Malware Pattern Scanning Schemes Secure Against Black-box Analysis

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

- Most of the malware codes are variants of known codes or original strains (Fortinet - 2006).

Introduction

- Most of the malware codes are variants of known codes or original strains (Fortinet - 2006).
- Attackers analyze antivirus software to bypass them.

Introduction

- Most of the malware codes are variants of known codes or original strains (Fortinet - 2006).
- Attackers analyze antivirus software to bypass them.
- Defense-side: no strict, reproducible antivirus evaluation methodology (Josse - 2006).

Introduction

- Most of the malware codes are variants of known codes or original strains (Fortinet - 2006).
- Attackers analyze antivirus software to bypass them.
- Defense-side: no strict, reproducible antivirus evaluation methodology (Josse - 2006).
- No efficient theoretical model for antiviral techniques.

Introduction

- Most of the malware codes are variants of known codes or original strains (Fortinet - 2006).
- Attackers analyze antivirus software to bypass them.
- Defense-side: no strict, reproducible antivirus evaluation methodology (Josse - 2006).
- No efficient theoretical model for antiviral techniques.
- Is is possible to model, evaluate and make more secure antiviral techniques?

Introduction (2)

Introduction (2)

- Antivirus systematically use sequence-based detection relying on scanning techniques.
  - either directly (1st generation techniques),
  - or through a validation step by means of more “sophisticated” techniques (2nd generation techniques, heuristics, meta-heuristics...).
- Practically, bypassing scanning techniques enables to bypass antivirus.

Summary of the talk

Summary of the talk

- Introduction.

Summary of the talk

- Introduction.
- Mathematical model of scanning.

Summary of the talk

- Introduction.
- Mathematical model of scanning.
- The malware extraction problem.

Summary of the talk

- Introduction.
- Mathematical model of scanning.
- The malware extraction problem.
- A malware pattern scanning scheme secure against black-box analysis.

Summary of the talk

- Introduction.
- Mathematical model of scanning.
- The malware extraction problem.
- A malware pattern scanning scheme secure against black-box analysis.
- Conclusion.

Definitions and notation

Definitions and notation

- Let $\mathcal{M}$ be a malware and $\mathcal{S}_{\mathcal{M}}$ be a malware detection pattern of size s.

$$\mathcal{S}_{\mathcal{M}} = \{b_1, b_2, \ldots, b_s\}.$$

- A file $\mathcal{F}$ is said to be infected by $\mathcal{M}$ if there exist s indices in $\mathcal{F}$ $\{i_1, i_2, \ldots, i_s\}$ such that

$$\mathcal{F}(i_j) = b_{\sigma(j)} \qquad 1 \leq j \leq s,$$

where σ describe a permutation of $\mathcal{S}_{\mathcal{M}}$ bytes.

Definitions and notation

- Let X_j ($1 \leq j \leq s$) be variables defined as follows:

$$X_j = \begin{cases} 1 & \text{if } \mathcal{F}(i_j) = b_{\sigma(j)} \\ 0 & \text{otherwise.} \end{cases}$$

$X_j = 0$ means that the malware writer has modified $\mathcal{S}_{\mathcal{M}}(j)$.

Definitions and notation

- Let be $f_{\mathcal{M}} : \mathbb{F}_2^s \rightarrow \mathbb{F}_2$ where $\mathbb{F}_2$. A detector $\mathcal{D}$ decides that $\mathcal{F}$ is infected by $\mathcal{M}$ with respect to $f_{\mathcal{M}}$ and $\mathcal{S}_{\mathcal{M}}$ iff:

$$f_{\mathcal{M}}(X_1, X_2, \ldots, X_s) = \begin{cases} 1 & \mathcal{F} \text{ is infected by } \mathcal{M} \\ 0 & \mathcal{F} \text{ is non infected by } \mathcal{M}. \end{cases}$$

- The detection function $f_{\mathcal{M}}$ is described by monotone DNF formulae.
- On the attacking side, the non-detection function $\overline{f_{\mathcal{M}}} = 1 \oplus f_{\mathcal{M}}$ is considered instead.

Malware detection pattern properties

Malware detection pattern properties

- **Entropy and transinformation.-** Evaluation of the uncertainty an attacker must face when dealing with the pattern extraction pattern.

$$I(\mathcal{S}_{\mathcal{F},\mathcal{M}}; f_{\mathcal{M}}, \mathcal{S}_{\mathcal{M}}) = I(\mathcal{S}_{\mathcal{F},\mathcal{M}}; \overline{f_{\mathcal{M}}}, \mathcal{S}_{\mathcal{M}}).$$

Malware detection pattern properties

- **Resistance.-** $\mathcal{C}(\mathcal{S}_{\mathcal{M}}, f_{\mathcal{M}})$ describes the difficulty to bypass detection based on a given detection pattern.

$$\begin{aligned}\mathcal{R}(\mathcal{S}_{\mathcal{M}}, f_{\mathcal{M}}) &= \frac{\{X = (X_1, X_2, \ldots, X_s) | f_{\mathcal{M}}(X) = 0\}}{s2^s} \\ &= \frac{\{X = (X_1, X_2, \ldots, X_s) | \overline{f_{\mathcal{M}}}(X) = 1\}}{s2^s}.\end{aligned}$$

Malware detection pattern properties

- **Efficiency**.- Capacity to uniquely detect a given malware (false alarm probability tends towards 0).
- Theoretical probability to find a random s-byte pattern is $\frac{1}{236^s}$.

Malware detection pattern properties

- File bytes are NOT Bernouilli variables (Markov process instead).
- Observed prob. for $s = 24$ in a 0.5 Go corpus: 0.34 (theo. prob = 0.85×10^{-49}.
- Inside a W32 EXE, $P[b_0 =' M'] = 1$ and $P[b_1 =' Z' | b_0 =' M'] = 1$.
- Generally $s \geq 12$.

The malware pattern extraction problem

The malware pattern extraction problem

- Goal: identify $\overline{f_{\mathcal{M}}}$ and $\mathcal{S}_{\mathcal{M},\mathcal{M}}$.
- Black-box analysis. No reverse-engineering.
- Direct work on $\overline{f_{\mathcal{M}}}$.
- Computing $\overline{f_{\mathcal{M}}}$ from $f_{\mathcal{M}}$ has exponential complexity.

Naive algorithm

Naive algorithm

Input: A malware sample $\mathcal{M} = \{m_1, m_2, \ldots, m_n\}$, a malware detector $\mathcal{D}$ and a malware name σ.

Output: The malware pattern $\mathcal{S}_{\mathcal{M},\mathcal{M}}$ (pattern indices).

$\mathcal{S}_{\mathcal{M},\mathcal{M}} \leftarrow \{\}$
for i = 1 to n do
    modify only byte m_i in $\mathcal{M}$
    If $\mathcal{D}(\mathcal{M}) \neq \sigma$ then
        $\mathcal{S}_{\mathcal{M},\mathcal{M}} \leftarrow \mathcal{S}_{\mathcal{M},\mathcal{M}} \cup \{m_i\}$
    end if
end for
return $\mathcal{S}_{\mathcal{M},\mathcal{M}}$

General optimized algorithm

General optimized algorithm

Input: $\mathcal{M} = \{m_1, m_2, \ldots, m_n\}$, $\mathcal{D}$ et σ. **Output**: $\mathcal{S}_{\mathcal{M},\mathcal{M}}$ et $\overline{f_{\mathcal{M}}}$ DNF.

$\mathcal{S}_{\mathcal{M},\mathcal{M}} \leftarrow \{\}$; $\text{DNF}_{f_{\mathcal{M}}} \leftarrow \{\}$; $S \leftarrow 2^{\lfloor \log_2(n) \rfloor + 1}$; $S' \leftarrow 2^{\lfloor \log_2(n) \rfloor + 1}$

While $S > 0$ do

  $S \leftarrow \frac{S}{2}$; $q \leftarrow \frac{S'}{S}$

  For i = 0 to q -1 do

    binf $\leftarrow i \times S$; bsup $\leftarrow$ binf $+ S$

    modify bytes in $I =$ [binf, bsup[in $\mathcal{M}$

    If $\mathcal{D}(\mathcal{M}) \neq \sigma$ then

      $\mathcal{S}_{\mathcal{M},\mathcal{M}} \leftarrow \mathcal{S}_{\mathcal{M},\mathcal{M}} + I$; $X = 1_I(X_1, X_2, \ldots, X_n)$

      $\text{DNF}_{f_{\mathcal{M}}} \leftarrow \text{DNF}_{f_{\mathcal{M}}} \cup X$

    End If

  End For

End While

$\mathcal{S}_{\mathcal{M},\mathcal{M}} \leftarrow$ CombinatorialMinimize$(\mathcal{S}_{\mathcal{M},\mathcal{M}})$;

$\text{DNF}_{f_{\mathcal{M}}} \leftarrow$ LogicalMinimize$(\text{DNF}_{f_{\mathcal{M}}}, \mathcal{S}_{\mathcal{M},\mathcal{M}})$

Computing complexities

Computing complexities

- Algorithm E-1 : complexity in $\mathcal{O}(n)$.
- Extraction of the following detection function only:

$$f_{\mathcal{M}}(X_1, X_2, X_3, \ldots, X_s) = X_1 \wedge X_2 \wedge X_3 \wedge \ldots \wedge X_s.$$

- Most frequent case.

Computing complexities

- Algorithm E-2 : complexity in $\mathcal{O}(sn + s2^s)$ where n is the size of the code.
- Optimized use of the *membership and equivalence queries* method.
- Algorithm E-2 extract the exact DNF formula whereas generic learning method only produce equivalent (non simplified) DNF formulae.

Examples of non trivial detection functions

Examples of non trivial detection functions

- *Wildcard technique*.- Let be the detection pattern

```
..... 07BB ??02 %3 33C9 .....
```

- The corresponding detection function is given by:

$$
\begin{aligned}
\ldots \quad & (X_i = 07) \wedge (X_{i+1} = BB) \wedge (X_{i+3} = 02) \wedge (X_{i+4} = 33) \wedge (X_{i+5} = C9) \\
\vee \quad & (X_i = 07) \wedge (X_{i+1} = BB) \wedge (X_{i+3} = 02) \wedge (X_{i+5} = 33) \wedge (X_{i+6} = C9) \\
\vee \quad & (X_i = 07) \wedge (X_{i+1} = BB) \wedge (X_{i+3} = 02) \wedge (X_{i+6} = 33) \wedge (X_{i+7} = C9) \\
\ldots &
\end{aligned}
$$

Examples of non trivial detection functions

- *Mistmatch technique*.- Let be the detection pattern `01 02 03 04` with mismatch value of 2.
- The corresponding detection function is given by:

$$\begin{aligned} \ldots \quad & (X_i = 01) \wedge (X_{i+1} = 02) \vee (X_i = 01) \wedge (X_{i+2} = 03) \\ \vee \quad & (X_{i+1} = 02) \wedge (X_{i+2} = 03) \vee (X_i = 01) \wedge (X_{i+3} = 04) \\ \vee \quad & (X_{i+1} = 02) \wedge (X_{i+3} = 04) \vee (X_{i+2} = 03) \wedge (X_{i+3} = 04) \end{aligned}$$

- For a pattern of size s with mismatch value μ ($\mu < s$), the DNF contains $\binom{s}{\mu}$ terms, each of them having $s - \mu$ literals ($(s - \mu)$-DNF).

Results

Results

- **Entropy and transinformation.-** For all tested products, the transinformation is maximal:

$$I(\mathcal{S}_{\mathcal{M},\mathcal{M}}; \overline{f_{\mathcal{M}}}, \mathcal{S}_{\mathcal{M}}) = H(\mathcal{S}_{\mathcal{M},\mathcal{M}}).$$

- Having the detector at one's dipsosal enables to suppress any uncertainty.
- Every antivirus we have tested are weak with respect this property.

Results

- **Resistance.-** It depends on the weight of the detection function.
  - For the AND detection function:

$$\mathcal{C}(\mathcal{S}_{\mathcal{M}}, f_{\mathcal{M}}) = \frac{1}{2^s - 1}.$$

  - For the other detection function, we have to evaluate wt$(\overline{f_{\mathcal{M}}})$.
- A resistant detection function must have a large weight.
- Nearly all tested antivirus are weak with respect to resistance.

Results

- **Efficiency**.- Nearly all tested AV products use small detection patterns.
- Trade-off between efficiency and resistance to consider for the AND function.
- The shorter a signature is, the easier the bypassing is but the higher false alarm probability is (AND function).
- Patterns are very similar from one AV products to another.

A secure scanning scheme

A secure scanning scheme

- We consider a s-byte detection pattern whose only a k-byte fixed sub-pattern will be used under the following constraints:
  - $k << s$,
  - sub-patterns are randomly chosen,
  - each sub-pattern is fixed for a given user/machine,
  - the whole pattern cannot be extracted except when knowing at least τ sub-patterns,
  - the number π of sub-patterns and τ must be large enough.

A secure scanning scheme

- Use of $2-(s,k,\lambda)$ combinatorial designs.
- We have $\mathcal{S}_{\mathcal{M}} = \mathcal{V}$, $s = v$, $\pi = b = \frac{\lambda v(v-1)}{k(k-1)}$ and $\mathcal{B}$ describes the set of all sub-patterns.
- We choose the detection function given by:

$$f(X_1, X_2, \ldots, X_s) = X_1 \oplus X_2 \oplus \ldots X_s.$$

A secure scanning scheme

- Initial step. When the antivirus is set up, following data are collected:
  - the processor ID (CPUID) and the hard disk ID (HDID);
  - the MAC address (MACadr);
  - the username (USRname) and its email address (@adr);
  - a secret value hiddent in the antivirus software (ν).

A secure scanning scheme

- The antivirus software computes

$$i = g(H(\text{CPUID} \oplus \text{HDID} \oplus \texttt{MACadr} \oplus \texttt{USRname} \oplus \texttt{@adr} \oplus \nu) \oplus \mathcal{N}).$$

- The antivirus sofware will use only the ith sub-pattern.

A secure scanning scheme

Proposition 0 *$\mathcal{S}_{\mathcal{M}}$ and $f_{\mathcal{M}}$ can be extracted when at least $\tau = \lceil \frac{s}{k} \rceil$ analysts collude.*

Proposition 0 *The detection function formula which has been extracted by a collusion of size $\tau = \lceil \frac{s}{k} \rceil$ contains at most $\frac{s}{k}(2^k - 1)$ terms.*

Proposition 0 *The knowledge of both $\mathcal{S}^i_{\mathcal{M}}$ and $f^i_{\mathcal{M}}$ enables to produce a variant which is still detected with a probability $P_{detection}$ such that*

$$\frac{1}{2} \leq P_{detection} \leq 1.$$

Implementation

Implementation

- Use of RBIBDs (*Resolvable BIBD*) and parallel classes.
- For a RBIBD, $P_{\text{détection}} = \frac{\pi-1}{\pi}$.
- Memory complexity in $\mathcal{O}(\frac{s^2}{k^2})$ for a $2-(s,k,\lambda)$ generic design.
- No significant reduction of scanning time.
- Optimization in progress.

Conclusion and open problems

Conclusion and open problems

- Every tested AV exhibit very weak scanning schemes.
- The black-box analysis is very easy to perform.
- Consequently, viral production is made easy as well.

Conclusion and open problems

- This approach has been successfully generalized to other detection techniques like behavior analysis (Filiol - Jacob - Leliard 2006).
- Metaheuristics and Boolean circuits.

Conclusion and open problems

Some open problems (to attract Ph. D students):

- classification and exploration of “good” detection functions $f_{\mathcal{M}}$,
- analysis of combinatorial properties of $f_{\mathcal{M}}$ that may increase pattern efficiency.
- study of “good” combinatorial objects (pairwise designs, parallel designs...).

Questions

Questions

Many thanks for your attention !