Stegomalware

Report Presentation

OLLSCOIL TEICNEOLAÍOCHTA
BHAILE ÁTHA CLIATH
TU DUBLIN
TECHNOLOGICAL
UNIVERSITY DUBLIN

“Metadata” of the project

Authors

- Danyil Tymchuk (B00167321)
- Illia Stefanovskyi (B00165280)
- Artem Surzhenko (B00163362)

Course

- Digital Forensics and Cyber Security (TU863/Y3)
- Module: Computer and Network Forensics

Submission date: 07/12/2025

Introduction

Introduction

- **Case Studies**
- **Experiment**
- **Steganalysis**
- **Mitigation**

Analysis of how malware uses steganography techniques and how those can be mitigated.

Background: How Steganography Actually Works

and Why It's a Nightmare for Forensics

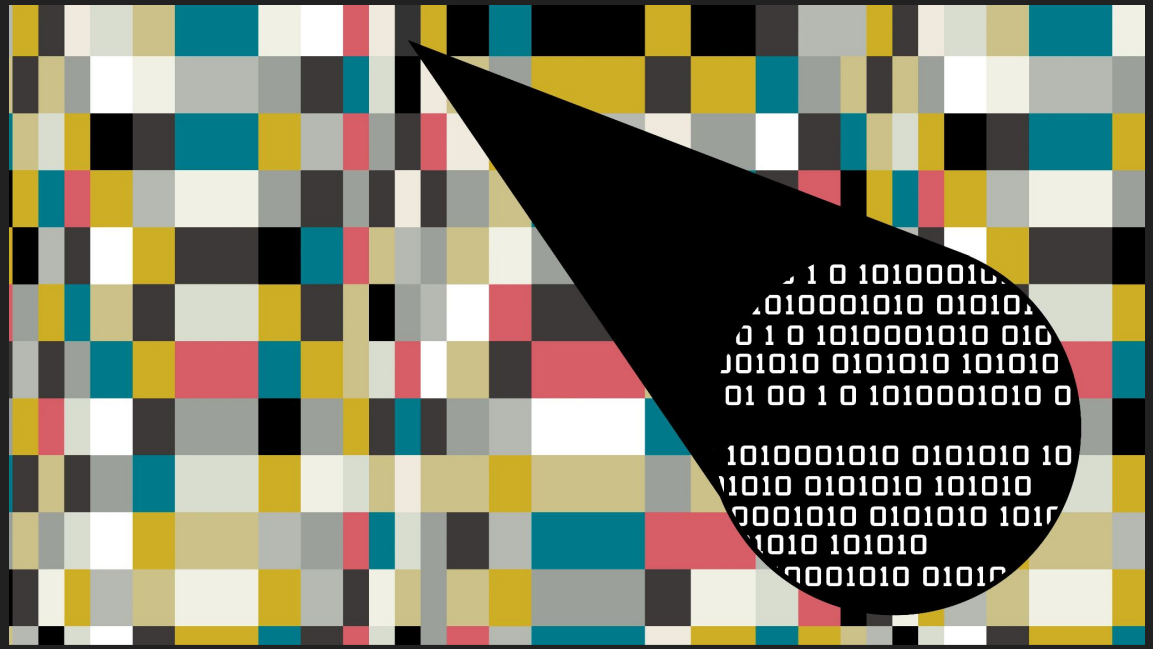


image taken from https://www.wired.com/story/steganography-hacker-lexicon/

Steganalysis: The Uphill Battle of Detection

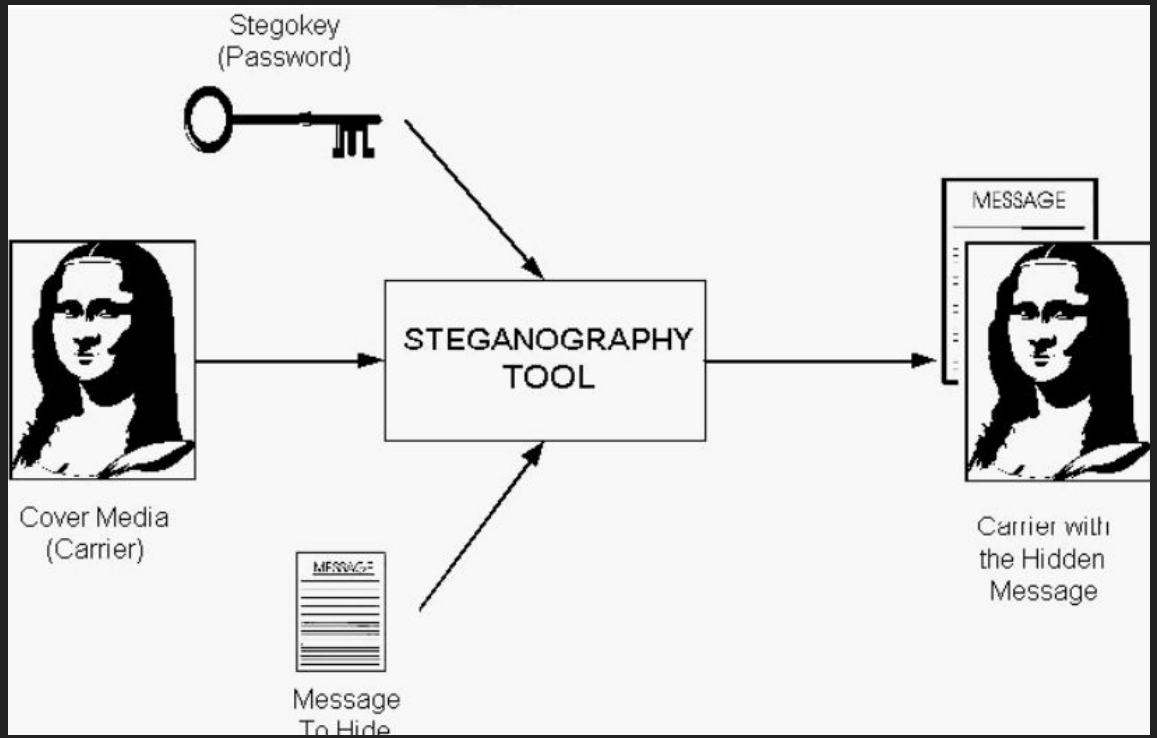


image taken from https://www.giac.org/paper/gsec/707/steganalysis-overview/101589

Case Studies

- Case Study 1 – Stegoloader
- Case Study 2 – The "ClickFix" Campaign

Case Study 1 – Stegoloader

camouflage its primary executable code among legitimate-looking PNG images

- discovered in 2013 by Dell SecureWorks CTU

Case Study 1 – Stegoloader

Technical Analysis

- deployment module
- main module
- optional add-on modules

Case Study 1 – Stegoloader

Technical Analysis

- ✓ deployment module
- ✓ main module
- optional add-on modules
  - Pony Password Stealer
  - Geolocation Module
  - Recent Documents Module
  - IDA-Related Module

Case Study 1 – Stegoloader

Indicators of Compromise (How to notice Stegoloader?)

- Network Traffic: HTTP GET/POST requests
- File Hashing
- Filenames

Case Study 1 – Stegoloader

Conclusion

Case Study 2 – The "ClickFix" Campaign

- Appeared late 2024 – major threat by 2025
- Malware hidden inside everyday images
- Uses fake software error messages
- Steganography + social engineering

The Hook: Fake Error Messages

- Looks like Chrome or Word errors
- Uses real logos and wording
- Tells users to press **Win + R**
- User unknowingly launches malware

How the Malware is Hidden

- Uses **mshta.exe → JScript → PowerShell**
- No files saved to disk (file-less attack)
- Malware hidden in **PNG or SVG images**
- Code hidden in **red pixel values**

What Happens to the Victim

- Encrypted code rebuilt in memory
- Injects into trusted processes
- Runs **LummaC2 / Rhadamanthys**
- Steals passwords, cookies, crypto

Why ClickFix is Dangerous

- No phishing, no attachments
- Bypasses many security tools
- Targets **human trust**
- Shows steganography is now mass-used

Another Example – AdGholas

bought real ad space on trustworthy websites and used image-based steganography to hide malicious JavaScript inside banner ads

Primary Research

Primary Research — Experiment

Simulation of Steganographic Payload Delivery

- Experiment A: The "Append" Method (Overlay Steganography)
- Experiment B: The “Embedding” Method (Steghide)

X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*

EICAR String

Experiment A: The "Append" Method

Overlay Steganography

Experiment A: The "Append" Method

Procedure

- bash: `cat image.jpg eicar.com.txt > infected_image_append.jpg`
- cmd: `copy /b image.jpg + eicar.com.txt infected_image_append.jpg`

Experiment A: The "Append" Method

Results

- Visual Inspection: The image is not corrupted
- Forensic Analysis: At the very bottom of the file → the plaintext EICAR string


UW PICO 5.09 File: infected_image_append.jpg Modified
$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*
^G Get Help ^O WriteOut ^R Read File ^Y Prev Pg ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where is ^V Next Pg ^U UnCut Text ^T To Spell

Experiment A: The "Append" Method

Hexadecimal Inspection

- JPEG ends with the byte sequence **FF D9**


File Edit Disk Options Tools Plug-Ins Window Help
0 1 2 3 4 5 6 7 8 9 A B C D E F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C
001A1523 12 DD DB DA F5 AA FE A1 E0 FD B5 B9 19 BD 0B 15 05 F0 30 9A 5E 31 D0 2E A2 DC A5 47 B6
001A1540 DC 8B 56 FC 0B 0C 6F 9D F8 9F 12 DC FF 00 3C 01 C9 59 8D B7 35 38 5A 5C 47 1C 55 B0 BD
001A155D 15 15 6C 25 B1 55 79 D8 51 D0 6F 54 3D CD B8 1F 3D B5 BA BB EC BE 97 D9 DE AA 2C 85 0C
001A157A 34 3A 0D E3 A1 A1 B5 00 12 F2 F4 71 56 C3 2B 9C ED 8B AD CD 44 B8 D4 BB 37 E8 F7 D7 7E
001A1597 0A C1 F2 AD C4 7C 07 53 D7 FD BA 5D 16 26 1B D5 47 1E 2A 3A 28 F3 BC 12 C2 9E 72 F3 15
001A15B4 45 B0 5C 8B F2 D7 19 74 BE 36 3E 0D F6 DF 05 95 F8 03 42 E8 1F 20 70 16 A5 2C 71 B8 8B
001A15D1 05 2C 74 2A 7E 70 71 51 68 16 C5 C7 45 82 D2 E3 C5 74 5E 39 2F 86 21 EA EF 1E F3 E8 C2
001A15EE 09 7A AC 8E F1 B6 F5 7B 0D FC 53 CC F1 45 BD 0A 2A 3A 38 B5 39 E3 15 95 A2 C1 62 F3 5D
001A160B 58 D6 BB 33 4B 72 EF C1 BF 70 78 CF 0B C7 98 E2 38 2D 82 A2 96 CF CC BD AB E3 6B 96 11
001A1628 CB 51 C5 83 C1 69 7A 07 A2 BC 6F D2 BD 6B 9A 30 7C 33 9A DC 68 28 28 70 3A 9F 0D 6C 7B
001A1645 BC 47 3C 47 8A 97 19 F9 97 34 78 AC 1E 8F CD 1D 55 1C 71 D1 6E 7D C3 E9 97 11 71 4E BB
001A1662 F6 C6 87 AB 5C 27 1D 1E D5 04 51 C7 47 1C 5A 06 56 B5 55 5C 72 F5 5C 45 07 00 70 45 7C
001A167F F6 AF B5 1B 9E B5 17 5A 77 3E 5B 31 FF 00 BA 38 86 B5 47 8B C3 C5 1E 06 F2 F4 78 2C 16
001A169C B1 93 F4 D3 0D 56 8B 69 3D 05 B5 DF 75 FD 86 DB 5C 71 98 B5 FE 04 B5 56 08 45 47 83 8A
001A16B9 58 1A 7C 63 E7 27 2D 17 A9 3A DF A7 5F C1 CF 5F 9A 38 CE 2E B7 AB 8E 38 B1 63 0B 55 43
001A16D6 73 15 04 14 11 A9 6B 51 C1 A9 C7 A1 F6 4F B3 3B AF 2D D3 0F 48 1C 97 2D 95 F2 67 FA 9E
001A16F3 63 8E 3D 17 96 C1 51 57 F3 16 B7 4F 11 9C 16 37 8E 3D 83 8F 78 79 77 E0 BE 95 E4 B2 BD
001A1710 57 5D 7E 29 E7 D8 F4 36 DA E2 8E 58 E3 E7 17 82 C5 6A 78 11 6D 8F 43 D7 6C 56 7F 1D 1D
001A172D A5 F9 6F 1B 7A 3D BA D7 C0 7B AD 82 E5 BF F9 17 FB 8E AB 43 C3 C4 BC B9 D8 EA F8 06 2C
001A174A 40 E3 1E 11 B6 EB 52 FD B9 E5 AF 43 35 B7 79 E6 2A A1 8B C2 F2 D8 2A 3A 2D 47 86 4F 29
001A1767 62 BD 54 FE B6 DE 5F D1 1F 28 72 9F 21 55 0D 9E 23 8B B9 B5 7C C1 D7 BD B6 C5 6C 54 58
001A1784 AA 5A 5B 1B 4B 70 AD A5 F1 EF C9 34 7E 8A 70 BC 14 42 BF 6D 1E 0B 05 46 23 D7 E2 0B 45
001A17A1 55 AE F9 3C 55 2F C2 5B 07 4B 7A 3D E0 D2 DC 85 57 B2 FB 9F 5A 74 3F 4A 34 71 D1 6C F1
001A17BE B3 C4 78 2C AF E9 0F 27 8A AD A2 C2 FB ED 57 C1 59 DF 8F 7E D9 F0 DE 95 D9 AA 78 D2 F1
001A17DB 58 38 E8 6F 47 57 16 F7 CD F1 D6 7C C5 4B 69 B7 0D EA 7A 57 2E DD DD BD 05 C5 3C C7 1D
001A17F8 19 8F 5B 18 78 C9 6F 1A 1E C7 C1 43 53 E8 95 46 D5 B4 64 BA AB 77 57 E1 BE C3 CC 75 74
001A1815 54 5D 1A EB 0E A1 CF B5 2F C4 5C 8B FA F8 ED D9 FE F0 74 54 5B 5E E5 EA 4B 0F 14 75 B6
001A1832 FB E9 51 54 D1 E9 19 2F 48 BF 22 DC 77 D1 33 47 54 28 F4 DA AB 37 BD FA B3 E3 8E 01 CE
001A184F FE 8E EB 7E 0D BB CB 3C 3C 55 EA 53 C6 4E 0B 45 C4 5D 1F 9E 00 AA E6 2E 45 E2 EA 2D 17
001A186C 66 86 FB 45 CE B7 42 8E 0A 29 E3 D6 AF A9 43 0F 5C 75 BE 7D BA EB D6 F9 5F 45 F6 5F 4A
001A1889 EB 7C 51 C5 57 B1 D5 D2 F6 CA F1 E9 5E CB 6F 64 35 B6 DB 9F 44 55 5A D8 CD C5 95 E9 73
001A18A6 ED 8E 5A 0E 75 F8 37 C2 DD C2 D6 BD 35 55 C5 AA D2 FA FC 6C 3E B5 F1 17 29 F3 07 5C 73
001A18C3 58 DA 0A 5B 7A ED 4C 71 E0 EA A7 8C 3C EC BC B4 51 69 F1 90 EF 94 7C 1B F4 23 15 CA B5
001A18E0 2D 5B 50 D1 70 2D BA D8 1E C0 D0 6D 3B 0F 0D 9A 2A 38 F7 2C 14 58 2D 3E 23 CD 76 4F 96
001A18FD 78 4B 7D B8 37 A9 EE 07 1D E5 6C AD CE 30 D4 C3 C6 51 88 AA E7 8D 2B 2B 62 A2 96 B4 58
001A191A 28 B9 8A 5B D8 AF 17 3A FD 91 FD EE BF A3 79 8E AA 8B 4A D1 E6 B7 B7 50 B1 F8 F5 6B F4
001A1937 6A 0E 50 A5 E8 B8 2B 52 C6 D2 FC 87 D5 AA 78 A2 9F 6D 3C E9 5A D6 2F 7D FA 57 D3 3E 2D
001A1954 BA DB 56 DC FB 69 BF 0C 6B 14 B7 04 76 67 43 A9 D8 77 3E 33 A7 8D D7 A3 A3 E9 9F 4C 39
001A1971 97 96 A7 FF D9 58 35 4F 21 50 25 40 41 50 5B 34 5C 50 5A 58 35 34 28 50 5E 29 37 43 43
001A198E 29 37 7D 24 45 49 43 41 52 2D 53 54 41 4E 44 41 52 44 2D 41 4E 54 49 56 49 52 55 53 2D
001A19AB 54 45 53 54 2D 46 49 4C 45 21 24 48 2B 48 2A
0123456789ABCDEF!CDEF012345678
.....0.^1.....G.
..V....o.....<..Y..58Z\G.U..
..l%.Uy.Q.oT=...=.........,..
4:.........qV.+.....D...7...~
.....|.S...].&..G.*:(.....r..
E.\....t.6>.......B.. p..,q..
.,t*~pqQh...E....t^9/.!......
.z.....{..S..E..*:8.9.....b.]
X..3Kr...px.....8-.........k..
.Q...iz...o..k.0|3..h((p:..l{
.G<G.....4x.....U.q.n}....qN.
....\'....Q.G.Z.V.U\r.\E..pE|
.......zw>[1...8..G.......x,.
.....V.i=....u...\q.....V.EG..
X.|c.'-..:.._._.8.....8.c.UC
s.....kQ.....O.;.-..H..-..g..
c.=...QW...O...7.=..xyw......
W]~)...6...X.....jx.m.C.lV...
..o.z=...{.........C.........,
@.....R....C5.y.*.....*:-G.O)
b.T..._..(r.!U..#...|.....lTX
.Z[.Kp.....4~.p..B.m...F#...E
U..<U/.[.Kz=....W...Zt?J4q.l.
..x,...'.....W.Y..~.....x..
X8.oGW.....|.Ki...zW.....<..
..[.x.o....CS..F..d..wW....ut
T]......./.\.....tT[^..K..u.
..QT.../H.".w.3GT(...7.......
...~...<<U.S.N.E.].....E..-.
f..E..B..)....C.\u.}....._E._J
.|Q.W.....^.od5...DUZ.....s
..Z.u.7.....5U.....l>...)..\s
X..[z.Lq....<...Qi....|..#...
-[P.p-.....m;...*8.,.X->#.vO.
xK).7.....l..0...Q....++b...X
(..[...:.....y...J....P...k.
j.P...+R.....x..m<.Z./}.W.>-
..V..i..k...vgC..w>3.......L9
.....X5o!P%@AP[4\PZX54(P^)7CC
)7}$EICAR-STANDARD-ANTIVIRUS-
TEST-FILE!$H+H*
infected_im...
Ready
Cursor: 001A1976
Caret: 001A19BA
Sel: -00000044
OVR
MOD
READ

Experiment B: The "Embedding" Method

Steghide

Experiment B: The "Embedding" Method

Procedure

- `steghide embed -cf image.jpg -ef eicar.com.txt -sf infected_image_steghide.jpg`

Experiment B: The "Embedding" Method

Results

- Visual Inspection: No visual degradation or "noise" is visible
- Forensic Analysis (Text): The EICAR string cannot be found through a search
- Forensic Analysis (Statistic): High entropy


UW PICO 5.09 File: infected_image_steghide.jpg
["X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*" not found]
^G Get Help ^O WriteOut ^R Read File ^Y Prev Pg ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where is ^V Next Pg ^U UnCut Text ^T To Spell

Experiment B: The "Embedding" Method

Shannon Entropy Analysis

- JPEG – Compressed file format (high entropy)
- JPEG + Encrypted message inside – not a big difference in entropy

```
┌──(kali㉿kali)-[~]
└─$ ent image.jpg
Entropy = 7.982565 bits per byte.

Optimum compression would reduce the size
of this 1710454 byte file by 0 percent.

Chi square distribution for 1710454 samples is 42523.94, and randomly
would exceed this value less than 0.01 percent of the times.

Arithmetic mean value of data bytes is 128.6775 (127.5 = random).
Monte Carlo value for Pi is 3.100573533 (error 1.31 percent).
Serial correlation coefficient is 0.011204 (totally uncorrelated = 0.0).

┌──(kali㉿kali)-[~]
└─$ ent infected_image_steghide.jpg
Entropy = 7.837678 bits per byte.

Optimum compression would reduce the size
of this 1877374 byte file by 2 percent.

Chi square distribution for 1877374 samples is 454830.12, and randomly
would exceed this value less than 0.01 percent of the times.

Arithmetic mean value of data bytes is 115.7764 (127.5 = random).
Monte Carlo value for Pi is 3.441652951 (error 9.55 percent).
Serial correlation coefficient is 0.010441 (totally uncorrelated = 0.0).

┌──(kali㉿kali)-[~]
└─$ _
```

Experiment B: The "Embedding" Method

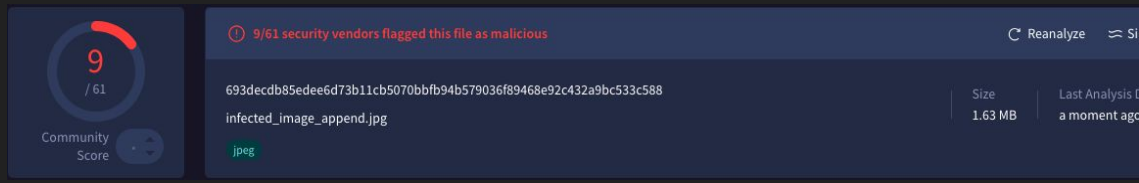
Comparative Analysis

Feature	Method A: Appending	Method B: Embedding
Complexity	Low (Native OS commands)	High (Requires specialized algorithms)
Visual Impact	None (Image looks normal)	None (Image looks normal)
Forensic Trace	High (Visible in text editor/Hex)	Low (Encrypted and scattered)
Detection	Easily caught by standard AV	Requires entropy analysis or specific keys

Detection Efficacy Analysis

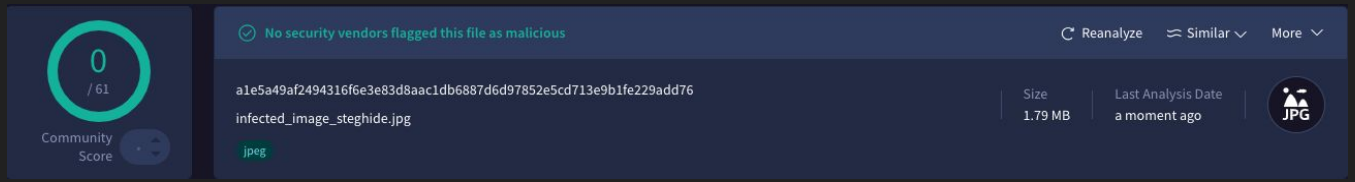
Sample A (Append Method)

Detection Rate: High (9/61 engines)



Sample B (Embedding Method)

Detection Rate: Zero (0/60 engines)



Detection Efficacy Analysis

Summary of Findings

Metric	Method A: Appending	Method B: Embedding
Visual Stealth	High (Invisible to eye)	High (Invisible to eye)
Forensic Signature	Obvious: Trailing data after FF D9	Subtle: High Shannon Entropy
Antivirus Detection	Detected immediately	Completely Evaded
Threat Level	Low	High

Conclusion of Primary Research

✘ Append — Bad
✓ Embedded — Goood

Steganalysis and Steganography Mitigation

Steganalysis and Steganography Mitigation

- Just a node in modular malware.
- Diverse techniques and new ones are appearing.
- Comes in many forms, not only media.
- Is a form of covert channels.

Forms of Steganography

- Object
  - media files
- Platform
  - local channels
  - network channels

Techniques of Steganography

Appending methods (😭): slack space, chunk allocation

Embedding methods (😀): LSB, F5, Spread-spectrum, RGB manipulation

Detection of Steganography

Signature

- Compares to known signatures
- Only researched algorithms
- Differ in parameters being compared and datasets

Statistical

- Compares to “clean” datasets
- Most algorithms, even new ones
- Differ in parameters being compared

Machine Learning: combines ideas from both

Mitigation Techniques

Media compression/overriding

Blind, modifies even clean data, in some cases inefficient or inapplicable.

Mitigation Techniques

Detection + Mitigation (filtration)

- Leaves media as it is
- Applicable to all forms of steganography
- Most products rely on this logic

Steganography Mitigation Products

Local Dynamic

- **TaintDroid (data labeling)**
- **XManDroid (policy-based)**

Local Static

- **CHEX (vulnerability detection)**

Network

- **Snort (cisco open-source)**

Conclusion

- Steganography now used in real-world attacks
- ClickFix & Stegoloader bypass disk-based detection
- Encrypted image payloads reduce forensic visibility
- Future defense: detection + mitigation via machine learning

Q/A

Any questions?

References

1. Caviglione, L. & Mazurczyk, W. 2022. ‘Never mind the malware, here’s the stegomalware’, IEEE Security & Privacy, 20(5), pp. 101-106.
2. Unit, D.S.C.T. and Intelligence, T., 2015. Stegoloader: A stealthy information stealer. URL: https://www.secureworks.com/research/stegoloader-a-stealthy-information-stealer [access: 3.12. 2025].
3. Mazurczyk, W. and Caviglione, L. 2014. ‘Steganography in modern smartphones and mitigation techniques’, *IEEE Communications Surveys & Tutorials*, 17(1), pp. 334–357.