Stegomalware: what is it and what we can do

Luca Caviglione

Institute for Applied Mathematics and Information Technologies

luca.caviglione@ge.imati.cnr.it

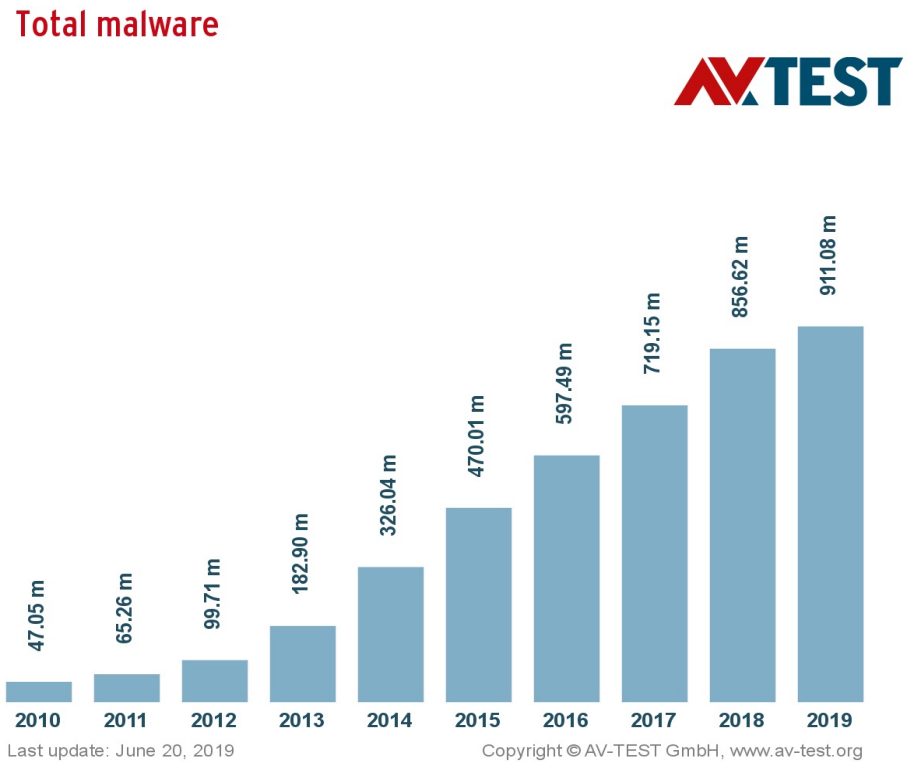
CUING 2021 - 5th International Workshop on Criminal Use of Information Hiding

Outline

- Some Facts About Malware
- What is Stegomalware?
- What We Can Do
- Conclusions

Some Facts About Malware

- Exponential grow of malicious software



Some Facts About Malware

- Exponential grow of malicious software
- Despite the effort of many security experts and researchers:
  - countermeasures are progressively showing limitations
  - only a fraction of threats is detected
  - malware increasingly operates **undisturbed** for **longer timeframes**

Malware	Discovered	Present since...
Stuxnet	06.2010	2007
Duqu	04.2011	2008
Flame	05.2012	2007
The Mask	2013	2007
Regin	2014	2003

Some Facts About Malware

- Exponential grow of malicious software
- Despite the effort of many security experts and researchers:
  - countermeasures are progressively showing limitations
  - only a fraction of threats is detected
  - malware increasingly operates **undisturbed** for **longer timeframes**
- How can malware developers avoid detection for long periods?

Giving an answer is **not** simple!

Some Facts About Malware

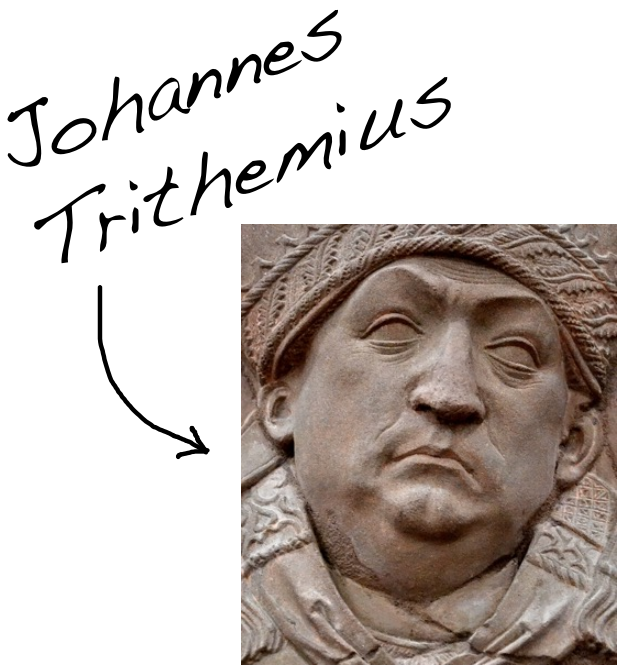
- Some possible reasons are:
  - Modular design for customization (e.g., Regin, Flamer, Weevil)
  - Multistage loading (e.g., Regin, Stuxnet, Duqu)
  - Cybercrime-as-a-Service models and Remote Access Trojans (e.g., Gh0st Rat)
  - **Information Hiding** techniques and **steganography** (e.g., Platinum APT)

Is the use of Information Hiding or steganography just a passing thing?

One Step Back: Information Hiding

- **Information Hiding** is part of a wide spectrum of methods that are used to make secret data difficult to notice
- **Steganography** is one of the most well-known subfields of Information Hiding
- Steganography vs **Cryptography**:
  - Steganography: information is difficult to notice
  - Cryptography: information is difficult to comprehend
- Information Hiding and cryptography can be used **jointly**

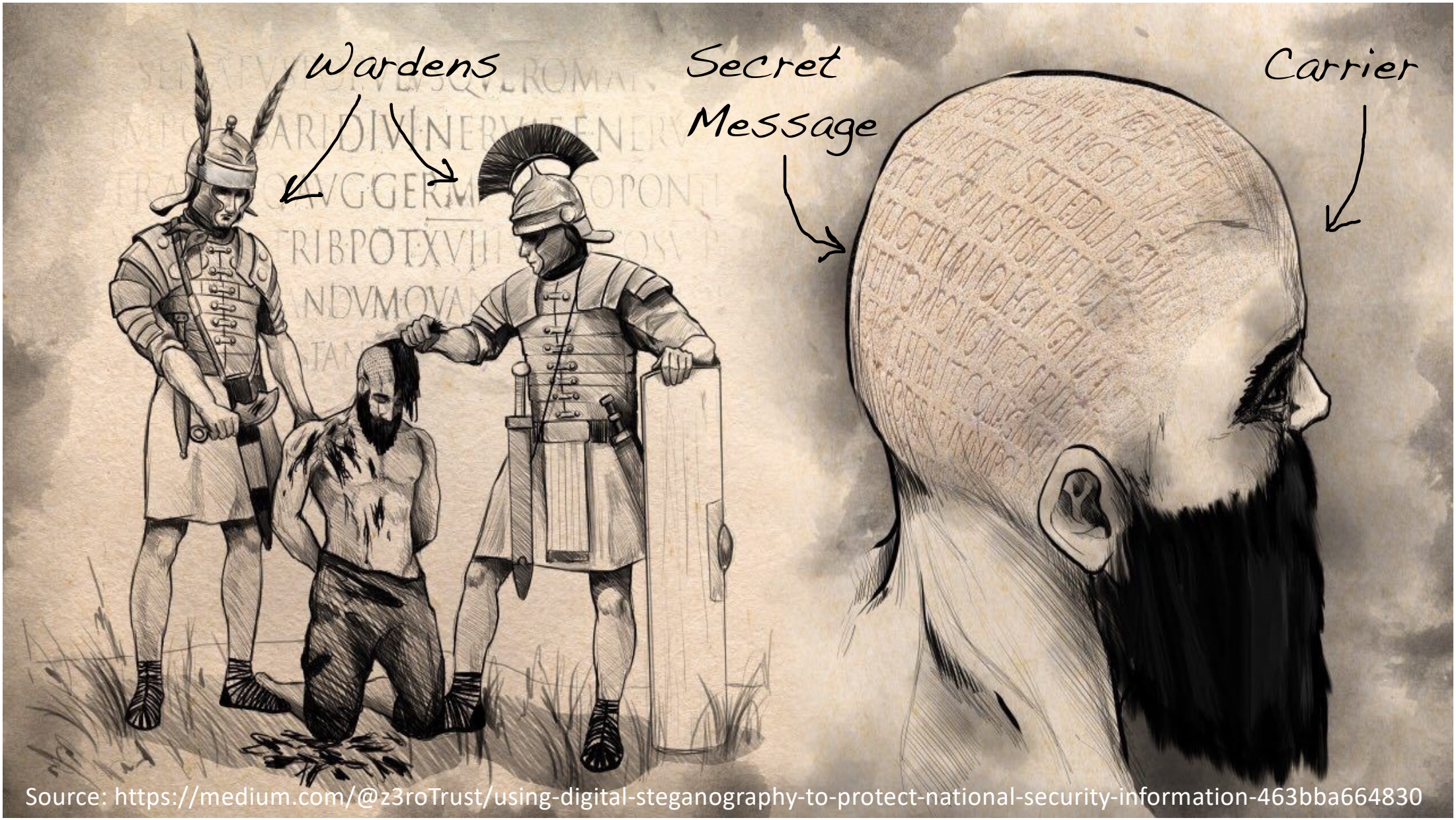
One Step Back: Steganography



Steganography

steganos (στεγᾰνός) + graphe (γραφή)

- The word **steganography** is the combination of:
  - *steganos* = covered, concealed
  - *graphe* = writing
- The first recorded use of the term:
  - in 1499 by Johannes Trithemius
  - book “Steganographia”, i.e., an essay on cryptography and steganography
- Mentioned in 440 BC by Herodotus in his Histories.
- **Cloak secret data into a suitable carrier**



Source: https://medium.com/@z3roTrust/using-digital-steganography-to-protect-national-security-information-463bba664830

Back to Malware…

- Information Hiding techniques have been **increasingly observed** in malicious software, for instance to:
  - elude detection techniques
  - covertly spread an infection or orchestrate attacks
  - exfiltrate sensitive data
  - bypass sandboxing mechanisms
  - implement covert channels
  - …

Back to Malware…

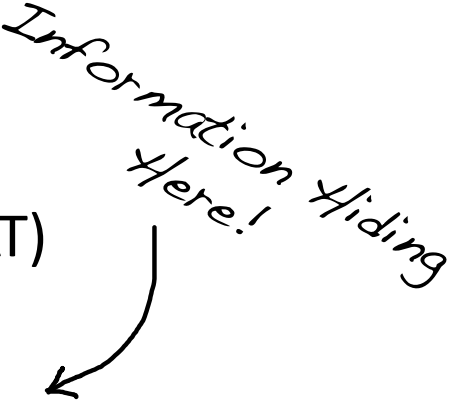
- Information Hiding techniques have been **increasingly observed** in malicious software, for instance to:
  - elude detection techniques
  - covertly spread an infection or orchestrate attacks
  - exfiltrate sensitive data
  - bypass sandboxing mechanisms
  - implement covert channels
  - …

Is the use of Information Hiding or steganography just a passing thing? (again)

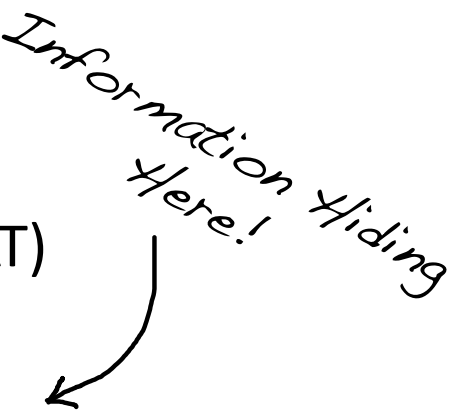

NO!

The Root of a Trend?

- Probably, Trojan.Downbot (circa 2006, Operation Shady RAT)
- The trojan created a back door and:
  - downloaded files appearing as real HTML pages or JPEG images
  - hidden data contained commands for remote servers

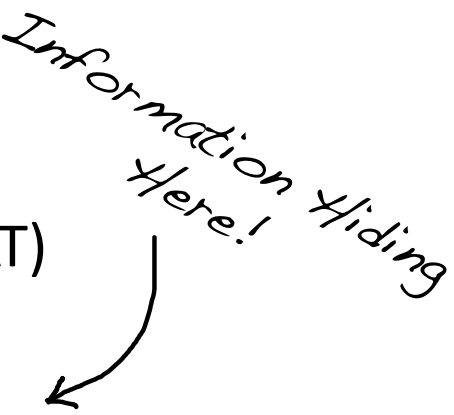


The Root of a Trend?



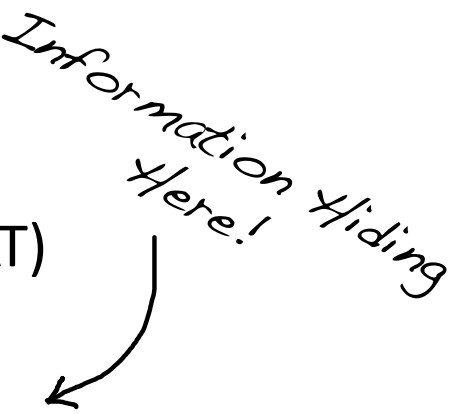
- Probably, Trojan.Downbot (circa 2006, Operation Shady RAT)
- The trojan created a back door and:
  - downloaded files appearing as real HTML pages or JPEG images
  - hidden data contained commands for remote servers
- Three attack stages:
  - Stage 1: phishing!

The Root of a Trend?



- Probably, Trojan.Downbot (circa 2006, Operation Shady RAT)
- The trojan created a back door and:
  - downloaded files appearing as real HTML pages or JPEG images
  - hidden data contained commands for remote servers
- Three attack stages:
  - Stage 1: phishing!
  - Stage 2: the trojan attempts to retrieve data from remote sources

The Root of a Trend?



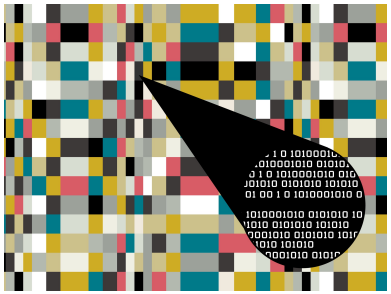
- Probably, Trojan.Downbot (circa 2006, Operation Shady RAT)
- The trojan created a back door and:
  - downloaded files appearing as real HTML pages or JPEG images
  - hidden data contained commands for remote servers
- Three attack stages:
  - Stage 1: phishing!
  - Stage 2: the trojan attempts to retrieve data from remote sources

```
www.comto[SANITIZED].com/wak/mansher0.gif
www.kay[SANITIZED].net/images/btn_topsec.jpg
www.swim[SANITIZED].net/images/sleepyboo.jpg
www.comto[SANITIZED].com/Tech/Lesson15.htm
```

The Root of a Trend?

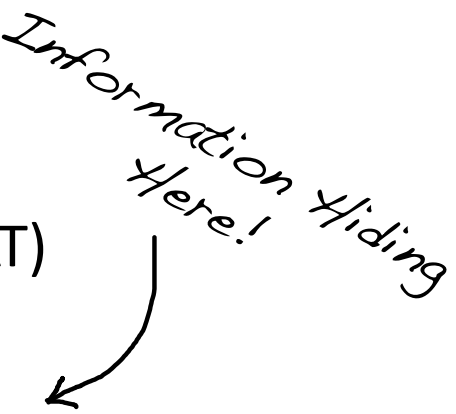
Information Hiding Here!

- Probably, Trojan.Downbot (circa 2006, Operation Shady RAT)
- The trojan created a back door and:
  - downloaded files appearing as real HTML pages or JPEG images
  - hidden data contained commands for remote servers
- Three attack stages:
  - Stage 1: phishing!
  - Stage 2: the trojan attempts to retrieve data from remote sources



Commands hidden in images via steganographic techniques

The Root of a Trend?

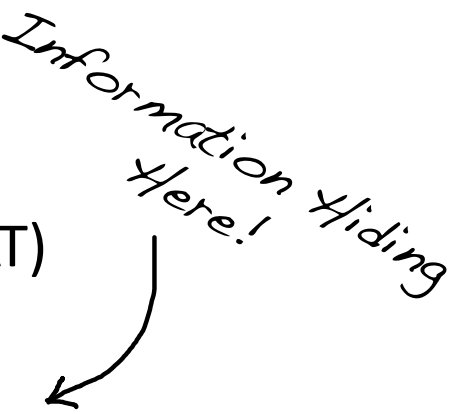


- Probably, Trojan.Downbot (circa 2006, Operation Shady RAT)
- The trojan created a back door and:
  - downloaded files appearing as real HTML pages or JPEG images
  - hidden data contained commands for remote servers
- Three attack stages:
  - Stage 1: phishing!
  - Stage 2: the trojan attempts to retrieve data from remote sources

```
<!-- {685DEC1O8DA731F1} -->
<!-- {685DEC1O8DA73CF1} -->
<!-- {eqNBb-OuO7WM} -->
<!-- {eqNBb-OuO7iM} -->
<!-- {eqNBb-OuOlOMOO++} -->
<!-- {eqNBb-Ou1lO+} -->
<!-- {eqNBb-Ou2Ra+} -->
<!-- {uGu~iWAl,Q(iNyn'/} -->
<!-- {ujQ~iY,UnQ[!,hboZWg} -->
<!-- {ujQ~iY,UnQ[!,hmoZWg} -->
<!-- {ujQ~iY,UnQ[!,hvoZWg} -->
```

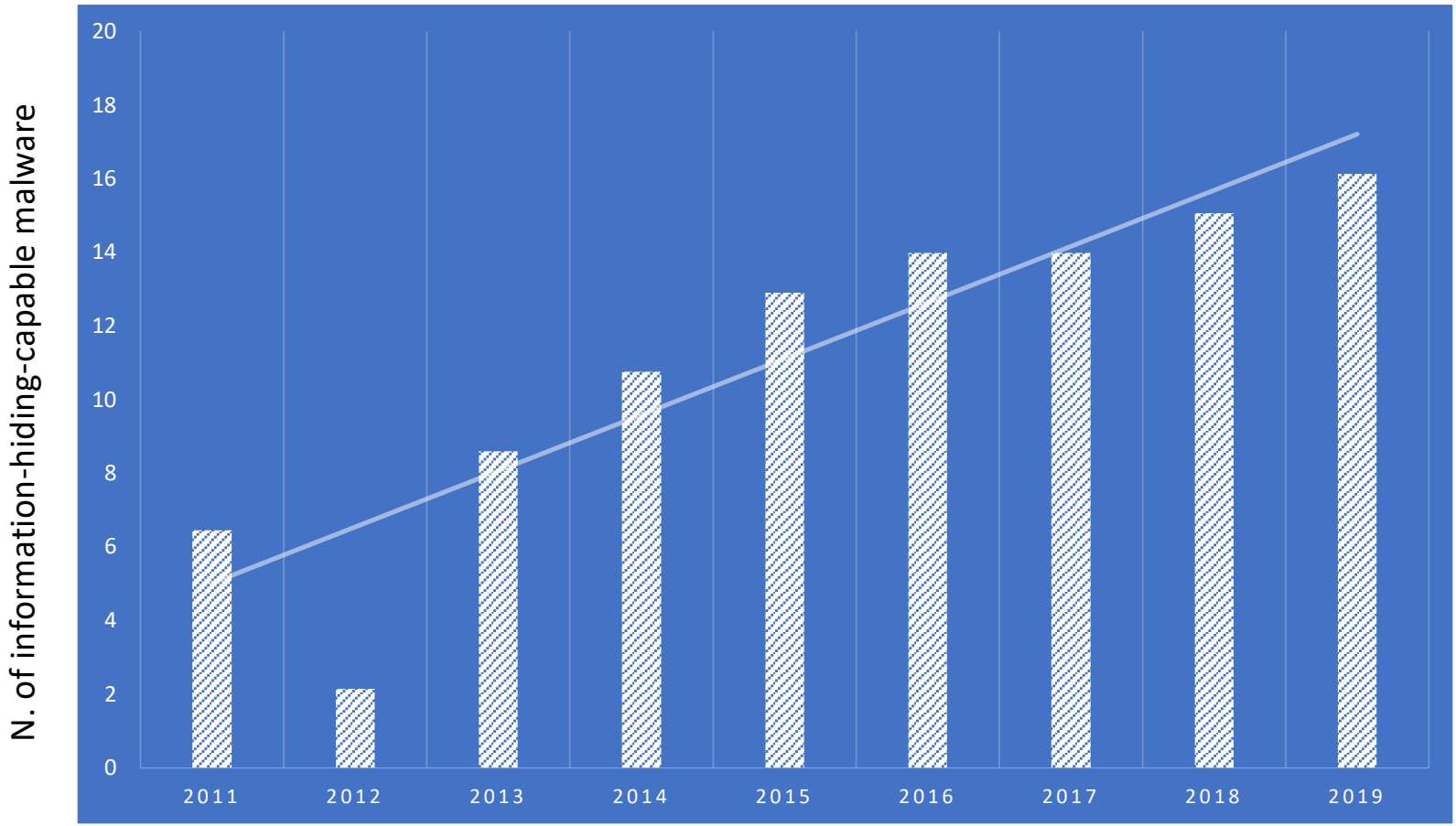
Commands hidden in HTML comments
(encrypted + base64 encoded)

The Root of a Trend?



- Probably, Trojan.Downbot (circa 2006, Operation Shady RAT)
- The trojan created a back door and:
  - downloaded files appearing as real HTML pages or JPEG images
  - hidden data contained commands for remote servers
- Three attack stages:
  - Stage 1: phishing!
  - Stage 2: the trojan attempts to retrieve data from remote sources
  - Stage 3: the trojan connects to a host and sets up a remote shell waiting for commands

Yes: it is a Trend!



Data collected by members of Criminal Use of Information Hiding initiative (CUING): cuing.org and cuing.eu

Yes: it is a Trend!

The impact of information-hiding-capable malware is heavily underestimated: security experts often do not correctly recognize and classify the used techniques

Stegomalware

- Many researchers are starting to identify this class of threats as:
  - **Stegomalware**: steganographic malware
  - "borrowed" from works on mobile security and covert social botnets*
- A possible (common) definition:
  - Stegomalware is a malware using some form of steganography to remain undetected

G. Suarez-Tangil, J. Tapiador, P. Peris-Lopez, "Stegomalware: Playing Hide and Seek with Malicious Components in Smartphone Apps", 10th International Conference on Information Security and Cryptology, pp. 496–515, 2014.
S. Nagaraja, A. Houmansadr, P. Piyawongwisal, V. Singh, P. Agarwal, B. Nikita, "Stegobot: A Covert Social Network Botnet", Information Hiding - 13th International Conference, pp. 299 – 313, 2011.

Stegomalware

- Many researchers are starting to identify this class of threats as:
  - **Stegomalware**: steganographic malware
  - "borrowed" from works on mobile security and covert social botnets*
- A possible (common) definition:
  - Stegomalware is a malware using some form of steganography to remain undetected
- Personally, I found it a bit reductive:
  - it narrows the scope too much
  - a bit ambiguous (Information Hiding vs steganography)
  - it is not only about detection (e.g., colluding applications)

G. Suarez-Tangil, J. Tapiador, P. Peris-Lopez, "Stegomalware: Playing Hide and Seek with Malicious Components in Smartphone Apps", 10th International Conference on Information Security and Cryptology, pp. 496–515, 2014.
S. Nagaraja, A. Houmansadr, P. Piyawongwisal, V. Singh, P. Agarwal, B. Nikita, "Stegobot: A Covert Social Network Botnet", Information Hiding - 13th International Conference, pp. 299 – 313, 2011.

Classification

- In 2015, a relevant corpus of research on “stegomalware” has started to emerge
- It considered:
  - attacks observed in the wild
  - prototypal/lab threats to explore new/potential vulnerabilities
- What has been inspected:
  - samples of real threats
  - attack reports
  - reversed binaries
  - …
- The following classification of malware using Information Hiding has been proposed*

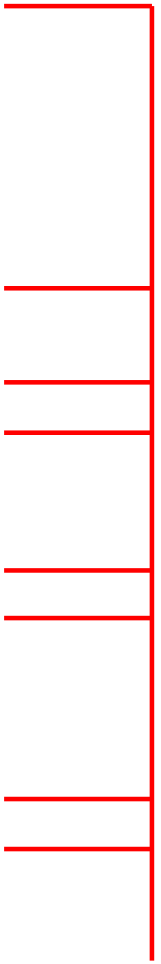
W. Mazurczyk, L. Caviglione, “Information Hiding as a Challenge for Malware Detection”, IEEE Security & Privacy, Vol. 13, No. 2, pp. 89-93, Mar.-Apr. 2015.

Classification

- **Group 1**: malware hiding information by modulating the status of shared hardware/software resources
- **Group 2**: malware injecting secret data into network traffic
- **Group 3**: malware embedding secret data by modifying a digital file structure or by using digital media steganography

W. Mazurczyk, L. Caviglione, “Information Hiding as a Challenge for Malware Detection”, IEEE Security & Privacy, Vol. 13, No. 2, pp. 89-93, Mar.-Apr. 2015.

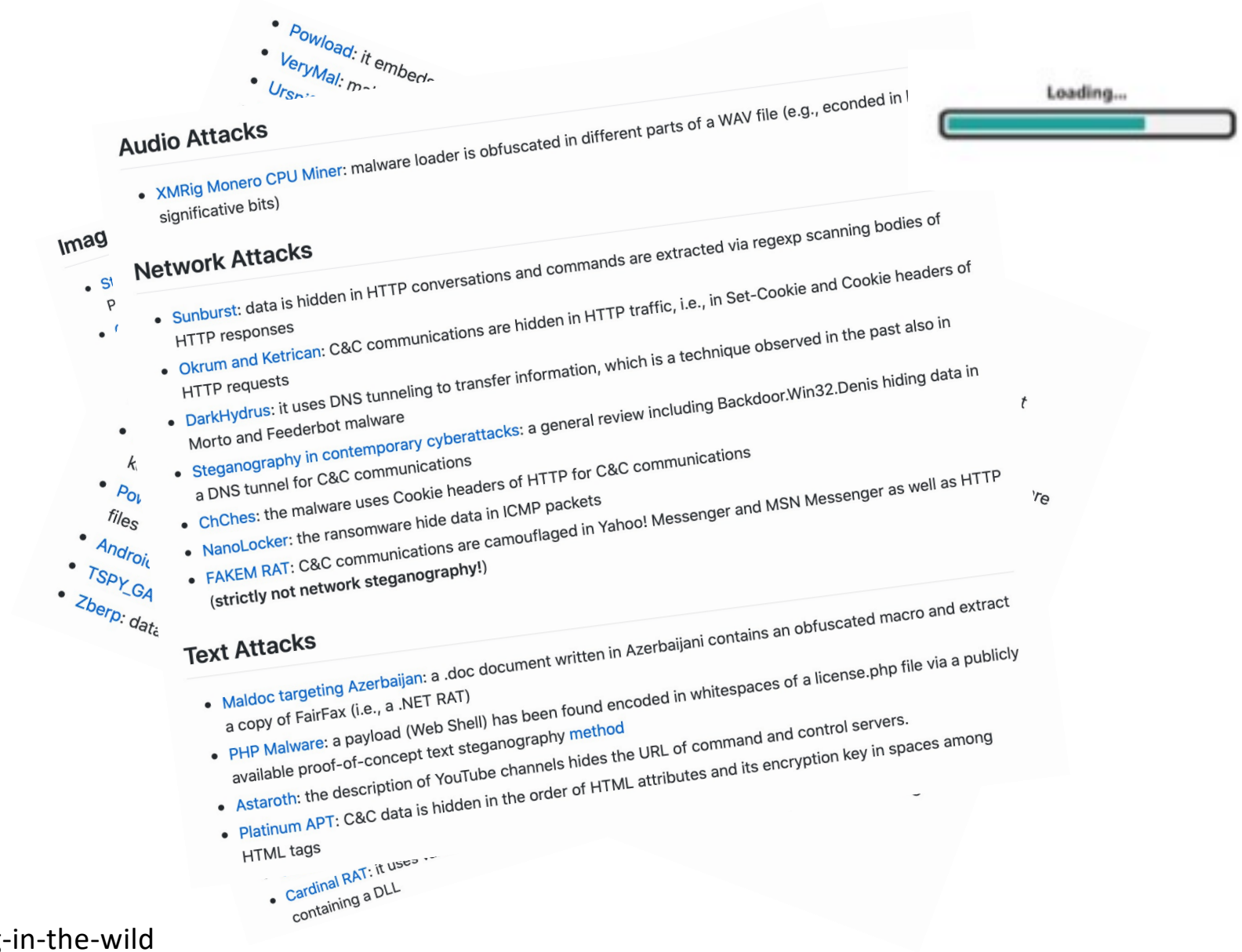
Malware name or developers	Group	Discovery/ proposal date	Desktop (D) or mobile (M)	Real-life (R) or academic (A) malware
Soundcomber	1, 2	Feb. 2011	M	A
Trojan.Downbot	3	May 2011	D	R
Feederbot	2	Aug. 2011	D	R
W32.Morto	2	Aug. 2011	D	R
Alureon	3	Sept. 2011	D	R
Duqu	3	Sept. 2011	D	R
Gasior and Yang[14,15]	2	Oct. 2011/Dec. 2012	M	A
Trojan:Android/FakeRegSMS.B	3	Jan. 2012	M	R
Marforio and his colleagues[16]	1	Dec. 2012	M	A
Sensor-based malware	1	May 2013	M	A
KINS Trojan (variant of Zeus)	3	June 2013	D	R
Linux.Fokirtor	2	Sept. 2013	D	R
Lalande and Wendzel[17]	1	Sept. 2013	M	A
Inaudible sound-based malware	1	Nov. 2013/Aug. 2014	D/M	A
Lurk	3	Feb. 2014	D	R
Trojan.Zbot	3	Mar. 2014	D	R
Oldboot.B	3	Apr. 2014	M	R
AirHopper	1	Oct. 2014	D/M	A
Smuggler[18]	2	Nov. 2014	D/M	A
Multilayer .NET malware	3	Nov. 2014	D	R
Regin	2	Nov. 2014	D	R



Most popular (and known) malware using Information Hiding in 2015

W. Mazurczyk, L. Caviglione, “Information Hiding as a Challenge for Malware Detection”, IEEE Security & Privacy, Vol. 13, No. 2, pp. 89-93, Mar.-Apr. 2015.

Updating



https://github.com/lucacav/steg-in-the-wild

Stegomalware in the Wild

- In general, real stegomalware exploits:
  - digital images (about 40%)
  - alteration of a digital content, e.g., file structure (about 28%)
  - network traffic (about 32%)
- The taxonomy of 2015 should be refined:
  - more focused on files rather than generic hardware/software artifacts
  - better highlight the domains exploiting digital images
- Examples:
  - images containing steganographic data to implement C&C communications
  - images for spreading an attack or dropping a payload
  - images used to locally obfuscate files
  - where the information is hidden
  - …

SteamHide: https://www.gdatasoftware.com/blog/steamhide-malware-in-profile-images
OceanLotus: https://www.bleepingcomputer.com/news/security/oceanlotus-apt-uses-steganography-to-load-backdoors/

Example: ZeusVM

- Discovered in 2014, it is an evolution of the Zeus/Zbot malware
- A variant has been also used in the Hammertoss APT isolated in 2015
- Attack phases:
  - the malware downloads an innocent JPG from a C&C server

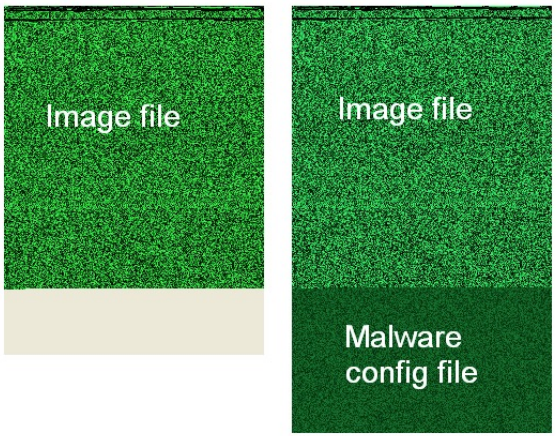
Example: ZeusVM

- Discovered in 2014, it is an evolution of the Zeus/Zbot malware
- A variant has been also used in the Hammertoss APT isolated in 2015
- Attack phases:
  - the malware downloads an innocent JPG from a C&C server

Source: https://blog.malwarebytes.com/threat-analysis/2014/02/hiding-in-plain-sight-a-story-about-a-sneaky-banking-trojan/

Example: ZeusVM

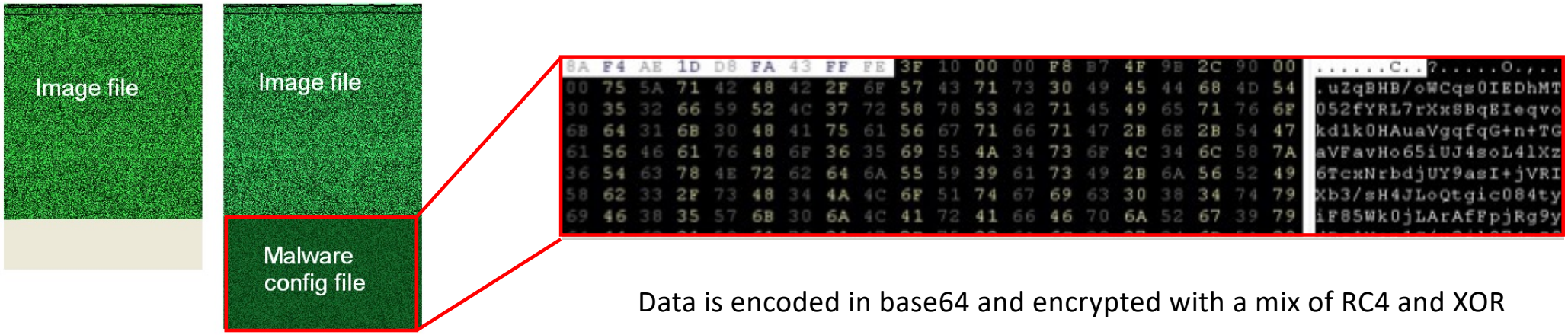
- Discovered in 2014, it is an evolution of the Zeus/Zbot malware
- A variant has been also used in the Hammertoss APT isolated in 2015
- Attack phases:
  - the malware downloads an innocent JPG from a C&C server
  - the image perfectly works but a configuration file is appended



Source: https://blog.malwarebytes.com/threat-analysis/2014/02/hiding-in-plain-sight-a-story-about-a-sneaky-banking-trojan/

Example: ZeusVM

- Attack phases:
  - the malware downloads an innocent JPG from a C&C server
  - the image perfectly works but a configuration file is appended



Data is encoded in base64 and encrypted with a mix of RC4 and XOR

Source: https://blog.malwarebytes.com/threat-analysis/2014/02/hiding-in-plain-sight-a-story-about-a-sneaky-banking-trojan/

Example: ZeusVM

- Attack phases:
  - the malware downloads an innocent JPG from a C&C server
  - the image perfectly works but a configuration file is appended

Source: https://blog.malwarebytes.com/threat-analysis/2014/02/hiding-in-plain-sight-a-story-about-a-sneaky-banking-trojan/

Example: ZeusVM

- Attack phases:
  - the malware downloads an innocent JPG from a C&C server
  - the image perfectly works but a configuration file is appended
  - trojan activates when traffic with the financial institutions provided in the configuration file is sensed
  - it steals user credentials by acting in a MitM fashion

Another Example: Invoke-PSImage

- Invoke-PSImage is a tool for encoding a PowerShell Script in pixels of a PNG image
- It uses Least Significant Bit (LSB) steganography

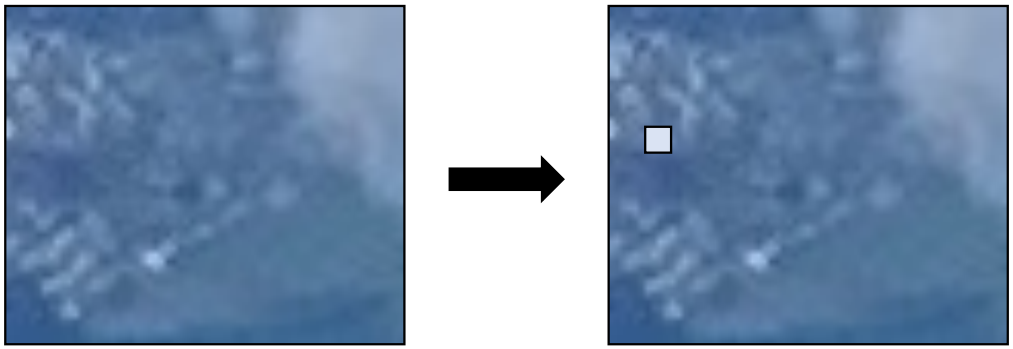
Invoke-PSImage: https://github.com/peewpw/Invoke-PSImage

One Step Back: LSB Steganography

MSB LSB

1	1	0	0	1	1	1	0	1
0	1	0	1	0	1	0	0	Green
1	1	1	1	0	0	1	1	Blu

Hide Secret Here



Another Example: Invoke-PSImage

- Invoke-PSImage is a tool for encoding a PowerShell Script in pixels of a PNG image
- It uses Least Significant Bit (LSB) steganography
- Invoke-PSImage has been released in Dec. 2017 and it has been used for a malware campaign just 1 week later
- Example:
  - Mimikatz
  - Ursnif

Invoke-PSImage: https://github.com/peewpw/Invoke-PSImage

Another Example: Invoke-PSImage

- Invoke-PSImage is a tool for encoding a PowerShell Script in pixels of a PNG image
- It uses Least Significant Bit (LSB) steganography
- Invoke-PSImage has been released in Dec. 2017 and it has been used for a malware campaign just 1 week later
- Attack Phases:
  - infected Excel is used to launch a malicious VB macro

Invoke-PSImage: https://github.com/peewpw/Invoke-PSImage

Another Example: Invoke-PSImage

- Invoke-PSImage is a tool for encoding a PowerShell Script in pixels of a PNG image
- It uses Least Significant Bit (LSB) steganography
- Invoke-PSImage has been released in Dec. 2017 and it has been used for a malware campaign just 1 week later
- Attack Phases:
  - infected Excel is used to launch a malicious VB macro
  - the macro downloads an image containing a PowerShell script

Invoke-PSImage: https://github.com/peewpw/Invoke-PSImage

Another Example: Invoke-PSImage

- Invoke-PSImage is a tool for encoding a PowerShell Script in pixels of a PNG image
- It uses Least Significant Bit (LSB) steganography
- Invoke-PSImage has been released in Dec. 2017 and it has been used for a malware campaign just 1 week later
- Attack Phases:
  - infected Excel is used to launch a malicious VB macro
  - the macro downloads an image containing a PowerShell script
  - the script is extracted and launched to retrieve the Ursnif loader

Invoke-PSImage: https://github.com/peewpw/Invoke-PSImage

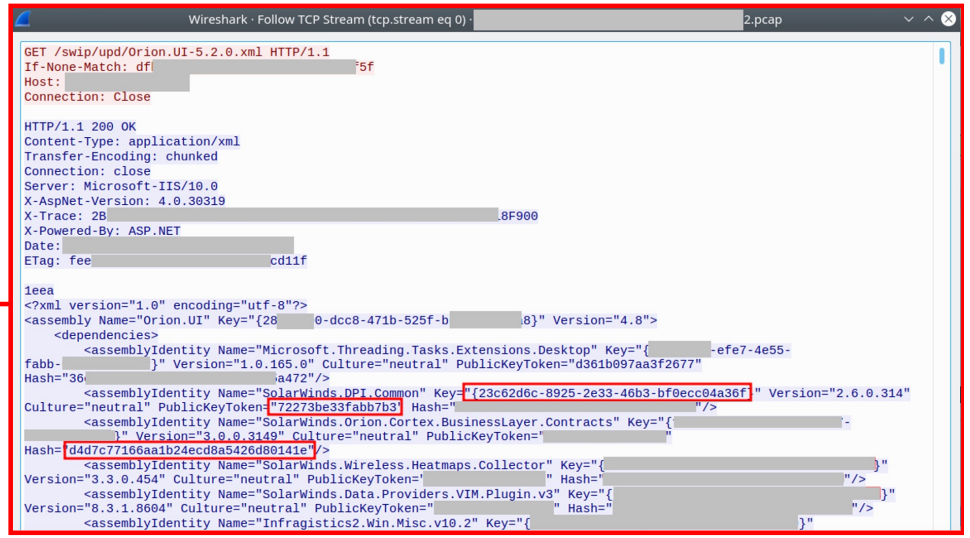
Yet Another Example: Sunburst

- Sunburst is a trojanized version of the Orion plugin (Solarwind)
- It targets HTTP traffic
- Attack Phases:
  - various checks to understand if an analysis tool is running

Yet Another Example: Sunburst

- Sunburst is a trojanized version of the Orion plugin (Solarwind)
- It targets HTTP traffic
- Attack Phases:
  - various checks to understand if an analysis tool is running
  - … (including, opening a backdoor)
  - creates a hidden C&C channel in HTTP

Sunburst uses HTTP GET or POST requests. The server hides data within HTTP response bodies mimicking benign XML/.NET files. Hidden data is spread across many IDs and strings and extracted via the *\{[0-9a-f-]{36}\}"|"[0-9a-f]{32}"|"[0-9a-f]{16}* regexep.



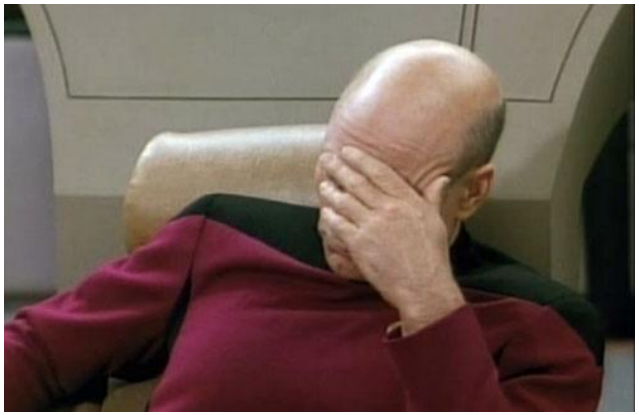
Source: https://www.fireeye.com/blog/threat-research/2020/12/sunburst-additional-technical-details.html

What We Can Do?

- Some facts:
  - carrier is **not** known *a priori* (e.g., images, network traffic, and text)
  - **heterogenous** set of protocols, files and data types
  - mixed **techniques** (LSB, metadata, comments, etc.)
  - **GDPR-like** constraints
  - **scalability**
  - …

What We Can Do?

- Some facts:
  - carrier is **not** known *a priori* (e.g., images, network traffic, and text)
  - **heterogenous** set of protocols, files and data types
  - mixed **techniques** (LSB, metadata, comments, etc.)
  - **GDPR-like** constraints
  - **scalability**
  - …
- Detection and mitigation are:
  - method-dependent
  - poorly generalizable
  - in a word: **hard!**




FRANK ZAPPA
and The
MOTHERS of INVENTION
ONE SIZE FITS ALL
Divan, Divan ... Weißt du wer ich bin
CRAB NEBULA GOES HERE
STARCLUSTER
M22(?)
TOO MUCH AIR

No!

(as today!)

Idea 1: Know Your Enemy

- Instead of chasing, a possible idea exploits prevention
- Possible actions:
  - clearly identify recurring patterns and address them

Seminal Work Here!

S. Wendzel, S. Zander, B. Fechner, C. Herdin, "Pattern-Based Survey and Categorization of Network Covert Channel Techniques", ACM Computing Survey, Vol. 47, No. 3, April 2015

Idea 1: Know Your Enemy

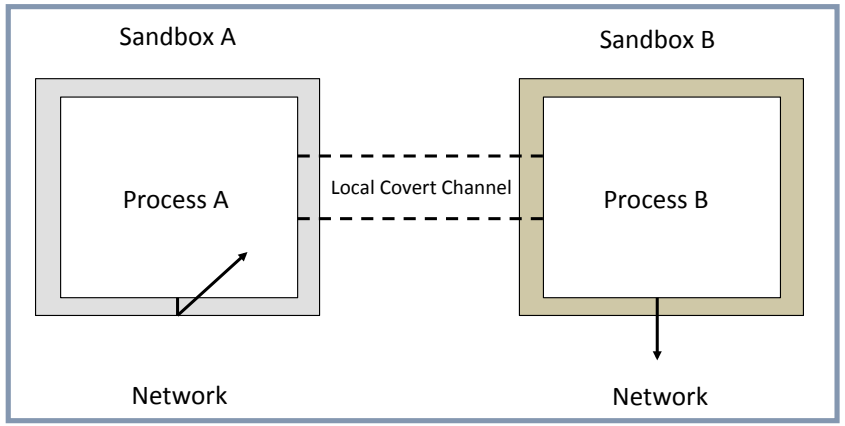
- Instead of chasing, a possible idea exploits prevention
- Possible actions:
  - clearly identify recurring patterns and address them
  - search for imperfect isolation or ambiguous implementations
  - develop a “formal” theory to make protocols, applications and contents information-hiding-resistant by-design

Idea 2: Abstraction

- Instead of being attack-specific:
  - increase the abstraction to describe multiple “stegomalware” with a reduced set of metrics
  - address threats per-behavior rather than per-carrier

Idea 2: Abstraction

- Instead of being attack-specific:
  - increase the abstraction to describe multiple “stegomalware” with a reduced set of metrics
  - address threats per-behavior rather than per-carrier
- Example:

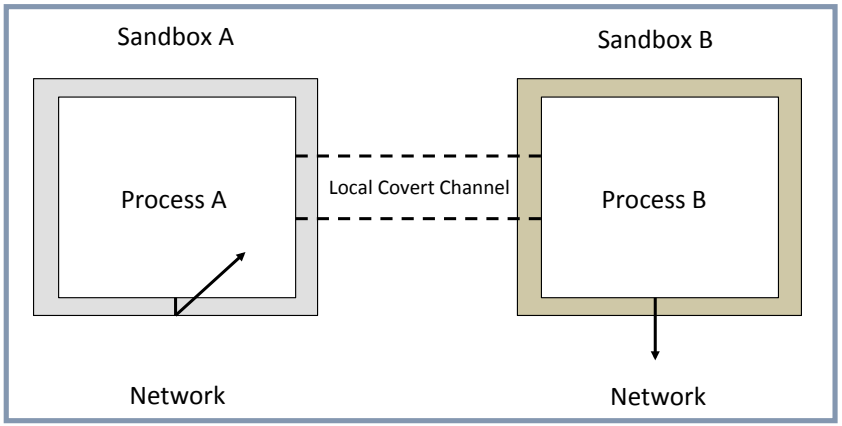


Information Hiding Here!

W. Mazurczyk, L. Caviglione “Steganography in Modern Smartphones and Mitigation Techniques”, *IEEE Communications Surveys & Tutorials*, Vol. 17, No. 1, pp. 334-357, Firstquarter 2015.

Idea 2: Abstraction

- Instead of being attack-specific:
  - increase the abstraction to describe multiple “stegomalware” with a reduced set of metrics
  - address threats per-behavior rather than per-carrier
- Example:

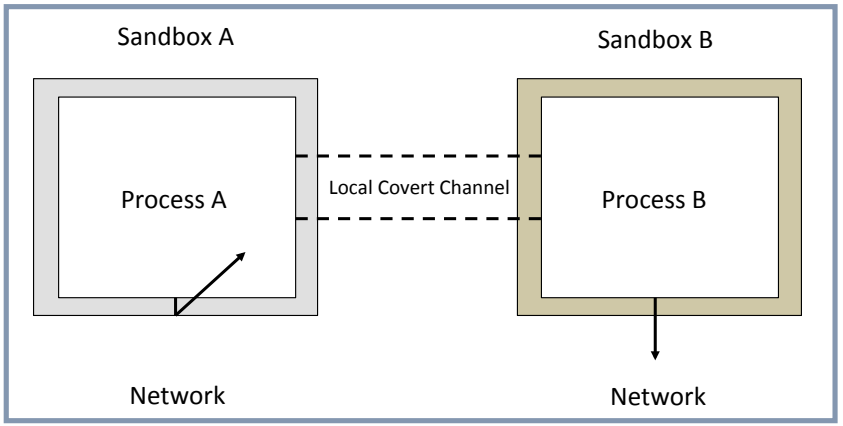


- The “Colluding Applications” threat:
  - both processes have visibility over a shared resource
  - a local covert channel is created by modulating its behavior
  - examples: vibration and volume settings (very popular in mobile devices), file permissions and sockets, free disk space, CPU load or RAM pressure, and abuse of legitimate IPC schema (e.g., Intentions in Android OS)
  - **not limited** to applications: also VMs, threads, etc.

W. Mazurczyk, L. Caviglione “Steganography in Modern Smartphones and Mitigation Techniques”, *IEEE Communications Surveys & Tutorials*, Vol. 17, No. 1, pp. 334-357, Firstquarter 2015.

Idea 2: Abstraction

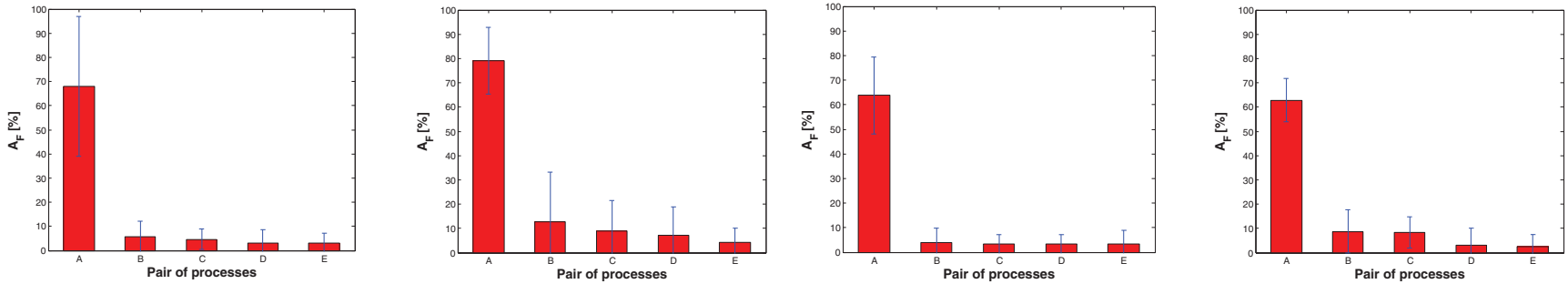
- Instead of being attack-specific:
  - increase the abstraction to describe multiple "s om ware" with a reduced set of metrics
  - address threats per-behavior rather t
- Example:



To create a local covert channel, the pair of colluding processes should be active "close" in time.

Idea 2: Abstraction

- Instead of being attack-specific:
  - increase the abstraction to describe multiple “stegomalware” with a reduced set of metrics
  - address threats per-behavior rather than per-carrier
- Example:

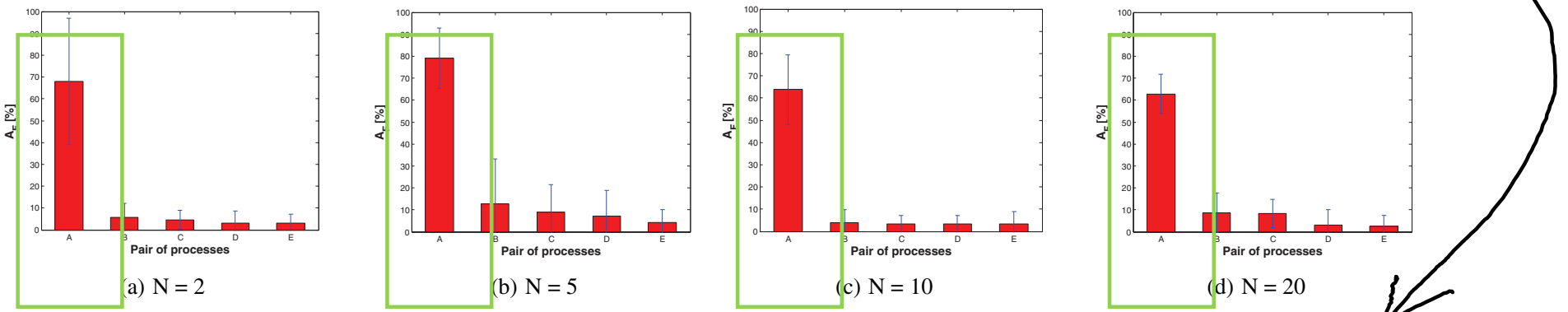


(a) N = 2 (b) N = 5 (c) N = 10 (d) N = 20

M. Urbanski, et al., “Detecting Local Covert Channels Using Process Activity Correlation on Android Smartphones”, Int. Journal of Computer Systems Science and Engineering, Vol. 32, No. 2, pp. 71-80, 2017.

Idea 2: Abstraction

- Instead of being attack-specific:
  - increase the abstraction to describe multiple “stegomalware” with a reduced set of metrics
  - address threats per-behavior rather than per-carrier
- Example:



(a) N = 2 (b) N = 5 (c) N = 10 (d) N = 20

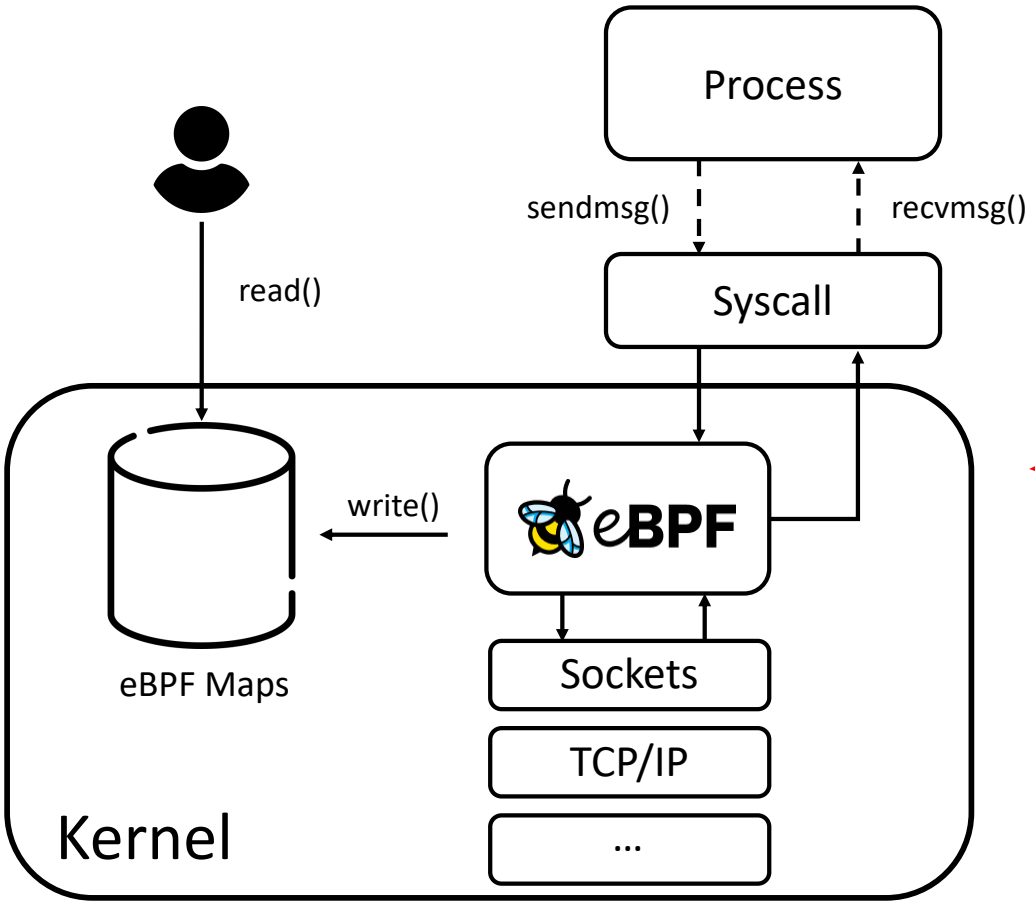
For the case of energy!

L. Caviglione, et. al, “Seeing the Unseen: Revealing Mobile Malware Hidden Communications via Energy Consumption and Artificial Intelligence”, *IEEE Trans. on Information Forensics and Security*, Vol. 11, No. 4, pp. 799-810, April 2016

Idea 3: Improve Visibility

- Being able to inspect multiple carriers should be considered as a good design rule when developing countermeasures against stegomalware
- Improved visibility over software, hardware and network could mitigate the challenge of not knowing what to check *a priori*
- Possible idea:
  - use the extended Berkeley Packet Filter (eBPF) to avoid bottlenecks or mitigate overheads
  - create datasets to feed AI-based frameworks

Idea 3: Improve Visibility



Not a "one-fits-all" solution, but at least a unique inspection technology!

L. Caviglione, W. Mazurczyk, M. Repetto, A. Schaffhauser, M. Zuppelli, "Kernel-level Tracing for Detecting Stegomalware and Covert Channels in Linux Environments", Computer Networks, Vol. 191, 2021

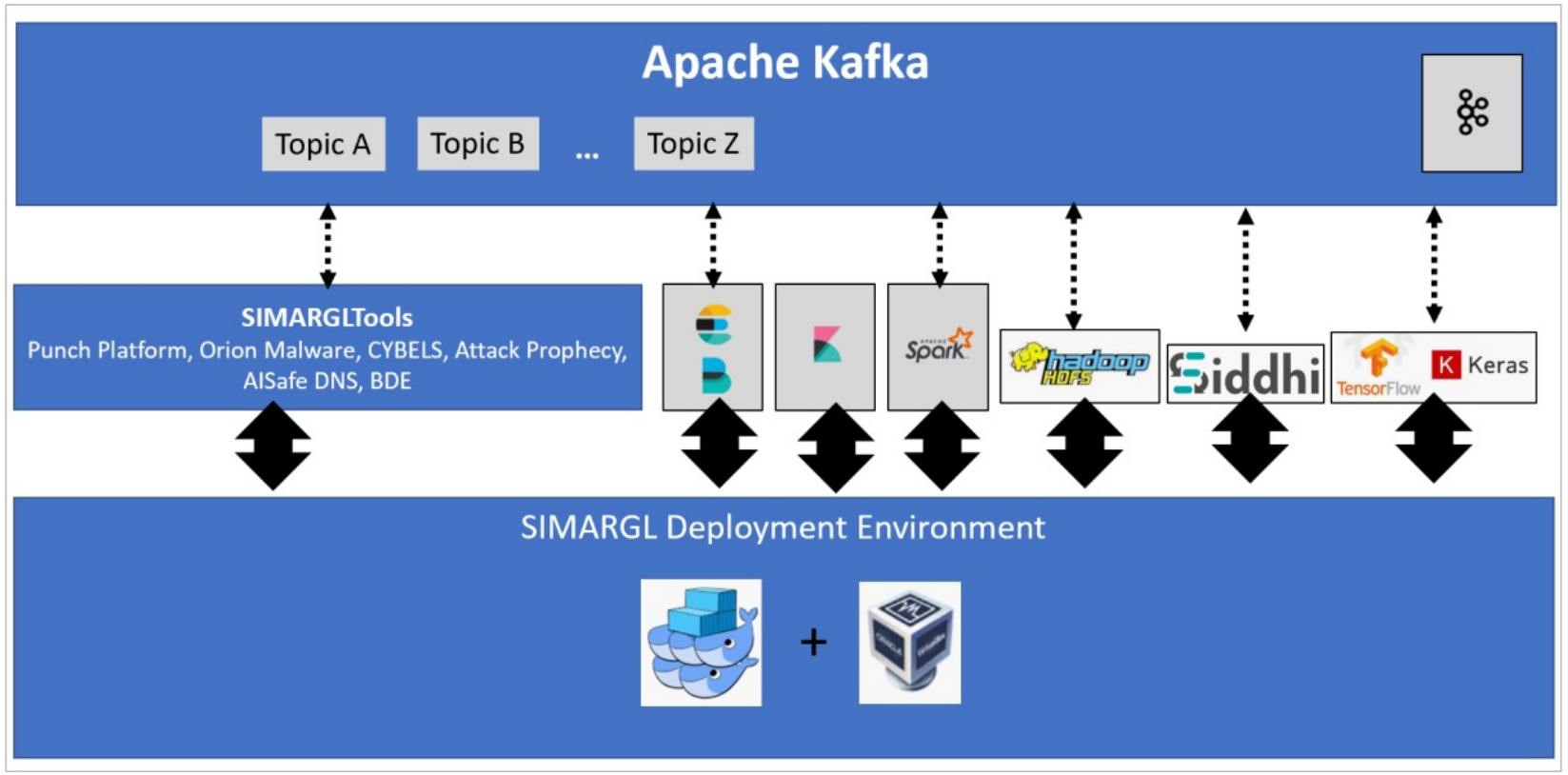
Idea 4: be Holistic!

Co-funded by the European Commission under the Horizon 2020 programme, the Secure Intelligent Methods for Advanced Recognition of malware and steGomaLware (SIMARGL) project joins 14 partners from 7 European countries.

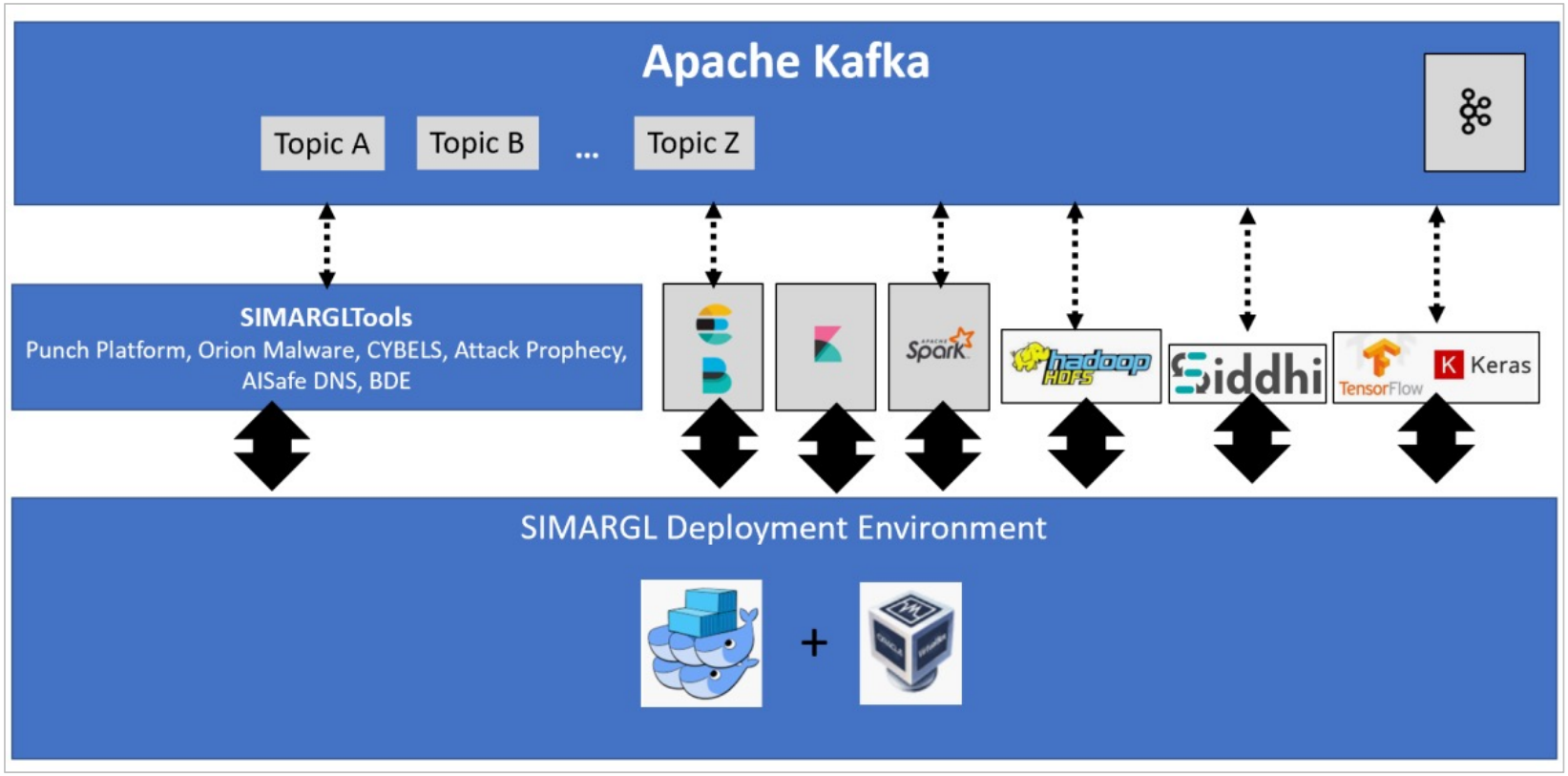
SIMARGL aims at tackling new challenges in the cybersecurity field, including Information Hiding techniques, network anomalies, stegomalware, ransomware and mobile malware.

SIMARGL exploits breakthrough methods and algorithms to analyze heterogenous network data and information.

Idea 4: be Holistic!



Idea 4: be Holistic!



Layered approach:

Deployment environment layer – Docker Swarm orchestration framework

Communication data bus - Apache Kafka to integrate functional components of the SIMARGL toolkit

Computational services layer – microservice-based approach to connect independently deployable components

Conclusions

- A **new trend** concerns the use of Information Hiding and steganographic techniques to empower malicious software
- Such threats are often called “stegomalware”: **they are here to stay!**
- Stegomalware is difficult to address:
  - it is emerging
  - it exploits mixed and heterogenous hw/sw features
- But:
  - we are working towards developing a “**theory**”
  - we can consider it in **early** design phases
  - we can try to mitigate it by being **general**

Conclusions

- A **new trend** concerns the use of Information Hiding and steganographic t oftware
- Such threats are **are here to stay!**
- Stegomalware is
  - it is emerging
  - it exploits mixe
- But:
  - we are working
  - we can conside
  - we can try to mitigate it by being **general**

M1ssing Register Access Controls Leak EL0 State
(CVE-2021-30747)
https://m1racles.com

Probably more for privacy leaking and moderately dangerous, but it is a covert-channel-based exploit

Conclusions

- A **new trend** concerns the use of Information Hiding and steganographic techniques to empower malicious software
- Such threats are often called “stegomalware”: **they are here to stay!**
- Stegomalware is difficult to address:
  - it is emerging
  - it exploits mixed and heterogenous hw/sw features
- But:
  - we are working towards developing a “**theory**”
  - we can consider it in **early** design phases
  - we can try to mitigate it by being **general**

Thank You!

@lucacaviglione

https://www.linkedin.com/in/lucacaviglione

luca.caviglione@ge.imati.cnr.it

simargl.eu