Behavioral Intrusion Detection

Stefano Zanero[⋆]

Dipartimento di Elettronica e Informazione, Politecnico di Milano,
Via Ponzio 34/5, 20133 Milano, Italy,
stefano.zanero@polimi.it

Abstract. In this paper we describe anomaly-based intrusion detection as a specialized case of the more general behavior detection problem. We draw concepts from the field of ethology to help us describe and characterize behavior and interactions. We briefly introduce a general framework for behavior detection and an algorithm for building a Markov-based model of behavior. We then apply the framework creating a proof-of-concept intrusion detection system (IDS) that can detect normal and intrusive behavior.

1 Introduction

The landscape of the threats to the security of computer systems is continuously evolving. Attacks and viruses are constantly on the rise. For this reason, it is important to design better systems for detecting infections and intrusions, making the reaction to security incidents quicker and more efficient.

In particular, we are realizing the limits of the misuse-based approach to intrusion detection. A misuse detection system tries to define what an attack is, in order to detect it. While this kind of approach has been widely successful and is implemented in almost all the modern antivirus and intrusion detection tools, its main drawback is that it is unable to properly detect previously unknown attacks (i.e., it is *reactive* and not *proactive*).

Antivirus vendors have responded with state of the art research facilities, round-the-clock response teams, and fast signature distribution methodologies. However, the diffusion of flash malware [1] is a hard to meet challenge. In the intrusion detection field maintaining a knowledge base of attack is impossible, both for the high number of new vulnerabilities that are discovered every day and for the even higher number of unexposed vulnerabilities that may not be immediately available to the experts for analysis and inclusion in the knowledge base (which, in general, does not happen for viral code).

Additionally, since there usually exist a number of ways to exploit the same vulnerability (polymorphism), it is difficult to develop compact signatures that detect all the variations of the attack and at the same time do not incur in false positives. Finally, many intrusions are performed by insiders who are abusing their privileges. In this case, since no attack against known vulnerabilities is performed, a misuse-based IDS is useless.

[⋆] Work partially supported by the FIRB-Perf italian project.

An obvious solution to all these problems would be to implement an anomaly detection approach, modeling what is *normal* instead of what is *anomalous*, going back to the earliest conceptions of what an IDS should do [2].

Anomaly detection systems have their own problems and show an alarming tendency to generate huge volumes of false positives. In addition, it has always been a difficult task for researchers to understand what to monitor in a computer system, and how to describe and model it. Even if not really successful in commercial systems anomaly detection has been implemented in a number of academic projects with various degrees of success.

In this paper, we will try to explore a behavioral approach to anomaly based intrusion detection. We will leverage an ongoing trend in knowledge engineering, which is called *behavior engineering* [3]. We draw concepts from the field of ethology to help us describe and characterize behavior and interactions. We briefly introduce a general framework for behavior detection and an algorithm for building a Markov-based model of multiple classes of behavior. We then apply the framework creating a proof-of-concept system that can detect normal and intrusive behavior.

The remainder of the paper is organized as follows. In Section 2 we introduce the problem of behavior detection, and we examine insights coming from ethology and behavioral sciences. In Section 3 we introduce a general framework for behavior detection problems, and we describe an algorithm for building a model of behavior based on Markov chains. In Section 4 we apply the model to the problem of intrusion detection and give proof-of-concept results. Finally, in Section 5 we will draw our conclusions and plan for future work.

2 The Problem of Behavior Detection

2.1 Introduction to Behavior Detection problems

We propose to consider anomaly based intrusion detection in the more general frame of *behavior detection* problems. This type of problems has been approached in many different fields: psychology, ethology, sociology. Most of the techniques applied in these areas are of no immediate use to us, since they are not prone to be translated into algorithms. However, some useful hints can be drawn forth, in particular by analyzing the quantitative methods of ethology and behavioral sciences [4].

In order to understand the problem and to transfer knowledge between these different fields, we must analyze parallel definitions of concepts we will be dealing with. The first term is "behavior", which ethology describes as the stable, coordinated and observable set of reactions an animal shows to some kinds of stimulations, either inner stimulations (or motivations) or outer stimulations (or stimuli). The distinction between "stimuli" and "motivations" is as old as ethology itself, being already present in Lorenz's work [5].

Our definition of "user behavior" is quite different. We could define it as the "coordinated, observable set of actions a user takes on a computer system in order to accomplish some task". Depending on the observation point we assume, we can give different definition of actions, but for the scope of this paper we will define them as the commands, the data communications and the inputs that the user exchanges with the system. We wish to make clear that our effort is not focused on the behavior of the computer system (which is by definition entirely predictable) but on the behavior of the user, which has relevant intentional components.

We will also make use of the concept of "typical behavior", which quantitative ethology would describe as the "most likely" one. In our definition, this behavior is the "normal" user behavior, as opposed to an "atypical" behavior which is not, however, always devious or dangerous.

2.2 Motivations for action and action selection

This consideration brings us to the point of analyzing the motivations of behavior. We are interested in detecting any anomalous behavior which is motivated by the desire to break the security policy of the system. Anomalous behaviour with no devious motivation is not a problem by itself; on the other hand perfectly normal, incospicuous network traffic, motivated by a devious goal, should in some way be detected by a perfect intrusion detection system.

Even if terminology varies from school to school in behavioral sciences, we can recognize three broad levels of increasing complexity in the analysis of behavior: reflex behavior (sensorial stimuli and innate reactions), instinctual behavior (genetically evolved, innate behavior of a species), and finally intentional behavior, with actions that an animal begins autonomously to reach its own goals.

Clearly, when dealing with computer misuse, we are mostly dealing with intentional behavior, and we need to define what *motivates* an action. The concept of motivation is crucial to ethology, and it has been a theme of a number of philosophical researches as well. Without getting deeply into the philosophical debate, we can define motivations as the dynamic factors of behaviors, which trigger actions from an organism and direct it towards a goal. We will try to recognize which motivations are behind a particular behavior of a user.

2.3 Fixed action patterns, modal action patterns, and ethograms

Closely associated with these concepts are *patterns*, elements shared by many slightly different behaviors, which are used to classify them. The concept of "behavioral pattern" is widely used in ethology.

Ethologists typically define as *Fixed Action Patterns* (FAP) the atomic units of instinctual behavior. FAPs have some well defined characteristics: they are mechanic; they are self-similar (stereotyped) in the same individual and across a species, and they are extensively present; they usually accomplish some objective. More importantly, they are atomic: once they begin, they are usually completed by the animal, and if the animal is interrupted, they are aborted.

A FAP must also be independent from (not correlated with) other behaviors or situations, except at most one, called a "releasor", which activates the FAP through a filter-trigger mechanism, called Innate Release Mechanism (IRM). The IRM can be purely interior, with no external observable input (emitted behavior), or it can be external (elicited behavior). In the latter case, sometimes the strength of the stimulus results in a stronger or weaker performance of the FAP (response to supernormal stimulus). In other cases, there is no such relation.

In [6], the whole concept of FAPs and IRMs is examined in detail. The author criticizes the rigid set of criteria defining a FAP, in particular the fact that the IRM must be different for each FAP; the fact that the IRM has no further effect on the FAP once it has been activated; and the fact that components of the FAP must fall into a strict order. Many behaviors do not fall into such criteria. Barlow proposes then to introduce MAPs, or *Modal Action Patterns*, action patterns with both fixed and variable parts, which can occur in a different order and can be modulated during their execution. Barlow suggests that the environment can modulate even the most stereotyped behavior. His definition of MAP is a "spatio temporal pattern of coordinated movement that clusters around some mode making it recognizable as a distinct behavior pattern". Unfortunately, the flexibility of a MAP is difficult to implement in a computer-based model of behavior.

A subset of FAPs, called "displays", are actually communication mechanisms. In an interesting chain of relations, a display can be the releasor of an answer, creating a communication sequence. An interesting characteristic of displays is the principle of *antithesis*, stating that two displays with opposite meanings tend to be as different as they can be. This is not necessarily true in behavior detection problems: for example, malicious computer users will try to hide behind a series of innocent-like activities.

We must also introduce the concept of an *ethogram*, which is an attempt to enumerate and describe correctly and completely the possible behavioral patterns of a species. On the field, an ethologist would observe the behavior of animals and list the different observed behavioral patterns in a list, annotated with possible interpretations of their meaning. Afterwards, s/he would observe at fixed interval the animals and "tick" the appropriate squares in an ethogram, generating a sequence data on the behavior of the observed animals. A similar discretization will be used also in our framework.

3 A Framework for Behavioral Detection

3.1 A methodology for behavioral detection

We will try to exploit the similarities we have found, in order to propose a framework for studying behavior detection and classification problems.

First of all, we need to specify which kind of displays of behavior we can detect and build appropriate sensors for detecting them. It is not difficult to collect and analyze the logs of a workstation, but detecting the behaviors of users in a virtual classroom environment could be difficult. For our example architecture we choose to use the interactions with a terminal. Other likely displays that could be analyzed are the logs of the interactions between a user and a web application, the sequence of system calls generated by user processes [7], or the generation of audit data (using for instance the syslog facilities of UNIX and similar systems).We refer the reader to one of our previous works [8] for considerations on network based anomaly detection systems. In this paper we will focus instead on host based anomaly detection.

As a second step, we must choose an appropriate model for representing the behavior. We could approach the problem at different levels of abstraction, making hypotheses on the action selection problem (as seen in 2.2) and analyzing the actual process which generates the behavior. However, we will use a traditional approach in quantitative behavior study, trying to model just the sequence of the displays of behavior, in order to infer various properties about the subject. In order to choose an appropriate model, we must understand if we want a binary classification, or a more complex one with several disjunct classes, or even one with overlapping categories.

Upon this model we must build an inference metamodel, which can help us learn actual parameters from observed data in order to tune the model. This is a classical instance of machine learning problem. Finally, we must set thresholds and logics that help us extract useful information from the observed behavior. Due to space constraints, we will now focus our discussion on how to build an appropriate model for representing the behavior. As a future work we will deal with the other steps required for building a complete behavior detection system.

3.2 Representing behavior: Markov Models

Markov models are widely used in quantitative behavioral sciences to classify and report observed behaviors. In particular, in ethology simple Markov Models are built on field observation results. A time domain process demonstrates a Markov property if the conditional probability density of the current event, given all present and past events, depends only on the K most recent events. K is known as the *order* of the underlying model. Usually, models of order $K = 1$ are considered, because they are simpler to analyze mathematically. Higher-order models can usually be approximated with first order models, but approaches for using high-order Markov models in an efficient manner have also been proposed, even in the intrusion detection field [9].

A first order Markov Model is a finite set of N states $S = \{s_1, s_2, \dots s_n\}$, each of which is associated with a (generally multidimensional) probability distribution. Transitions among the states are governed by a set of probabilities called transition probabilities $a_{i,j} = P\{t = k + 1,\, s_j \,|\, t = k,\, s_i\}$ (whereas in order K models the probability depends on the states in the K previous steps, generating a $K + 1$-dimensional array of probabilities). We consider a time-homogeneous model, in which $A = a_{i,j}$ is time-independent.

In a Hidden Markov Model, in any particular state, an outcome or observation o_k can be generated according to a probability distribution associated to the state ($b_{j,k} = P\{o_k \,|\, s_j\}$), in an alphabet of M possible observations. These probabilities obviously form a matrix $B = b_{j,k}$ which we also suppose to be time independent. Only the outcome, and not the state, is visible to an external observer; therefore states are “hidden” from the outside. The definition also implies an assumption which is probably not true: the output is assumed to be statistically independent from the previous outputs. If the observations are continuous, then a continuous probability density function is used, approximated by a mixture of Gaussians. However, ethologists discretize animal behavior using FAPs and MAPs and ethograms, in order to simplify the model. In our case, user-computer interactions are mostly discrete sequences of events. Obviously, non hidden Markov models are special, simple cases.

In order to use HMMs in behavior detection, we need to solve two common problems associated with HMMs [10]. The first is the *evaluation problem*, which means, given a sequence of observations and a model, what is the probability that the observed sequence was generated by the model. The second is the *learning problem*: building from data a model, or a set of models, that properly describe the observed behavior. A third problem, the so called *decoding problem*, is not of particular interest to us.

3.3 An algorithm for building Markovian models of behavior

The *evaluation problem* is trivial to solve in the case of a normal model, more complex to solve in the case of an HMM: in this case, the naive approach yield a complexity of N^T, where T is the length of the sequence of observations. The so-called forward algorithm [11] can be used, which has a complexity of N^2T.

The *learning problem* is more complex, in particular if we do not know the structure of the model. First of all, we need to choose the order of the model we will use. Often a first-order approximation is used for simplicity, but more complex models can be considered. A good estimate for an HMM can be extracted from data using the criteria defined in [12]; for normal Markov models, a χ^2-test for first against second order dependency can be used [13], but also an information criterion such as BIC or MDL can be used.

In order to estimate the correct number of states for an HMM, in [14] an interesting approach is proposed, by eliminating the time dependency and constructing a classification by means of clustering of the observations, considering each state as a generation mechanism.

Once we have chosen the model structure, learning a sequence of T observations means to find the matrices $\{A, B\}$ that maximize the probability of the sequence: $maxP[o_1o_2 \ldots o_T|A, B]$. This is computationally unfeasible, however the Baum-Welch algorithm [15] can give a local maximum for that function. Another approach to the parameter estimation problem is proposed in [16]. If the model is not hidden, however, the calculations become simple.

In many earlier proposals for the use of Markovian models in intrusion detection [17] the authors either build a Markov model for each user and then try to find out masquerading users (users accessing illicitly the account of another user); or they build a Markov model for the generic user and flag as anomalous any user who behaves differently. The first approach brings an explosion of models, lacking generalization or support for users who are not identified uniquely to the system, while the second approach ignores the existence of different classes of users on the system.

In order to account for the existence of different *classes* of user behaviors, we propose the following algorithm, based on a Bayesian approach. Denoting with M a generic model and with O a sequence of observations, $P(M|O) \propto P(O|M)P(M)$. This means that, if we have a set of I models $M_1, M_2 \dots M_I$, the most likely model for the sequence of observations O is given by: $max_i\, P(M_i|O) = max_i\, P(O|M_i)\, P(M_i)$

We need now to choose an appropriate prior $P(M_i)$ for the models. Let us suppose that this procedure is iterative, which means that we have built the existing I models out of K observation sequences $O_1 \dots O_K$, iteratively associating each sequence with the best-fitting model and retraining the model with the new observations. This also means that we need to define a criterion for choosing whether it is appropriate to associate the new observations O_k with an existing model, or to create a new model for representing them.

A common decomposition for studying the prior of the model would be $P(M_i) = P(\theta_i|M_s)P(M_s)$, denoting with $P(\theta_i)$ the probability of the particular parameter set of M_i given a basic structure M_s and with $P(M_s)$ the probability of the structure itself. However, this type of approach leads to very complex calculations.

Using a simpler approach, we could proceed as follows. Let us call O_i the union of the observation sequences that have generated model M_i. We can build a non-informative prior criterion such as:

$$P(M_i) = \left(\frac{|O_i| + |O_k|}{(\sum |O_i|) + |O_k|} \right)^{log(|O_k|)}$$

which penalizes more particular models, favoring more general ones. Inserting the exponent $log(|O_k|)$ is necessary in order to account for the fact that different length of observation strings will generate different orders of magnitude in posterior probability. This generates also a simple criterion for the creation of new models. In fact, denoting with M_{I+1} a new model built on the new observations O_k, we would choose: $max_i P(M_i|O_k) = max_i P(O|M_i)P(M_i)$ with $1 \leq i \leq I+1$, defining:

$$P(M_{I+1}) = \frac{|O_k|}{(\sum |O_i|) + |O_k|}$$

In this way, the prior biases the probability towards more general models instead of more fitting but less general ones, averaging out the fact that less general models tend to have an highet posterior probability $P(M_i|O_k)$. Once we have selected which model the k-th sequence O_k will be associated with, we re-train the model including in training data the new sequence.

Afterwards, we may optionally include a *merging* step, which means we will try to find couples of models M_i, M_j such that, denoting with $M_{i,j}$ the "merged" model and with O_i and O_j the observations associated with M_i and M_J:

$$P(O_i \cup O_j|M_{i,j})P(M_{i,j}) > P(O_i|M_i)P(M_i)$$
$$P(O_i \cup O_j|M_{i,j})P(M_{i,j}) > P(O_j|M_j)P(M_j)$$

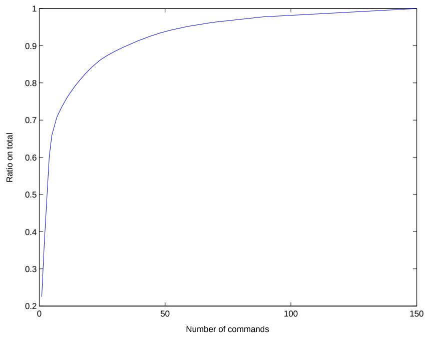

Fig. 1. Cumulative distribution of commands

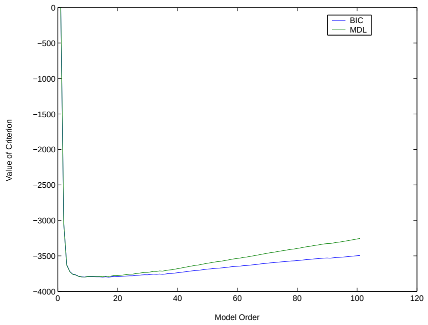

Fig. 2. Information criteria: MDL and BIC

In this case, a suitable criterion for selecting models to merge and for merging them must be also researched. There are some examples in literature of criteria for measuring a distance between two Markov models, for instance in [18] the following (asymmetric) distance is proposed: $D(M_i, M_j) = 1/T[logP(O^{(i)}|M_i) - logP(O^{(i)}|M_j)]$, where $O^{(i)}$ is a sequence of observations generated by model M_i. Criteria for merging HMM models can be found in [19] [20], where they are proposed as a suitable way to induce the models by aggregation.

If we wish to incorporate the insights from section 2.3 on the presence of FAPs and MAPs in behavior, we will need to use higher order models, because we need to express the probability on the base of a history. A suggestion that we may borrow from Barlow's studies on modal components of behavior, however, is that we may also want to detect clusters of states in the Markov chain that exhibit the following properties: they have "similar" outgoing transition probabilities and "similar" symbol emission probabilities (if we are dealing with an HMM). These states can be collapsed together in a single state, with simple probability calculations that we omit. This method is also applied in quantitative behavioral science, see [21].

4 Case Study: Behavioral Intrusion Detection

We acquired test data from a limited number of users of two different terminal systems, with about 10 users for each system and some months of data. We prepared the data by discarding command options and encoding each different command with a number. In one of these systems, for example, on 2717 interactions, 150 unique commands were used. However, as we can see in Figure 1, a significative fraction of the interactions consists of a limited subset of frequently used commands, so we can set a minimum threshold below which we will group all the commands together as "other".

In order to estimate the optimal order for the model, we used both the BIC and MDL criteria, and both agree on order $k = 4$ as being the optimal

Commands	Our algorithm		Naive Markov
	Fitting	Detection	Detection
60	90.0	95.9	90.0
40	89.2	95.6	87.8
30	87.8	94.8	86.3
20	84.8	92.9	78.9
10	67.4	81.1	65.6
8	61.1	78.5	59.3
6	38.1	63.3	51.5
4	20.4	55.9	50.7

Table 1. Performance of our algorithm vs. naive application of Markov Models

value. However, as a first approximation we will use a first-order model to fit the observations (approximation supported by the steep descent in criteria curves, which can be observed in Figure 2). Also, since the observations are finite in number, we use a normal Markov chain and not an HMM, to fit it.

We trained a set of Markov models following the basic algorithm outlined above. We experimented with various combinations of thresholds and parameters: we show the results in Table 1, compared with a naive application of Markov models (by pre-labeling the traces and building a transition matrix for each user, or class of user). For our models, we show also a measure of the overfitting of the model classes on the training sequences (the higher the fitting, the lower the generalization capacity of the algorithm). Creating Markov models with a high number of nodes increases both detection rate (because users are identified by relatively uncommon commands they perform) and overfitting. Using only 6 types of commands, we obtain a much better generalization and still a 63.3% detection rate. The rate may seem overall low, but it is still much higher than the detection rate of a naive application of Markov models. The computation time for building the model is quite higher than the naive one (about 6 times higher), but still in the order of seconds. At runtime, there is no difference in complexity between our model and a naive one.

5 Conclusions

In this paper, we have described a behavioral approach to anomaly detection. By analyzing and leveraging concepts from the field of ethology, we have presented a general framework to build behavior detection systems. We have also introduced a simple algorithm to build a Markov-based model of multiple classes of behavior. We have shown that a behavior classifier built with this algorithm is capable of detecting intrusion attempts on computer systems. Future extensions of this work could include a deeper analysis of the prior for the Bayesian choice between different models and applications to other problems in the area of behavior detections.

References

1. Serazzi, G., Zanero, S.: Computer virus propagation models. In Calzarossa, M.C., Gelenbe, E., eds.: Performance Tools and Applications to Networked Systems: Revised Tutorial Lectures - MASCOTS 2003, LNCS Springer-Verlag (2004)
2. Anderson, J.P.: Computer security threat monitoring and surveillance. Technical report, James P. Anderson Company, Fort Washington, Pennsylvania (1980)
3. Colombetti, M., Dorigo, M., Borghi, G.: Behavior analysis and training: A methodology for behavior engineering. IEEE Trans. on Systems, Man and Cybernetics **26** (1996) 365–380
4. Martin, P., Bateson, P.: Measuring Behaviour: An Introductory Guide. 2 edn. Cambridge University Press, Cambridge, UK (1993)
5. Lorenz, K.Z.: The comparative method in studying innate behaviour patterns. In: Symposia of the Society for Experimental Biology. (1950) 226
6. Barlow, G.W. In: Ethological units of behavior. Chicago University Press, Chicago (1968) 217–237
7. Jha, S., Tan, K., Maxion, R.A.: Markov chains, classifiers, and intrusion detection. In: 14th IEEE Computer Security Foundations Workshop (CSFW'01). (2001) 0206
8. Zanero, S., Savaresi, S.M.: Unsupervised learning techniques for an intrusion detection system. In: Proc. of the 2004 ACM Symposium on Applied Computing, ACM Press (2004) 412–419
9. Ju, W.H., Vardi, Y.: A hybrid high-order Markov chain model for computer intrusion detection. J. of Computational and Graphical Statistics **10** (2001) 277–295
10. Rabiner, L.R.: A tutorial on hidden Markov models and selected applications in speech recognition. In: Proc. of the IEEE. Volume 77. (1989) 257–286
11. Baum, L.E., Eagon, J.A.: An inequality with applications to statistical prediction for functions of Markov process and to a model of ecology. Bull. American Math. Soc. (1967) 360–363
12. Merhav, N., Gutman, M., Ziv, J.: On the estimation of the order of a Markov chain and universal data compression. IEEE Trans. Inform. Theory **35** (1989) 1014–1019
13. Haccou, P., Meelis, E.: Statistical analysis of behavioural data. An approach based on timestructured models. Oxford university press (1992)
14. Cheung, Y.M., Xu, L.: An RPCL-based approach for Markov model identification with unknown state number. IEEE Signal Processing Letters **7** (2000) 284–287
15. Baum, L.: An inequality and associated maximization technique in statistical estimation for probalistic functions of Markov processes. Inequalities (1972) 1–8
16. Moore, J.B., Krishnamurthy, V.: On-line estimation of hidden Markov model based on the Kullback-Leibler information measure. IEEE Trans. on Signal Processing (1993) 2557–2573
17. Yeung, D.Y., Ding, Y.: Host-based intrusion detection using dynamic and static behavioral models. Pattern Recognition **36** (2003) 229–243
18. Juang, B.H., Rabiner, L.: A probabilistic distance measure for hidden Markov models. AT&T Technical Journal **64** (1985) 391–408
19. Stolcke, A., Omohundro, S.: Hidden Markov Model induction by bayesian model merging. In: Advances in Neural Information Processing Systems. Volume 5., Morgan Kaufmann (1993) 11–18
20. Stolcke, A., Omohundro, S.M.: Best-first model merging for hidden Markov model induction. Technical Report TR-94-003, 1947 Center Street, Berkeley, CA (1994)
21. te Boekhorst, I.R.J.: Freeing machines from Cartesian chains. In: Proceedings of the 4th International Conference on Cognitive Technology. Number 2117 in LNCS, Springer-Verlag (2001) 95–108