2011 International Conference on Computer and Software Modeling
IPCSIT vol.14 (2011) © (2011) IACSIT Press, Singapore

Detecting Malicious Codes: A Signature-Based Solution

Razvan Bogdan[1+]

"Politehnica" University of Timisoara

Bd. V. Parvan, nr. 2, 300223, Timisoara, Romania

Abstract. Malicious codes are one of the most destructive pieces of software that can attack a computer or network. They are self-duplicating and self-propagating, so their behavior is repetitive and automated. Different methods of detecting such attackers have been proposed. The paper is presenting a method to detect this software based on the digital signature of the malware. Our aim is to obtain a dedicated detection scheme for different state-of-the-art digital signatures. Such detection scheme should be optimal from performance point of view.

Keywords: attacks, malware, detection scheme, intrusions detection, performance

1. Introduction

Malicious codes (malware) – viruses, worms, Trojan horses, rootkits and so on - are one of the most destructive pieces of software that can attack a computer or network. Malicious software such as Beagle, Code Red, Nimda, NetSky, Witty have infected millions of computers [1]. The damages they have caused are estimated at billions of dollars. Such software is self-duplicating and self-propagating, but also repetitive and automated. Therefore, by proper means, it can be detected and blocked.

Malware detection techniques are organized in two categories, namely anomaly-based and misuse-based [1]. Anomaly-based techniques are capable of detecting novel threats, but are error prone and require a significant amount of resources, such as computer power. The second category aims at generating a specific, known signature, but different studies [1], [5] have demonstrated that this category offers a higher performance in terms of resources and time.

A very important problem in the state-of-the art literature regarding signature-based malware detection [2], [3] is related to the increasing number of signatures that are generated in order to combat potential malwares. Only in 2008, Symantec created over 1.6 million new signatures. Dealing with such a huge amount of data is very demanding in terms of necessary resources, networks' scaling and cache utilization [2]. Other research papers [4], [5] address the problem of signature-based detection in terms of signature accuracy, but also the involved complexity necessary in order to obtain a certain signature. In the same trend, the speed of signature scanning is of peculiar importance because of the amount of time and required resources necessary to scan a message.

This paper aims at constructing a detection scheme for different signatures, from different types of attacks. These attacks can be targeted towards a certain type of files. Our aim is to offer a proper modality in terms of complexity and time that can detect a group of malicious software. Such a Group Detection Scheme (GDS) can be incorporated in an existent overall Intrusion Detection System (IDS) or an entire IDS can be constructed based on the method proposed in this paper.

[+] razvan.bogdan@cs.upt.ro

The rest of this paper is organized as follows: next section presents previous work in terms of GLFSR-based detection schemes. Following next the GDS solution will be detailed, including discussions regarding the time and performance. The last section is devoted to conclusions and future work.

2. Construction of Detection Schemes based on GLFSR

A solution of obtaining detection schemes based on GLFSR is presented in [6]. It should be noted that by applying the method described in [6] a detection scheme for every particular attack will be provided.

Such a solution is based on a Generalized Linear Feedback Shift Register, which is a pattern generator with $n = (\delta \; x \; m)$ outputs. This structure is designed over GF(2^{δ}) and all its elements are part of GF(2^{δ}). The general form of the GLFSR can be represented as a feedback polynomial by the equation (1). Every coefficient of the polynomial equation from (1) defines the feedback connections specific to a GLFSR.

$$\Phi(x) = x^m + \Phi_{m-1}x^{m-1}+...+\Phi_1 x+\Phi_0 \quad (1)$$

The problem that has been stated in [6] was to design a method in order to verify that each message exchanged in the communication process is not malicious in some way. In other words, the goal is to acknowledge if a message is compromised by a certain, single attack. Such an attack can be identified by its signature. The architecture of the solution to this problem is illustrated in Fig. 1. It can be noted how outputs from the GLFSR are being transformed by a dedicated mapping logic into the signatures of the attacks (test vectors). In order to generate the patterns for the mapping logic, the GLFSR is initially loaded with a certain seed, so as to provide an optimum coverage of the faults. In this case, the GLFSR is used as a pseudo pattern generator (PRPG) [7].

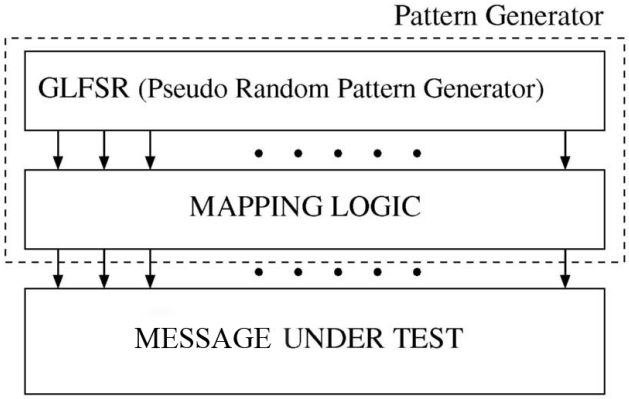

Fig. 1: The process of verifying every message [6].

Following next, the above method can be further refined into obtaining a unique detection scheme for diverse attacks recognized by different signatures. Such a solution can be named Group Detection Scheme.

3. Group Detection Scheme

In order to obtain a single detection scheme for different signatures, we propose to adopt a *column approach* for matching of the PRPG patterns into target patterns [9]. These target patterns are the typically introduced faults of an attack, namely attack's signature.

The steps needed to be employed in order to obtain a unique detection scheme for multiple attacks can be formulated as follows:

1. Design a composite pattern generator by using a GLFSR functioning as a PRPG engine
2. Propose a dictionary of attacks signatures that should be detected by the outcome scheme
3. Establish a minimal set of PRPG patterns such as that they should test the determined fault list (Target Patterns)
4. Obtain the Target Patterns (attacks' signatures) by applying a column matching process
5. Design an efficient combinational circuit with the PRPG so that the combined pattern generator produces all the signatures from the above constructed dictionary. In this way, obtain a Group Detection Scheme as the outcome of the applied steps.

The column matching process will be considered complete when those columns that cannot be matched will be minimized so as all the desired outputs to be obtained. Each output can be acquired from the column values of the input patterns.

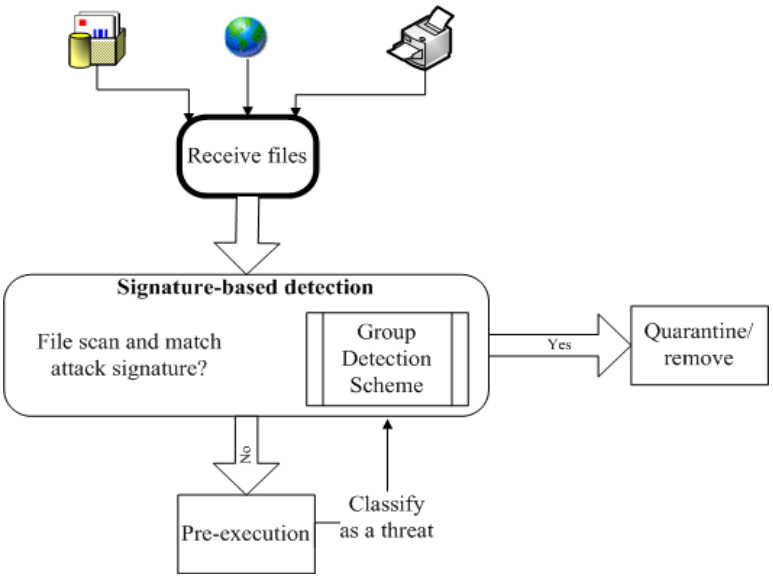

Fig. 2: Integration of GDS in an Intrusion Detection System.

Such a method as proposed above can be used in order to generate a scheme for detection of multiple signatures. Further on, the appliance of GDS can be made as part of an existing IDS or even by constructing an entire IDS based on this method. In Fig. 2 we have shown how the integration of GDS can be made in the example presented in [4], namely a file received from an exterior source (e.g. email, internet, a multimedia center etc.) is scanned by an IDS. Such an IDS has a special feature implemented with GDS. GDS will be able to detect multiple threats, depending on the length of the dictionary. In other words, this technique offers a tailored detection scheme for different attacks identified by unique signatures.

3.1. Practical Results

The next step we approached was the appliance of the proposed method of GDS to some up-to-date attacks signatures. In this regard, we aimed at different faults introduces by malicious software in the executable files. Such signatures are to be found in [3] and [4] and summarized in the dictionary of attacks signatures of Table 1.

The feedback polynomial used in order to generate the output combinations of the GLFSR is $G(x) = x^{16} + x^{14} + x^{13} + x^{11} + 1$, while the initial seed is FE1C. In Fig. 3 it can be noted the column matching process in order to obtain the Target Patterns. Such Target Patterns are corresponding to the attacks' signatures in Table 1. In Fig. 4 is presented the scheme for detecting the signatures from the entire dictionary.

Table 1: Dictionary of attacks signatures.

Attack Type	Signature
Exe-file_1	B802
Exe-file_2	3DBA
Exe-file_3	CD21

The appropriate combinations in the column matching technique can be found by applying the following steps:

1. Divide the PRPG vector in m number of bits
2. Search the group obtained at Step 1 in the desired fault (the attack's signature)
3. If the group is found in the fault (signature), provide it at output
4. Otherwise, complete the rest of the outputs by proper inputs' combinations.

By applying these steps, every output (C_i) has been provided by the correct combination. In Fig. 3 it can be noted that different outputs (*e.g.* C_2 and C_{15}) can be loaded by the same input column.

3.2. Performance Discussion

One of the most important aspects when talking about an IDS is related to the performance of the implied method of detecting and counteracting an attack. The result of not detecting on time a malware will determine an exponential damage and cost [2]. The complexity of the proposed method called Group Detection System can be calculated as being of *O(N.logNNk)*. When applying different search algorithms in order to find a digital signature of a possible attack, the performance complexity would vary from *O(n)* to *O((n-m+1)m)* [8].

Hex Value	Input Vectors																Output Vectors (Target Patterns)															Hex Value
	X_0	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	
B8E9	1	0	1	1	1	0	0	0	1	1	1	0	1	0	0	1	1	0	1	1	1	0	0	0	0	0	0	0	0	1	0	B802
3DF8	0	0	1	1	1	1	0	1	1	1	1	1	1	0	0	0	0	0	1	1	1	1	0	1	1	0	1	1	1	0	1	0	3DBA
F821	1	1	1	1	1	0	0	0	0	0	1	0	0	0	0	1	1	1	0	0	1	1	0	1	0	0	1	0	0	0	0	1	CD21

$C_0 = X_0$
$C_1 = C_{15} = X_1$
$C_2 = C_3 = C_{14} = X_8$
$C_4 = X_4$
$C_5 = C_7 = C_{10} = X_1 + X_5$
$C_6 = C_9 = X_6$
$C_8 = X_5$
$C_{11} = C_{12} = X_{11}$
$C_{13} = X_{13}$

Fig. 3: Column matching technique applied for dictionary of attacks signatures.

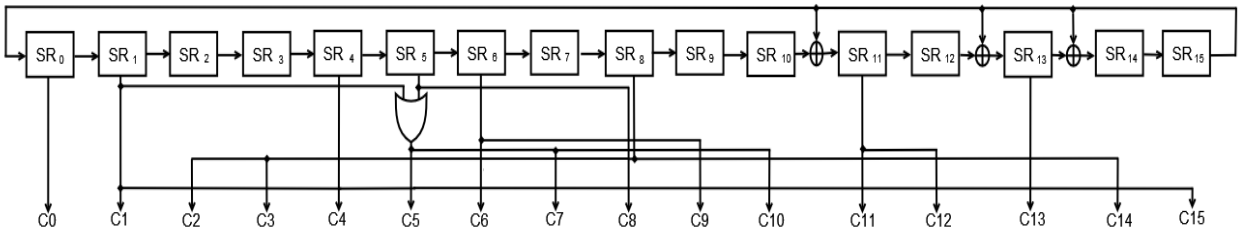

Fig. 4: Unique detection scheme for the entire dictionary of attacks signatures.

It can be noted that the technique presented in this paper can be applied successfully not only from the point of view of the detection rate (which is 100%), but also from performance point of view. The detectability of a fault has been previously defined in [7] *"in terms of the fraction of all possible test vectors that can detect the particular fault"*, in our case a particular attack's signature. Such a customized detection scheme would offer a maximum detectability of attacks that can be traced by their digital signatures. The problem of false detection [1] is automatically solved. In the same time, such a method is posing an excellent performance complexity giving the high rate of attacks' detection.

4. Conclusions

Please acknowledge collaborators or anyone who has helped with the paper at the end of the text. Please acknowledge collaborators or anyone who has helped with the paper at the end of the text.

This paper addresses the problem of detecting multiple attacks identified by their signature. In this regard a method for constructing a Group Detection Scheme has been provided and its appliance revealed a detection scheme for different state-of-the-art attacks. The peculiarity of this method is that it offers a unique detection scheme for several distinct signatures. In terms of performance complexity this technique demonstrated to be similar or superior to the signature searching methods. On the other hand, this method assures a complete detectability because it proposes a tailored detection scheme. The appliance of the GDS method has been illustrated taking into consideration up-to-date attacks and their respective signatures. In other words, this paper also presents a modality of how to detect state-of-the-art attacks by a single detection scheme.

As future work, we propose to research the choosing of a proper feedback polynomial in terms of execution time and complexity. In this way it can be firmly concluded which is the most appropriate feedback polynomial that can be used in order to generate the detection scheme.

5. References

[1] K. Simkhada, T. Taleb, Y. Waizumi, A. Jamlipour and Y. Nemoto. Combating against Internet worms in large-scale networks: an autonomic signature-based solution. *Security and Communication Networks*. edited by Hsiao-Hwa Chen, 2009, Vol. 2, p. 11-28.

[2] S. K. Cha, I. Moraru, J. Jang, J. Truelove, D. Brumley and D. G. Andersen. SplitScreen: Enabling Efficient, Distributed Malware Detection. *Journal of Communications and Networks*. 2011, 13(2): 187-200.

[3] Symantec global internet security threat report. [Online]. Available: http://www.symantec.com/about/news/release/article.jsp?prid=2009041301

[4] O. Sukwong, H. Kim and J. Hoe. Commercial Antivirus Software Effectiveness: An Empirical Study. *Computer*. 2011, 44(3): 63-70.

[5] P. Li, M. Salaour and X. Su. A survey of Internet worm detection and containment. *IEEE Communications Surveys & Tutorials*. 2008.

[6] R. Bogdan and M. Vladutiu. Intrusions Detection in Intelligent Agent-Based Non-traditional Grids. *Proceedings of the International Conference on Education Technology and Computer (ICETC09)*. Singapore, 2009: 116-122.

[7] M. Chatterjee and D. K. Pradhan. A BIST Pattern Generator Design for Near-Perfect Fault Coverage. *IEEE Transactions on Computers*. 2003, 52(12): 1543 - 1558.

[8] Z. Alqadi, M. Aqel and I. El Emary. Multiple Skip Multiple Pattern Matching Algorithm (MSMPMA). *IAENG International Journal of Computer Science*. 2007, 34(2).

[9] R. Bogdan. *A Data Security Perspective on Information Transmission Over Distributed Systems*. Publication place: Politehnica Publishing, 2009.