

A stylized graphic of blue circuit lines with circular nodes, extending horizontally from the left and right sides of the central black box.

DIGITAL FORENSICS

THE ART OF UNCOVERING

DE DANSKE CYBERMESTERSKABER

Kvalifikationen er i gang!

Løs 6/15 opgaver for at gå til de regionale mesterskaber

Join fællesskabet på Discord:

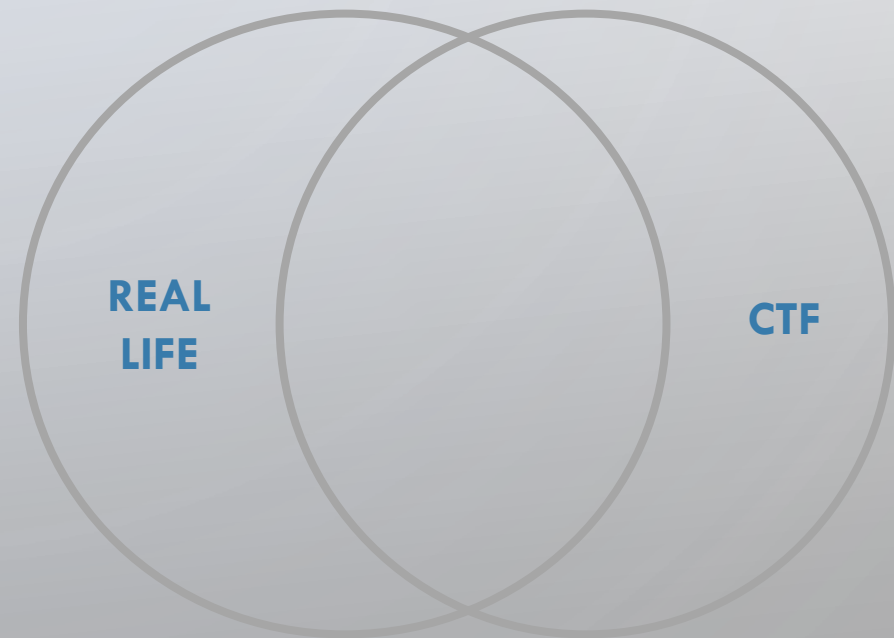
DDC: <https://discord.gg/WtSuA3AR68>

CyberSkills: <https://discord.gg/NM8rjxuqPS>

FORENSICS?

Real life

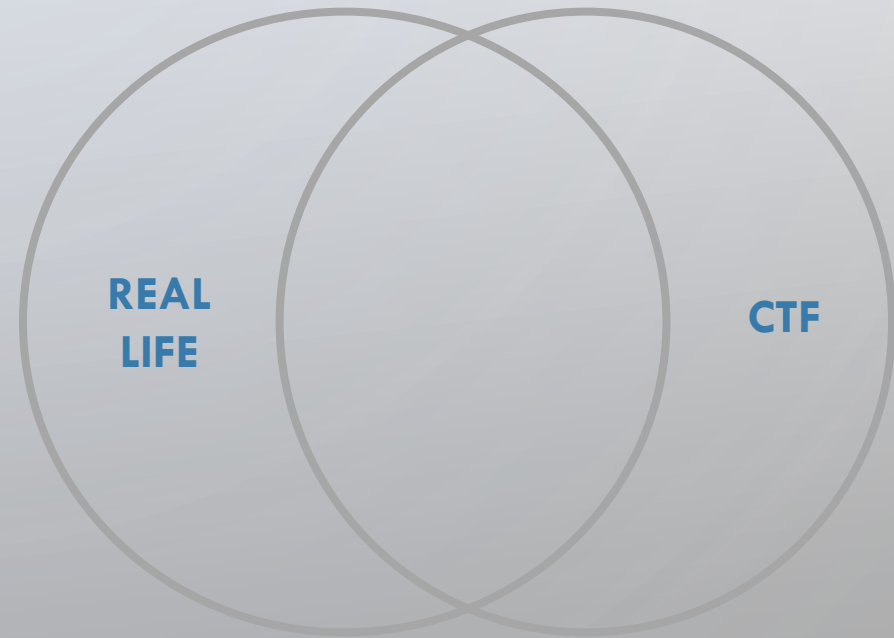
- Efterforskning
 - Indsamling og analyse af digital data
 - Kriminal­sager og incident response
- Finde og fortolke fakta
- Genskabe hændelsesforløb
- Hvad, hvem, hvornår, hvordan, hvorfor



FORENSICS?

CTF

- Finde skjult information i filer eller metadata
- Gendanne tabt eller slettet data
- Rekonstruere beskadigede filer
- Genkende filstrukturer og identificere filformater
- Forstå et hændelsesforløb ud fra netværkslogs eller memory dumps
- Hash cracking



MINDSET

- Digital data: rå bits
- Vi skal selv tillægge meningen – og indbygge den i computeren
- Filer, filsystemer, protokoller osv. er struktur, vi tillægger dataen
- Forensics bliver nemmere, når du forstår at binær information eksisterer uafhængigt af de kendte abstraktioner
- Ting er ikke altid hvad de ser ud til!
- Kend dine teknikker og værktøjer (Linux terminalen er din ven)

PROGRAM

PART 1

File Analysis 25 min

Øvelser 25 min

Spørgsmål 5 min

Steganography 30 min

Øvelser 30 min

Spørgsmål 5 min

BREAK

PART 2

Memory Analysis 10 min

Øvelser 60 min

Spørgsmål 10 min

The background features a light gray gradient with faint, large-scale circular patterns. In the corners, there are decorative elements resembling circuit board traces or neural network connections, consisting of thin blue lines and small circles.

FILE ANALYSIS

ENCODINGS

Encode: konvertere data fra en form til en anden

Samme data kan encodes på forskellige måder

```
01000110 01101111 01110010 01100101 01101110  
01110011 01101001 01100011 01110011 00100001
```

Encodings:

- Decimal: 70 111 114 101 110 115 105 99 115 33
- Hex: 46 6f 72 65 6e 73 69 63 73 21
- Octal: 106 157 162 145 156 163 151 143 163 41
- ASCII: Forensics!
- Base64: Rm9yZW5zaWNzIQ==
- Base85: 7W3<YDKBN%F!1

HEX

Fleste filtyper bruger bytes uden for ASCII range

Nemmest at læse og analysere i hex

- Hex bruger 0-9A-F
- Én hex karakter svarer til 4 bits (en "nibble")
- Så én byte = 2 hex karakterer (f.eks. 10111110 11101111 = BE EF)

Tools:

- hexdump / xxd: lav et hexdump af en fil
- hexedit: CLI hex editor til at ændre hex filer
- GUI hex editors: ghex (Linux), HxD (Windows)
- Scripting language, fx Python
- CyberChef: online toolkit

BASE64

Base64: encoding scheme der repræsenterer binær data som ASCII tekst

Cybermesterskaberne → Q3liZXJtZXN0ZXJza2FiZXJuZQ==

Input splittes i blokke af 6 bits, der mappes til en karakter i A-Za-z0-9+/-

Eks.: Base64(DDC) = RERD

- DDC = 01000100 01000100 01000011
- 010001 000100 010001 000011
- R E R D

Paddes med = el. == hvis sidste blok mangler hhv. 2 eller 4 bits

Bruges til at sende og gemme binær data (f.eks. billeder)

Hint: alfanumerisk streng, der evt. slutter på = el. == er base64 encoded

FILE ANALYSIS – FILE TYPE

Filer er bare binær data, vi tillægger en bestemt struktur

Filens *filtype* fortæller os, hvordan vi skal fortolke dataen

- F.eks. PNG, PDF, DOCX, WAV

Filtypen er ofte indikeret af filens *extension* i filnavnet, f.eks. .png, .mp4

- Typisk hvad OS bruger for at vurdere hvordan filen skal åbnes/fortolkes
- Stol ikke på extensions! Kan ændres for at snyde OS til at fortolke data forkert

Filtypen indikeres i filens indhold med en filsignatur – et *magic number*

- Hex streng ved et bestemt offset
- Fx PNG-filer: 89 50 4e 47 (sidste tre hex er PNG i ASCII)
- Tool: `file`
- Mere troværdig end extension, men kan også nemt forfalskes

FILE ANALYSIS - METADATA

Filtypen er én form for *metadata*: data om data

Ekstra information om en fil udover selve indholdet

- Generelle: Filnavn, extension, størrelse, oprindelsestidspunkt, permissions
- Specifikke: GPS-data i billeder, antal frames i GIF, CPU arkitektur i executables, osv.

Husk at analysere metadata!

- Kan gemme på vigtig info – måske endda info, der skulle have været skjult
- I nogle tilfælde endnu vigtigere end indhold – fx ved krypteret HTTPS trafik

Tool: exiftool

FILE ANALYSIS – FILE FORMAT

En filtype har et bestemt *format* – filens struktur

Typisk struktur:

- Fil signatur – magic number
- Header – typisk info, der skal bruges til at forstå indholdet (metadata)
- Evt. metadata
- Data
- Evt. trailer, der afslutter filen

Formatet er præcist defineret i et specification doc – ofte offentligt tilgængeligt

- Beskadigede filer: sammenlign fil med specifikation, ret forskelle med hex editor
- Ukendte filtyper: led efter spor fra et filformat

ØVELSER

STEGANOGRAPHY

STEGANOGRAPHY

Steganografi: skjule hemmelig information i ikke-hemmeligt data

Simpel teknik: tilføj ekstra data i slutningen af en fil

- Ændrer ofte ikke håndteringen af filen
- Opdages kun ved at inspicere filens indhold

Simpleste udgave: tilføj tekststreng

- `cat` (print som ASCII), `strings` (print ASCII strings fra filen)
- Søg/filtrer i resultatet med `grep` / `bgrep`

Kan tilføje hele ekstra filer

- Fx ZIP-fil gemt i bunden af en PNG
- PNG slutter med traileren `IEND`, så billedeviseren er upåvirket

FILE CARVING

File carving: extracte filer på baggrund af filformatet

- Kig efter filsignaturer, headers, trailers osv.
- Oprindeligt brugt ifbm. extraction af filer fra disk images og memory dumps
- Brugbart til at extracte filer gemt i andre filer i stego challenges

File carving tools:

- binwalk
- foremost
- dd (manuel extraction)

```
dd if=input.png of=output.txt bs=1 skip=1000 count=32
```

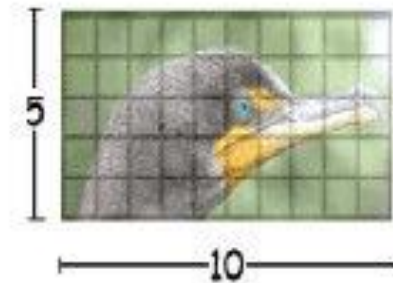
IMAGE STEGANOGRAPHY

Mediefiler er særligt
egnede til steganografi

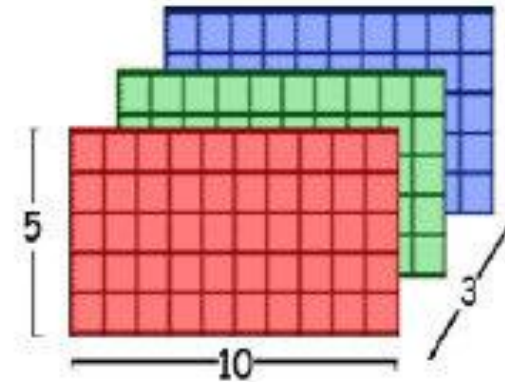
Pixels

Farvekanaler: RGB

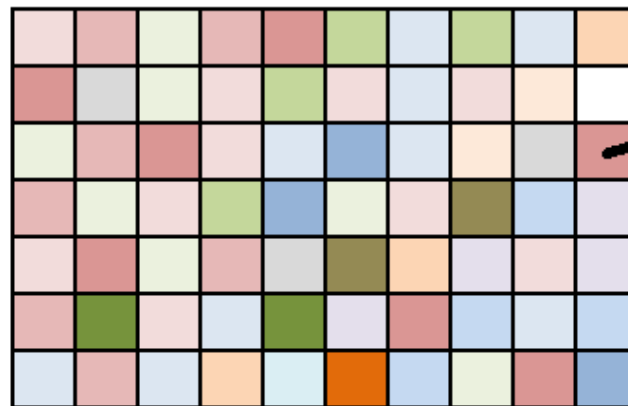
Planer: typisk 8



Original Color Image



RGB Matrix



RGB (218, 150, 149)

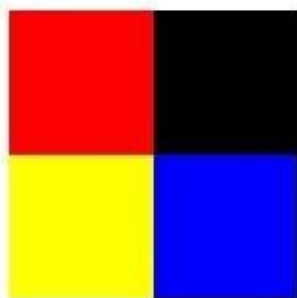
R = 11011010

G = 10010110

B = 10010101

IMAGE STEGANOGRAPHY

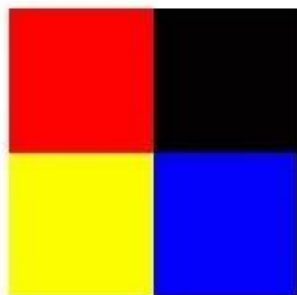
Original Image



11111111	00000000
00000000	00000000
00000000	00000000
11111111	00000000
11111111	00000000
00000000	11111111

Least Significant Bit Steganography

Stego Image



111111 01	000000 11
000000 10	000000 01
000000 00	000000 10
111111 00	000000 11
111111 01	000000 01
000000 01	111111 00



c	a	t
01 10 00 11	01 10 00 01	01 11 01 00

IMAGE STEGANOGRAPHY

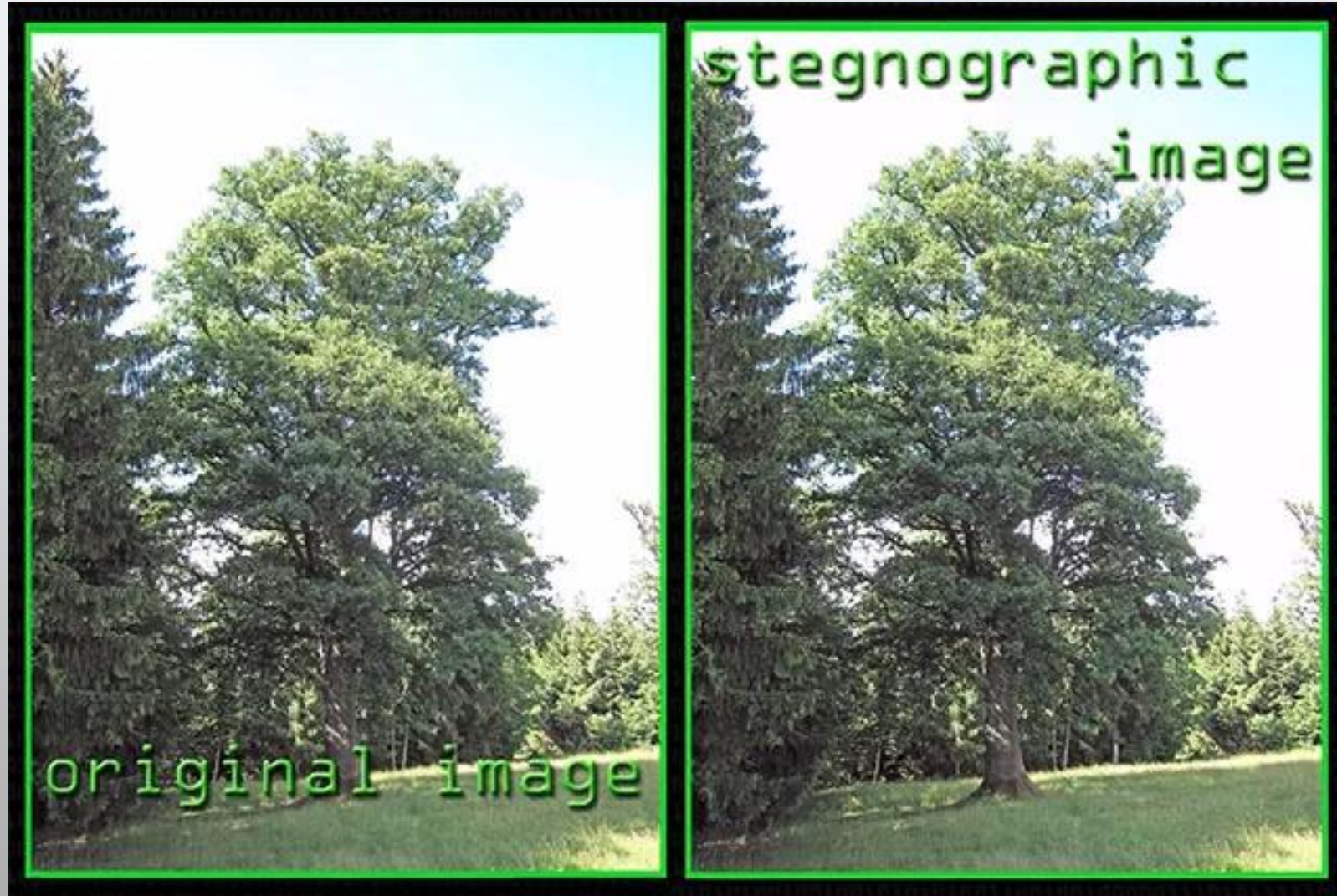


IMAGE STEGANOGRAPHY

Cover Image



Image to Merge



Merged Image



Unmerged Image



IMAGE STEGANOGRAPHY

Tools

- stegsolve: inspicér billeders individuelle farvekanaler og bitplaner
 - Især brugbart til billeder gemt i andre billeder
- zsteg: find gemt tekstdata i PNG og BMP
- steghide: embed/extract data i billede- og lydfile
 - Mere sofistikeret metode end LSB
 - Optional password ved embedding, kræver samme password ved extraction
 - Tip: password kan være tomt
 - Tip: brug stegseek til at brute force passwords
- Stego Toolkit: <https://github.com/DominicBreuker/stego-toolkit>

ØVELSER

MEMORY ANALYSIS

MEMORY ANALYSIS

Traditionel computer forensics = filesystem forensics

- Persistent data, hard disk/USB
- "Dead box forensics"

I dag er fokus også på memory forensics på volatil data

- Volatil data: ikke-permanent data, forsvinder når strømmen går
- Typisk indholdet af main memory – RAM
- "Live box forensics"
- Analyse foregår på et memory dump – giver et øjebliksbillede

MEMORY ANALYSIS

Data der kan findes i volatil hukommelse

- Kørende processer og services
- Åbne filer
- Netværksforbindelser
- Kørte kommandoer
- Passwords, keys
- Ukrypteret data, der er krypteret på disken, men skal bruges i dekrypteret tilstand i memory
- Stateless malware – malware, der kun lever i hukommelsen
- Endda ting som et basic screenshot eller brugerens clipboard

MEMORY ANALYSIS

Tool: Volatility

Volatility 2

- Skal bruge den rigtige *profil* for at analysere dumpet
 - `imageinfo` – foreslår mulige profiler
 - `kdbgscan` – identificér korrekt profil
- `python2 vol.py -f [image] --profile=[profile] [plugin]`

Volatility 3

- Bruger et library af symbol tables – ikke profiler
- Hurtigere og nemmere
- Men! Mangler en række plugins
 - `notepad`, `iehistory`, `screenshot`, `clipboard` mm.
- `python3 vol.py -f [image] [plugin]`

CASE: MALPDF

FORENSICS TIPS

Vær struktureret, analytisk – skriv alle findings ned

Led efter sammenhænge – og efter afvigelser

Bliv fortrolig med dine tools og med et scripting language

Lær at genkende og forstå fil formater, protokoller og encodings

Vær kritisk over for dine egne forudindtagelser! Ikke alt er som det ser ud!

Don't do it twice: gem resultatet af dine scanninger

Have fun!

LINKS

CTF Field Guide – Forensics: <https://trailofbits.github.io/ctf/forensics/>

Forensics Labs (network, memory, disk, malware, etc.): <https://cyberdefenders.org/>

HackTricks – Forensics: <https://book.hacktricks.xyz/forensics/>

13Cubed: <https://www.youtube.com/c/13cubed>

BlueTrainer (CTF): <https://discord.gg/2eaBBbgxRK>

DDC Discord: <https://discord.gg/WtSuA3AR68>

CyberSkills Discord: <https://discord.gg/NM8rjxuqPS>

DDC kvalifikationsrunde: <https://cybermesterskaberne.dk/>

The image features a light gray background with a subtle, large-scale geometric pattern of overlapping triangles. In the four corners, there are decorative elements consisting of thin, dark gray lines that branch out like circuit traces, ending in small circles of varying sizes.

SPØRGSMÅL