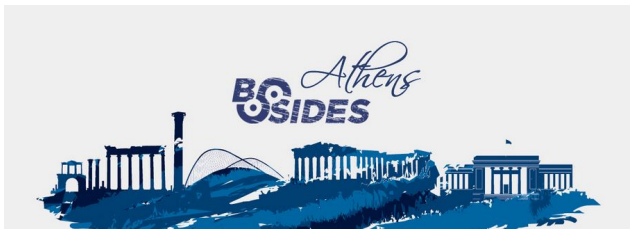
Recent Advances in Malicious Cryptography & Mathematics and Their Potential Impact on Malware Activity

Eric Filiol
efll@protonmail.com
https://ericfiliol.site

Thales Digital Factory, Paris, France

June 29^{th}, 2024



Summary of the talk

Agenda

Introduction

GENERAL DISCLAIMER

Any views or opinions presented in this talk are personal and are the results of my own research work. They belong solely to the speaker and, in any case, they do not represent those of people, institutions, companies or organizations that the speaker may or may not be associated with in professional or personal capacity (including past, present and future employers).

- This research work was conducted from mid-2019 to end of 2020.
- All codes and PoC developed are TLP:RED (responsible disclosure).

Introduction: Working Environment & Scenarios

CYBERSECURITY DIVE Deep Dive Library Events Press Releases

Strategy Breaches Vulnerability Cyberattacks Threats Leadership & Careers Poli

DIVE BRIEF

National Cyber Director echoes past warnings: Nation-state cyber threats are mounting

State-linked actors with ties to China and Russia are growing more sophisticated in their efforts to disrupt critical infrastructure, Harry Coker Jr. said during a CyberUK conference keynote.

Published May 15, 2024

David Jones
Reporter

- We focus in our talk to APT or Nation state attacks.
  - Optimization of the risk/effort ratio.
  - The worst risk is to be detected and identified.
- From the defender perspective, the analysis and safeguards are
  - Automated or semi-automated analysis.
  - Manual/ad hoc analysis.

Attacker's Issues to Solve

- The attacker has to face several critical issues than can trigger alerts and block malware action:
  - 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 Malware binary code may be analysed and reverse-engineered.
  - 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 be defeated.

Aim of the Present Work

- Explore the potential of malicious cryptography & mathematics.
- Evaluate and assess forthcoming approaches by malware designers (APT threats, state malware).
  - Especially, assess risk far beyond already existing identified threats (such *Lambload, VerbleCon*...)
- Identify and test possible mitigation techniques.
- We present "unitary attack bricks" for clarity but they can be combined in whole or in part.

Agenda

Malicious Cryptology & Mathematics

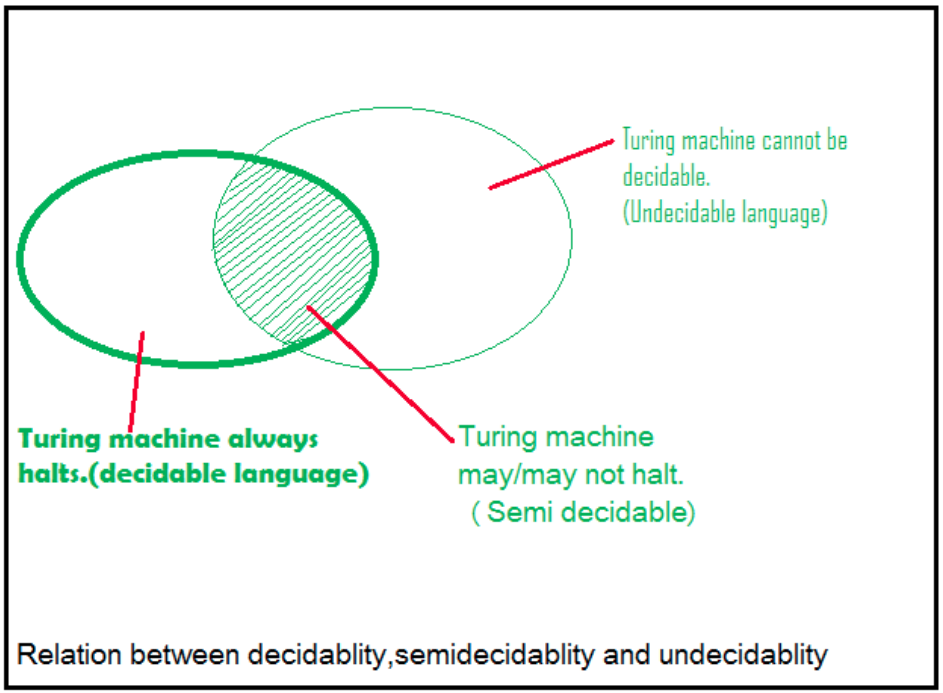
- Malicious Cryptology and Malicious Mathematics (MCMM) is an emerging domain initiated in (Filiol, 2012)
  - Generalization of Young & Yung's (2004) crypto virology Young (limited case of extortion malware/ransomware).
- MCMM defines the interconnection of attack techniques with cryptology and mathematics 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 (malware code and operations protection thanks to strong cryptography-based tools).
    - Partial or total invisibility features. The programmer intends to make his code and actions to become invisible by using statistical simulability

Malicious Cryptology & Mathematics (continued)

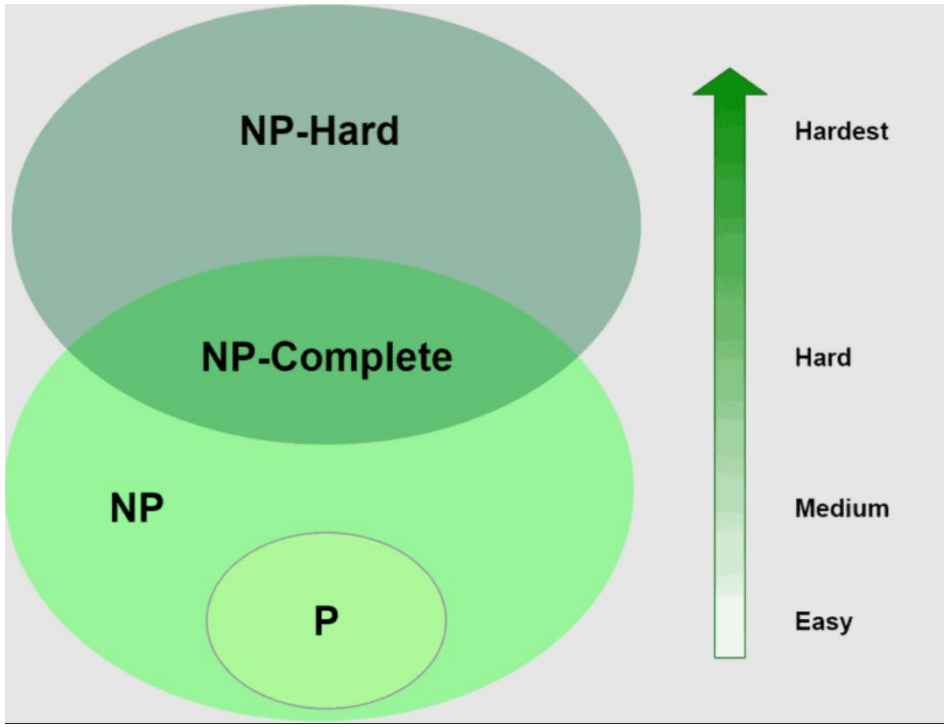
- **Use of complexity theory or computability theory to design undetectable malware.**
- Use of malware to perform cryptanalysis operations (steal secret keys or passwords), manipulate encryption algorithms to weaken them on the fly in the target computer memory. The resulting encryption process will be easier to break/bypass.
- Recon in target environments (e.g. processor-dependent malware)
- Design and implementation of encryption systems with hidden, undetectable mathematical trapdoors (see a real instance in [Filiol & Bannier, BlackHat Europe, 2017]).

See bibliography slide for extended references [1].

Decidability and Complexity Classes



https://www.geeksforgeeks.org/undecidability-and-reducibility-in-toc/



https://www.baeldung.com/cs/p-np-np-complete-np-hard

Known Complex Malware Techniques

Techniques	Class	Author
Contradictory malware	Undecidable	(Cohen, 1986)
Short-key Deniable Cryptography	Undecidable	(Filiol, 2019)
Crypto Extorsion (Ransomware)	NP-Hard	(Young & Yung, 1996)
Polymorphic malware (bounded length)	NP-complete	(Spinellis, 2003)
Polymorphic malware	NP-complete	(Zuo & Zhou, 2005)
Formal grammar-based Metamorphic Malware	At least NP-Hard	(Filiol, 2007)
K-ary Malware	NP-hard	(Filiol, 2007)
Anti-DLP evasion techniques	At least NP Hard	(Filiol, 2019)
Information Leakage over IPSEC Tunnels	NP Hard	(Filiol et al., 2008)

Agenda

Case-study

- We consider a two-step attack.
  1. A malware precursor is deployed. It does not contain malicious functions so far. It gathers technical information on the target (intelligence step) and sends it to a Command & Control (C&C).
  2. The C&C constantly analyses data provided by the precursor and malware to evaluate whether they are under analysis or not.
  3. The C&C updates, modifies and provides data to the malware as often as required for the attack evolution.

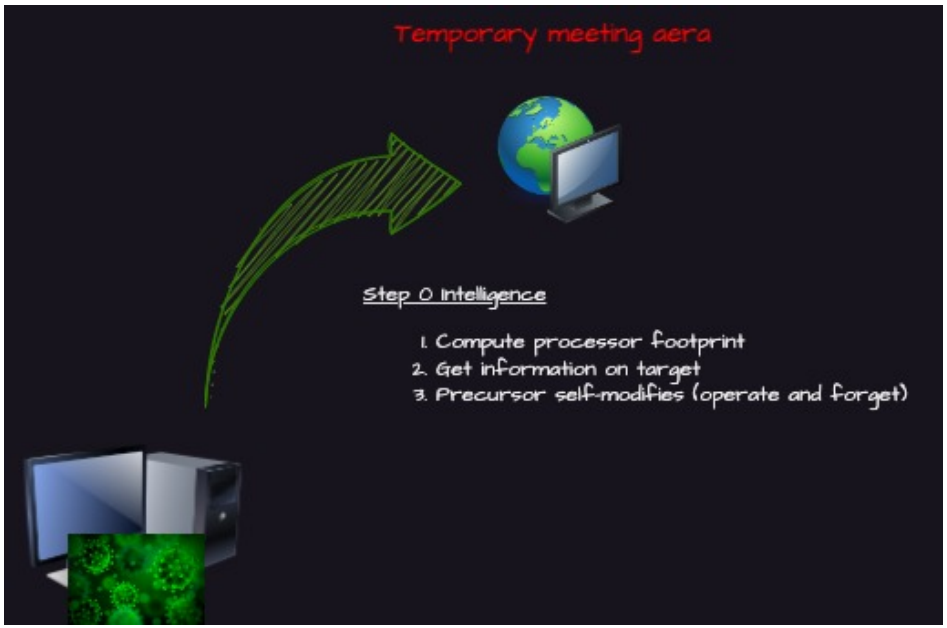
Case-study

- We consider a two-step attack.
  1. A malware precursor is deployed. It does not contain malicious functions so far. It gathers technical information on the target (intelligence step) and sends it to a Command & Control (C&C).
  2. The C&C constantly analyses data provided by the precursor and malware to evaluate whether they are under analysis or not.
  3. The C&C updates, modifies and provides data to the malware as often as required for the attack evolution.
- Malware and C&C communicate through a buffer neutral zone (mutating IPs, protection of the C&C). Use of a sophisticated protocol to manage potentiel analysis on target size.

Case-study

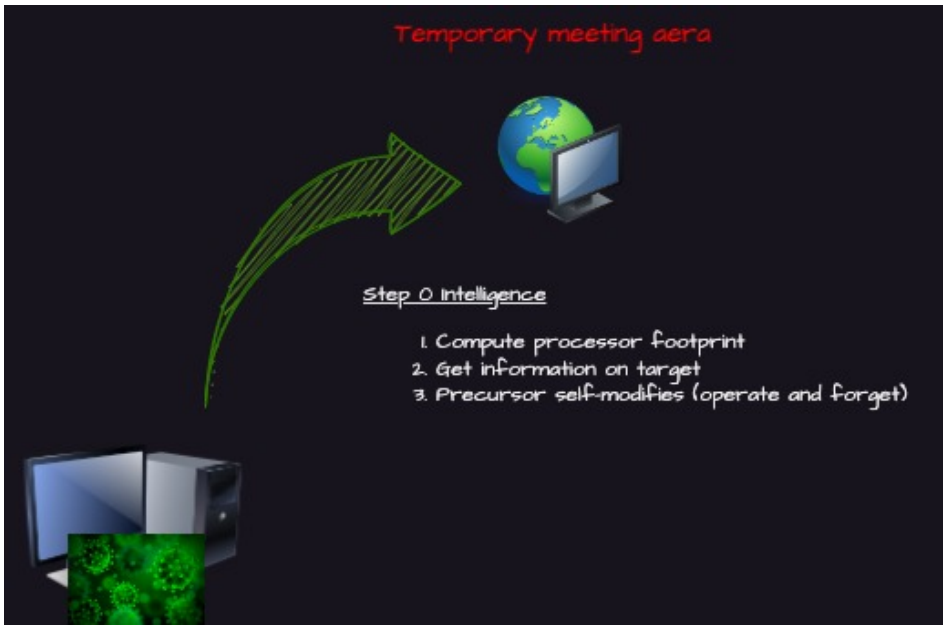
- We consider a two-step attack.
  1. A malware precursor is deployed. It does not contain malicious functions so far. It gathers technical information on the target (intelligence step) and sends it to a Command & Control (C&C).
  2. The C&C constantly analyses data provided by the precursor and malware to evaluate whether they are under analysis or not.
  3. The C&C updates, modifies and provides data to the malware as often as required for the attack evolution.
- Malware and C&C communicate through a buffer neutral zone (mutating IPs, protection of the C&C). Use of a sophisticated protocol to manage potentiel analysis on target size.
- The precursor evades detection (it does not contain actual malicious functions). Subsequent modifications aim at providing persistence without detection.

Secure Clueless Protocol (precursor/malware side)



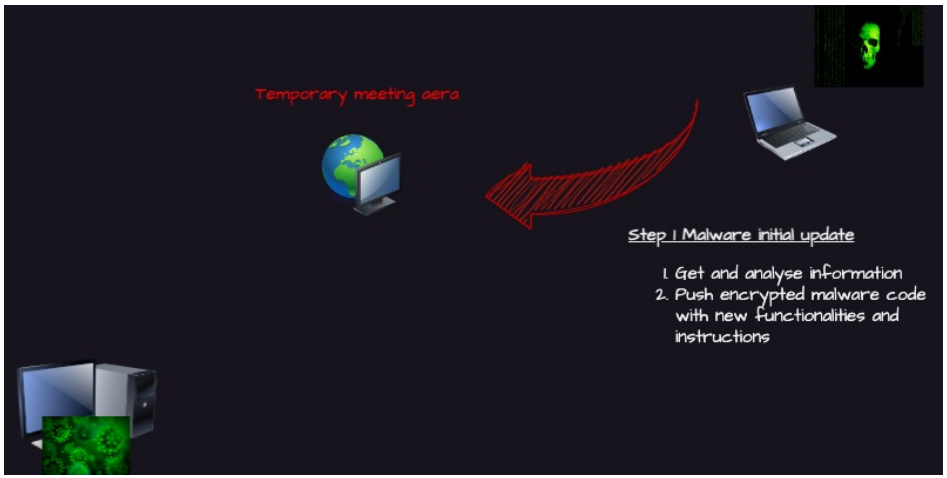
- Module M: computes complex mathematical functions (processor fingerprint [1]). Results v_i and computation time t_i are sent to the C&C.
  - For instance $\sin(10^{37}\pi_j)$ where π_j are different precision values of π.
  - A lot of possible functions and precision levels enable a very accurate processor fingerprint.

Secure Clueless Protocol (precursor/malware side)



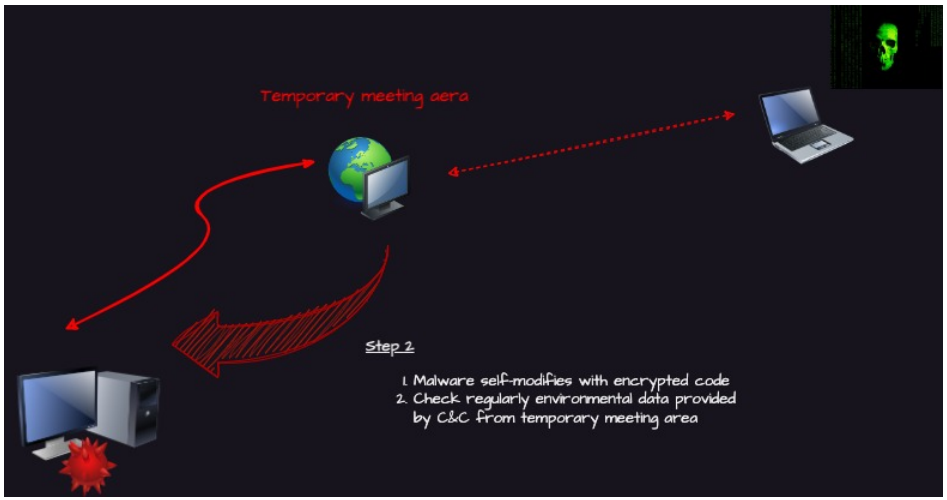
- Module *I*: gathers technical data (and the collection time) on the local target:
  - CPUID, HD Serial numbers, MAC address, SID, GUID, RAM fingerprint, Thermal status...
  - The collection time for each parameter is also sent.
- Module *V*: compute a virtualization detection index according to [4].
- Upon completion, the precursor self-modifies (operate and forget). **From now on any analysis would reveal nothing**.

Secure Clueless Protocol (C&C side)



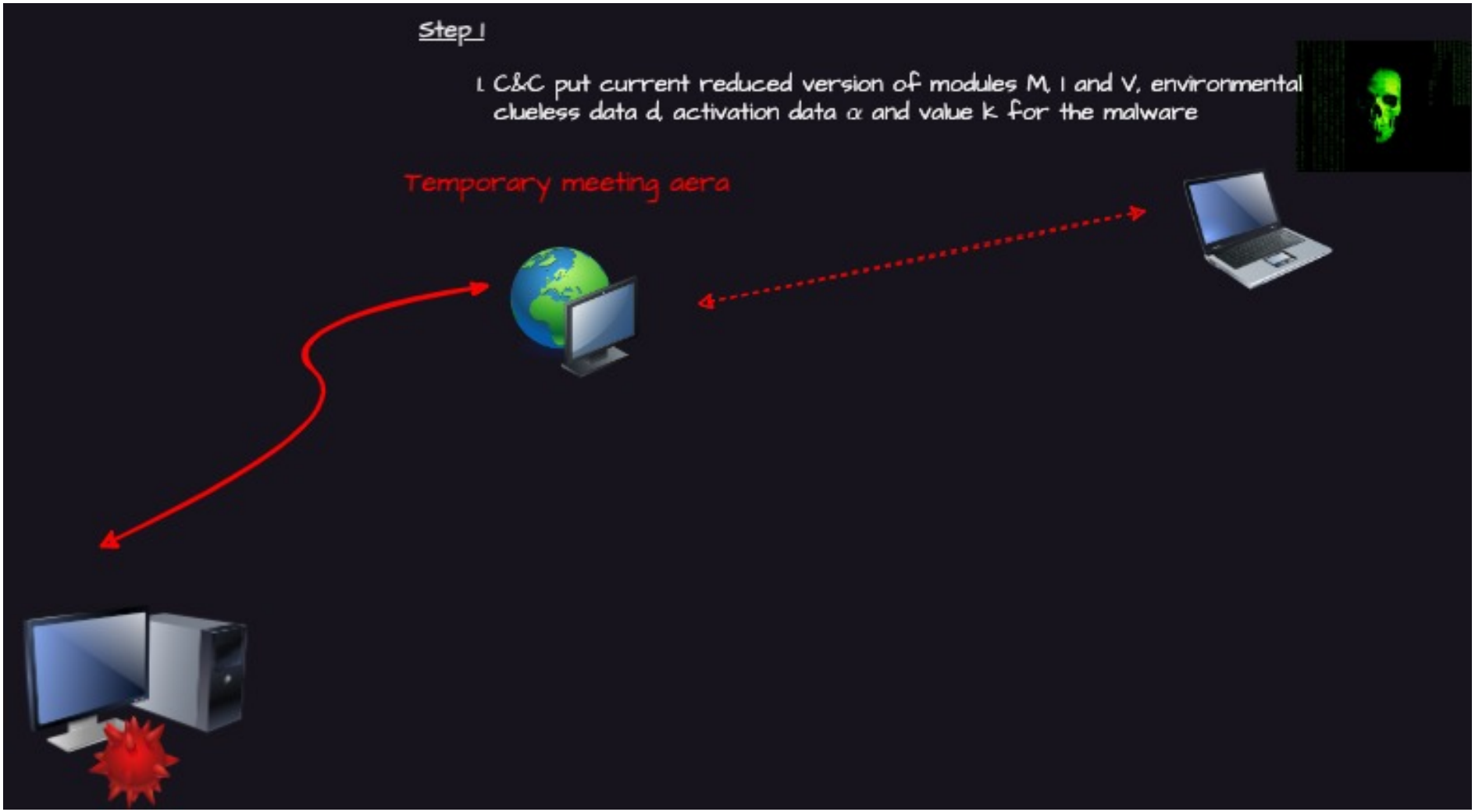
- The C&C analyses data collected and build a target computation model $\mathcal{M}_T$.
  - $\mathcal{M}_T$ can be compared to a reference model database.
  - $\mathcal{M}_T$ enables the C&C to build and use further clueless challenges for the malware at times fixed by the C&C.
  - $\mathcal{M}_T$ is not simulable.
- The actual encrypted malware is pushed to the precursor.

Attack Management Protocol

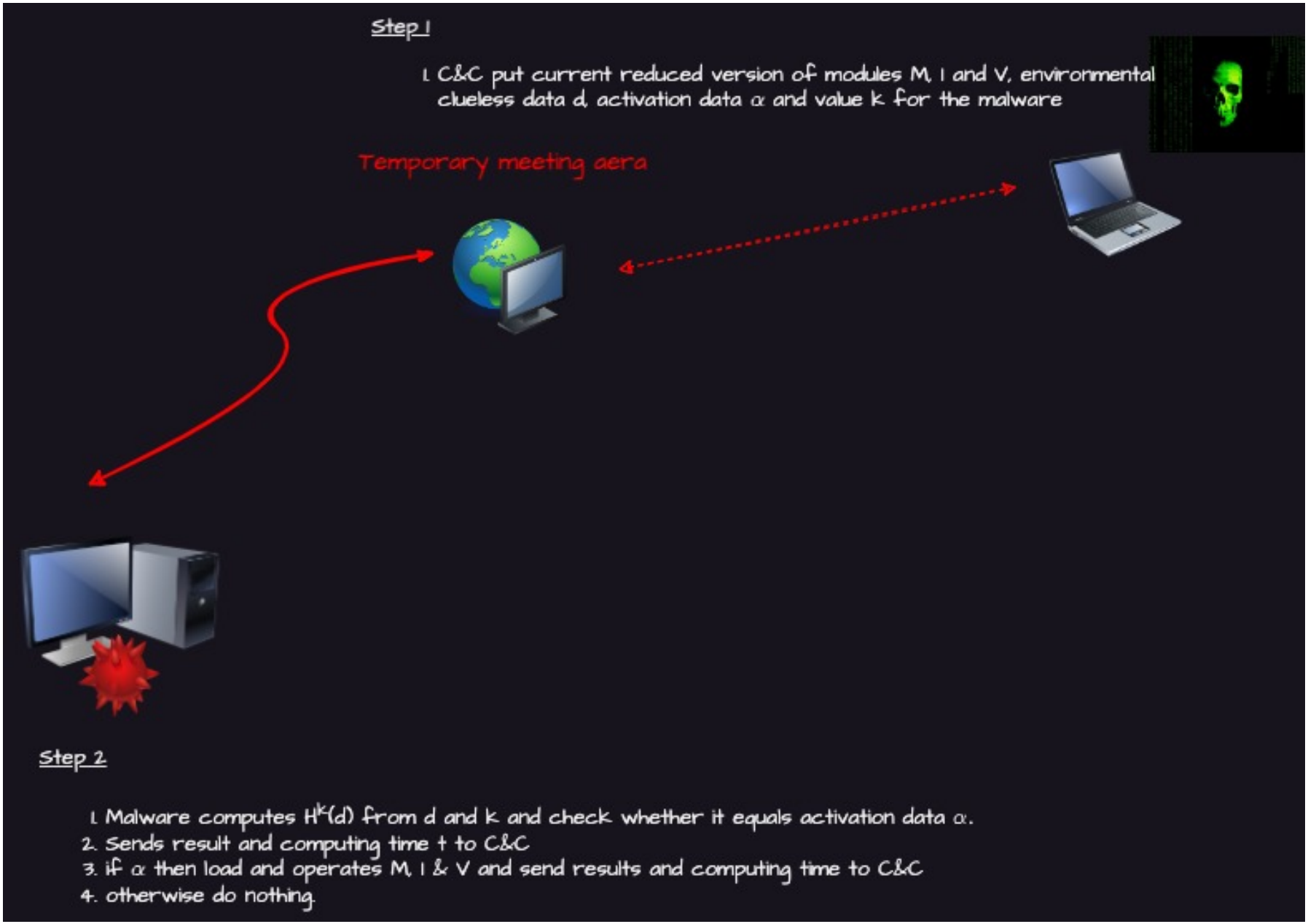


- Precursor self-updates with the malware code (actual attack part is encrypted).
- Environmental data are used to request frequent computations from the malware.
  - Random computation and data gathering (values and computation time) wrt modules M, I and V.

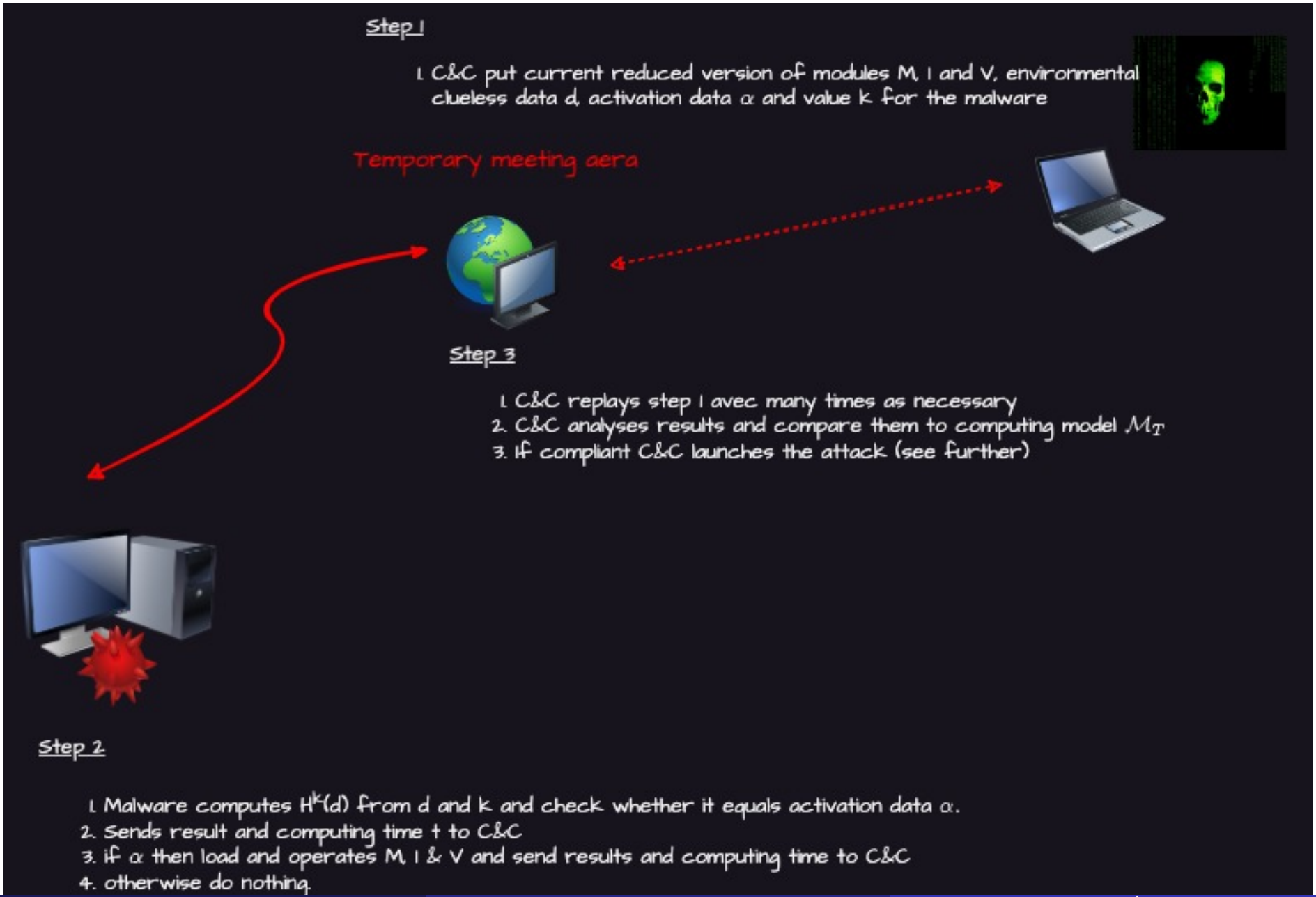
Activation Protocol



Activation Protocol



Activation Protocol



Agenda

Non-Trivial Deniable Cryptography: Principles

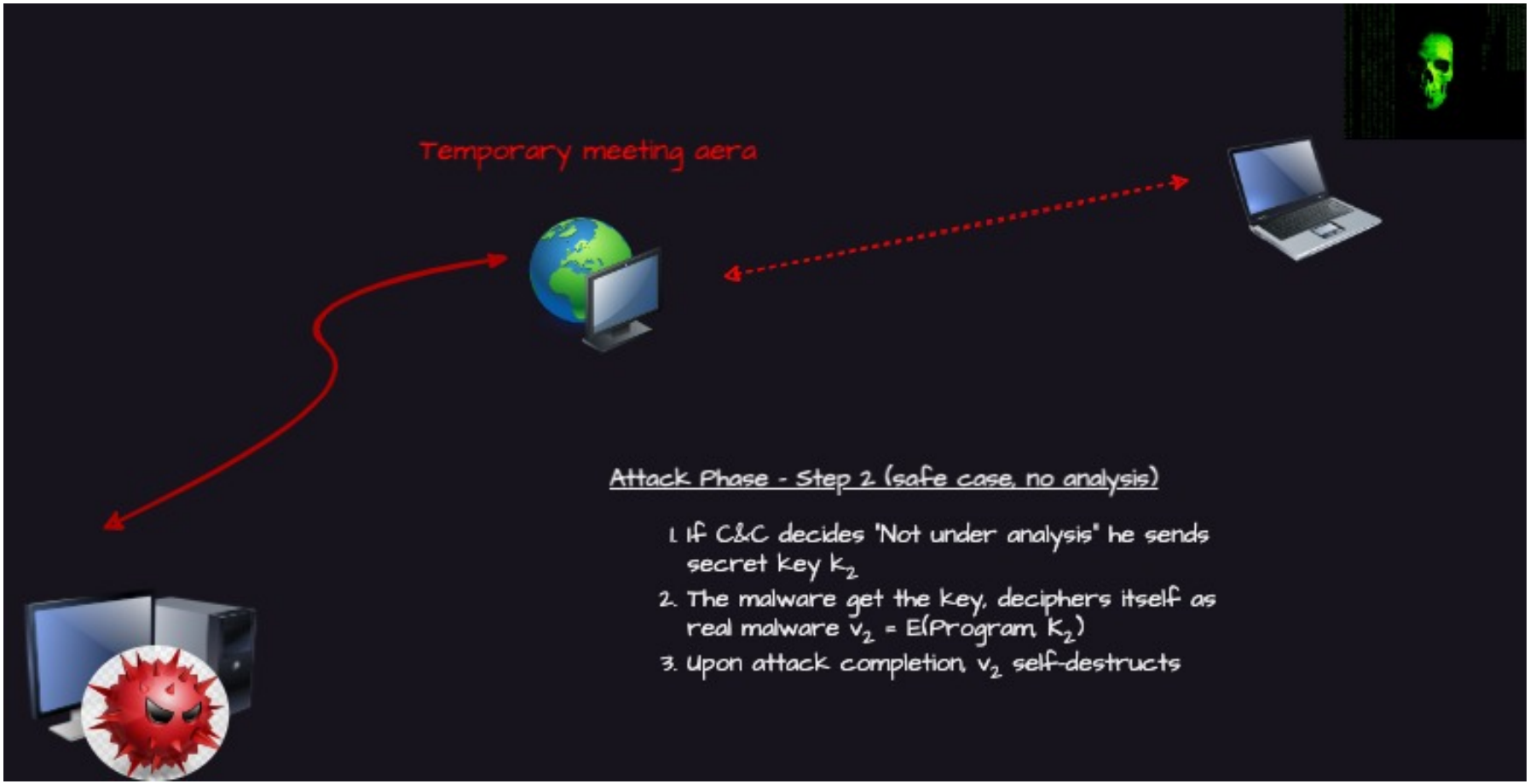
- Building effective encryption algorithm to realize **practical** deniable cryptography was until very recently still an open problem.
  - Only known case is trivial (one-time pads or OTP).
  - Since the "keys" are as long as the two (or more) plaintexts, not a valid solution (OTPs must be hardcoded).
- 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. **No secret element hardcoded**.
  - k is far smaller than N (so OTPs 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:
  - Code protection against static and dynamic analysis (see next slide).
  - Anti-forensics techniques.
  - Multiple communication channels in a single one (flow deniable cryptography).
  - ...
- **Demo**...

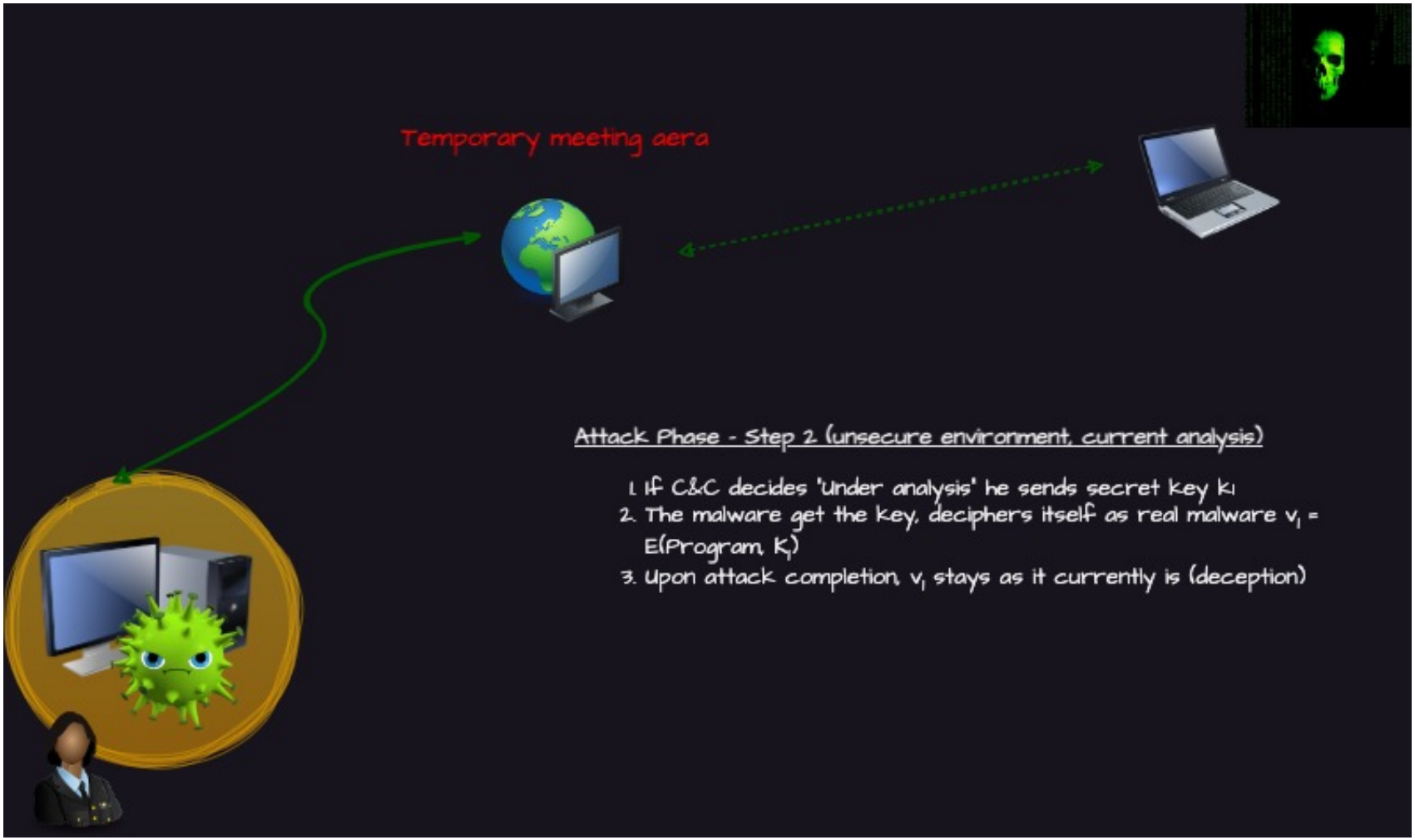
Deniable Cryptography-based Malware

- The most critical part of the malware is encrypted (C). Needs an external key from the C&C to operate.



Deniable Cryptography-based Malware

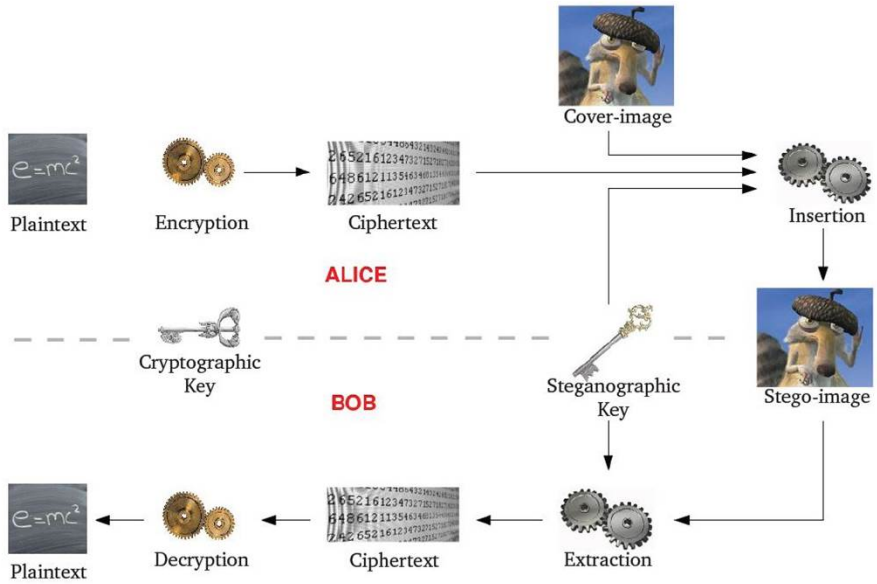
- The most critical part of the malware is encrypted (C). Needs an external key from the C&C to operate.



Agenda

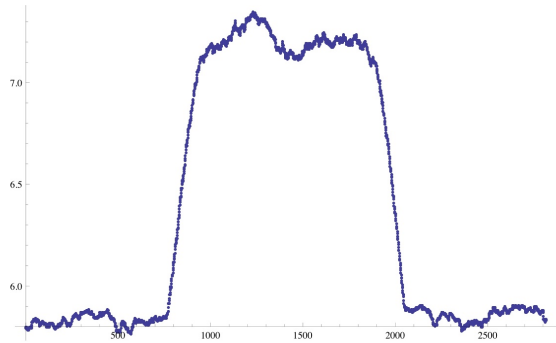
Cryptography versus Steganography

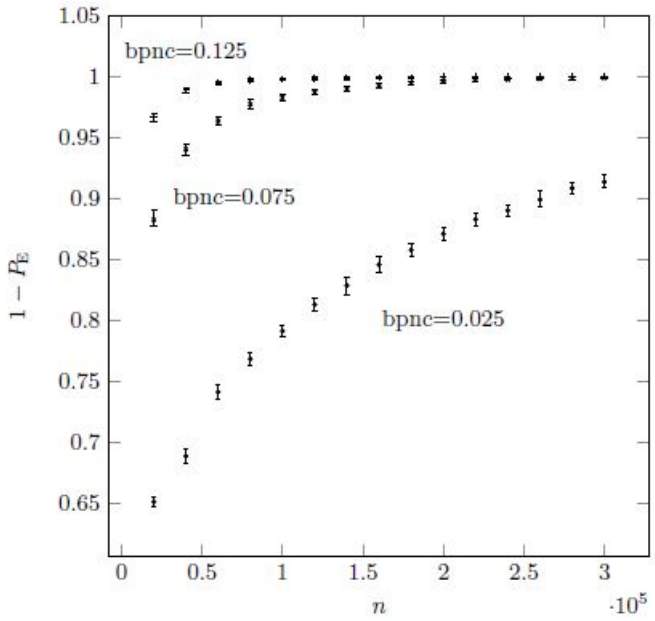
- Transform data with an encryption algorithm and a secret key into a random data.
- Accessing plaintext data is supposed to be practically intractable without the key.



- 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

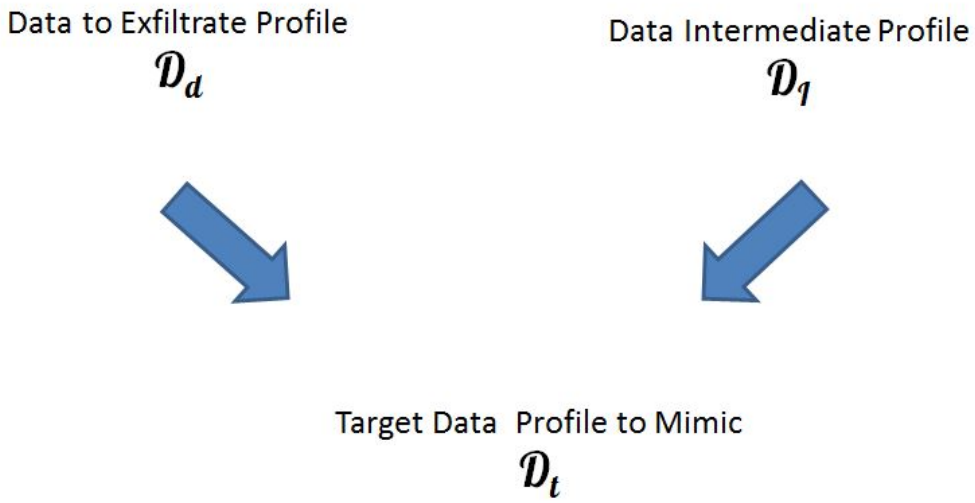




- Detection of cryptography is straightforward using the entropy profile [2].
  - Plaintext data: entropy $H(X) \approx 4$
  - Packed/compressed data: $H(X) \approx 6$
  - Encryption data: $H(X) = 8$

- 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).

Entropy and Statistical Profile Mimicking



- 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)
  - 128-bit key dependent transformation
  - Target distribution: FW log language
  - Target file of 8852 bytes (embedding rate of 17 %, no optimization)
  - FW log files are prone to be exchanged outside

Perfect Data Exfiltration

```
- Team Bad Dream
- "Attacks from Saudi Arabia"
Twitter : @x1337ksa
#opQatar

#     كلمة المرور - الاسم   الايميل   اسم المستخدم   نوع المستخدم   فعال   [تعديل]
1001        Yes   Administrator   admin   admin@admin.com   Administrator |pass=*******
1004        Yes   Staff   emp   sada@m.com   intranet |pass********
1006        Yes   Administrator   y.jassim   y.jassim@moj.gov.qa   y.jassim |pass********
1007        Yes   Administrator   mona   mona@moj.gov.qa   mona |pass*********
1009        Yes   Staff   maryam   maryam@moj.gov.qa   maryam |pass********
1010        Yes   Administrator   alaa   a.eid@moj.gov.qa   alaa |pass*******
1011        Yes   Administrator   saeed   s.alsuwaidi@moj.gov.qa   aseed |pass********
1012        Yes   Staff   alreem   Aa.Almalki@moj.gov.qa   alreem |pass*********
1013        Yes   Staff   aisha   a.alkobisi@moj.gov.qa   aisha alkbisi |pass*******
1014        Yes   Staff   muhsin   m.kandiyil@moj.gov.qa   muhsin |pass************
1015        Yes   Staff   alkhalifa   dsafsad@dsfsa.com   |pass********
1016        Yes   Staff   fatima   f.abdullraheem@moj.gov.qa   fatima |pass**********
1017        Yes   Staff   Zahra   z.alghanim@moj.gov.qa   Zahra Masoud Alghanim |pass*********
1018        Yes   Staff   loloa   l@moj.gov.qa   lolo |pass******
1019        Yes   Staff   Mohammed   m.alhandasi@moj.gov.qa   Mohammed Khusaif Alhandasi |pass*****

\ The End /
```

Perfect Data Exfiltration

```
Oct 25 13:17:11.31139 rule 84/0(match): block in on x10 8.8.158.145 : 20015 -> 10.0.0.3 : 80
Oct 25 13:18:12.990 rule 84/0(match): block in on x10 3.35.111.112 : 25077 -> 10.0.0.3 : 80
Oct 25 13:19:13.97116 rule 84/0(match): block in on x10 97.114.95.45 : 49427 -> 10.0.0.3 : 80
Oct 25 13:20:14.109111 rule 84/0(match): block in on x10 106.46.103.111 : 60887 -> 10.0.0.3 : 80
Oct 25 13:21:15.46113 rule 84/0(match): block in on x10 97.46.116.120 : 1877 -> 10.0.0.3 : 80
Oct 25 13:22:16.0125 rule 84/0(match): block in on x10 148.205.142.211 : 25304 -> 10.0.0.3 : 80
Oct 25 13:23:17.16199 rule 84/0(match): block in on x10 239.149.250.14 : 38282 -> 10.0.0.3 : 80
Oct 25 13:24:18.114110 rule 84/0(match): block in on x10 104.226.118.155 : 37883 -> 10.0.0.3 : 80
Oct 25 13:25:19.33181 rule 84/0(match): block in on x10 8.78.104.15 : 52744 -> 10.0.0.3 : 80
Oct 25 13:26:20.123225 rule 84/0(match): block in on x10 235.48.93.59 : 30854 -> 10.0.0.3 : 80
Oct 25 13:27:21.111226 rule 84/0(match): block in on x10 164.196.9.75 : 48495 -> 10.0.0.3 : 80
Oct 25 13:28:22.2368 rule 84/0(match): block in on x10 139.208.190.200 : 5665 -> 10.0.0.3 : 80
Oct 25 13:29:23.14160 rule 84/0(match): block in on x10 213.158.246.77 : 25189 -> 10.0.0.3 : 80
Oct 25 13:30:24.18397 rule 84/0(match): block in on x10 108.167.85.27 : 21632 -> 10.0.0.3 : 80
Oct 25 13:31:25.211214 rule 84/0(match): block in on x10 246.100.254.254 : 38849 -> 10.0.0.3 : 80
Oct 25 13:32:26.22725 rule 84/0(match): block in on x10 119.64.230.28 : 39049 -> 10.0.0.3 : 80
Oct 25 13:33:27.1216 rule 84/0(match): block in on x10 140.60.207.113 : 35126 -> 10.0.0.3 : 80
Oct 25 13:34:28.23913 rule 84/0(match): block in on x10 200.201.172.40 : 20606 -> 10.0.0.3 : 80
Oct 25 13:35:29.3486 rule 84/0(match): block in on x10 36.204.51.73 : 32408 -> 10.0.0.3 : 80
Oct 25 13:36:30.161100 rule 84/0(match): block in on x10 130.204.114.88 : 18951 -> 10.0.0.3 : 80
Oct 25 13:37:31.56233 rule 84/0(match): block in on x10 247.230.215.162 : 61667 -> 10.0.0.3 : 80
Oct 25 13:38:32.12078 rule 84/0(match): block in on x10 158.144.233.39 : 46803 -> 10.0.0.3 : 80
Oct 25 13:39:33.21073 rule 84/0(match): block in on x10 172.160.223.123 : 25026 -> 10.0.0.3 : 80
Oct 25 13:40:34.4595 rule 84/0(match): block in on x10 65.1.121.191 : 7767 -> 10.0.0.3 : 80
Oct 25 13:41:35.99162 rule 84/0(match): block in on x10 159.183.213.175 : 12225 -> 10.0.0.3 : 80
Oct 25 13:42:36.161250 rule 84/0(match): block in on x10 83.223.212.235 : 13699 -> 10.0.0.3 : 80
Oct 25 13:43:37.104214 rule 84/0(match): block in on x10 95.209.186.173 : 51685 -> 10.0.0.3 : 80
Oct 25 13:44:38.122248 rule 84/0(match): block in on x10 189.254.81.61 : 16147 -> 10.0.0.3 : 80
Oct 25 13:45:39.109213 rule 84/0(match): block in on x10 155.234.30.117 : 44389 -> 10.0.0.3 : 80
Oct 25 13:46:40.81183 rule 84/0(match): block in on x10 65.79.117.91 : 62098 -> 10.0.0.3 : 80
Oct 25 13:47:41.227160 rule 84/0(match): block in on x10 211.83.175.241 : 27127 -> 10.0.0.3 : 80
Oct 25 13:48:42.83111 rule 84/0(match): block in on x10 52.112.247.194 : 64311 -> 10.0.0.3 : 80
Oct 25 13:49:43.7164 rule 84/0(match): block in on x10 254.102.38.252 : 21781 -> 10.0.0.3 : 80
Oct 25 13:50:44.206188 rule 84/0(match): block in on x10 195.5.238.250 : 11248 -> 10.0.0.3 : 80
Oct 25 13:51:45.11956 rule 84/0(match): block in on x10 116.137.125.94 : 59714 -> 10.0.0.3 : 80
Oct 25 13:52:46.133237 rule 84/0(match): block in on x10 140.73.145.10 : 23535 -> 10.0.0.3 : 80
Oct 25 13:53:47.22880 rule 84/0(match): block in on x10 100.57.218.160 : 36680 -> 10.0.0.3 : 80
Oct 25 13:54:48.10963 rule 84/0(match): block in on x10 53.173.115.129 : 32531 -> 10.0.0.3 : 80
```

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). Care about some tags to avoid errors however.
  - 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).
- 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.
- Similar approach requiring to analyse the main behavioural distributions used in DLPs.
  - These distributions are often public or known.
  - A malware can analyse specific distributions in a preliminary identification/intelligence phase.

Agenda

Summary

- The potential of MCMM is huge and is bound to almost systematically defeat pure technical solutions and mitigations.
- The defender faces complexity and computing issues especially if the malware embeds adaptative behaviours and techniques.
- The huge potential of MCMM is very 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.
- Suitable security policies should be adopted especially whitelisting techniques (including time fencing, geofencing, portknocking...)

Conclusion

Thank you for your attention

Questions & Answers

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. 11–22, December 2008.