STUDIA INFORMATICA
Nr 1-2(14) Systems and information technology 2010

Intrusion Detection Systems. Model and implementation of a knowledge base of intrusions into the computer system

Andrzej Barczak, Grzegorz Tokajuk
Institute of Computer Science
Siedlce University of Natural Sciences and Humanities
3 Maja Str. 54, 08-110 Siedlce, Poland

Abstract. There are many complementary approaches to detecting intrusions e.g. behavior based approach, knowledge-based (KB) approach. The paper described the last one. The essential problems of determination of a KB for intrusion detection systems (IDS), prepared of threat signature, architecture of database containing the rules are considered.

Keywords. IDS, knowledge base systems

1 Introduction

In intrusion detection systems, knowledge bases, are appearing in many different forms. They determines the way to profiling users and systems, storing attack signatures used to intrusion detection and storing information that is considered useful for processing, correlation and analysis of potential intrusion. An important issue in research with regard to knowledge bases is to develop a general method of determining the characteristics of encoding profiles and information about the attacks.

The knowledge base is a component of intrusion detection system, which stores information about the attacks in the form of signatures and text strings. Beside the definition of known attacks can be found here information about system and users behaviors in the form of profiles and information about the risks that are described by signatures. The knowledge base should be equipped with mechanisms of query processing in order to fully support the functions of intrusion detection. It must also be possible to supplement it with signatures of new attacks. Knowledge base should have adequate capacity and must be properly protected.

2 Information stored in intrusion detection systems knowledge bases

Profiles of normal user and system behavior

This information includes descriptions of users and protected systems, attending as an indicator of an attack in the detection of abnormal behavior paradigm. Creating a profile of user behavior often consists on indicate an activity pattern such as types of used services and commands, typical duration of operation and CPU utilization degree. Profiling system behavior often involves on indicate periods of maximum and minimum load, and degree of utilization CPU and memory.

Signatures of known network threat

This information represents the signature for provided sequences of harmful actions. The sources of such signatures include public archives of information about the safety, cracker appearing at the conference and publications. The key issue here is a representation of those attacks.

Text strings, which can be regarded as suspicious in any request for service on the network, can be signatures. Rules that are in knowledge base, are based on practical experience of computer security specialists and contain attack signatures. Signature analysis consists in matching parameters of the system and its objects, and network traffic to the attacks database. Most intrusion detection systems implements signature analysis based on attacks database, developed by the manufacturer of the system. In many cases, however, is the ability to add their signatures.

Text sequences which means suspicious patterns in package

This involves checking whether the package is sent text strings, which may indicate an attempt to attack. These suspicious strings are stored in the knowledge base and compared with the studied packages.

The information used to start the reaction

Such information is usually guides the desire processing and reactions, which correspond to the various anomalies and attacks arrives at the intrusion detection system. At centers of the intrusion data processing, which can support a lot of sources of data, you can keep the various reacting processes for every source.

Furthermore, the knowledge base should store information about detected threats. This may be helpful in detecting intrusion attempts and help administrators gather information about intrusion attempts have already been made. The information contained in the database can be correlated with each other and used to create various types of reports so that administrators can take appropriate action to improve the security of the protected system.

3 Updating the knowledge base

Due to the constant evolution of methods applied by burglars a possibility of updating the knowledge base of new attack signatures is a condition of the effectiveness of detecting hacking in the longer time. He has this paramount meaning

since only the regular and on time updates can provide the desired level of effectiveness of intrusion detection. It would be well if there was no delay between the moment of the publication of the information about the new attack and the moment of including its signature in the database of intrusion detection system. Even better would be if the manufacturers have updated their systems prior to their publication of information about vulnerabilities. In this way users would not even stay for a moment at the back behind burglars. However, in practice updates rarely proceed so efficiently. Independently of how quickly the producer is responding to news reports on new attacks and gaps in components of computer systems, between the appearance of such a gap and taking it into consideration in the database of signatures must pass the while needed for the producer for the preparation, testing and distribution of the updating package. Meanwhile a minimization of this delay is a base of the effectiveness of the intrusion detection system. To enable this some knowledge bases enable the user to define own signatures of attacks. Such signatures can be created using a special attack description language, or by simply setting the parameters specific to the attack. Mechanisms of independent supplementing the knowledge base intrusion detection system on own definitions of attacks and the vulnerabilities are priceless for administrators which systematically are following bulletins publishing news reports on attacks and gaps. They can then quickly make new rules for attack detection signatures, immediately immunize the system to new threats. Some of the knowledge base intrusion detection systems have implemented mechanisms for automatic generation of signatures based on collected information.

4 Determination of the knowledge base for intrusion detection systems

Intrusion detection systems operating at the network perform on the basis of network traffic observations. Creating rules for matching the characteristics of this movement, one must take into consideration both of information from internal, as well as outside sources. External sources are roots of the following information characterizing transmission:

- addresses (source and destination)
- port numbers (source and target)
- packet size
- connection status
- data fields
- header tags (in case of TCP)

Collection of attack signatures can be developed in several ways. You can, for example, select the signatures from all those that are found in the network by including all available signatures to signature database and monitoring cases of fitting them via the controlling console. However, this approach ignores the possibility of omission of some signatures in the examination of network traffic.
The different approach consists of few strides:

1. Creating the list of protocols, port numbers and addresses into given section of the network. If the intrusion detection system is supposed to protect the section which is connected to a public network and the network is protected

by a firewall, it is possible to be based on the list used at the firewall configuration.
2. Creating a list of operating systems and software installed in the protected network. The list should include all software installed on nodes with its updates.
3. Information gathered in the previous steps should be placed in the table as a basis for a set of rules for intrusion detection system.
4. Knowledge base intrusion detection system should be limited to only those signatures which reflecting attacks are representing contents of the table.
5. In case of the lack of signatures covering attacks on some programs or services from the list, it is possible to attempt creating own signatures.
6. Finally, adjust the properties of the individual selected for the collection of signatures for intrusion detection system.

Adjusting to your needs signatures of attacks, make sure that both of the signatures and their matching rules operate on numeric representations of sender and recipient addresses, transmission, rather than symbolic addresses, including domain-names. Otherwise, incorrect configuration of DNS server (or its failure provoked in order to and burglar carried) may prevent correct identification of the source. At least a numerical representation of addresses isn't too legible that is more reliable. Best, if symbolic representations of addresses are applied while configuring the system and then their explanation of the numerical figure follows. This greatly simplifies the configuration process. For Intrusion Detection System all signatures and rules of fitting them should not be available, because this reduces the packet processing performance. Similarly, performance can be also reduced by the excessive number of filters defined by the operator. Hence the necessity of the identification of protocols and operating systems used in the given section networks. An excessive number of rules slows down intrusion detection system and increases the probability of configuration errors.

5 Automatic manufacturing signatures of network threat

Manufacture of network threat signatures, which are used in intrusion detection systems are in most cases a manual process, and thus prone to errors and slow. So that the protecting system is effective, the reaction to new threats should be fast and effective so that doesn't bring undesirable incidental effects behind itself. The knowledge base of such system can have implement function of automatic producing threat signatures which can achieve this purpose. Signature can be defined as a set or subset of characteristics of given threats. For example, features may be information from the packet header fields, the contents of packets, analyzing the frequency of occurrences of characters, triggering system functions or temporal relations between events. The essential features of the signature are the low rate of false alarms, attacks detection and speed of manufacturing the signature. It is very important that as far as possible the signature should be independent of application layer protocols. Thanks to that, for effective applying the signature, there won't be necessary to understanding by intrusion detection system an application protocol, which is the signatures relates. This allows you to achieve the universality of signatures, because it

can be applied in a large number of systems or also used towards brand new protocols. From the viewpoint of preventing an attack, it is desirable to extracting the signature attack vulnerability, which it uses. Such a signature is a more general, independent from exploit used to carry out an attack. However, in terms of information it is advisable to also obtain a signature uniquely identifying an exploit. The first type of signature is much more difficult to obtain.

The basis for signature generation is to identify network traffic that is supposed to be submitted for processing. The first step is to detect anomalies in network traffic and classify the movement as an attack or not. It is assumed that a new threat is characterized by a repetition of the measures necessary for propagation. If an available harvest of flows is connected with threatening data you are ready to try to describe the characteristics of the threat and thus generate the signature. Result of the signature generation process is then be verified against the samples of traffic, about which it is obvious that it isn't bringing threats with himself. In this way a quality of the produced signature is estimating. After generating and measuring the quality signature is classified on the basis of similarity to the previously generated signature.

The simplest method of attack identification is to compare and catalog packets in network using cryptographic abbreviations (ex. MD5). Often repeating itself the same packages managed from different sources to different hosts can be an indication of new threat. In this case the MD5 abridgement is becoming the signature of the threat. This method, however, involves performance problems and a large number of false alarms if it is running on a simple network with large capacity. Large amounts of false alarms result from it that in such networks the majority of the move is legal. Applying this method becomes more effective in the HoneyNet environment. In such networks the movement is smaller and in the majority connected with threats. However such an approach to automatic generation of signatures has great flaw. Any minor modification of the attack will produce a new MD5 hash. In this case, sequence of bytes constituting the essence of the attack is not separated, but the whole package. In order to not consider the entire package a mechanism of the moving window is applying. This allows better variants identification and attack isolation. At constant sliding window of specified length in the same package footprint is calculated, which next is being compared among packages. After exceeding the threshold of a repeated fragments defined by the number of source and destination hosts the alarm about the potential threat detection is indicated. In this method, for every package there are many imprints counted, of which the number is dependent on the length of the package and the breadth of the window. However, the need of counting a number of impressions for each packet using cryptographic abbreviations is the performance problem. In their place we used Rabin imprints, which are the basis of Rabin-Karp algorithm. This algorithm is one of the fastest to search for strings. Rabin imprints are more effective than imprints calculated by cryptographic method, because it enabling calculate the window shortcut offset by one byte basis on previous window calculations. The shorter imprint is calculated the better probability of detecting an attack. However, with the shorter imprint the possibility of a false alarm increasing. A way to improve performance of Rabin moving windows is to monitor only one side of the flow. The point here is that you can monitor only

site that was the initiator of connection, when there is suspicion that this site can send threatened packets, ignoring packets that are responding to packets from the initiator. The undoubted advantage of the use of Rabin fingerprints, resulting from their high efficiency, is possibility of using them not only in honeynet environments but also in production networks. Comparing the imprints can be used for both packet classification and for the signature manufacturing. In the case of Rabin windows set when a lot of imprints are associated with a single package, there is appropriate to set the repeat counter so as to they are associated with a set of imprints not a single imprint. Then Rabin imprint is becoming signature.

For detecting repeating sequences it is possible to use not only a method of imprints. Algorithm known as the LCS (Longest Common Substring) is algorithm used to search the longest common string between packages. A weak productivity of the algorithm and large amounts of signatures which he is producing and which later it is hard to manage are the problem connected with this method. This is caused by the fact that this algorithm compares all flows with all stored in the memory, without preliminary classification. A system based on this algorithm is suitable only for monitoring the space consisting of no more than a few individual IP addresses. To further streamline the process of creating an attack signature it can be used a method, which is based on both the Rabin imprints algorithm and the LCS. Rabin imprints are used to the initial classification of packets. On the basis of this classification anomalies is detected. Finished flows are grouped by their similarities in terms of Rabin. For example, the group can provide all the completed flows, which have 20% similar imprints. Additionally, you may check number of unique source IP addresses within a certain period of time, forming one group. After the initial classification and distribution of packets to a group, they are undergoing to further analysis, which deals LCS algorithm. In the group of similar packages are sophisticated the longest common bytes sequence. If there are compared summary contents from different flows one threat signature is being get. However, if there is a compared content of packages within a flow, then the signature is obtained exploit. This approach allows for the use of Rabin windows with small length, which increases the likelihood of searching common strings. In this case, the final signature algorithm determines a more flexible LCS, which increases the detected signature.

The quality of generated signature is irrelevant if it is used only for informational purposes for understanding the context of a particular phenomenon. In the case, when such signature is supposed to be used for the protection in the intrusion detection system, it is necessary to make sure that the signature won't block the legal traffic in the network. It is necessary to storage the move pool, as it is known that does not include flows related to the threats, and then check whether the using generated signature is not blocking legal traffic.

For finishing the entire process of producing signatures it is necessary to classify every newly created. You can perform this process by comparing the newly generated signature with the signatures already scheduled. You should check whether the produced signature already exists and whether she is classified, or she is a subset of the existing signature, and if no, whether is similar to some from existing. To streamline this process you should choose one signature (representative) from every class of ready signatures in order to compare the newly created signature

only with the representative of the class, rather than with every signature individually.

6 Knowledge base

An application, that is associated with this article, base on the database. Its diagram shows the Figure 1. In it are stored all rules and information about detected threats. Attack responding module has access to rules contained in knowledge base. When an attack is detected the information is saved in the database. This information is later used to create reports and gather information about the flaws in the protected system.

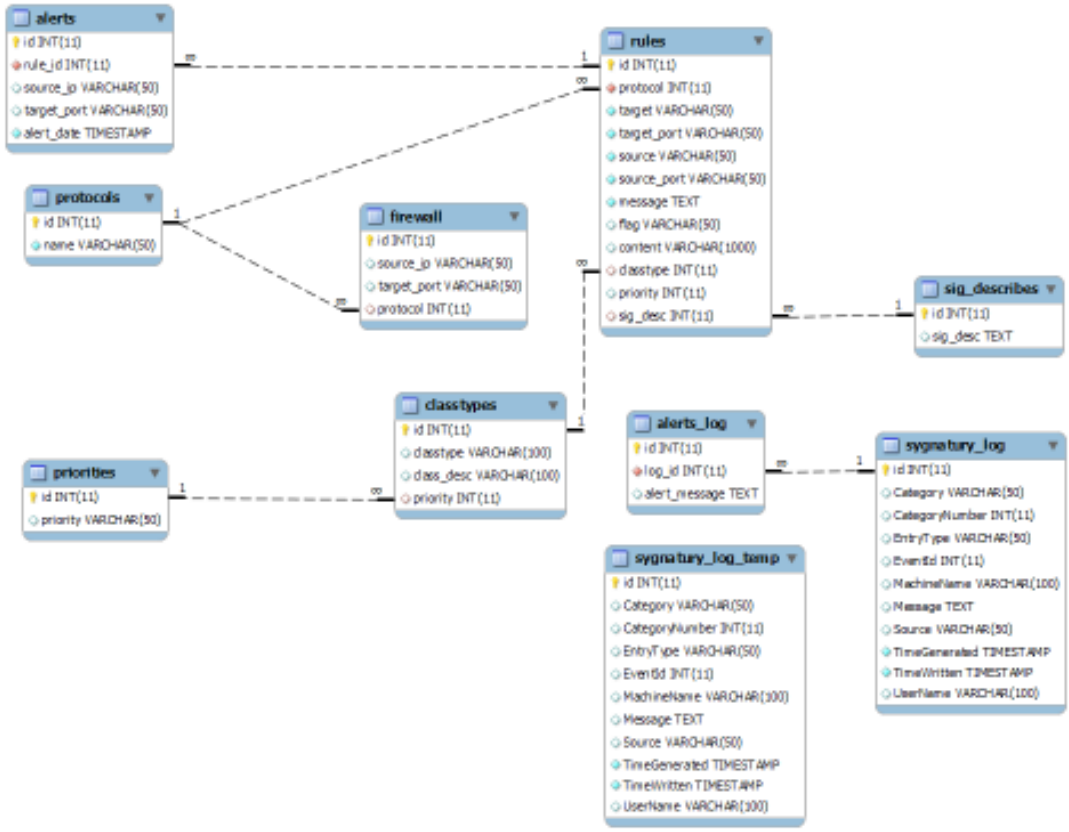

Figure 1. Knowledge base

The main elements of knowledge base are rules based on which attack response module detect and prevent intrusion attempts. An example rule might look like this:

```
TCP EXTERNAL_NET ANY --> HOME_NET ANY (msg:”ATTACK-RESPONSES directory listing” content:”Volume Serial Number” classtype:” Potential traffic error”)
```

This rule means that if TCP packet is detected, which came from an IP address belonging to the external network and is sent from any port and is directed to address

from home network range and is aimed at any port, then it will be checked for content. A fragment of rule, which is located in the brackets, is control information. In case of rule stored above, if text string saved in “content” field will be located in content of the package, then this packet will be blocked. Packages that in their content have a sequence identical to the sequence at value of the "content" field and have flags set such as at "flags" field, that occurs in some rules will be blocked. In cases where the rule is not taken into these fields, all packets meeting the first part of the rule (outside the brackets) will be blocked. Other values in parentheses are the information about attack, which is described by rule. The "msg" field is a value giving name of the attack or brief information about it. The "classtype" field assigns attack to one of the classes which enable determination of attack risk degree and quickly identify the type of threat. Additionally, you can set the "priority" field, by which you can set the priority of risk, which is different than assigned to a class. Each class of attacks has given priority threat.

Information about the attacks, which were detected by the response module are important element of the knowledge base. Thanks to correlate this information with rules describing known risks we can generate reports on which system administrator can gain valuable information about vulnerabilities of the protected system and take steps towards to improved security. The knowledge base provides users information about attack time, type, degree to which current attack threatening the system and displays description of the attack, thanks to which we can better understand it nature and threat that brings. In addition, for some attacks, administrator can obtain information about how you can remove the effects of the attack.

The structure of database used in the application is described below.

Alerts table.

It includes information about the attacks, which were detected by the system. Table fields:

Id – attack identifier,

rule_id – rule identifier, which describes attack,

source_ip – ip address of computer which from was sent dangerous package.

target_port – port number, to which was sent threatening package,

alert_date – date of detection dangerous package

Rules table

It contains the rules by which packets are classified as attacks or passed away. Table fields:

id – rule identifier,

protocol – protocol identifier,

target – destination ip address, on which package was sent.

target_port – destination port, on which package was sent.

source – source ip address, from which package was sent.

source_port – source port, from which package was sent.

message – attack name or brief information which identifies attack.

flag – flag field values in package, which may indicate risk.

content – text string in package, which may indicate risk.

classtype – attack class identifier.

Priority - risk priority (if set to 0 then is taken into account the priority assigned to the class)
Sig_desc – attack description identifier

If, after comparison rule with package values of protocol, target, target_port, source, source_port, flags and content column will agree then this package will be classified as an attack. Packet classification deals with the response module.

Protocols table
It includes names of protocols. Table fields:
Id – protocol identifier
Name – protocol name

Firewall table
It includes firewall blocking rules. If incoming packet after compare with rule meet all its condition then it is immediately blocked. Table fields:
Id – rule identifier
Source_ip – source address of package to be blocked
Target_port – numbers of ports to be blocked for incoming packets from the specified address,
Protocol – protocol identifier

Classtype table
It includes classes, on which divided the identified attacks. Table fields:
id – class identifier,
classtype – class name,
class_desc – class description,
priority – priority identifier.

Priorities table
Including the priorities assigned to classes. Table fields:
id – priority identifier.
priority – priority name.

sig_describes table
It includes descriptions of attacks described by the rules. Table fields:
id – description identifier,
sig_desc – attack description

alerts_log table
It include information about irregularities detected in the system logs. Table fields:
id – alarm identifier
log_id – log identifier
alert_message – message assigned to the alarm.

Sygnatury_log table
It includes information about irregularities detected in the logs that may indicate an attack. If this information is repeated and response module will classified they as a threat then information about this is saved to the table alerts_log. Table fields:
id – log identifier
Category – log category
CategoryNumber – log category number
EntryType – type of entry
EventId – event identifier

MachineName – computer name
Message – message
Source – log source
TimeGenerated – time to generate
TimeWritten – time to write to log diary
UserName – user name

Sygnatury_log_temp table

It contains supporting data for attack response module. Table fields are the same as in table sygnatury_log.

7 Conclusion

The knowledge base is an integral part of intrusion detection. It includes both the information needed for intrusion detection system, and administrators to better understand the existing threats and better secure protected system. Taking into consideration fact that the automatic detection of intrusion into computer systems is a relatively young field of computer science, it should be remembered that that it is still developed area. More and more organizations are specializing in describing known gaps, creating signatures and supplementing knowledge bases available on the market of intrusion detection. The most popular free license system is Snort. It is worth to study their documentation in order to better understand the operation of such systems. Additionally it is possible to see how signatures of known threats looks what can interest the user wants to write own signature of threats detected by himself. However, please note that not every intrusion detection system provides mechanisms for self-definition signatures and not every system uses the same language descriptions of signature as Snort.

References

1. Amoroso Edward: *Sieci: Wykrywanie intruzów,* Wydawnictwo RM, Warszawa 1999, ISBN 83-7243-039-X
2. Kijewski Piotr: *Metody automatycznego wytwarzania sygnatur zagrożeń sieciowych*, CERT Polska SECURE Conference, Październik 2005
3. Lukatsky Alex: *Wykrywanie włamań i aktywna ochrona danych,* Wydawnictwo Helion, Gliwice 2005, ISBN 83-7361-666-7