Behavioral Intrusion Detection Indicators

Jacques Saraydaryan, Luc Paffumi, Veronique Legrand and Stephane Ubeda

Abstract Monitoring and analysing Information system(IS)'s security events has become more and more difficult in the last few years. As IS complexity rises, the number of mandatory monitoring points has increased along with the number of deployed probes. Consequently, a huge amount of information is reported to the analyst which subsequently floods him and implies the implementation of very complex event analysis engines. In the behaviour analysis context in which sequences of events are studied, this information quantity issue makes it difficult to build automatable - not too complex - models. In order to cope with this increasing amount of information, we will describe a method to reduce the observation perimeter through the selection of most relevant indicators. Such indicators, which are defined thanks to users and attackers behaviour analysis, represent different actions that users or attackers perform in the IS. This method implies neither information loss nor significant detection rate decline. We experienced this indicators selection with a behaviour anomaly detection engines injecting few days of events. Results show that model complexity issues are significantly reduced while keeping detection rate almost the same.

Jacques Saraydaryan
ARES INRIA / CITI, INSA Lyon, F69621, France, Exaprotect, 149 bd Stalingrad 69100 Villeurbanne France e-mail: Jsaraydaryan@exaprotect.com

Luc Paffumi
Exaprotect, 149 bd Stalingrad 69100 Villeurbanne France e-mail: lpaffumi@exaprotect.com

Veronique Legrand
ARES INRIA / CITI, INSA Lyon, F69621, France, Exaprotect, 149 bd Stalingrad 69100 Villeurbanne France e-mail: vlegrand@exaprotect.com

Stephane Ubeda
ARES INRIA / CITI, INSA Lyon, F69621, France e-mail: stephane.ubeda@insa-lyon.fr

Please use the following format when citing this chapter:

Saraydaryan, J., et al., 2008, in IFIP International Federation for Information Processing, Volume 278; *Proceedings of the IFIP TC 11 23rd International Information Security Conference*; Sushil Jajodia, Pierangela Samarati, Stelvio Cimato; (Boston: Springer), pp. 317–331.

1 Introduction

The information security interest greatly increased in the last ten years. Securing large Information System (IS) communications, mobile users and sensitive data became one of the top priorities of private and governmental institutions. Helped by a multitude of security tools, security analysts and administrators organize and manage their IS defense. Classical security tools like firewalls, Host and Network Intrusion Detection Systems (HIDS-NIDS) are focused on local parts of the IS and are not sufficient anymore. These approaches are efficient for local detection but still need to be investigated to provide new methods to reduce data volume, increase alert semantics and detect global attack scenarios. Industrial and research communities show a great interest in the global Information System vision. Recent literatures aim at modeling and discovering global attack scenarios and Information System dependencies. Working on the global vision introduces two main limitations: the volume of computed data that can reach thousands of events per second and the complexity of attacks scenarios and IS dependencies that increase very quickly with the volume of data. Recent works provide three main functions to reduce the large volume of incoming events: normalization, aggregation and correlation. [4] describes an ontology of all actions (called moves) occurring on IS components. By normalizing all incoming events, redundant information are deleted and analysis and IS action modeling are possible. The aggregation approaches [12] gather events sharing the same semantics or same attributes. Correlation solutions follow the same objective by grouping incoming events through predefined models [15], precomputed data, or automatically generated models [7]. All these approaches aim at reducing the amount of data presented to the analysts and used for IS modeling. However, this data reduction is not always sufficient for an administrator's analysis as some hundreds of alerts can remain, the data volume remains an important issue for global analysis especially for global behavioral Intrusion Detection where all the events information are not relevant. In this paper, we introduce the notion of necessary transit actions for an attacker to achieve his objectives: these actions are called "checkpoints". We propose a selection of specific monitoring points (called indicators) in order to focus our analysis on specific local points in the IS. With the extraction of critical and relevant key points, we only provide necessary information for a global behavioral intrusion detection analysis. The section 2 introduces a survey of different observation points involved in anomalies detection. A user behavior analysis is realized on section 3. Section 4 and section 5 provide a classification and selection of behavioral indicators. Experimental results show our complexity reduction of anomaly models in section 6, followed by our conclusion in section 7.

2 Behavioral observation points

Since the beginning of the behavioral intrusion detection [8], several approaches aim at discovering anomalous behavior reflecting attacker's activities. In this section, we enumerate behavioral indicators described in Behavioral Intrusion Detection. This one tends to collect all types of behavioral observations in order to specify the relevance of each ones. Behavioral Intrusion Detection can be divided into two categories: Host intrusion detection System (HIDS) and Network intrusion detection system (NIDS).

2.1 Behavioral HIDS indicators

Traditional behavioral HIDS focus their analysis on particular points of observation inside the host. [14] realized an overview of intrusion detection methods and data sources. It describes information used by HIDS: system access information, system usage information, files usage information, application usage information and security violation information. System access information describes how someone or something accesses the system, how relevant information can be monitored like user or process/terminal ids, connection modes (local, remote) and time relation between connections. System usage information is focused on interactions between users and systems. [9] determinates the frequency of each user commands. User names, command types and times are the main properties of the used command. [6] models system command sequences. Each command is defined with pre and post conditions (file name, kind of agent, address and host name, source port, etc.). [1] determinates anomalies of proxylets by comparing CPU usage and memory use with actual CPU and memory load. File usage information determines how file can be accessed like access time, types (open, close, read, modify, etc.). As explained in [13], common intruder actions are visible thanks to the file manipulation monitoring. Intruders who successfully enter the IS often modify data. [13] stresses out four categories of *warning signs*: data/attribute modification, update pattern (deviation of “rotated” log file or modification of previous one), content integrity and suspicious content. Information hold by used applications gives detail about application properties installed on the host. [10] focus their work on system call modeling during application run to discover potential application misuse. [17] models temporal application relations of each user to discover unusual sequence application uses.

A last indicator, security violation information, defines anomaly behavior as a violation of specific rules. [5] defines specific policies for regulating access to system resources. Some strategic file access or bad privilege command execution are relevant about suspicious behaviors.

2.2 Behavioral NIDS indicators

While HDISs focus on the monitoring of specific components of the IS, NIDS became a complementary mechanism by monitoring network traffic. They are located in strategic network points of IS like network DMZ, firewall front or back head area. Although signature based NIDS are most popular inside commercial products, behavioral analysis becomes an alternative way to detect unknown attacks. Computing network flows modeling or determining threats threshold, behavioral NIDSs discover network flows anomalies. [11] classified network anomaly detection following three criteria: the Network feature Analyzed, the Behavior Model and the Analysis Scale. Network feature Analyzed expresses which data is monitored and modeling inside the IS. Behavior Model defines how behavioral NIDSs model usual IS activities (learnt models, specification-based models, etc.). Finally, Analysis Scale provides information about the level of analysis abstraction. For example, the monitoring of the number of packet can be viewed as a low level, the monitoring of connections or packet streams as a medium level and high level can be performed by the monitoring of several connections and event correlation within the whole network.In our context, we particularly focus on the Network feature Analyzed which defines different behavioral observation points. Network feature Analyzed criterion is divided into two main groups, the network traffic data source and the Network Elements/topology information. The Network Elements/topology information is not currently usable for classical wired networks and find its importance in emerging architecture like ad hoc networks. The Network Traffic data source constitute the major source of network behavioral analysis. Two Network Traffic properties (flows or protocols) can be analysed. On one side, flow analysis are characterized by the study of the evolution of traffic flows. Related data sources are various: number of bytes sent/received during a fixed time interval by a given final system, number protocol packets (TCP/UDP,etc.) packets sent/received, number of TCP/UDP connection, number of HTTP/DNS request,etc. On the other side, protocol analysis consists in discovering protocol misuse at different levels [2](Data Link, Network, Transport, Application). The analysis is realized on sequences of protocol steps and protocol transaction evolutions.This section tends to cover the monitored data frequently used for the behavioral analysis. All these data do not have any correspondence between each other. The next sections will explain logical links between behavioral observations and propose a selection of essential points of observations. These logical links will improve the interpretations of global behavioral anomalies study.

3 User-Oriented Behavioral Analysis

We focus our study on the selection of indicators for the global Behavioral Analysis of users actions. In order to define key points in the IS, we study the behavior of each users family in the IS. The IS behavior can be defined thanks to two main elements: the behavior of actions realized on the IS and the components states of the IS. Four different entities can be distinguished for the IS behavior definition: the behavior of classical user, the behavior of administrator of the IS, the behavior of the attacker and the behavior of the IS itself. Usually, the IS security is evaluated through a method which associates a number with attributes (called metrics) like:

Confidentiality, Integrity, Availability (CIA). Users' actions affect these metrics by increasing or decreasing their values (e.g. an Administrator increases confidentiality by configuring authentication, an Attack decreases the availability by flooding a server) (figure 1). In the following paragraph, we are going to describe each users family approach and involved actions affecting IS metrics.

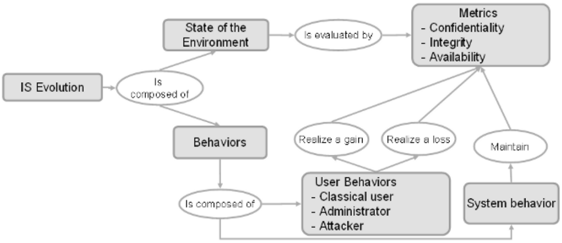


Fig. 1 Information System Composition

3.1 User actions modeling

Most of this work relies on the [4] ontology that provides a user actions description. It implies four parameters; the intention of the user, the realized actions, the target impacted by the action and the result of actions. An IS user has typically four types of goal: collecting information about a target (Recon), accessing the IS (Authentication), accessing IS resources (Authorization) and affecting IS resources (System). These four goals (called intention) describe the reasons why the user performs an action. In order to achieve his aim, the user (including classical users, administrators or attackers) performs actions directed on a target. These actions are differentiated according to their modes (activity, config, attack) and their natures (login , read, execute,etc.). Finally, each action has a result that reflects the gain of the user on the system i.e. whether the user succeeded its action attempt or not. For instance, a user that logs into an SSH server would be modeled in the ontology by a four-uplets : Authentication (referring to the intention), Activity Login (referring to the realized action), SSH (referring to the target) and Success (referring to the result). The resulted action model is noted Authentication.Activity.Login.SSH.Success.

3.2 Classical users approach

Classical users use the IS for professional or personal interest but always with respect to security policies. As described in section 3.1, we base our work on [4] and [3] which describe a wireless networks users analysis. We only extract the intention and the types of actions for our analysis, the other ones being too much specific and useless for the global behavior analysis. The *Recon* intention is not often achieved by classical users. For our study, we merge both authentication and authorization (often performed successively) in a unique authentication intention. The *System* intention is usually performed by classical users. As for the realized actions, we only use the *Activity* action for the classical user, the other ones representing attacker or administrator behaviors (described in section 3.3 and 3.4). Classical users activities could be summed up into two main action families; the local or remote connections/ authentications (services, applications of the IS ,etc.) and the use of local or remote resources. Two sequences of actions can be distinguished, local action sequences (authentication, resources use, disconnection) and remote action sequences (remote authentication, remote connection, remote resources use, remote service disconnection, system disconnection).

3.3 Administrator approach

An Administrator is a special IS user. Administrator inherits classical users behaviors and has special additional properties. One of the administrator characteristics is his ability to switch between classical user activities and administrative tasks. In addition to its classical user activities, an administrator has to manage and configure the IS. These functions need the use of special actions on specific targets, theoretically not accessible for classical users. In order to take into account these specific activities, the realized actions *Config* of [4] are used to describe a policy or configuration modification, add or deletion. Moreover, the intention *Recon* is also used to describe administrator activities as an administrator needs to get some information on its IS in order to monitor and manage it as well as possible. Configuration and maintenance actions form the administrator's main activities. He connects itself to the system with special logins (root) giving him full rights to the System. With such rights, an Administrator can effectively modify configurations and policies in the IS. Moreover, tests of accessibility and vulnerability (which take part of the *Recon* intention) can be launched in order to verify the performance of the system.

3.4 Attacker approach

The attacker's behaviour is the most complex and unpredictable one.[4] defines the first steps an attacker usually performs to enter an IS: information gathering and vulnerabilities exploitation. Sometimes, attackers can perform alternative initial actions; the attacker can also take advantage of backdoors or virus installed by users unaware of security risks during the metastasis phase.Despite various objectives, we can enhance the definition of necessary steps to achieve the attackers objectives. So as to mask his actions, an attacker would try to hide his malicious behaviour by

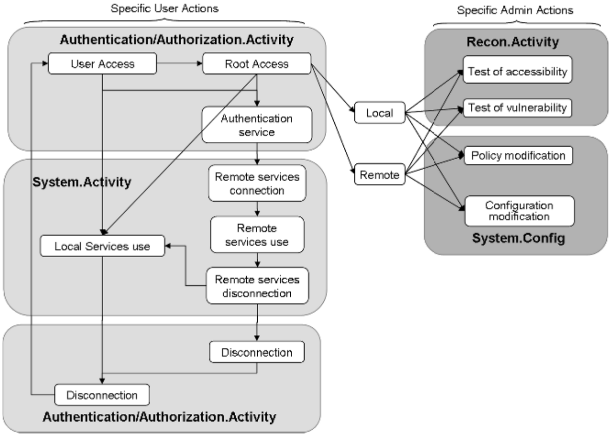


Fig. 2 Administrator approach

faking a classical user behavior. An attacker behavior inherits from classical user actions. An intermediate goal of an attacker is to gain administrative privileges in order to modify, alter or steal data in the IS. The figure 3 shows possible actions of an attacker on the IS. We can distinguish five objectives of an attacker; gain of privilege, gain of access, deny of service, bounce or data stealing (spying). In order to achieve these goals, following steps are essential. In the first step, an attacker has to locate the targeted system. In a second step, he needs to collect information about the system. Once the target located and analyzed (environment and vulnerabilities discovery), the attacker exploits a vulnerability to reach his final goal. An alternative sequence of action using automatic tools (malware like virus) exists. These malwares would automatically try to exploit some vulnerabilities. Once the system penetrated, the attacker would have a behavior close to classical users or administrators. All attacker actions are not always detectable (new attacks, miss of specific probes or non adequate probes in some locations of the IS), however it is possible to reveal important information about deviating users or administrator behaviors.

3.5 User-Attacker comparison

Normal IS users (classical users and administrator) share similar actions with attackers. These actions are bottlenecks in IS for all users. These actions reveal attackers checkpoints inside the user or administrator approach. As defined in section 1, these checkpoints define necessary actions for an attacker to reach his objectives. They constitute an essential behavioral monitoring for the detection of deviating behaviors subject to belong to an attacker. The figure 4 shows the interactions between the classical user/administrator approaches and attacker strategy. Arrows going through Vertical separators between both approaches represent merged points of both ap-

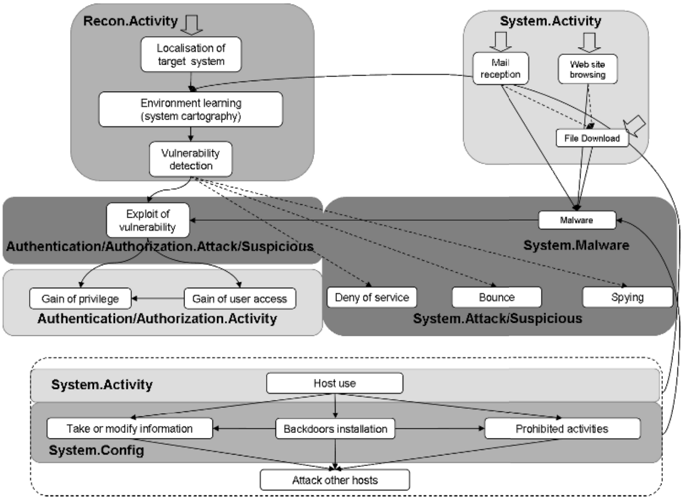


Fig. 3 Attacker approach

proaches. Two main checkpoints are enhanced. The connection action in the user approach side defines the first checkpoint. An attacker has no alternative way to reach his objective, he has to pass through classical authentication points to go into the IS. The second checkpoint is the transition from the user approach to the attacker strategy at the end of the attacker scenario. In order to achieve his malicious goals, the attacker would deviate from the original user or administrator behavior. This second checkpoint is less relevant because the deviance detection at this stage is not necessarily possible. For most of time, slight behavioral deviation on IS are not detectable and only effects of these actions differentiate attackers from normal IS users.

3.6 System Approach

The IS has to maintain its level of Confidentiality, Integrity and Availability (CIA). IS actions will represent internal actions by the system itself and not by a user interaction. Each component's behavior can be modeled as follow: after the physical start of the component, this one will launch services needed for its own operations or will try to load external services or information (load DHCP address, load list of services to start...). The started services could modify internal configuration of the system depending of the policies of the IS (network parameters modifications, restriction of some functionalities). After this starting step, the component will wait for external environment interactions. As soon as a user interaction occurred, the IS checks the security permission for the asked operation. The permissions are checked and the operation is launched. Information of the launched operation may be revealed through the state of the component (e.g. CPU overload representing by System.Information in [4]). The component can also execute operation by its own (component basic functions) as forwarding information, mail/packet reception and dispatch,etc. Then the component can be halted by an external intervention (System.Information.Stop) or by itself (System.Activity.Stop).

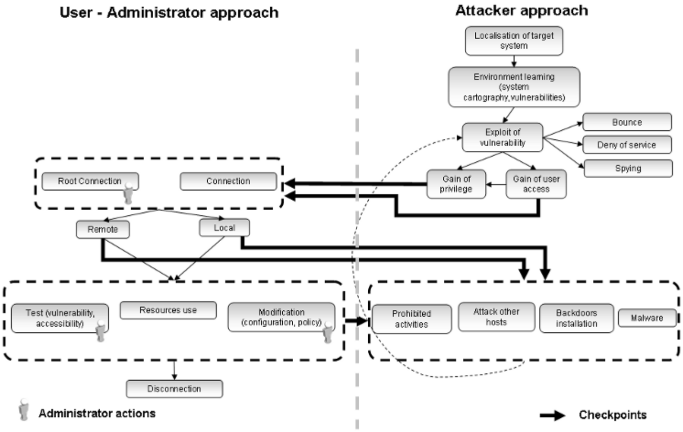


Fig. 4 Checkpoint

4 Indicators Selection

The description of IS user behaviour highlights different groups of actions. The main groups reflect actions about system access, usage/modification of authorization (called rigths) and usage/modification of the system. All users interact with the system through success or failure of these three groups. To determine which actions of success/failure are important for the user behaviour monitoring, we select all events reflecting system-user interactions. Trying to detect behavioral anomalies, our groups of observations (called meters) are focused on the evolution of classical user's and administrator's behaviors. Following users approach (described in section 3), each group is divided into different activities reflecting usage, modification or state information. The next paragraphs detail each group content in depth.

4.1 Indicators for the Access Meter

The access activities will be represented through three indicators: authentication activities, connection activities and modification of authentication configurations. The authentication activities correspond to the activities of login of an user or an administrator. The connection activities reflect the system activities during the process of authentication. These activities include the connection to an external service of authentication and other necessary transactions to identify the connected account. The monitoring of the modification of configuration is essential to detect potential intrusion activities since this one is the entry point for all IS users (section 3). All actions concerning the modification of authentication configuration will be taken into account.

4.2 Indicators for the Rights Meter

The rights activities will be represented through two indicators: rights activities and modification of rights configurations. The rights activities explain the use of special rights by a user in order to achieve an operation. The modification of rights activities describes the activities of modification by a user of its own rights or rights of other people or objects.

4.3 Indicators for the System Use Meter

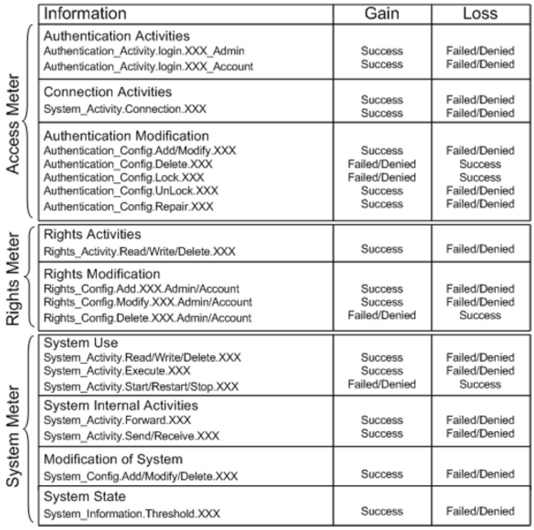
Different activities can be operated on an IS. Files or objects can be of course read, written, deleted, but other special operations can be realized like executing a command, launching an application, starting/restarting/stopping a service. We can differentiate two IS activities; activities of user interactions and activities of IS itself (automated actions). Activities of user interaction include usage of services or applications and also services/applications configuration. Activities of IS describe related actions operate by IS itself like necessary packets transactions, files or configurations upload, etc. Moreover, the IS state reflects the actual load/status of service or IS components. This monitoring point can enhance the detection of abnormal action effects.The figure 5 sums up the selection of indicators for all meters.

5 Indicators Selection Refinement

Previous sections provide a set of relevant indicators to monitor user behaviors. Nevertheless, the set is still too large to be efficiently used as an anomaly detection. In order to reduce this set, this section presents two processes. One based on the selection of checkpoints (key points of observation), and the other one based on the delimitation of the observation perimeter.

5.1 Checkpoints Selection

As explained in section 3.5, attackers' actions pass through bottlenecks called attackers' checkpoints. In order to select the most relevant checkpoints, we analyzed each kind of attacks and determined their checkpoints. The attack-centric taxonomy defined by the DARPA advocates an attack classification through the attack effects. Five effect classes are distinguished: User to Root, Remote to local, Denial of service, Surveillance/probe, System Access/Alter data. We enrich these classes by adding for each class all possible attackers actions leading to a specific effect.



	Information	Gain	Loss
Access Meter	Authentication Activities Authentication_Activity.login.XXX_Admin Authentication_Activity.login.XXX_Account	Success Success	Failed/Denied Failed/Denied
	Connection Activities System_Activity.Connection.XXX	Success Success	Failed/Denied Failed/Denied
	Authentication Modification Authentication_Config.Add/Modify.XXX Authentication_Config.Delete.XXX Authentication_Config.Lock.XXX Authentication_Config.UnLock.XXX Authentication_Config.Repair.XXX	Success Failed/Denied Failed/Denied Success Success	Failed/Denied Success Success Failed/Denied Failed/Denied
Rights Meter	Rights Activities Rights_Activity.Read/Write/Delete.XXX	Success	Failed/Denied
	Rights Modification Rights_Config.Add.XXX.Admin/Account Rights_Config.Modify.XXX.Admin/Account Rights_Config.Delete.XXX.Admin/Account	Success Success Failed/Denied	Failed/Denied Failed/Denied Success
System Meter	System Use System_Activity.Read/Write/Delete.XXX System_Activity.Execute.XXX System_Activity.Start/Restart/Stop.XXX	Success Success Failed/Denied	Failed/Denied Failed/Denied Success
	System Internal Activities System_Activity.Forward.XXX System_Activity.Send/Receive.XXX	Success Success	Failed/Denied Failed/Denied
	Modification of System System_Config.Add/Modify/Delete.XXX	Success	Failed/Denied
	System State System_Information.Threshold.XXX	Success	Failed/Denied

Fig. 5 Indicators selection

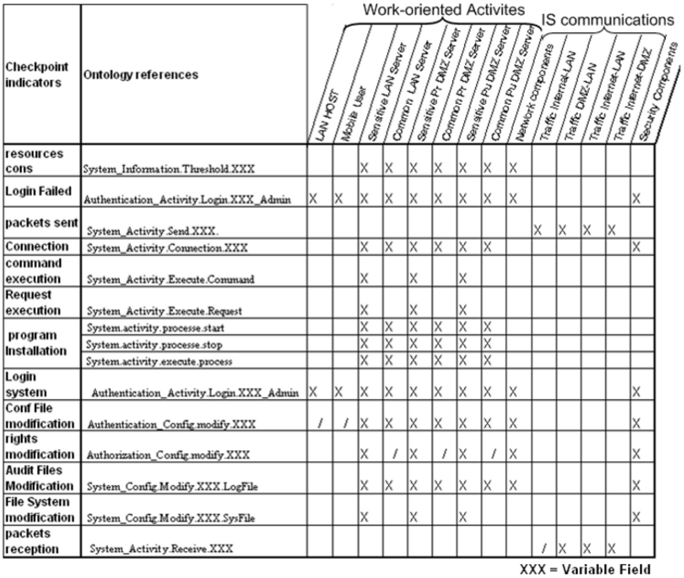
We analyzed all checkpoints of all possible actions leading to one of these five effects. In order to illustrate our approach, we detail the attacker checkpoints for the User to Root effect. One of the main objectives of an attacker is to reach root rights on a host. The authentication and the connection to the IS are two requirements to achieve this objective. As soon as the attacker is connected, several actions are possible to realize this exploit. Checkpoints exist for each of these possible actions. Six actions can lead to a User to Root effect. The gain action consists in modifying the rights of a running session. The associated checkpoint is the modification of rights for this session. Another way of privilege escalation is the realization of an injection. An injection consists in launching an operation through a started session or service. The checkpoint needed to achieve this operation is the launch of the command. Furthermore, an overflow overloads a service in order to execute a command, can lead to gain upper rights.By flooding a buffer or a service, an attacker can overwrite another memory space and execute commands or scripts.To do this, a command or a request needs to be launched after the packet sending for the overload. The bypass attacker action which consists in bypassing authentication and rights by an exploit, is more difficult to be detected. Usually, a command execution is realized before an exploit. Finally, an attacker can elevate his rights using a virus or a Trojan. These Malwares are programs installed in the system with more rights and obviously need a program installation. The installation of a program or service defines a checkpoint for the Malwares usage. The checkpoint study leads to reduce the number of unavoidable checkpoint monitoring at a considerable degree. The final list of checkpoints are presented in figure 6.

5.2 Monitoring Perimeter Delimitation

The second reduction process consists in focusing on the monitoring of filtered indicators on special IS components. The relevancy of each indicator depends of the location where there are collected. The perimeter of observation of these indicators depends of the nature of the IS component from which there are collected. We separate IS components into three classes: components involved in work-oriented activities, components used for IS communications and one focused on security components. We differentiate components involved in work-oriented activities with their location in the IS and their criticality regarding their business importance (User Host LAN, Mobile User Host, Classical/sensitive LAN Server, Classical/sensitive private DMZ Server,Classical/sensitive public DMZ Server). The components of IS communications include the network components that manage and maintain the network activity (Network equipments). Moreover, we specify different network traffics, witnesses of network exchanges in the IS. (Internal LAN Traffic , LAN-DMZ traffic, LAN-Internet Traffic, DMZ Internet traffic). Then, components involved in the security management, detection and configuration compose the security components group (Firewall, Antivirus, IDS/IPS, Security Information Management). This classification of IS components allows defining appropriate information monitoring regarding their nature, location and sensitivity. The figure 6 sums up a proposition of indicators regarding the perimeter delimitation and checkpoints selection.

6 Experimentation

Anomaly detection used to test our approach is trained with normal event sequences [16]. Then, these sequences are transformed in a Bayesian Network where nodes represent events and linked nodes represent sequences of events. This approach highlights three anomaly classes: node anomalies (identification of unknown events), state anomalies (identification of unknown sources of events) and probability anomalies (identification of unfrequent sequences of events). We compared the detection rates of [16] with and without our indicators selection. The experimentation was realized on two datasets (training and test). The training data set is composed of 1500 events collected on heterogeneous probes. The test dataset is composed of 60 usual events and 30 events composed of attack scenarios (DoS, Bruteforce, Trojan contamination) and unusual system use. The resulting Bayesian Network model has been reduced by 36.60% for nodes and 54.83% for links.



Checkpoint indicators	Ontology references	Work-oriented Activites									IS communications				
		LAN HOST	Mobile User	Sensitive LAN Server	Common LAN Server	Sensitive Pr DMZ Server	Common Pr DMZ Server	Sensitive Pu DMZ Server	Common Pu DMZ Server	Network components	Traffic Internal-LAN	Traffic DMZ-LAN	Traffic Internet-LAN	Traffic Internet-DMZ	Security Components
resources cons	System_Information.Threshold.XXX			X	X	X	X	X	X	X					
Login Failed	Authentication_Activity.Login.XXX_Admin	X	X	X	X	X	X	X	X	X					X
packets sent	System_Activity.Send.XXX.										X	X	X	X	
Connection	System_Activity.Connection.XXX			X	X	X	X	X	X						X
command execution	System_Activity.Execute.Command			X		X		X							
Request execution	System_Activity.Execute.Request			X		X		X							
program Installation	System.activity.processe.start			X	X	X	X	X	X						
	System.activity.processe.stop			X	X	X	X	X	X						
	System.activity.execute.process			X	X	X	X	X	X						
Login system	Authentication_Activity.Login.XXX_Admin	X	X	X	X	X	X	X	X	X					X
Conf File modification	Authentication_Config.modify.XXX	/	/	X	X	X	X	X	X	X					X
rights modification	Authorization_Config.modify.XXX			X	/	X	/	X	/	X					X
Audit Files Modification	System_Config.Modify.XXX.LogFile			X	X	X	X	X	X	X					X
File System modification	System_Config.Modify.XXX.SysFile			X		X		X							X
packets reception	System_Activity.Receive.XXX										/	X	X	X	

XXX = Variable Field

Fig. 6 Perimeter Sum UP

Figure 7 shows the difference between detection rates. White columns represent the number of detected events through the anomaly detection engine without indicators selection. Grey ones represent the number of detected events through the anomaly detection engine with indicators selection. Black columns are the number of scenarios' events inside the test data set. We can notice that, for the events detection of attack scenarios, the same detection rates is found. However, the unusual system use detection rate is slightly lower. Moreover the indicators selection reduces the false positive rate. False positives have been reduced from 10.0% to 6.6%.

To enhance the complexity reduction, we compared the model computation time with and without our indicators reduction. The model computation was based on the training data set previously described and enriched by three other training data sets of 3000, 4500 and 6000 events. These training data sets share same types of events proportions. Figure 8 shows a significant reduction of the model computation time when only relevant indicators are selected. We can notice that for a low number of events, time reduction is of about 20% whereas for a higher number of events, time reduction is of about 40%. These results can be explained by the fact that new relations between events appear when the number of events is high. By deleting irrelevant information, indicators selection also deletes irrelevant relation. The invariability of the time reduction for a high number of events is due to the fact that no new relation between events has been found. The lack of new relations can be explained by the composition of the events which maintains the same proportion of alert's types.

More complex experimentations are under construction. We modeled a set of 86000 events following the [16] approach and highlight a node reduction of 24.40% and a link reduction of 34.12%. We presently work on the comparison of detection rates on large attack scenarios datasets.

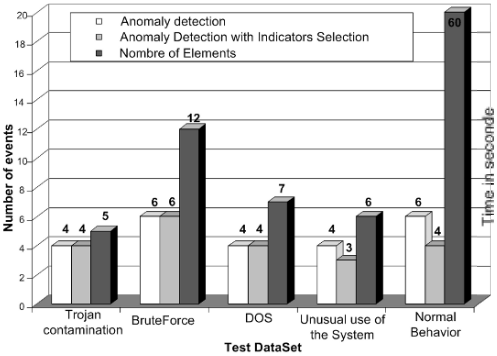


Fig. 7 Detection Rate Comparison

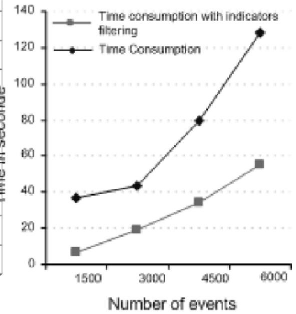


Fig. 8 Modeling Time Consumption

7 Conclusion and Discussions

In this paper, we presented both the benefit of the IS monitoring point selection (called indicators) that reduces the volume of data to handle and the selection method. After analyzing the significance and the relationships between each monitoring point (based on the state of the art) within IS user approaches, we emphasized necessary indicators for the monitoring of relevant behavioral deviances. Focused on the attackers' checkpoints, our set of indicators represents bottlenecks of attackers and legitimate users in the IS. Combined with business perimeter delimitations, we considerably lower the number of indicators, thus the number of sources observations. This complexity reduction allows a better scalability and the creation of more detailed models. We already tested our approach on anomaly detection engines and decreased significantly the complexity of the normal behavior model, slightly reducing the detection rate. We intended to realize deeper experimentations on large datasets with other anomaly detection engines.

The indicators reduction could also be applied to some global anomaly detections to compare and enhance our first results. In the same way, our further works tend to specify relevant sequence of events. This approach will enhance the complexity of the reduction of anomaly detection engines which exploit sequences of events.

References

1. Adballah Abbey Sebyala, Temitope Olukemi,Lionel Sacks. Active platform security through intrusion detection using naive bayesian network for anomaly detection. In *Proceedings of the London Communications Symposium 2002*, 2002.
2. A. Alharby and H. Imai. Security protocols protection based on anomaly detection. *IEICE Transactions on Information and Systems*, E89-D(1):189–200, 2006.
3. Anand Balachandran, Geoffrey M. Voelker, Paramvir Bahl, and P. Venkat Rangan. Characterizing user behavior and network performance in a public wireless lan. In *SIGMETRICS '02: Proceedings of the 2002 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, pages 195–205, New York, NY, USA, 2002. ACM Press.
4. Fatiha Benali, Veronique Legrand, Sephane Ubeda. An ontology for the management of heteregenous alerts of information system. In *SAM*, 2007.
5. S. Chari and P. Cheng. Bluebox: A policy-driven, host-based intrusion detection system, 2003.
6. Frédéric Cuppens, Fabien Autrel, Alexandre Miège, and Salem Benferhat. Recognizing malicious intention in an intrusion detection process. In *HIS*, pages 806–817, 2002.
7. Olivier M. Dain, Robert K. Cunningham. Building scenarios from a heterogeneous alert stream. In *IEEE SMC Information Assurance Workshop*, 2001.
8. D. Denning. An intrusion detection model. In *IEEE Transactions on Software Engeneering*, pages SE–13:222–232, 1987.
9. William DuMouchel. Computer intrusion detection based on bayes factors for comparing command transition probabilities. Technical report, National Institute of Statistical Sciences (NISS), 1999.
10. Eleazar Eskin, Wenke Lee, and Salvatore J. Stolfo. Modelling system calls for intrusion detection with dynamic window sizes. In *the DARPA Conference and Exposition on Information Survivability. DISCEX '01*, 2001.
11. Juan M. Estvez-Tapiador, Pedro Garcia-Teodoro, and Jess E. Daz-Verdejo. Anomaly detection methods in wired networks: a survey and taxonomy. *Computer Communications*, 27(16):1569–1584, 2004.
12. Annarita Giani, Ian Gregorio De Souza,Vincent Berk, George Cybenko. Attribution and aggregation of network flows for security analysis. In *FloCon 2006*, 2006.
13. Adam G. Pennington, John D. Strunk, John Linwood Griffin, Craig A.N. Soules, Garth R. Goodson, Gregory R. Ganger. Storage-based intrusion detection: Watching storage activity for suspicious behavior. In *the 12th USENIX Security Symposium*, 2003.
14. Ludovic Me and Cdric Michel. La dtection d'intrusions : bref aperu et derniers dveloppements. Actes du congrs EUROSEC'99, 1999.
15. B. Morin and H. Debar. Correlation of intrusion symptoms: an application of chronicles. In *6th International Conference on Recent Advances in Intrusion Detection (RAID'2003)*, 2003.
16. Jacques Saraydaryan, Veronique Legrand & Sephane Ubeda . Behavioral anomaly detection using bayesian modelization based on a global vision of the system. In *7eme Conference Internationale sur les NOuvelles TEchnologies de la REpartition (NOTERE 07)*, 2007.
17. Alexandr Seleznyov and Seppo Puuronen. Anomaly intrusion detection systems: Handling temporal relations between events. In *Recent Advances in Intrusion Detection*, 1999.