt i november en ung mand i København og sigtede ham for at stjæle et hemmeligt dokument fra BBC. Ifølge BBC er det meste af dokumentet volapyk, men det gemmer på en hemmelighed.

Et USB-stik er under afhøringen af den anholdte blevet beslaglagt og sendt til teknisk analyse. Politiet har vurderet det meget sandsynligt, at dokumentet har ligget på USB-stikket og har brug for at finde det før retssagen. Den tekniske afdeling har dog kun fundet en række Monty Python memes og et manuskript.

Kan du hjælpe politiet med deres efterforskning?

drive\_usb\_1gb.zip<sup>6</sup>

#### 2.8.2 Løsning

- 1. Download drive\_usb\_1gb.zip og udpak den.
- 2. Undersøg nu usb.001 med et forensic tool, så som foremost der er godt til store mængder af data.
- 3. Åben programmet Terminal i haaukins. Skriv derefter foremost usb.001.
- 4. Undersøg nu hvad der ligger i output mappen.
- 5. Udpak 00880064.zip som ligger i mappen zip.
- 6. Åben Lorem\_Ipsum.rtf og find en base64 encoded tekst, RERDezcxNV9idTdfNF81Y3I0N2NofQ==.
- Decode RERDezcxNV9idTdfNF81Y3I0N2NofQ== i f.eks CyberChef<sup>7</sup> eller google "base64 decode".
- 8. Flaget er DDC{715\_bu7\_4\_5cr47ch}.

 $<sup>^{6}</sup> https://nextcloud.ntp-event.dk:8443/s/s5HzksPmrQbdkje/download/drive\_usb\_1gb.zip$ 

<sup>&</sup>lt;sup>7</sup>https://gchq.github.io/CyberChef/

## 2.9 Browserauth

BrowserAuth handler om at man skal analysere netværkspakke gennem et værtøj som Wireshark, så man kan finde den URL som indeholder flaget, samt hvilken User Agent man skal have for at tilgå den URL.

#### 2.9.1 Beskrivelse på Haaukins

BrowserAuth the game changing passwordless authenticationmethod without any vulnerabilities... or so they say. Let's prove them wrong. I've captured some interesting packages, that you could try to take a look at.

 $Download^8$ 

http://browserauth.hkn

#### 2.9.2 Løsning

- 1. Download og installer WireShark
- 2. Åben capture.pcap i Wireshark
- 3. Wireshark viser nu en liste over netværkspakker man kan undersøge.
- 4. Den 4. pakke er en http pakke. På den kan man højreklikke  $\rightarrow$  Follow  $\rightarrow$  HTTP Stream (Se Figur 6)
- 5. I det vindue der popper op (Se Figur 7) kan man i de første 3 linjer se "GET /secrets/YWdlbnRfMDA3/flag HTTP/1.1" og "User-Agent: BrowserAuth/9472.2843.8275.1753". Det betyder at der en en der har forsøgt at tilgå http://browserauth.hkn/secrets/YWdlbnRfMDA3/flag med en specefik user agent.
- 6. Forsøger man at tilgå http://browserauth.hkn/secrets/YWdlbnRfMDA3/flag får man en fejlbesked som siger at man skal have den top hemmelige browser for at kunne tilgå siden. Browsere bruger http headeren "user agent" til at finde ud af hvilken browser man bruger, det kan man dog forfalske.
- I firefox kan man skifte user agent ved at gå til about:config og søg efter general.useragent.override. Vælg String click på + iconet og skriv BrowserAuth/9472.2843 som vi fandt i tideligere.
- 8. Nu har du forfalsket din browser til en der kan tilgå siden. Gå til http://browserauth.hkn/secrets/YWdlbnRfMDA3/flag og få flaget. HKN{wlerzQMwJIdhTm5WCWbSD}

 $<sup>^{8}</sup> https://nextcloud.haaukins.com/s/fxGdBLfADn2kAwN/download/capture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pcapture.pc$ 

	capture(3).pcap	- 0 ×						
File Edit View Go Capture Analyze Statistics Telephony	Wireless Tools Help							
$\blacksquare \blacksquare \measuredangle \circledcirc ( 1 = 1 ) \blacksquare \blacksquare \blacksquare \blacksquare ( 2 + 2 ) + 4 $								
tcp.stream eq 0		+ 💌 🖘						
No.         Time         Source         Destination           -         10.000000         192.108.1.3         192.108.1.3           20.000413         192.108.1.3         192.108.1.3           -         10.000435         192.108.1.3         192.108.1.3           -         10.000435         192.108.1.3         192.108.1.3           0.000445         192.108.1.3         192.108.1.3         192.108.1.3           0.000549         192.108.1.3         192.108.1.3         192.108.1.3           0.000549         192.108.1.2         192.108.1.3         192.108.1.3           10.000549         192.108.1.2         192.108.1.2         192.108.1.2           11.0.144498         192.108.1.2         192.108.1.2         192.108.1.2           12.0.144498         192.108.1.2         192.108.1.2         192.108.1.2           12.0.144498         192.108.1.2         192.108.1.2         192.108.1.2           13.0.144492         192.108.1.3         192.108.1.2         192.108.1.2           14.0.144931         192.108.1.3         192.108.1.2         192.108.1.2           13.0.144921         192.108.1.3         192.108.1.3         192.108.1.2           14.0.144931         192.108.1.3         192.108.1.3         192.108.1	Protocol         Length Info           TCP         74<66814         680         157K1         Seq=0         Min=64246         Len=0         MSS=1466         SACK         PERFECT           TCP         74<68614         680         ASS         Seq=0         Min=64246         Len=0         MSS=1466         SACK         PERFECT         Min         Seq         Ack         Min=6426         Len=0         MSS=1466         SACK         PERFECT         Min         Min							
	SCTP >							
	Follow ICP stream Ctrl+Alt+Shift+L							
	Copy DDP Stream Ctr1+Alt+Shift+0							
Frame 4: 387 bytes on wire (3096 bits), 387 bytes ca Ethernet II. Src: Decomput 49:04:01 (09:00:27:49:04)	Protocol Preferences HTTP Stream Citri+Alt+Shift+H	*						
<ul> <li>Internet Protocol Version 4, Src: 192.168.1.2, Dst:</li> </ul>	Show Dasket in New Window HTTP/2 Stream							
<ul> <li>Transmission Control Protocol, Src Port: 60814, Dst</li> <li>Hypertext Transfer Protocol</li> </ul>	QUIC Stream							
<pre>&gt; GET /secrets/WailbackM0A3/flag HTP2/1.1x\n Host: 192.1681.3v\n User-Agent: BrowserAuth/9472.2843.8275.1753\r\n Accept: Lenguage: en-68, en;q=0,St\n Accept:Language: en</pre>								
$ \begin{array}{c} \begin{array}{c} \begin{array}{c} \begin{array}{c} \begin{array}{c} \begin{array}{c} \begin{array}{c} \begin{array}{c}$		• •						

Figur 6: Skærmbillede af capture.pcap åbnet i Wireshark.

	Wireshark · Follow HTTP Stream (tcp.stream eq 0) · capture(3).pcap – ر-	×	
	GET /secrets/YWdlbnRfMDA3/flag HTTP/1.1 Host: 192.168.1.3 User-Agent: BrowserAuth/9472.2843.8275.1753 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8 Accept-Language: en-GB,en;q=0.5 Accept-Encoding: gz1p, deflate Connection: keep-alive Upgrade-Insecure-Requests: 1	3)	
	HTTP/1.1 200 OK Content-Length: 788 Content-Type: text/html; charset=utf-8 Date: Sun, 15 Oct 2023 16:47:26 GMT Server: waitress html <html lang="en"></html>		
	<pre><head>                                                                                                                                                                                        <td <td<="" td=""><td>er V:</td></td></head></pre>	<td>er V:</td>	er V:
	 <body></body>	12 =0	
	<img class="bg-image" src="/static/key.jpg"/>	I.	
	<pre><div class="masthead">     <div class="masthead-content text-white">         <div class="container-fluid px-4 px-lg-0"></div></div></div></pre>	Lr V:	
1	<hi class="fst-italic lh-1 mb-4">Access granted</hi> Welcome back Alice. Your stored secret is: HKN{XXXXX}	=6 [1 /j	
ł			
ł			
00	 GET /static/index.css HTTP/1.1 Host 192,168.1.3		
	Oser Agent: BrowserActory 1/2:2043.02/5.1/33         Accept: Language: en-68, en;q=0.5         Accept:Language: en-68, en;q=0.5         If-Modified: Since: Non, 09 0ct 2023 19:00:10 GMT         If-None-Match: "1696878010.2701066-240102-1422985190"		
	HTTP/1.1 304 NOT MODIFIED	- 5	
1	2 client pkts, 2 server pkts, 3 turns.		
613	End End New		
	Find Next		
1	Printer Out This Stream Print Save as Back Close		

Figur 7: Skærmbillede af "Follow HTTP" lokation.