ISSN 1870-4069

Optimation through Automation of Malware Update Process, Capable of Evading Anti-Malware Systems

Daniel Soto Carabantes[1], Cristian Barría Huidobro[2], David Cordero Vidal[3]

[1] Universidad Adolfo Ibáñez, Center for Social and Cognitive Neuroscience (CSCN),
School of Psychology, Santiago,
Chile

[2] Pontificia Universidad Católica de Valparaíso, Valparaíso,
Chile

[3] Universidad Andrés Bello, Santiago,
Chile

daniel.sotoc@uai.cl, cristian.barria@udp.cl, d.cordero.v@gmail.com

Abstract. Implementation and maintenance of malware protection measures imply high resources usage. Such is the case of Information Security Management Systems (ISMS), whose suggested structure is described by ISO Standard 27.001:2013. In this standard, work with malware is contemplated for penetration testing (pentesting) purposes, allowing to evaluate the response of computer systems against this kind of events. The present document approaches one of the existing malware usage methods for this purpose: encrypted malware obfuscation, through dead code insertion. This method is evaluated in terms of monetary cost and required time, through simulation, to later evaluate those metrics against an automated model, tested through a prototype software. The optimization of this process through the proposed automation, yielded a significant reduction of the monetary cost and time needed.

Keywords: Malware, obfuscation, automatization.

1 Introduction

Achieving a precise estimation of the actual economic cost caused by malware is an expensive and very complex task, which has been discussed by many authors and institutions [1, 2, 3, 4, 5, 6, 21], but that has not resulted in the development of a tool nor system solid enough to be accepted worldwide, although there is a consensus that such costs are high and are increasing [7].

In 2010, economic losses related to cyber crime were estimated at USD $70 millions (at consumer-level), amount that reaches USD $30 millions at business-level. In 2014, annual losses at a global scale went up to USD $400 billions [1], evidencing the explosive impact of these kind of crimes in the world.

Bernik states that a large portion of the costs that malware causes to organizations, are not related to the crimes per se, but to the implementation and maintenance of protection systems [1], making it harder for organizations with less resouces to get access to such protection measures.

Latin America has not been oblivious to this situation, although the countries in this region acknowledge the importance of approaching this issue, current knowledge about cyber threats in this region is reduced [9]. With these limitations in mind, it has been estimated that for 2013, economic costs related to cyber crimes went up to USD $113 millions [10].

In Chile, this situation is present too: there is awareness at a government-level of the problem, but authorities themselves stated in 2013, that they did not have enough information to provide numbers about increases or decreases of the occurrence of those events in the country.

2 Malware, ISMS and ISO Standards

For crimes happening in the cyber space, malware is a leading component [20], which has transformed these activities into actual business models, developing a market on its own for development, purchases and sales, including whole kits with user-friendly interfaces [11].

For this research, we consider malware as any software that deliberately achieves the goals of an attacker, in order to cause some sort of loss to a target [12].

In order to face this threat, it is crucial to have specialized systems for the protection of the different elements of the computer systems in an organization. A suggested structure for such systems, called Information Security Management Systems (ISMS), is described by ISO Standard 27,001:2013 which, among many other aspects, includes security self-analysis, through penetration tests (pentesting). Those tests can make authorized use of malware, under monitored conditions in other to accurately measure the response capabilities the target system [13].

3 Related Work

As mentioned before, there are different opinions about how to quantitatively estimate worldwide economic impact of malware. In 2012, Anderson et al. conducted one of the first systematic studies about this subject, where they propose a multi-level structure, categorizing the economic impact of malware as follows [8]:

- Direct Losses: The economic equivalent of all losses, damages or other suffering perceived by a victim, as a consequence of a cyber crime.
- Indirect Losses: The losses and opportunity costs imposed to society by the fact that a certain attack is carried out.
- Defense Costs: The economic equivalent of prevention efforts conducted to avoid new losses.

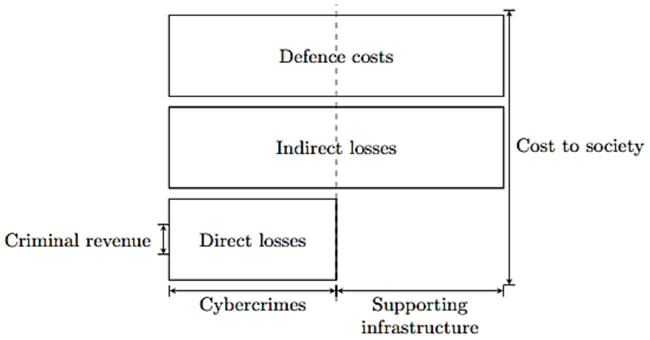

Fig. 1. Costs measure scheme (Source: Adapted from the work of Anderson et al. 2012).

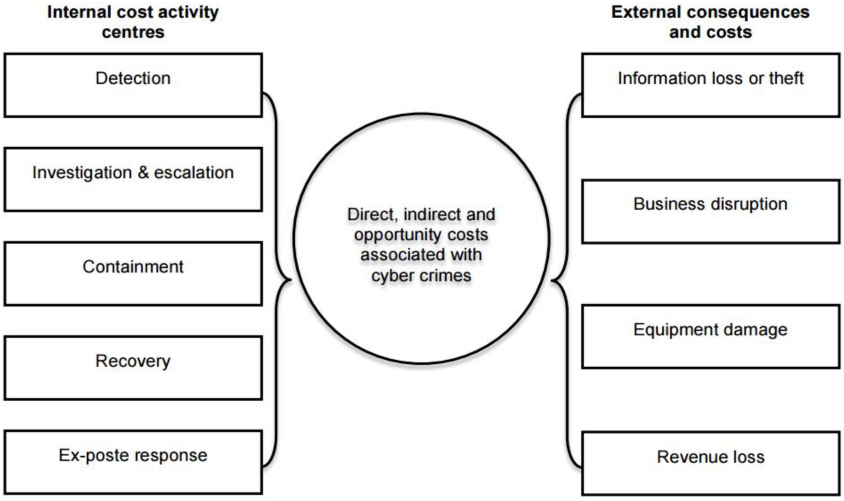
Internal cost activity centres
Detection
Investigation & escalation
Containment
Recovery
Ex-poste response
Direct, indirect and opportunity costs associated with cyber crimes
External consequences and costs
Information loss or theft
Business disruption
Equipment damage
Revenue loss

Fig. 2. Cyber crime costs scheme (Source: Ponemon 2015).

Another classification model, proposed by Ponemon Institute (2015) [14], considers two main and separated costs steams: Internal Costs Activity Centers, and External Consequences and Costs. The internal stream is divided into five points:

- Detection,
- research and escalation,
- Containment,
- Recovery,
- Ex-post Response.

On the other hand, the external stream is divided into four main points:

- Costs due to information loss or theft,

- Costs due to business disruption,
- Costs due to Equipment Damages,
- Revenue losses.

These streams converge in the affected organization, in the form of costs and losses, both direct and indirect.

4 Malware Taxonomy and Strategies to Avoid Detection

It is convenient to understand the characteristics of malware in order to understand the strategies applied on it when evading anti-malware systems. Although it is common to wrongly refer to malware as "viruses", it is important to note that a virus is a type of malware. In general terms, there are four main malware classifications [15]: 1) viruses, 2) worms, 3) botnets y 4) trojan horses.

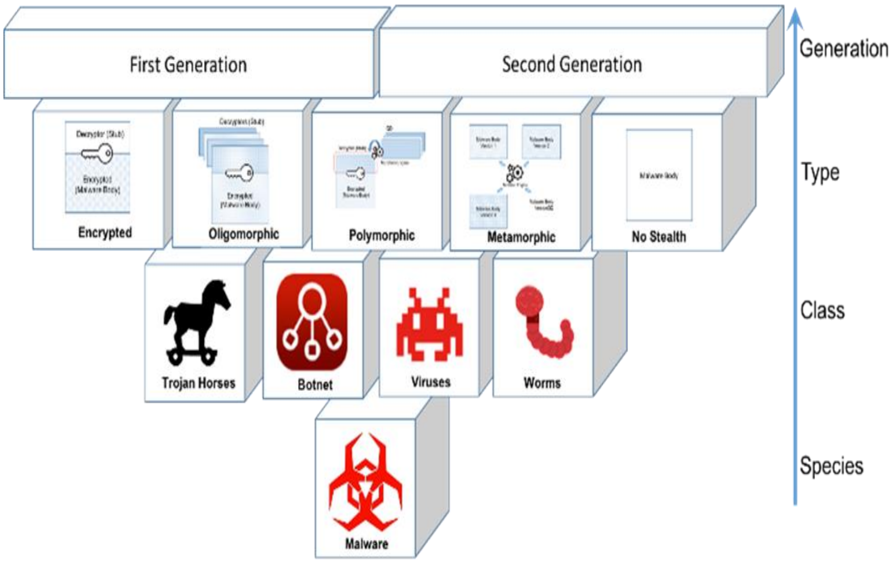

Fig. 3. Inverted pyramid model for malware classification (Source: Barría et al., 2016).

There are other criteria to classify malware, organized by Barría et al. into one big classification structure, in the form of an inverted pyramid model [16]:

The different characteristics of each malware type determine the strategy used to avoid anti-malware systems (commonly known as anti-virus systems). However, in order to understand the reasoning behind each strategy, it is necessary to understand how anti-malware systems works to achieve their goal of successfully detect malware.

Those systems mainly work based on two different approaches: Signature-based Detection, and Heuristic Detection. In the Signature-based Detection, the anti-malware engine looks for known data patterns, inside the executable code of each file. On the other hand, Heuristic Detection involves behavior analysis, structure analysis and other attributes, which is more efficient for mutated malware detection, but also usually more time needed to perform the analysis [17].

For malware to successfully evade signature-based detection, there are different strategies [18], among which we can mention:

- Encryption: It aims to change the malware appearance, consisting of two basic sections: a “stub” (encryption algorithm) and a “body” (encrypted malware code).
- Oligomorphism: Like encryption, this strategy has two basic sections, including a body, but it has a set of different stubs, randomly executed on each iteration.
- Polymorphism: This strategy can be understood as a improved version of Oligomorphism, by adding a third section, called mutation engine, which generates a new stub version on each iteration.
- Metamorphism: Unlike the previous strategies, metamorphism does not use a stub, because it directly uses a mutation engine on the malware body, making it very hard for signature-based anti-malware systems to detect this kind of malicious code.

All these strategies can be complemented with other techniques to make the malware code harder to analyze and, as consequence, harder to identify as a threat. Among the main techniques to achieve this goal, we can find the code obfuscation, which is basically the application of code transformations (over the source code or the binary), whose change the malware appearance through a set of steps, keeping its functionality intact. At the same time, there are different techniques for obfuscating code [18], among which we can mention:

- Dead code insertion: Is the addition of code aiming to modify the binary sequence of a program, without affecting its functionality.
- Code transformation: Is the reordering of the original code sequence, without impacting its behavior.
- Sub-routine reordering: Random changes in the order of the sub-routines of a given code.
- Instruction substitution: A library of equivalent instructions is set, allowing instruction substitution in the code, without affecting its functionality.
- Code integration: It is the generation of new structures of the malware body on each iteration.
- Encryption/Decryption: Systematic encapsulation of the information, through a two-parts structure: a) Encrypted body, and b) Decrypted code.

To implement these techniques, a set of instructions and tasks, called procedures, are used. Among the most common procedures we can find: AvFucker, DSplit, RIT, Hexing y XoR, and others.

For this study, we have chosen the strategy of encryption, using the AvFucker procedure to update a test malware, based on the dead code insertion technique.

5 Current Malware Update Process

The process of taking a program whose signature is already recognized by anti-malware systems as a threat, and transforming it in such a way that is no longer detected, without losing functionality, is described by Barría et al. as an iterative process of malware "update" [19].

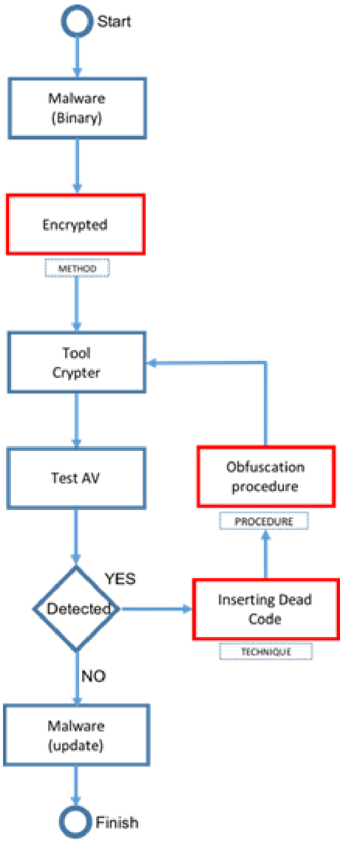

Fig. 4. Malware update scheme (Source: Barría et al. 2016).

This process involves a procedure cycle, where malware is encrypted thanks to a Crypter, to then be analyzed by one or more anti-malware systems. In the case of being detected, the malware is inserted with dead code, supported by some obfuscation process, and then encrypt the resulting code again with a Crypter.

Finally another test is conducted against the anti-malware system and the cycle is repeated until a program which signature is no longer detected, is obtained. A graphical explanation of the previously mentioned process, is provided below:

Nowadays, iteration of this model is executed with the help of different tools, including crypters, hexadecimal editors, anti-malware systems engines, among others. It It is also common to use a tool which allows to generate "n" number of copies of the original file, modifying each copy in a different offset range, replacing those locations with dead code. Although thhe usage of that kind of tools means an important reduction of the time required for this process, compared to the “manual” way to do that process (for example, editing the offset range by using an editor), this part of the process still iss the one that adds the most amount of time to the whole process, thus increasing its final cost.

Having identified this difficulty, the possibility of automating this iteration arises as a natural option when it comes to reduce the costs in terms of money and time.

6 Testing through Prototype Software

A Python version 2 was coded, with the adition of the PyWinAuto library, in order to automate the process in study.

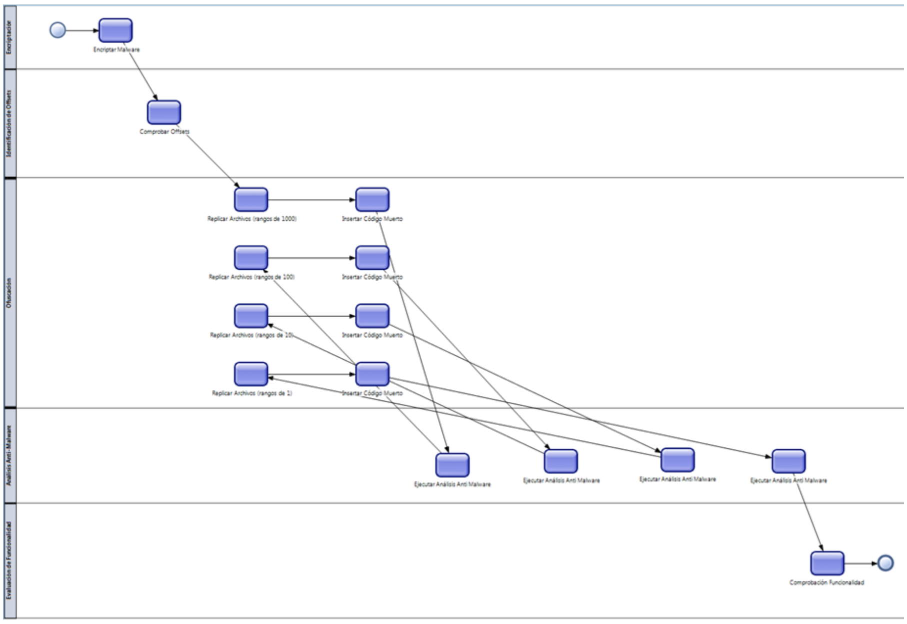
Encriptación
Encriptar Malware
Identificación de Offsets
Comprobar Offsets
Ofuscación
Replicar Archivos (rangos de 1000)
Insertar Código Muerto
Replicar Archivos (rangos de 100)
Insertar Código Muerto
Replicar Archivos (rangos de 10)
Insertar Código Muerto
Replicar Archivos (rangos de 1)
Insertar Código Muerto
Análisis Anti-Malware
Ejecutar Análisis Anti Malware
Ejecutar Análisis Anti Malware
Ejecutar Análisis Anti Malware
Ejecutar Análisis Anti Malware
Evaluación de Funcionalidad
Comprobación Funcionalidad

Fig. 5. Automated process scheme (prepared).

7 Metrics Comparison

Having the prototype ready, we proceeded to compare the time required to finish a whole instance of the process, from beginning to end, both in the case of the automated process and the original one (manual process), and then use a simulation software which allowed us to evaluate the required time for 50 instances.

In order to homologate the experimental conditions for both cases, the same tools and the same base file with the malware signature were used, as described below:

- Base file,
- Crypter tool,
- Hexadecimal edition and file replication tool “Offset Locator”,
- Anti-malware engine “Eset Nod32”.

Both processes were executed in the same computer, under the same operative system (Windows 7).

Finally, to evaluate the economic dimension of each process, it was given to each working hour the value of CLP $1100 each one.

7.1 Experimental Evaluation

The execution of the manual process lasted 420 minutes (7 hours), while the prototype software took 5 minutes to finish the whole instance. In both cases we obtained a set of files whose successfully evaded the anti-malware engine.

Considering the value assigned to each working hour, we obtained the following results presented in Table 1.

Table 1. Comparison chart for time and cost between both processes, experimentally evaluated (prepared).

	Manual Process	Automated Process
Time (hours)	7	0,083
Cost ($ CLP)	7700	1100

7.2 Simulated Evaluation

The execution of 50 whole instances generated the following results presented in Table 2.

Table 2. Comparison chart for time and cost between both processes, evaluated through simulation (prepared).

	Manual Process	Automated Process
Time (hours)	350	4,2
Cost ($ CLP)	385000	5500

8 Conclusions

The observed time reduction in the automated process has an evident impact in the final cost of it. The fact that the tests were performed with ligh files (5KB) allows us to deduce that, if applied to larger files, the benefits of this process will be even more significant.

The possibility of reducing the cost for this kind of processes is, undoubtedly, a great chance to bring this kind of tools closer to smaller organizations, thus helping them to gain access to security evaluations for their computer systems, contributing to create a safer environment for all kind of organizations.

Being possible to approach this kind of process from a modular perspective, the doors for future software versions are left open, potentially including other tools, allowing for future experimental comparisons of performance, efficiency, to name a few. This could lead to potential improvements for each module, in terms of the process itself, or execution times, among other points.

Furthermore, the path is open for approaching other techniques and strategies, expanding the range of tools whose time and money costs can be reduced.

References

1. Bernik, I.: Cybercrime: The Cost of Investments into Protection. Varstvoslovje, 16(2), pp. 105 (2014)
2. Agrafiotis, I., Bada, M., Cornish, P., Creese, S., Goldsmith, M., Ignatuschtschenko, E., Roberts, T., Upton, D.: Cyber Harm: Concepts, Taxonomy and Measurement. Said Business School Research Papers (2016)
3. Mandiant.: M-Trends 2015: A View from the Frontlines (2015)
4. McAffe.: Net losses: Estimating the Global Cost of Cybercrime. Center for Strategic and International Studies (2014)
5. Guruswamy, K.: Measuring the High Costs of Web Malware Protection. Menlo Security. Web Article, available at: http://www.itproportal.com/features/measuring-the-high-costs-of-web-malware-protection (2016)
6. Ponemon Institute: 2014: A Year of Mega Breaches (2015)
7. Euronews Internet of Things: Cyber crime on the rise. Web article, available at: http://www.euronews.com/2016/02/02/internet-of-things-cyber-crime-on-the-rise (2016)
8. Armin, J., Thompson, B., Kijewski, P.: Cybercrime Economic Costs: No Measure, No Solution. Springer International Publishing Switzerland (2016)
9. Trend Micro.: Reporte de Seguridad Cibernética e Infraestructura Crítica de las Américas (2015)
10. Symantec.: Tendencias de Seguridad Cibernética en América Latina y el Caribe (2014)
11. Calleja, A., Tapiador, J., Caballero, J.: A Look into 30 Years of Malware Development from a Software Metrics Perspective. Springer International Publishing Switzerland (2016)
12. Bai, J., Wang, J., Zou, G.: A malware detection scheme based on mining format information. The Scientific World Journal (2014)
13. ISO/IEC JTC 1/SC 27. ISO/IEC 27001 (2013)
14. Ponemon Institute.: Cost of Cyber Crime Study: United Kingdom (2015)
15. CISCO.: What Is the Difference: Viruses, Worms, Trojans, and Bots? Web article, available at: http://www.cisco.com/c/en/us/about/security-center/virus-differences.html (2016)
16. Barría, C., Cordero, D., Cubillos, C., Collazos, C. A.: Proposed Classification of Malware, based on Obfuscation (2016)
17. Vemparala, S., Di Troia, F., Aaron, V., Austin, T., Stamp, M.: Malware Detection Using Dynamic Birthmarks. In: Proceedings of the 2016 ACM on International Workshop on Security And Privacy Analytics, ACM (2016)
18. Barría, Cordero, Cubillos, Collazos.: Obfuscation method for the manual update of Malware, based on Crypter: A Survey

19. Barría, C., Cordero, D., Cubillos, C., Osses, R.: Obfuscation Procedure, based on the Insertion of Dead Code into Crypter (2016)
20. Lee, T., Kwak, J.: Effective and Reliable Malware Group Classification for a Massive Malware Environment. International Journal of Distributed Sensor Networks (2016)
21. Levi, M.: Assessing the trends, scale and nature of economic cybercrimes: overview and Issues. Springer Science+Business Media Dordrecht 2016 (2016)