Applied evaluation methodology for anti-virus software

Jean-Baptiste Bédrune
Sogeti / ESEC R&D
jean-baptiste.bedrune(at)sogeti.com

Alexandre Gazet
Sogeti / ESEC R&D
alexandre.gazet(at)sogeti.com

EICAR Conference 2009

Roadmap

CSPN Evaluation framework

Why a new evaluation framework?

- `Common Criteria` (CC) evaluation framework is sometimes not well adapted
- The process may be too long, too expensive; too much formal, it does not reflects the real effectiveness of a product, etc.

CSPN key concepts

- Reference time: 20 days
- Effective compromise between evaluation depth and evaluation cost
- Ensuring all the security functions reach a basic resistance level
- Product is considered into its environment

How does it work?

The actors

- The DCSSI
- The silent partner: the one who asks for the evaluation
- The evaluation center

The steps

1. The **Security Target (ST)** is validated by the DCSSI
2. The evaluation center carries out the evaluations *wrt* the **ST**
3. It sends an **Evaluation Technical Report** (**ETR**) to the DCSSI
4. Based on the **ETR**, the DCSSI delivers the certification

Antivirus software distinctive characteristics

- Viral code detection, from a formal point of view, is proven to be undecidable
- The answer of an evaluation cannot be as simple as a binary answer

An evaluation must then be performed to assess how much the anti-virus software is imperfect, to evaluate the difficulty for an attacker to thwart the protection

Our objectives

- To propose an evaluation methodology for antivirus software
- That fits into the **CSPN** framework
- That take into consideration antivirus software distinctive characteristics
- Both a formal (conformity) and a true technical and effectiveness-oriented evaluation

First step: we have to define a security target

What is an antivirus?

Silly question?

- What do we expect from them ⇒ functionnal requirements
  - Scanning (obvious)
  - Audit
  - Management
  - Self-protection
- Many products offer additional functionalities:
  - secure backup, performance tweaking, etc.

A reference document

US Government Protection Profile - Anti-virus Applications for Workstations in Basic Robustness Environments Version 1.2 25 July 2007.

Security target

We have defined a security target from a reduce protection profile

Functional requirement	Description
FAV_ACT_SCN.1	**Anti-Virus scanning**
FAV_ACT_SCN.1.1	Real-time scanning or resident protection
FAV_ACT_SCN.1.2	Real-time on-demand scans
FAV_ACT_SCN.1.3	Scheduled scans
FAV_ACT_EXP.1	**Anti-Virus actions**
FAV_ACT_EXP.1.1	Prevent execution upon detection of a memory-based virus
FMT	**Security management**
FMT_MOF	Management of security functions behaviour
FMT_SMR	Definition of security roles

Our evaluation methodology

1. Product identification
2. Product specification
3. Product installation
4. Conformity analysis:
   1. Functionalities
   2. Documentation
5. Robustness of cryptographic mechanisms
6. Form analysis review
7. Behavioural analysis review
8. Operational attack scenario
9. Vulnerability analysis

Viral detection review

One or many detection scheme(s)

- Form analysis: signatures, heuristics, etc.
- Behavioural analysis: experts systems, states machines, etc.

Reference works

- `DrWeb` and `OneCare` evaluations made by ESAT laboratory
- "How to assess the effectiveness of your anti-virus", Sébastien Josse (`EICAR 2006`)
- Previous works by Filiol, Jacob, Josse and Quenez,

Cryptographic mechanisms review

- Cryptographic mechanisms are a critical resource
- Most anti-virus software are closed source software
- Communication between the editor and the evaluation center[a]

[a] *cf.* Information required for the analysis of cryptographic mechanisms by DCSSI

Evaluation with respect to?

US `NIST FIPS 140-2` standards from US Government

FR Rules and Recommendations for Cryptographic Mechanisms with Standard Robustness from the DCSSI

Open questions

- How to handle extra functionalities (secure backup, performance tweaking), how to evaluate their impact?
- Rootkit detection?
- Vulnerability exploitation detection?
- Where is the boundary between antivirus and `HIPS/IDS`

Roadmap

1. Definition of a methodology
2. Pilot evaluation
   - Introduction
     - Product description
     - Test environment
     - Security target
     - Differences with a real evaluation
   - Form analysis
   - Behavioural analysis
   - Attack scenario
   - Vulnerability analysis
3. Conclusion

NOD32 v3

- Antivirus with few annex features
- Renowned, widely used by home users
- Very few vulnerabilities published
- Known for the quality of its scan engine and its speed

New version available: v4

- Final version not available when evaluation started
- Vendor preferred version 3 to be evaluated

Architecture setup

VMware virtual machines

- Tests environment more flexible
- Tests can be reproduced
- **Sometimes a problem when evaluating an antivirus**
  - Several malware detect they are executed in a VM
  - Choose carefully your set of malware samples

Evaluation under Windows XP SP3 32bits

Conformity analysis

Features proposed by the product have been evaluated according to the proposed security target $\Rightarrow$ conformity analysis

2 non-conformities

A non-privileged user can:

- update the signatures base
- read machine logs

Contrary to the specification of the role separation: these actions should be reserved to an administrator.

Nevertheless, the product verifies all the other items of the security target.

Not a real evaluation

- Security target not validated by the DCSSI
- No reverse engineering
  - We did not ask ESET
- Cryptographic specifications not asked to ESET
  - Not evaluated
- **No certification**

Performance

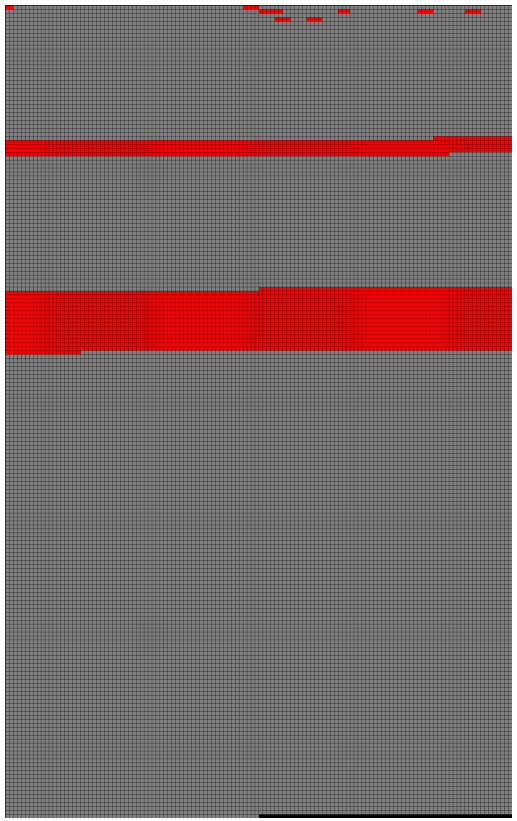
- Set of 4421 samples
- Detection rate? Scan speed?

Product	Detection rate	Time taken
NOD32 (Signatures)	53%	3min 39s
NOD32 (Heuristics)	88%	30min 3s
Another product	93%	46min

Signature extraction

Extraction method

- Each byte of a given malware is modified
  - Bit inversion
- Scan of each of the mutated files
- Signature is directly extracted



Advanced heuristics

Test of heuristics

- "Signature" extraction
- Comparison with the previous result

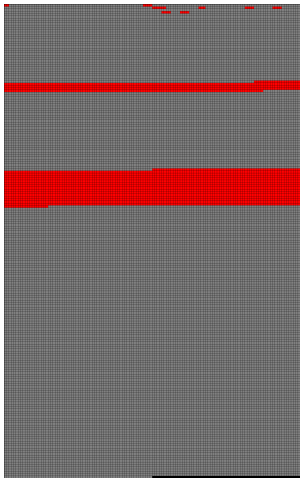


Figure: Signatures only

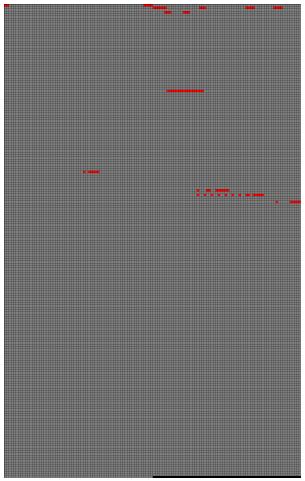


Figure: Advanced heuristics

Bypassing heuristics

How?

"Signature" = important data. Undetected executables do not run correctly

- Entry point
- Data needed by the virus

⇒ Modifications?

Polymorphism on the entry point	Detected
Data modification	Detected

⇒ Bypassing the antivirus with modified strains is not easy

File formats

Methodology

- Set of files of common format
  - Archives: `zip`, `gzip`, `bz2`, etc.
  - Disk images: `iso`, `bin/cue`, etc.
  - ...
- `eicar.com` is inserted into these files
- Tests on our set of files

File formats: results

Type	Format	Result
Archives	zip, rar, tar, 7z, ace, cab, bzip, gzip	Detected
Images	ISO, BIN/CUE, Nero	No scan
Other formats	tnef, sis, nsis, chm, etc.	Detected

We do not know why disk images are not scanned.
No answer from the editor.

Packers

The same method has been used.

Packer name	Handled?	Detection
Armadillo 6.24	No	-
ASPack 2.12	Yes	Win32/Small.NAN.Worm
ASProtect 1.4	Yes	Probably a variant of Win32/Agent.NAS.Worm
FSG 2.0	Yes	Win32/Small.NAN.Worm
JDPack v2.00	No	-
PECompact v2.98	Yes	Win32/Small.NAN.Worm
UPX 3.01	Yes	Win32/Small.NAN.Worm

Very good results

Method

Functional polymorphism

- Mutation of a given behaviour into an equivalent behaviour
- Example: a surinfection marker
  Test if a mutex is present ⇒ Test if an event is present

Creation of a set of modified strains

- Common strain, with many features: `MyDoom.A`
- Functional polymorphism applied to the whole strain
- Signature is removed from the binary

Results

- None of the variants has been detected
- As most antivirus, NOD32 is mainly based on its form analysis
- It confirms the results of a previous work (Evaluation methodology and theoretical model for antiviral behavioural detection strategies, E. Filiol, G. Jacob, M. Le Liard)

Attack context

Method

- Simulation of a real attack
- Infection vector: USB key
- Backdoored Word documents are used
- Three exploits:
  - Two of them are public
  - The other one comes from a targeted attack

Document	Exploit detected?	Virus detected?
File 1	Variant of `Exploit.MSWord.Smtag`	N/A
File 2	No	Variant of `Hupigon Trojan`
File 3	No	No

Existing vulnerabilities

CVE List

- A few vulnerabilities on versions 1 and 2
- No vulnerability on version 3

Local privilege escalation on version 3 (patched)

- Source: `www.orange-bat.com`

Methodology

- Not a lot of time spent on vulnerability research
- Mainly fuzzing on handled file formats
  - Simplistic approach: random modifications on existing files
  - Checksum correction
  - Scan of the newly created files

ACE archive parser

- Denial of Service on the scan engine
- Pointer on the beginning of file can be controlled with a crafted archive
- Exception handled by the engine, that scans the file again

$\Rightarrow$ Infinite loop

No reverse engineering
$\Rightarrow$ Exploitation?

Corrected within a day by the vendor

UPX executables parser

- Denial of Service on the scan engine
- Creation of an "infinite" compressed stream
- Stream is written on the disk
- Engine can not be stopped until the disk has been fullfilled

Corrected within a day by the vendor

Admin password

Storage

- Hash saved in the registry
  - Modified `CRC32`
  - Readable by **all the users**
- collisions obtained **almost** instantly
  - Printable chars
  - Encoding: `UTF-16`

Collision

$$H(\texttt{"mypassword"}) = H(\texttt{"myBD3miKqm"})$$

It is easy to get solution administrator privileges

$\Rightarrow$ Biggest vulnerability found during the evaluation

Not corrected

Conclusion of the evaluation

- Conformance with the security target
- Good documentation
- Impressive form analysis
  - Fast
  - Good detection of variants
- Ineffective behavioural analysis
- Several vulnerabilities found
  - ...including a very embarrassing one
  - Vendor's reactivity must be pointed out

Roadmap

1. Definition of a methodology
2. Pilot evaluation
3. Conclusion

Conclusion

- Our methodology is thought to be a compromise between
  - the formalism of a CC evaluation: a rigorous security target
  - the applied aspect of a CSPN evaluation, effectiveness oriented
- It is a complete, while open, evaluation framework
- The pilot evaluation was very interesting:
  - Discerning understanding of the product (strength and weakness)
  - Constructive dialog with the editor

Conclusion

Thanks for your attention.

Any questions?