Behavioral Detection of Malwares: From a Survey Towards an Established Taxonomy -Extended Abstract-

Grégoire Jacob[1/2], Hervé Debar[1], Eric Filiol[2]

[1] France Télécom R&D, Caen, France
{gregoire.jacob|herve.debar}@orange-ftgroup.com

[2] French Army Signal Academy,
Virology and Cryptology Lab., Rennes, France
eric.filiol@esat.terre.defense.gouv.fr

March 9, 2007

1 Introduction to the behavioral principles

This paper was originally motivated by a simple observation. There is no global survey covering the domain of behavioral detection whereas we observe an increasing activity both in commercial products and research. The multitude of behavior-based detection systems is striking, and so is the inconsistency in the vocabulary and the designations used. Basically, behavioral detection differs from the appearance or form-based detection in that it relies on an identification of the actions performed by the malware rather than syntactic markers. Identifying these malicious actions and interpreting their final purpose is a complex reasoning process. This idea is not really new and was already evoked within the first formal works of F. Cohen [1]. He puts forward the fact that viruses, just as any other running program, use the services provided by the system. The prediction of the viral nature of a program according to its behavior is then equivalent to defining what is and what is not a legitimate use of the system facilities.

This definition gives two opposite approaches to address the problem. The first one, mostly used by intrusion detection systems, is to provide a model for legitimate usage [2][3]. Unfortunately, modelling a generic behavior for every kind of program proves to be unmanageable in our present case. It explains why, in virology, the opposite approach of modelling and detecting suspicious behaviors is mainly used. We have willingly adopted the virology point of view and will thus implicitly consider the modelling of suspicious behaviors all along our speech. We have organised our paper as follows: Section 2 explains the recent interest in behavioral detection by the predicted failure of appearance detection, Section 3 describes a generic behavior-based detection system, Section 4 introduces the taxonomy and describes the different classes of detectors, and Section 5 illustrates our speech with an overview of both existing commercial products

and research prototypes. In the complete paper, detailed descriptions of the different classes of systems will be given as well as more numerous references which have been omitted in this abstract.

2 Why behavioral detection may success where appearance detection will undeniably fail

Historically, appearance detection, also called form-based detection, has been the first technology used to fight against malwares and still remain at the heart of nowadays antiviruses. Their functioning principle is the search in files for suspicious byte patterns stored in a base of signatures. As a consequence, these form-based techniques are bound to detect known malwares contrary to the behavioral detection. Unfortunately, the signature extraction process is often manual and thus time consuming. Once extracted, the signature must still be distributed. This proves to be a major drawback since malware propagation speed and release frequency are increasing at an alarming pace. The analysts are completely overwhelmed by this phenomenon. It is partly explained by the fact that the production of new version from an original strain is made easier by weak signature schemes [4]. The problem is even exacerbated by the existence of mutation engine such as polymorphism and more recently metamorphism. The detection of these mutating viruses has been proven NP-complete, leaving few hopes to eradicate them completely [5].

Behavior signatures are no longer simple byte patterns but carry a semantic interpretation. As a consequence, they prove to be more generic and thus resilient to simple modifications. Moreover, a single signature extraction should be sufficient for several strains using the same viral techniques. Since the release of innovative techniques is more scarce, the behavioral approach should increase the time allotted to the analysis and decrease the distribution frequency.

3 Generic description of a behavioral detectors

A behavioral detection system identifies the different actions of a program using the system resources. Based on its knowledge of malwares, it must be able to decide whether these actions betray a malicious activity or not. Information on system use is mainly available in the host environment thus explaining that behavioral detectors work at this level. Whatever the considered detector, its architecture can be split into four main components as shown in figure 1. We have chosen to consider indifferently dynamic capture and static extraction for data collection as in both modes, the intended actions of a program can be observed. Behavioral detectors working at a higher level of description than simple appearance, collected data need to be analyzed and interpreted before to be fed in the matching algorithm.

It is also fundamental to define the important properties of a behavioral detection system since they will provide the basis for efficiency assessment. Actual certifications simply confront malware detectors to known viral strains thereby assessing solely appearance detection. A recent paper has introduced a first basis for a test method specifically dedicated to behavioral detection using functional

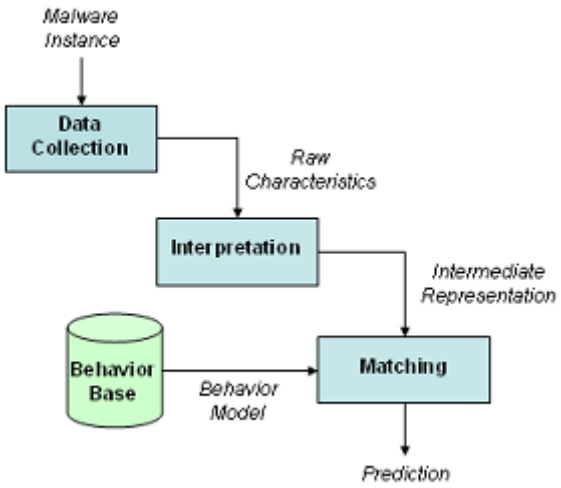

Figure 1: Generic description of a behavior-based detector.

metamorphism [6]. In order to introduce new test directions, we have defined several properties that a behavior-based detector should exhibit:

- Performance (resource use),
- Completeness (false negative),
- Accuracy (false positive),
- Adaptability (new behavior),
- Resilience (anti-analysis techniques),
- Fault-tolerance, Unobtrusiveness and Timeliness specifically for dynamic detectors.

4 Taxonomy of behavioral detector

The concepts we use to classify behavior-based systems derive directly from their generic description. The four main axes described in figure 2 correspond to the four components forming the detector: data collection, capture interpretation, matching algorithms and behavior models. As a matter of fact, every combination of components is not possible. Different models and algorithms are used whether the input data is collected dynamically or extracted statically.

We first describe the capture conditions and the nature of the collected data. In the case of dynamic monitoring, we have distinguished four conditions to collect system call traces: real-time, sandboxes, virtual machines [7] and real-time with action recording [8]. For each condition, we have stated its main advantages, measured its impact on the performance and described possible attacks. In the case of static extraction, we have described the process which leads to the extraction of control flow graphs used as intermediate representation. By observing the global assets and limitations of both modes, we conclude that

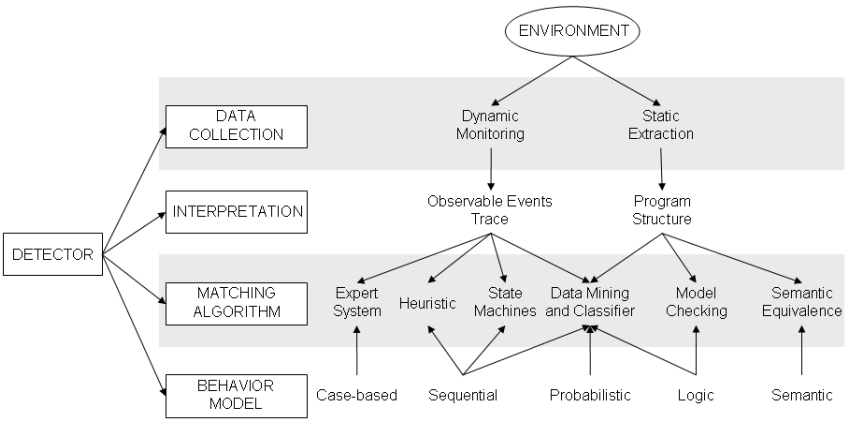

Figure 2: Characteristics of behavioral detection systems.

they are strongly complementary.

The matching algorithms and behaviors models will then be described simultaneously. This is explained by the fact that the engine nature will determine the behavior modelling, the intermediate representation as well as the confrontation method. We have identified the following engines already mentioned in the figure 2:

- Expert systems based on case-based rules [9]. Every separated action taken by the observed program will be confronted to the related rules. The target and the privilege level of the caller are additional factors to take into account because they often draw the distinction between a legitimate action and a malicious one [10].

- Heuristic engines based on sequential models [11][12]. Basically, heuristic engines are made up of three parts: an association mechanism labelling the collected data, a database of rules containing the behavior descriptions and a strategy defining the rule exploration.

- State machines based on sequential automata [13]. From an initial state, the collected data are evaluated step-by-step making the automaton progress. If during its progression, the automaton reaches an accepting state, a malicious behavior has been discovered.

- Data mining and classifiers based on three paradigms: rule induction, Bayesian statistics and clustering [14]. Properly speaking, classifiers combined with data mining techniques are not simple detection engines. They should rather be qualified of automatic learning mechanisms building classification rules.

- Semantic verifiers based on semantic templates [15]. The detection consist in checking that the semantic description of a program satisfies the given behavior template. A program and a template are equivalent if their execution have an identical impact on memory.

- Model checkers based on temporal logic formulae [16]. The verification algorithm explores enumeratively the possible execution paths in order to find the intermediate states satisfying the given formula.

5 Panorama of existing behavioral detectors

As an illustration, we have classified several existing behavior-based systems of detection according to the elements of our taxonomy. The classification has been applied to several engines coming both from the research domain and the commercial products. As a result, we have been able to bring into light the current trends in behavioral detection. In particular, a convergence of the antiviruses using behavioral detection with host-based intrusion prevention systems (HIPS) can be observed. This is not really surprising since virology and intrusion detection are strongly connected security domains.

6 Conclusion

The main idea to retain of this paper is that under the terms of behavioral detection lies a whole set of heterogeneous techniques relying on a common principle of functionality identification. In particular, we observe in the taxonomy a clear distinction between the static and dynamic modes. Yet these modes are complementary as they exhibit opposite strengths and weaknesses.

Several researchers have already thought of means to combine the static and dynamic modes in order to take advantage of their respective assets. Dynamic analysis makes it possible to determine a reduced perimeter where a static analysis would be worth deploying. Based on this principle, a system has already been put forward in order to detect spywares parasiting web browsers. The dynamic phase is used to find the processing routines associated to the different web events. Once localized a static analysis is deployed to detect any malicious activity [17]. Generally speaking, a static analysis could be deployed at each reached branching to explore the alternative execution paths that will not be executed.

If we want to combine efficiently both modes it remain necessary to evolve towards a common model of reference. This model could then be slightly adapted according to the class of system considered, while remaining compatible with others. Unfortunately, such a model is still missing.

References

[1] F. Cohen, *Computer Viruses*. PhD thesis, University of South California, 1986.

[2] H. Debar, M. Dacier, and A. Wespi, "Towards a taxonomy of intrusion-detection systems," *Computers Networks, Special Issue on Computer Network Security*, vol. 31, no. 9, pp. 805–822, 1999.

[3] L. Mé and B. Morin, "Intrusion detection and virology: an analysis of differences, similarities and complementariness," *Journal in Computer Virology*, vol. 3, no. 1, Special Issue WTCV'06, Coming in 2007.

[4] E. Filiol, "Malware pattern scanning schemes secure against black-box analysis," *Journal in Computer Virology*, vol. 2, no. 1, pp. 35–50, 2006.

[5] D. Spinellis, "Reliable identification of boundedlength viruses is np-complete," *IEEE Transactions on Information Theory*, vol. 49, pp. 280–284, 2003.

[6] E. Filiol, G. Jacob, and M. L. Liard, "Evaluation methodology and theoretical model for antiviral behavioural detection strategies," *Journal in Computer Virology*, vol. 3, no. 1, Special Issue WTCV'06, Coming in 2007.

[7] U. Bayer, C. Kruegel, and E. Kirda, "Ttanalyze: A tool for analyzing malware," in *Proceedings of EICAR*, 2006.

[8] M. E. Wagner, "Behavior oriented detection of malicious code at run-time," Master's thesis, Florida Institute of Technology, 2004.

[9] M. Debbabi, "Dynamic monitoring of malicious activity in software systems," in *Proceedings of the Symposium on Requirements Engineering for Information Security (SREIS)*, 2001.

[10] C. Kruegel, D. Mutz, F. Valeur, and G. Vigna, "On the detection of anomalous system call arguments," in *Proceedings of the European Symposium on Research in Computer Security*, pp. 326–343, 2003.

[11] M. Schmall, *Classification and Identification of Malicious Code Based on Heuristic Techniques Utilizing Meta-languages*. PhD thesis, University of Hamburg, 2002.

[12] F. Veldman, "Heuristic anti-virus technology," in *Proceedings of the International Virus Protection and Information Security Council*, 1994.

[13] B. L. Charlier, A. Mounji, and M. Swimmer, "Dynamic detection and classification of computer viruses using general behaviour patterns," in *Proceedings of the Virus Bulletin Conference*, 1995.

[14] M. G. Schultz, E. Eskin, and E. Zadok, "Data mining methods for detection of new malicious executables," in *Proceedings of IEEE Symposium on Security and Privacy*, pp. 38–49, 2001.

[15] M. Christodorescu, S. Jha, S. A. Seshia, D. Song, and R. E. Bryant, "Semantic-aware malware detection," in *Proceedings of IEEE Symposium on Security and Privacy*, pp. 32–46, 2005.

[16] J. Kinder, S. Katzenbeisser, C. Schallhart, and H. Veith, "Detecting malicious code by model checking," *Lecture Notes in Computer Science*, vol. 3548, pp. 174–187, 2005.

[17] E. Kirda, C. Kruegel, G. Banks, G. Vigna, and R. Kemmerer, "Behavior-based spyware detection," in *Proceedings of the 15th USENIX Security Symposium*, 2006.