Perfect Data exfiltration Techniques

Eric Filiol
efiliol@hse.ru, efll@protonmail.com
https://ericfiliol.site

Thales Digital Factory, France & Higher School of Economics, Russia

November 11^{th}, 2022

Summary of the talk

Summary of the talk

Introduction: Key Security Issue

- Most attacks now include data exfiltration.
  - From a few credentials...
  - ... to complete databases.
- For the attacker, the difficulty depends on the target environment.
  - Unconnected environments require air-gap attacks $\Rightarrow$ limited amount of data, low data rate, low security awareness.
  - Connected/networked environments (any protocol) require bypassing traffic surveillance $\Rightarrow$ Possibly high amount of data, high data rate and medium / high security awareness.
- From the defender perspective, the analysis and safeguards are
  - Automated or semi-automated analysis (DLP)
  - Manual/ad hoc analysis (*post mortem*)

Introduction: Working Environment & Scenarios

- The operational context is twofold:
  - Either relatively weak targeted attacks for environments with only automated detection in place.
  - Or strongly targeted attacks (single or very few targets) where a manual analysis of malware/traffic data can be performed.
- We focus on connected environments (network with any protocol) but these techniques apply to air-gap environments as well.
- If we primarily focus on bypassing automated analysis, we cannot neglect the ad hoc/manual analysis.
  - The initial step consists in deploying a malware (most of the times) or a dedicated device
  - We have consequently to manage this analysis step as well. This analysis must not reveal the actual nature of the attack (and ultimately that an attack is under way) at much as possible.

Attacker's Issues to Solve - Defender's Challenges

- The attacker has to face several critical issues than can trigger alerts and block data exfiltration:
  - I1 Data may be analysed so semantic detection (keywords, statistical profile, Data Leak Prevention [DLP]) can be enforced.
  - I2 Encrypting data before exfiltration is likely to be detected (entropy profile test, however rarely in place)
  - I3 Encryption implies a secret key (can be recovered during malware analysis or during the process performing the data exfiltration).
  - I4 All outbound traffics may be encrypted automatically (IPSec VPN) and this encryption it out-of-control for this attacker (this is the case in sensitive networks for instance)
  - I5 Known exfiltration techniques (steganography, covert channels) drastically limit the amount of data to exfiltrate without detection (up to a few kilobytes)
  - I6 Users behaviours may be analysed to detect suspect actions betraying data exfiltration
- Any analysis by the defender must fail or at least be delayed enough.

Case of Issue I4

- This case has been already fully addressed and solved with malware enforcing IPSec hijacking techniques.
- See my talk at CanSecWest 2011.
  - Updates presented at Hacktivity 2021 and PHDays 2022.

Aim of the Present Work

- Showing how an attacker could exfiltrate sensitive data while bypassing all these issues by using malicious cryptography & mathematics.
- Evaluating forthcoming approach by malware designers/attackers to make malware/ransomware techniques evolve in a more critical way.
- Identify and test possible mitigation techniques to prevent the presented techniques in a proactive way (precautionary principle)
- We present "unitary attack bricks" for clarity but they can be combined in whole or in part.
- All codes and PoC developed are TLP:RED

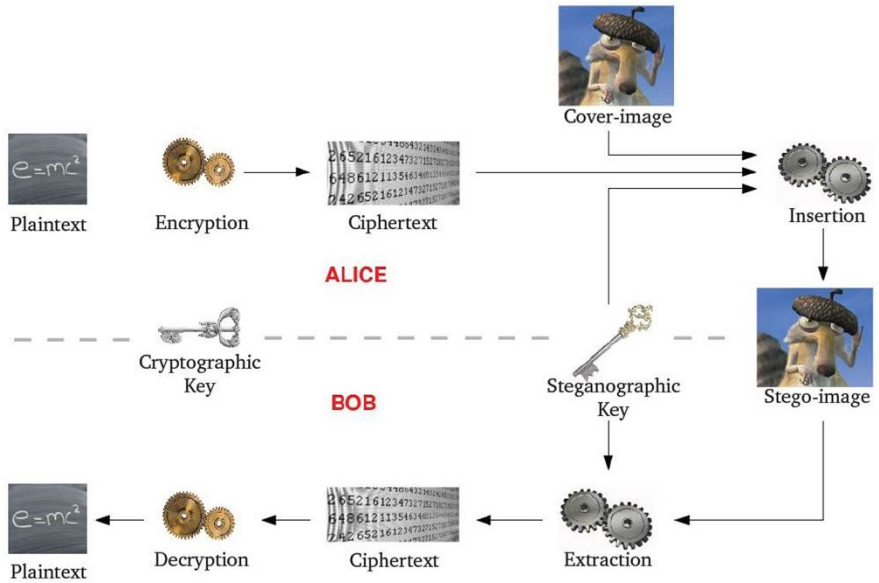
Summary of the talk

ComSec versus TranSec

Two different views in Information Security (NATO terminology)

- **ComSec** (*Communications Security*) ensures the security (confidentiality and integrity) of telecommunications. ComSec refers to the security of information transmitted regardless of the communication channel.
  - Cryptography, Tempest/EMSEC, physical security (network, rooms...)
- **TranSec** (*Transmission Security*) ensures the protection of the channel itself and especially the existence of secret data being exchanged (prevent interception, disruption of reception, communications deception, and/or derivation of intelligence by analysis of transmission characteristics such as signal parameters or message externals...).
  - TranSec is a field of COMSEC which deals with the security of communication transmissions (the channel), rather than that of the information being communicated.
  - Steganography, Tempest/EMSEC, traffic flow security, routing protocols...
- Our techniques intends to combine both views.

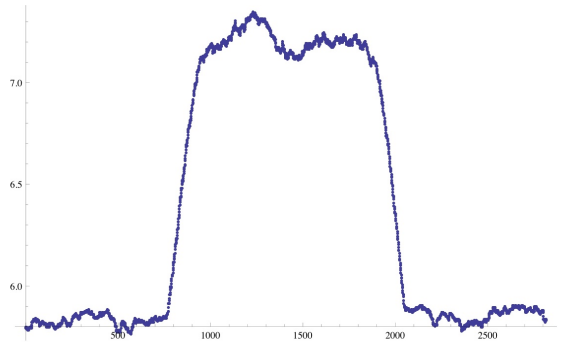
Cryptography versus Steganography

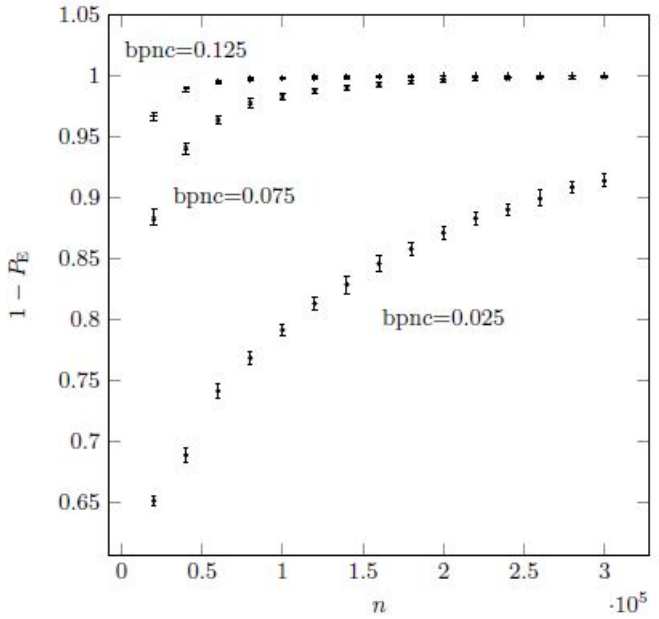


- Transform data with an encryption algorithm and a secret key into a random data.
- Accessing plaintext data is supposed to be practically intractable without the key.

- Secret key-dependent encryption and random insertion to hide a message into an innocent-looking cover without altering (too much) its statistical profile.

Cryptography versus Steganography: detection





- Detection of cryptography is straightforward using the entropy profile [2].
  - Plaintext data: entropy $H(X) \approx 4$
  - Packed/compressed data: $H(X) \approx 6$
  - Encryption data: $H(X) = 8$
- Beyond an insertion rate of 0.03, detection is efficient with modern techniques [3]
- Size of secure payload is limited (to $\sqrt{n}$; n is the number of usable coefficients for embedding).

Malicious Cryptology & Mathematics

- Malicious Cryptology and Malicious Mathematics (MCMM) is an emerging domain initiated in (Filiol, 2012)
  - Generalization of Young & Yung's (2004) crypto virology Young (limited case of extortion malware which prefigures ransomware).
- MCMM can be defined as the interconnection of attack techniques with cryptology and mathematics for their mutual benefit. Covers several fields and topics (non exhaustive list):
  - Development "super malware" able to evade any kind of detection by implementing:
    - Optimized propagation and attack techniques (e.g. by using biased or specific random number generator).
    - Sophisticated self-protection techniques. The malware code protects itself and its own functional activity by using strong cryptography-based tools.
    - Partial or total invisibility features. The programmer intends to make his code to become invisible by using statistical simulability

Malicious Cryptology & Mathematics (continued)

- Use of complexity theory or computability theory to design undetectable malware.
- Use of malware to perform cryptanalysis operations (steal secret keys or passwords), manipulate encryption algorithms to weaken them on the fly in the target computer memory. The resulting encryption process will be easier to break/bypass.
- Recon in target environments (e.g. processor-dependent malware)
- Design and implementation of encryption systems with hidden mathematical trapdoors. The knowledge of the trap (by the system designer only) enables to break the system very efficiently. Despite the fact that the system is open and public, the trapdoor must remain undetectable (see a real instance in [Filiol & Bannier, BlackHat Europe, 2017]).

See bibliography slide for extended references [1].

Summary of the talk

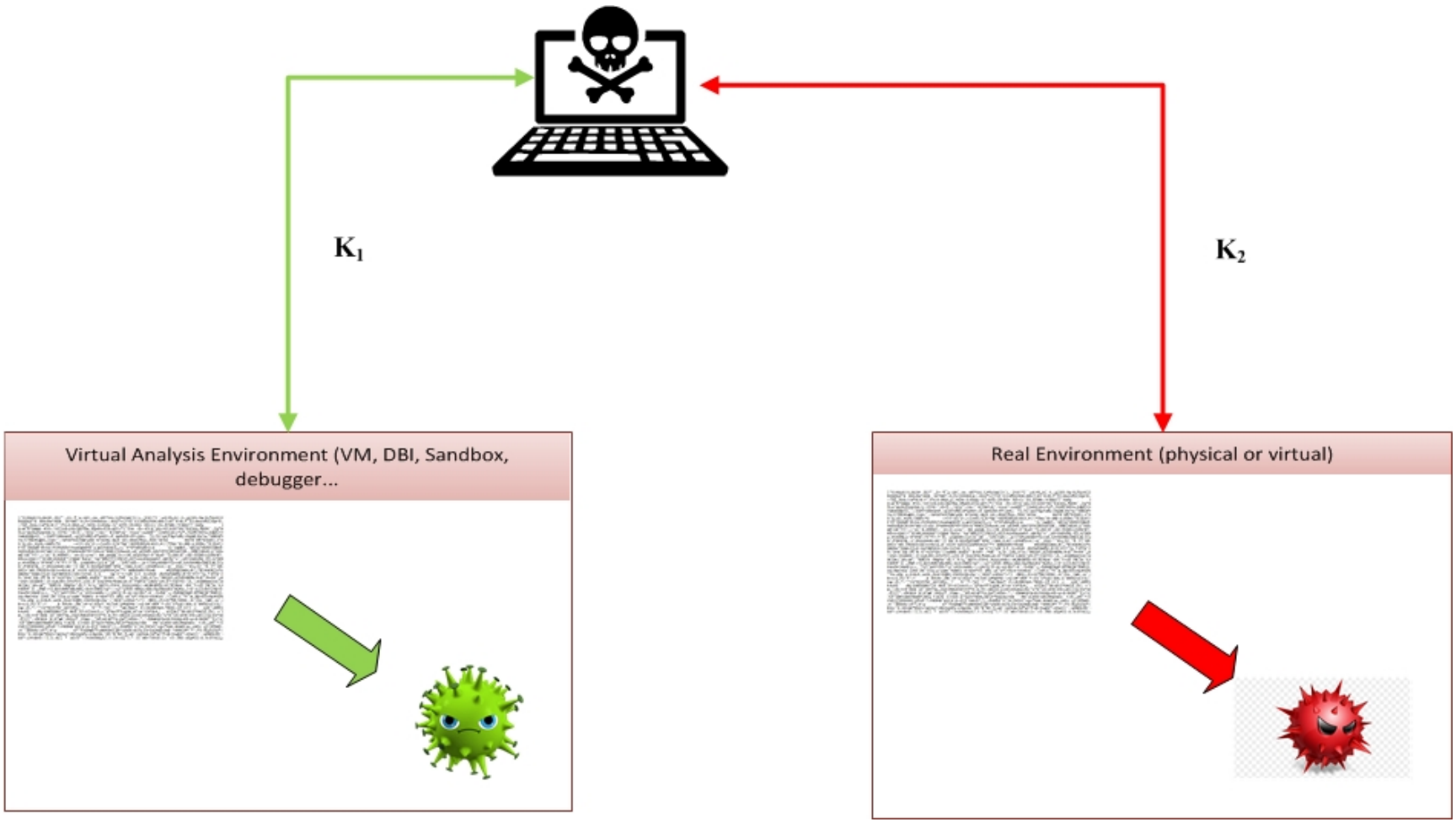
Non-Trivial Deniable Cryptography: Principles

- Building effective encryption algorithm (deterministic algorithms) to realize practical deniable cryptography was until very recently still an open problem.
  - Only known case is trivial (one-time pads).
  - Since the "key" is as long as the two (or more) plaintexts, this solution is not valid (one-time pad must be in the code).
- Let C be a ciphertext of length N, a unique algorithm E and any two different arbitrary plaintexts P_1 and P_2. We built a C framework to build encryption algorithms (within seconds from given plaintexts) enabling effective deniable cryptography with short keys (128 - 256 bits).
  - E is a deterministic encryption algorithm (stream cipher or block cipher). It is supposed to be public and therefore resistant to known cryptanalysis techniques. Keys are k-bit long.
  - k is far smaller than N (so one-time pad is not considered).
  - We have $C = E(K_1, P_1) = E(K_2, P_2)$
  - The scheme can be extended to a finite number of plaintexts P_i

Non-Trivial Deniable Cryptography: Applications

- The cryptographic security analysis of these algorithms have confirmed the resistance against the following attacks:
  - Guess P_1 and P_2 from the ciphertext C (in other words, retrieving keys K_1 and/or K_2).
  - Find P_1 knowing P_2 and conversely.
- Awesome number of applications (developed at OPS4SEC):
  - Code protection (malware or legitimate program) against static and dynamic analysis (see next slide).
  - Anti-forensics techniques.
  - Multiple communication channels in a single one
  - ...
- Demo

Deniable Cryptography-based Malware



Deniable Cryptography-based Malware

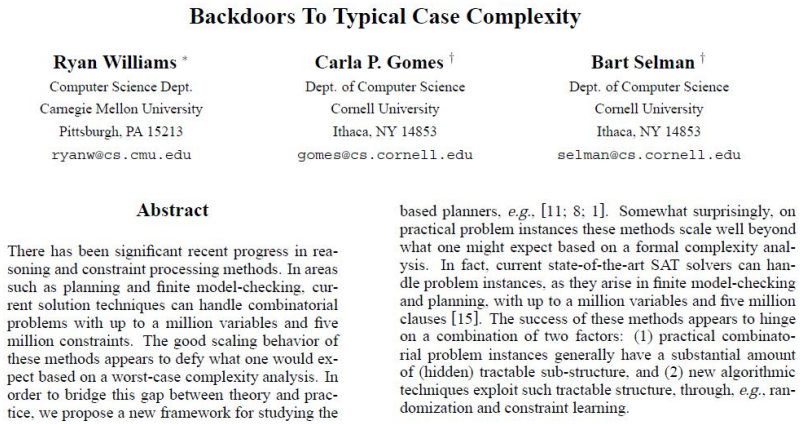
- The most critical part of the malware is encrypted (C) and needs an external key from the C&C .
- Secure and complex communication protocol between malware and C&C (fingerprint, time index, time obfuscation, random connexions, environment conditions...). The malware is clueless wrt this protocol.
- The malware is able to detect that it is under analysis (see [4] for instance)
- The malware analyses its environment and requests an external key according to the connexion protocol.
- If no analysis is under way, the malware receives key K_2 and then decrypts itself as $P_2 = E(K_2, C)$. This is actually the real malware.
- If analysis is detected and/or connexion conditions are not fulfilled, the malware receives key K_1 and then decrypts itself as $P_1 = E(K_1, C)$. This is either a goodware or an alternative malware (to fool the malware analyst).

Summary of the talk

Metadata & Document Formats/Internals Permissiveness

- Most documents formats includes metadata and rather rich and complex formats or internals:
  - Most DLPs do not check metadata or are very weak (easy to bypass) at analysing them.
  - Underlying format languages/internal either are not properly specified or the compliance to the internals is partially or not checked.
- Most document formats have a large permissiveness with respect to data embedding.
- Depending on the document format, it is possible to silently exfiltrate from a few tens of bytes to several megabytes
  - Documents are extremely interesting natural carriers for data exfiltration.
  - It is possible to split data into several documents (of possible different formats)
- To some extent but in limited way (bytes to kilobytes of data), the same approach applies to network protocol metadata (see Drzymaa & al. from Warsaw University of Technology for instance)

PDF File as Carrier: Illustration



Backdoors To Typical Case Complexity

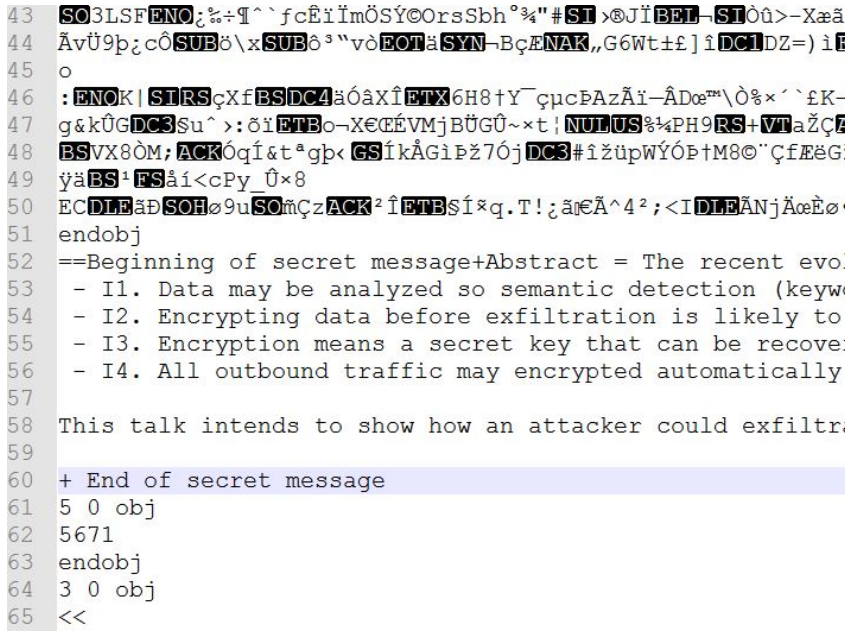
Ryan Williams *
Computer Science Dept.
Carnegie Mellon University
Pittsburgh, PA 15213
ryanw@cs.cmu.edu

Carla P. Gomes †
Dept. of Computer Science
Cornell University
Ithaca, NY 14853
gomes@cs.cornell.edu

Bart Selman †
Dept. of Computer Science
Cornell University
Ithaca, NY 14853
selman@cs.cornell.edu

Abstract

There has been significant recent progress in reasoning and constraint processing methods. In areas such as planning and finite model-checking, current solution techniques can handle combinatorial problems with up to a million variables and five million constraints. The good scaling behavior of these methods appears to defy what one would expect based on a worst-case complexity analysis. In order to bridge this gap between theory and practice, we propose a new framework for studying the based planners, *e.g.*, [11; 8; 1]. Somewhat surprisingly, on practical problem instances these methods scale well beyond what one might expect based on a formal complexity analysis. In fact, current state-of-the-art SAT solvers can handle problem instances, as they arise in finite model-checking and planning, with up to a million variables and five million clauses [15]. The success of these methods appears to hinge on a combination of two factors: (1) practical combinatorial problem instances generally have a substantial amount of (hidden) tractable sub-structure, and (2) new algorithmic techniques exploit such tractable structure, through, *e.g.*, randomization and constraint learning.



```
43 SO3LSFENQ¿%÷¶^`ƒcÊïÏmÖSÝ©OrsSbh°¾"#SI›®JÏBEL¬SIÒû>-Xæã
44 ÃvÜ9þ¿cÔSUBö\xSUBô³"vòEOTäSYN¬BçÆNAK„G6Wt±£]îDC1DZ=)ì
45 o
46 :ENQK|SIRSçXfBSDC4äÓâXÎETX6H8†Y¯çµcÞAzÃï—ÂDœ™\Ò%×´`£K-
47 g&kÛGDC3$u^›:õïETBo¬X€ŒÉVMjBÜGÛ~×t¦NULUS¾¼PH9RS+VTaŽÇ
48 BSVX8ÒM;ACKÓqÍ&tªgþ‹GSÍkÅGìÞž7ÓjDC3#îžüpWÝÓÞ†M8©¨ÇfÆëG
49 ÿäBS¹FSåí<cPy_Û×8
50 ECDLEãÐSOHø9uSOmÇzACK²ÎETB§Í¤q.T!¿ãœÃ^4²;<IDLEÃNjÄœÈø
51 endobj
52 ==Beginning of secret message+Abstract = The recent evo
53  - I1. Data may be analyzed so semantic detection (keyw
54  - I2. Encrypting data before exfiltration is likely to
55  - I3. Encryption means a secret key that can be recove
56  - I4. All outbound traffic may encrypted automatically
57
58 This talk intends to show how an attacker could exfiltr
59
60 + End of secret message
61 5 0 obj
62 5671
63 endobj
64 3 0 obj
65 <<
```

- Existing PDF file after leak injection.
  - Lame example for illustration purposes.
  - Fully working PDF (no alert)
  - Leaked data not encoded (see further) for visibility purposes.
- In case of document integrity in place (rarely), documents may be created from scratch.

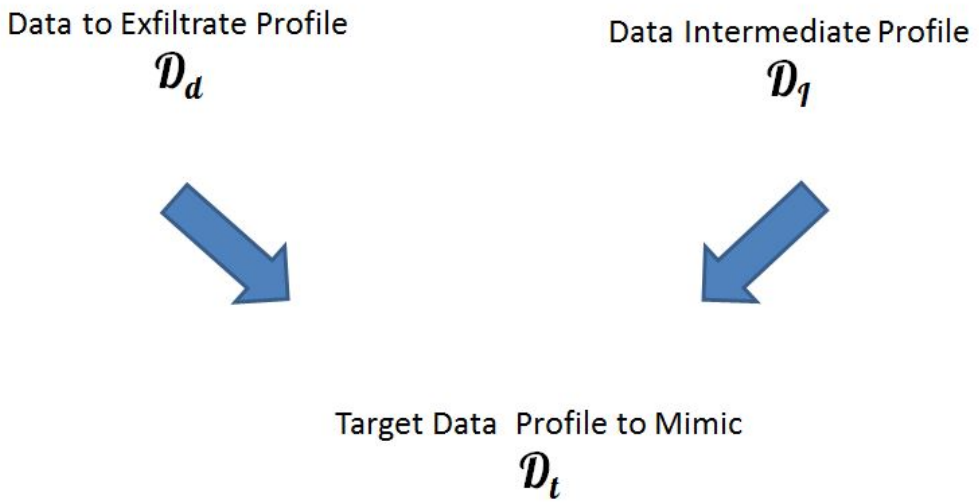
Entropy & Statistical Profiles

- First step of detection is generally automated. A statistical end entropy analysis may be enforced (even if most DLPs actually do not)
- So any data exfiltrated must exhibit innocent-looking profiles or at least compliant with the target environment.
- The general approach must
  - Consider key-dependent transformations of the data to be exfiltrated (otherwise it is just encoding which can be detected and broken rather easily)
  - Prevent any manual analysis to reveal/recover the secret keys used.
  - Be able to mimic/simulate any target statistical/entropy profile
- In the rest of the talk, without loss of generality we focus on character entropy or statistical profile but any other profile can be considered (n-gram profiles)

Secret Keys Management

- Whenever using key-dependent transformations (e.g. encryption), keys are the critical parts to protect from analysis.
- The solution is to use random keys generated from the environment (e.g. `/dev/random/` or equivalent)
  - For each file m, the malware generates a random key K_m of size $|K|$.
  - Size $|K|$ must enable exhaustive search for the attacker. For instance, we tested $|K| = 40$.
- It is also possible to use encryption algorithms with backdoor to use longer key size (see my webpage for references).

General Principle of Data Transformation



- Data to exfiltrate exhibit entropy profile $H(\mathcal{D}_d)$ (e.g. *.pdf files)
- Data to mimic exhibit entropy profile $H(\mathcal{D}_t)$ (e.g. *.text files).
- We then compute *a priori* an intermediate (transition) entropy profile $H(\mathcal{D}_I)$ such that $H(\mathcal{D}_t)$ is the joint entropy profile $H(\mathcal{D}_d, \mathcal{D}_I)$ (here $H(\mathcal{D})$ describes the entropy profile of source data with distribution $\mathcal{D}$).

Examples

Without loss of generalities, let us consider the following text to evade

```
THIS IS A SECRET MESSAGE IN TEXT
```

The result of transformation toward different text distributions gives:

- Second-order Markov character distribution
  ```
  HEMB THAT WILSDOM ABOARICE AMOLL ELETS XEDEAT GIRLS ESSE
  OFTE AGENT
  ```
- First-order Markov word distribution gives
  ```
  ACTING THIS AND BEARING SECRET DEFENSE IS A NATURAL AGE IN
  METHOD OF TEXT FOR THE LETTERS IN MESS BE THOSE
  ```

Here we achieve nearly 30 % of embedding rate (similar to concept of steganographic rate) without detection.

Evolution : Perfect Data Exfiltration

- Examples of the previous slides are perfect against any automated analysis BUT might be detected by human analysis (infeasible in practice however but possible during forensics steps)
- With more complex models and distributions, we succeeded in defeating even manual analysis (reading)
  - However, depending on the target distribution (target data profile), the embedding rate may drops to 8-10 % (which still better than steganography).
- Demo: critical database exfiltration (1494 bytes)
  - Target distribution: FW log language
  - Size of the target file 8852 bytes (embedding rate of 17 %, no optimization)
  - FW log files are prone to be exchanged outside

Comments - Security Analysis

- There exist a quite infinite number of possibilities:
  - You can change the language (English to French for instance)
  - You can change the format (TEXT to WORD or PDF). However you may have to care about some tags to avoid errors.
  - You can make the distribution order vary...
- Note that the higher the distribution order the lower the data rate.
- Preventing this is impossible. It would require to recode/transcode any data before transmission preventively and by default (computing resources issues). But advanced techniques enables to bypass even reconding/transcoding techniques.
- No DLP tool is able to detect this nowadays and will likely never.

DLP Behaviour Analysis

- DLP techniques also rely on behavioural analysis (frequency, volume, nature or format of documents, etc.). They must be taken into account.
- The approach is the same and requires the availability of the main behavioural distributions used in general (or specifically by some DLPs)
    - These distributions are often public or known.
    - A malware can analyse specific distributions in a preliminary identification/intelligence phase.

Summary of the talk

Summary

- Exfiltrating data without detection (IDS, DLP, flow analysis...) is still easy.
- Detection faces complexity and computing issues especially if the malware embeds adaptative behaviors and techniques.
- The huge potential of malicious mathematics/cryptography is likely to see new malware technologies arise very soon (if not already for APTs).
- Prior security assessment, secure architecture, data assessment, tight rights management are necessary to lower the issues but in no way sufficient.

Conclusion

Thank you for your attention

Questions & Answers through emails

Bibliography

- E. Filiol (2012). *Malicious Cryptography and Mathematics*. `https://www.intechopen.com/chapters/29700` (open access)
- E. Carrera (2007). *Scanning data for entropy anomalies*. `http://blog.dkbza.org/2007/05/scanning-data-for-entropy-anomalies.html`
- J. Fridrich (2010). *Steganography in Digital Media*. Cambridge University Press
- F. Plumerault, B. David (2021). *DBI, debuggers, VM: gotta catch them all*, Journal of Computer Virology and Hacking Techniques, vol. 17, issue 2.
- E. Filiol, F. Jennequin & G. Delaunay. "Malware-based Information Leakage over IPSEC Tunnels", Journal in Information Warfare, volume 7, issue 3, pp. 1122, December 2008.