A Survey of Cryptologic Issues in Computer Virology

When Cryptology becomes malicious...

Eric Filiol

.

`efiliol@esat.terre.defense.gouv.fr`

`http://www-rocq.inria.fr/codes/Eric.Filiol/index.html`

Laboratoire de virologie et de cryptologie

Ecole Supérieure et d'Application des Transmissions

Introduction

Introduction

- Cryptology is the deep core of every computer security mechanism.

Introduction

- Cryptology is the deep core of every computer security mechanism.
- Dual of cryptoloy is essential and critical in computer virology.

Introduction

- Cryptology is the deep core of every computer security mechanism.
- Dual of cryptoloy is essential and critical in computer virology.
- Cryptologic techniques can put antiviral detection at check very easily.

Introduction

- Cryptology is the deep core of every computer security mechanism.
- Dual of cryptoloy is essential and critical in computer virology.
- Cryptologic techniques can put antiviral detection at check very easily.
- Until now they are not used a lot or very poorly implemented in practice:
  - There is worst in store... unless if it not already the case.

Plan

Plan

- A (very) Short Introduction to Cryptology and Computer Virology.

Plan

- A (very) Short Introduction to Cryptology and Computer Virology.
- Disseminating Codes: Random Generation for Worms.

Plan

- A (very) Short Introduction to Cryptology and Computer Virology.
- Disseminating Codes: Random Generation for Worms.
- Code Mutation: Polymorphism by Encryption.

Plan

- A (very) Short Introduction to Cryptology and Computer Virology.
- Disseminating Codes: Random Generation for Worms.
- Code Mutation: Polymorphism by Encryption.
- Code Armouring: the BRADLEY Technology.

Plan

- A (very) Short Introduction to Cryptology and Computer Virology.
- Disseminating Codes: Random Generation for Worms.
- Code Mutation: Polymorphism by Encryption.
- Code Armouring: the BRADLEY Technology.
- Some Other Aspects and Conclusion.

Taxonomy - Terminology

Cryptology

- Two main domains:

Taxonomy - Terminology

- **Cryptography.**- The study of optimal mathematical primitives and properties that can be used to design efficient algorithms to protect the confidentiality of Information.
  - Symmetric cryptography.
  - Asymmetric cryptography.

Taxonomy - Terminology

- **Cryptography.**- The study of optimal mathematical primitives and properties that can be used to design efficient algorithms to protect the confidentiality of Information.
  - Symmetric cryptography.
  - Asymmetric cryptography.

- **Cryptanalysis.**- The set of <u>mathematical</u> techniques which aim at attacking the core encryption algorithm to illegitimately access the encrypted message either directly or by recovering the secret key first.

Taxonomy - Terminology (2)

Taxonomy - Terminology (2)

- **Applied Cryptanalysis.**- The set of techniques which aim at attacking encryption mechanisms at the implementation level or at the key/algorithm management level: issue of the (armoured) security door on a paper wall.

Taxonomy - Terminology (2)

- Physical attacks: DPA, Timing Attack, BPA...
- Computer attacks: cache attacks, spying malware, CORE/PageFile....
- Human attacks: key compromission...

Taxonomy - Terminology (3)

Anti-antiviral techniques:

Taxonomy - Terminology (3)

Anti-antiviral techniques:

- **Stealth.**- Techniques aiming at convincing the user, the operating system and antiviral programs that there is no malicious code in the machine while indeed there is some.

Taxonomy - Terminology (3)

Anti-antiviral techniques:

- **Code mutation.**- Ability to make its own code change (encryption, rewriting) to bypass the sequence-based detection. Includes Polymorphism and Metamorphism.

Taxonomy - Terminology (3)

Anti-antiviral techniques:

- **Armouring.**- Ability to delay or forbid code (human-driven or software-driven) analysis through disassembly/debugging.

Random Generation and Worm Propagation

Random Generation and Worm Propagation

- To propagate, worms need to randomly generate target IP addresses.

Random Generation and Worm Propagation

- To propagate, worms need to randomly generate target IP addresses.
- The propagation must be time and space homogeneous (for most of classical worms).

Random Generation and Worm Propagation

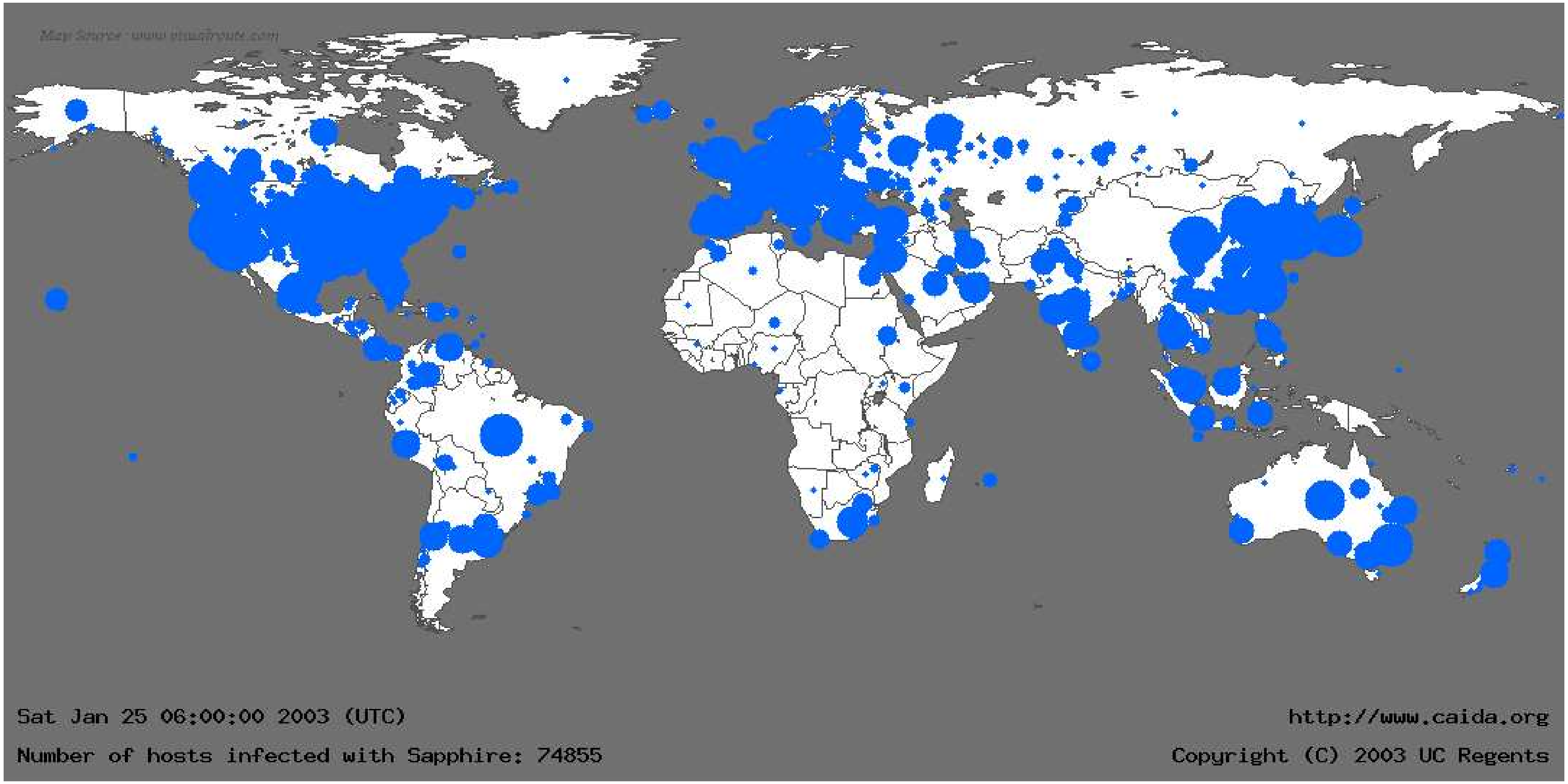
- To propagate, worms need to randomly generate target IP addresses.
- The propagation must be time and space homogeneous (for most of classical worms).
- The random generation process must be weighted and as good as possible.
    - IP addresses should be uniformly distributed, at least locally.

Random Generation and Worm Propagation

- To propagate, worms need to randomly generate target IP addresses.
- The propagation must be time and space homogeneous (for most of classical worms).
- The random generation process must be weighted and as good as possible.
  - IP addresses should be uniformly distributed, at least locally.
- Use of encryption primitives/algorithms to generate randomness.

The Sapphire/Slammer Case

The Sapphire/Slammer Case



The Sapphire/Slammer Case

- The randomness is very bad, due to a programming error.

```
DATA:00402138 mov esi, eax ;
DATA:0040213A or ebx, ebx ;
DATA:0040213C xor ebx, 0FFD9613Ch ;
```

The Sapphire/Slammer Case

- The worm uses the Microsoft modular congruential generator:

$$x_{n+1} = (x_n * 214013 + 2531011) \text{ modulo } 2^{32}.$$

The Sapphire/Slammer Case

- Register `EBX` should contain the constant value 2531011.
  - In fact, it contains the value `0FFD9613CH` xored with the *GetProcAddress* API address, in other words `77f8313H, 77e89b18H` or `77ea094H`.

The Sapphire/Slammer Case

- Second error: the increment value `0FFD9613CH` corresponds in fact to -2531011.
- Consequently this increment value is always either odd or even $\Rightarrow$ strong bias !
  - According to the parity of the x_0 initial value, the 32-bit values produced are either all even (even seed) or odd (odd seed).

The Sapphire/Slammer Case

- The bad quality of the random generation of IP addresses strongly hindered the own worm propagation.
- Strong concentration of the worm attacks in Asia.
  - South Korea has been disconnected from Internet during 24 hours.

The Blaster Worm Case

The Blaster Worm Case

- Weighted random generation of IP addresses.
- Very good randomness quality achieved.
- Nearly 1,000,000 targets infected during the 24 first hours.

The Blaster Worm Case

Let us consider a IPv4 address A.B.C.D, a random number N is produced:

- if $N < 12$ (proba = 0.6), random generation of bytes A, B and C ($D = 0$).
  - Addresses of type [1..254].[0..253].[0..253].0 (spreading to C subclass networks).
- otherwise (proba = 0.4), if byte C of local address > 20, le worm substracts 20 to C and D set to 0.

Code Mutation through Encryption

Code Mutation through Encryption

- Sequence-based detection is mostly used nowadays (Filiol - 2006; Filiol, Jacob, Le Liard - 2006).
  - Scan of more or less complex invariant patterns.

Code Mutation through Encryption

- Principle: the code encrypts/decrypts itself by means of a key that is different every time.

Code Mutation through Encryption

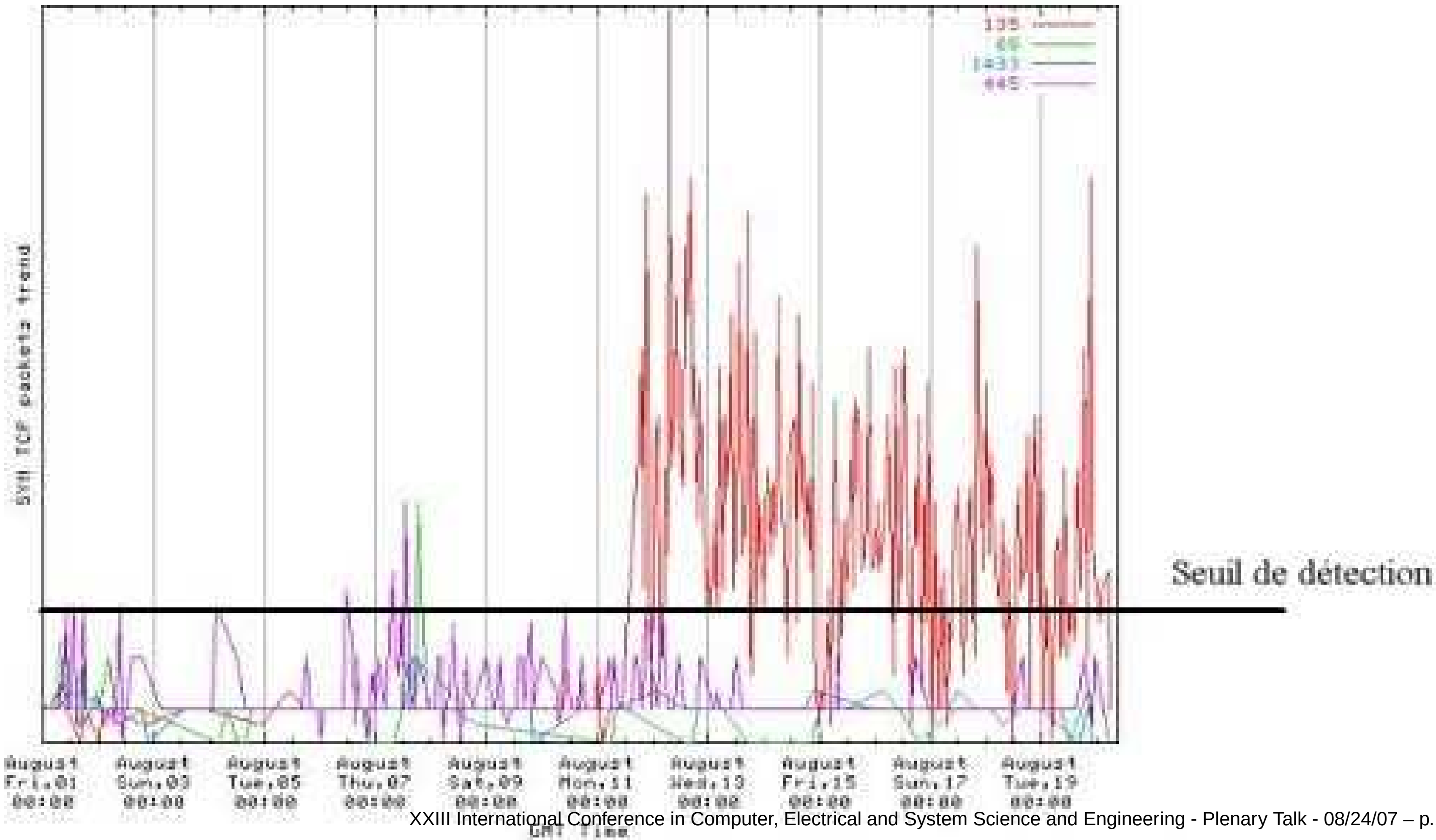
```
MOV EDI, OFFSET START_ENCRYPT ; EDI = viral
body offset
ADD EDI, EBP
MOV ECX, 0A6BH ; viral code size
MOV AL, SS:Key[EBP] ; the key (one byte)
DECRYPT_LOOP:
XOR [EDI], AL ; encr./decryp.  constant xor
INC EDI ; LOOP DECRYPT_LOOP

JMP SHORT START_ENCRYPT ; jump to the code

start
```

Code Armouring (1)

Code Armouring (1)

- Any (malicious or not) code can be analysed by (human-driven) disassembly/debugging.
- A high virulence enables the initial detection.
- The analysis enables to understand the attack and to update antivirus.

Code Armouring (1)



Code Armouring Techniques

Code Armouring Techniques

Definition 0 *(Armoured Code)Code which contains instruction or programming techniques whose purpose is to delay, make more complex or forbid its own analysis (generally by disassembly and/or debugging).*

Code Armouring Techniques

Different techniques used:

- *Code Obfuscation*: transform a program into another one which is functionally equivalent but more complex to analyse.
- Code mutation by rewriting.
- Code mutation by encryption.

Code Armouring Techniques

All these techniques are limited by nature:

- They are deterministic. They delay analysis at most.
- As for encryption, generally weak cryptographic primitives are used.
- Very poor key management.

Code Armouring Techniques

Whale Virus (September 1990) - First example known.

- Limited virulence.
- Encryption techniques of code in memory.
- Multi-layer encryption/obfuscation/code interleaving.
- Very poor cryptographic algorithms and no key management however.
- Able to detect a debugger in use and react accordingly.

Environmental Key Manegement

Environmental Key Manegement

- Cryptographic are built from environmental data only.

Environmental Key Manegement

- Cryptographic are built from environmental data only.
- The code itself ignores which data are used to build the key.

Environmental Key Manegement

- Cryptographic are built from environmental data only.
- The code itself ignores which data are used to build the key.
- The key is built when needed only.

Environmental Key Manegement

- Cryptographic are built from environmental data only.
- The code itself ignores which data are used to build the key.
- The key is built when needed only.
- The security model assumes the attacker (e.g. the code analyst) may have total control over the environment.

Some Constructions

Some Constructions

- N an integer corresponding to an environmental observation.
- $\mathcal{H}$ a one-way function.
- $M = H(N)$. The value M is carried by the code.
- R a random nonce.
- K a key.

Some Constructions

- if $\mathcal{H}(N) = M$ then $K = N$.
- if $\mathcal{H}(\mathcal{H}(N)) = M$ then $K = \mathcal{H}(N)$.
- if $\mathcal{H}(N_i) = M_i$ then $K = \mathcal{H}(N_1, N_2, \ldots, N_i)$.
- if $\mathcal{H}(N) = M$ then $K = \mathcal{H}(R_1, N) \oplus R_2$.

BRADLEY *Codes*

▪

BRADLEY *Codes*

.

- Family of proof-of-concept codes designed and tested in order to prove the existence of, study and evaluate the operational capability of total code armouring.

.

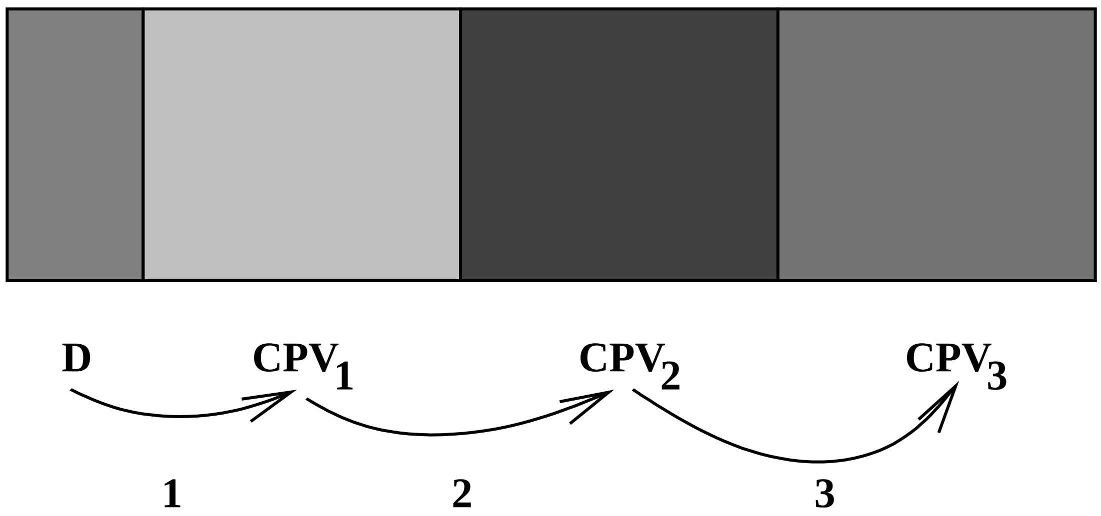
- Two main classes:
  - Class A.- Targeted codes to attack a specific group of users/machines.
  - Class B.- Targeted codes to attack a very small number of users/machines.

.

- Why using total armouring (from the malware writer's side)?
  - To forbid antivirus update.
  - To hide the malware actions.

BRADLEY *Codes*

.



BRADLEY **Codes**

.

- A decryption procedure D collects activation data, tests and evaluate them. If result is OK, D deciphers the different parts of the code.
- Code part EVP$_1$ (key K_1).- Anti-antiviral techniques (active and passive).
- Code part EVP$_2$ (key K_2).- Infection and propagation + metamorphism.
- Code part EVP$_3$ (key K_3).- Payload (optional; in our case to monitor the code activity).

Key Maganement Protocol

Environmental activation data (class A):

- local DNS address (e.g @company.com) denoted α,
- clock time (hh only) and system date (mmdd) denoted δ,
- a specific data which is present within the target system, denoted ι,
- a fixed specific data under the attacker's control's only; it is externally accessible to the code (e.g. a fixed data whose access is time-limited), denoted π.

Key Maganement Protocol

Class B:

- The data ι is a public key which is present into the target system (*pubring.gpg*).
- The code may target a very specific user.

Key Maganement Protocol

- D collects environmental data and computes

$$V = \mathcal{H}(\mathcal{H}(\alpha \oplus \delta \oplus \iota \oplus \pi) \oplus \nu)$$

where ν describes the first 512 bits in EVP$_1$.

Key Maganement Protocol

- If $V = M$ (M *activation data*) then

$$K_1 = \mathcal{H}(\alpha \oplus \delta \oplus \iota \oplus \pi)$$

otherwise D halts and the code self-disinfects.

- D deciphers EVP$_1$ to give VP$_1 = D_{K_1}($EVP$_1)$ and then executes it. Then D computes

$$K_2 = \mathcal{H}(K_1 \oplus \nu_2)$$

where ν_2 describes the first 512 bits in VP$_1$.

Key Maganement Protocol

- D deciphers EVP$_2$ to give VP$_2 = D_{K_2}($EVP$_2)$ and runs it. Then D computes

$$K_3 = \mathcal{H}(K_1 \oplus K_2 \oplus \nu_3)$$

  where ν_3 describes the first 512 last bits in VP$_2$.
- D deciphers EVP$_3$ to give VP$_3 = D_{K_3}($EVP$_3)$ and runs it.
- Once the code has operated, it totally self-disinfects.

Key Maganement Protocol

- From replication to replication, the whole has mutated (including D and M).
- Keys K_1, K_2 and K_3 may involve more environmental data.
- More sophisticated protocols and codes structures have been designed and successfully tested (e.g. detection of honeypots).

Mathematical Analysis

To evaluate the code analysis complexity, two cases have to be considered:

- the analyst has the binary code at his disposal,
- he has not.

The second case is the most realistic one (since the code self-disinfects). Let us however consider the first case.

Mathematical Analysis

Proposition 0 *Analysis of* BRADLEY *has an exponential complexity.*

Mathematical Analysis

- Decipherment procedure D leaks only:
  - the activation value $V = M$,
  - the fact that the system date and time are required,
  - the fact that data α, ι and π are required.
- A successful analysis needs to recover the exact secret key K_1 used by the code.

Mathematical Analysis

- Classical cryptanalysis.- For a (n, m)-hash function, we must perform $2^{\frac{3n-2m}{2}}$ operation.
- Dictionary attack.- We must perform 2^n operations.

All things being considered, the overall complexity is $\min(2^n, 2^{\frac{3n-2m}{2}}) = 2^n$ operations (2^{512} for SHA-1).

Tests

Tests

- Total Armouring combined with a limited virulence, effectively forbids code analysis.
- This concepts has been successfully tested in close network without any detection by existing AVs.
  - Attack launched at time t.
  - Effective propagation complexted at time $t + 15'$.
  - The data π was active between time $t + 1'$ and time $t + 15'$ only.
- A number of other cases have been tested (see bibliography).

Tests

- No technical solution against BRADLEY-like codes.
- Only solution: critical networks must be isolated.
- Strong security policies.

Other Aspects

Other Aspects

- Cryptology may be considered for the payload.

Other Aspects

- Cryptology may be considered for the payload.
- Retaliation or money extorsion (cryptovirus):
  - Virus *Ransom.A* and Trojan horse *Trojan.PGP.Coder* (2005).
- Applied cryptanalysis:
  - *Magic Lantern* worm (FBI - 2001).
  - *Ymun* codes (ESAT - 2002).

Other Aspects (2)

Other Aspects (2)

- Use of efficient cryptanalysis techniques to implement τ-obfuscation (Beaucamps - Filiol 2006):

Other Aspects (2)

- Use of efficient cryptanalysis techniques to implement τ-obfuscation (Beaucamps - Filiol 2006):
- The code encrypts itself and “throws” the key away.
- When executed, the code performs a cryptanalysis to recover the key.

Other Aspects (2)

- Use of efficient cryptanalysis techniques to implement τ-obfuscation (Beaucamps - Filiol 2006):
- The code can accept a significantly large operation time τ but not the antivirus.
  - Current improvement of E0 zero knowledge-like crytpanalysis (Filiol - 2006).
  - Other such cryptanalysis are under current research.

Conclusion

Conclusion

- Cryptology becomes a critical issue in modern computer virology.

Conclusion

- Cryptology becomes a critical issue in modern computer virology.
- There is a strong need to develop and maintain capability and skills in the cryptanalysis field.
  - Until now, the complexity of most of the underlying problems is still too high for an efficient antiviral action.
- Security policies must be strengthened to compensate.
  - This is the only solution at the present time!

Questions

Questions

Thanks for your attention!

References

- E. Filiol - Computer Viruses: from Theory to Applications, IRIS International Series, Springer, 2005 - ISBN 2-287-23939-1.
- E. Filiol - Techniques virales avancées, collection IRIS, Springer, 2007. An English translation is pending for end of 2007.
- Journal MISC - Le journal de la sécurité informatique - ISSN 1631-9030.