IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 16, Issue 1, Ver. IV (Jan. 2014), PP 81-90
www.iosrjournals.org

Design and Development of Web-based Expert System to Detect Network Intrusion

Asghaiyer Mohamed Asghaiyer, Purnomo Budi Santoso, Harry Soekotjo Dachlan
(Dept. Communication and Information System, Brawijaya University, Indonesia)

Abstract: *It is very important that the security mechanisms of a system are designed so as to prevent unauthorized access to system resources and data. However, completely preventing breaches of security appear, at present, unrealistic. We can, however, try to detect these intrusion attempts so that action may be taken to repair the damage later. Expert system (ES) has been proposed to be the beter solution of reasoning on the existing problems on the basis of expert knowledge.*

Experts 'brain' are stored in a database called a knowledge base and then made the rules to give decisions in accordance with the knowledge base.In this research, knowledge base taken from the expert and journal. Expert system will be used to determine or predict the occurrence of intrusion detection systems based on the feature above. By simple means, any data packet is going to be examined by ES, which initially stores knowledge about intrusion characteristics from database as variables. The result of this research is anomaly packet already detected by the application according to the rule based on the expert and the journal.

Keywords: *anomaly packet, expert system, intrusion, knowledge base, network packet.*

I. INTRODUCTION

In the last three years, networking revolution has finally come of age. More than ever before, we see that the Internet is changing computing as we know it. More over, today, most people are connected to internet in nearly 24 hours a day. The possibilities and opportunities are limitless; unfortunately, so too are the risks and chances of malicious intrusions. A computer system should provide confidentiality, integrity and assurance against denial of service. However, due to increased connectivity (especially on the Internet), and the vast spectrum of financial possibilities that are opening up, more and more systems are subject to attack by intruders. These subversion attempts try to exploit flaws in the operating system as well as in application programs and have resulted in spectacular incidents like the Internet Worm incident of 1988.

Intrusion is something that may harm computers which connected to the network if it is being abandoned. Many kinds of attack comes from the vulnerability of network, of which without proper intrusion detection system. If a network has a good intrusion detection system, it has the capability to prevent any future damage or theft. Each data packets is grabbed by ES to compare its characteristics with the variables (intrusion characteristics), and the result is a decision taken by ES whether to hold the data packet or pass it to client.The scheduling method to grab-compare-decide by expert system should be able to make sure that all data packets are checked without any significant time consumption.

II. LITERATURE REVIEW

2.1 Definition of Expert System

Abraham (2005), stated that the basic components of an expert system are illustrated in Figure 2.1. The knowledge base stores all relevant information, data, rules, cases, and relationships used by the expert system. A knowledge base can combine the knowledge of multiple human experts. A rule is a conditional statement that links given conditions to actions or outcomes. A frame is another approach used to capture and store knowledge in a knowledge base. It relates an object or item to various facts or values. A frame-based representation is ideally suited for object-oriented programming techniques. Expert systems making use of frames to store knowledge are also called frame-based expert systems.

2.2 Knowledge Base

The highly specialized knowledge of the problem area is located in the knowledge base. This module contains the problem facts, rules, concepts, and relationships. The first step to build the knowledge base is to gather the knowledge from human experts, which should be stored in the knowledge base. After this step the knowledge must be coded in a form which is useful for automatic processing, which is subject of the knowledge representation. There are several techniques as to how to store the knowledge. As an example we could look at a rule-based representation. A rule is a if/then structure that logically relates information contained in the 'if' part of the rule to information in the 'then' part. An example above shows a sample rule-based knowledge representation.

2.3 Working Memory

The working memory contains the facts about a problem that are collected during one consultation of the expert system. When a new problem has to be solved, the user enters information about the problem into the working memory. The expert system then uses this information together with the facts in the knowledge base to infer new facts. These new facts are also stored in the working memory. The content of the working memory can also consist of facts that have been collected from external storage like databases, spreadsheets, or sensors, beside the information that is taken from the knowledge base.

2.4 Inference Engine

In logic, a rule of inference, inference rule, or transformation rule is the act of drawing a conclusion based on the form ofpremises interpreted as a function which takes premises, analyzes their syntax, and returns a conclusion (or conclusions). For example, the rule of inference modus ponens takes two premises, one in the form of "If p then q" and another in the form of "p" and returns the conclusion "q". The rule is valid with respect to the semantics of classical logic (as well as the semantics of many other non-classical logics), in the sense that if the premises are true (under an interpretation) then so is the conclusion. Typically, a rule of inference preserves truth, a semantic property. In many-valued logic, it preserves a general designation. But a rule of inference's action is purely syntactic, and does not need to preserve any semantic property: any function from sets of formulae to formulae counts as a rule of inference. Usually only rules that are recursive are important; i.e. rules such that there is an effective procedure for determining whether any given formula is the conclusion of a given set of formulae according to the rule.

2.5 Characteristics of Expert Systems

The most important characteristic of any expert system is that the knowledge is separated from control logic. This distinguishes an expert systems from a conventional program, where it is often not possible to add knowledge without changing the source code of the program at the same time. The tasks of modifying and maintaining the system are easier with this separation. Another important feature of an expert system is that it embodies the expertise of a human expert. The knowledge of such an expert is used as well as his problem-solving skills. Because expert systems focuses on knowledge of a clearly defined domain they poorly perform outside their area of expertise. At each time in an interaction between the system and the user it should be possible for the user to ask the systemhowit has reached the actual conclusion. This is important because an expert system does not have a well-defined structure like a conventional program that makes it easy to understand what the system is doing. Lets look at an example :

Expert System: The battery is bad.
Person: How?
Expert System: Since your car won’t start,
I assumed there was a problem with the electrical system.
Once I found the battery voltage was below 10 volts,
I knew that the battery was bad.

An expert system can also explain why it is asking a specific question. The explanation of the system makes the user feel more comfortable because he can reproduce why the system is asking this question. Lets look at an example :

Expert System: Will the car not start?
Person: Why?
Expert System: If I know that the car won’t start,
then I usually assume the problem is in the electrical system.

2.6 Building Expert Systems

The development of an expert system is a highly iterative process. Most of the time it is not possible to design first the whole system and then just implement the design. Instead it is needed to build a first prototype, then use this prototype to get more knowledge from the human experts and re-implement the prototype. This process is repeated until there is a final system that is able to do what the users are expecting it to do.

2.6.1 Knowledge Engineering

One major part is the Knowledge Engineering. Usually there are human experts who know much about their specific domain, and application developers who have a lot of knowledge in programming but don’t know much about the domain where the expert system is used in the future. In the process of Knowledge Engineeringboth groups of people are involved and try to find an efficient way as to how to encode the human experts knowledge in the expert system.

2.6.1.1 Assessment

During the assessment phase, studies are made to determine the feasibility and justification of the candidate problem. After this process the overall goals of the project have to be defined. These goals together with the information gathered by the process of Knowledge Engineeringare then used to identify the sources of needed knowledge.

2.6.1.2 Knowledge Acquisition

Knowledge Acquisitionis the process of acquiring, organizing, and studying the knowledge from human experts. In the early stage of expert system development the objective of this step is to uncover key concepts and general problem-solving methods used by the expert. In a later stage the results of the testing are used to explore for more detailed information. Knowledge Acquisitionhast long been recognized as the bottleneck in expert system development.

2.6.1.3 Design

After the knowledge acquisition, the design phase is used to define the overall structure and organization of the system's knowledge. Methods are defined for processing the knowledge and the appropriate software tool is chosen. After this step an initial prototype system is built, which serves as the focal point for further interviews with the human experts. As mentioned above this task is repeated several times until a final design is reached

2.6.1.4 Testing

The Testing phase is not a separate task. It is rather a continuous process throughout the whole project. After every step in building the expert system, it is tested and new knowledge is added to it. These tests should not only be done by the programmers, but also by the end-users because it is very important that the end-users are able to use the system and that the system is well adapted to the user's needs.

2.6.1.5 Documentation

The Documentation addresses the need to compile all of the project's information into a document. The Documentation must also support the knowledge engineer during the development of the system. It should contain a knowledge dictionary that provides a well organized representation of the system's knowledge and the included problem solving procedures.

2.6.1.6 Maintenance

Because of the highly iterative process in developing an expert system, it is also important to periodically maintain the system. The system's knowledge may need to be refined or updated to adapt the system to the actual circumstances. If the knowledge is separated from the control, this process is more flexible and easier as if the whole knowledge is included in the logic.

2.6.2 Knowledge Representation

Knowledge is represented in some symbolic form that can be manipulated by the expert system and by the users of the expert system. There exist many different forms of knowledge representation. The knowledge engineer must choose the knowledge representation technique best suited for the given application. Some of the most important types of knowledge are presented in the following subsection.

2.6.2.1 Types of Knowledge

Knowledge can be divide into the following types:

a. Procedural knowledgedescribes how a problem is solved, e.g. rules, strategies, and procedures.
b. Declarative knowledgedescribes what is known about a problem, e.g. concepts and objects.
c. Meta-knowledgedescribes knowledge about the knowledge.
d. Heuristic knowledgedescribes a rule-of-thumb.
e. Structural knowledgedescribes knowledge structures, e.g. rule sets and rela-tionships of concept.

It is important to use an appropriate type of knowledge for a specific expert system application, because only a well adjusted structure is able to support effective problem solving. The next subsection describes different knowledge representation techniques.

2.7 Rules

In logic, a rule of inference, inference rule, or transformation rule is the act of drawing a conclusion based on the form ofpremises interpreted as a function which takes premises, analyzes their syntax, and returns a conclusion (or conclusions). For example, the rule of inference modus ponens takes two premises, one in the form of "If p then q" and another in the form of "p" and returns the conclusion "q". The rule is valid with respect to the semantics of classical logic (as well as the semantics of many other non-classical logics), in the sense that if the premises are true (under an interpretation) then so is the conclusion.

2.8 Forward Chaining

Forward chaining starts with the available data and uses inference rules to extract more data (from an end user for example) until a goal is reached. An inference engine using forward chaining searches the inference rules until it finds one where the antecedent (If clause) is known to be true. When found it can conclude, or infer, the consequent (Then clause), resulting in the addition of new information to its data.

III. METHODOLOGY

In this research, rule-based system is chosen because it has the advantages of using less time to perform the detection. The fast detection can be performed due to that the computer is not ‘thinking’ about another algorithm than just passing data packets to its rules, checking each data packet to sample of network intrusion, of which stored in memory, and lastly, decision is made after a data packet is done checked.

3.1 Application Flowchart

Figure 3.1 is the flowchart for the network intrusion detection using expert system.

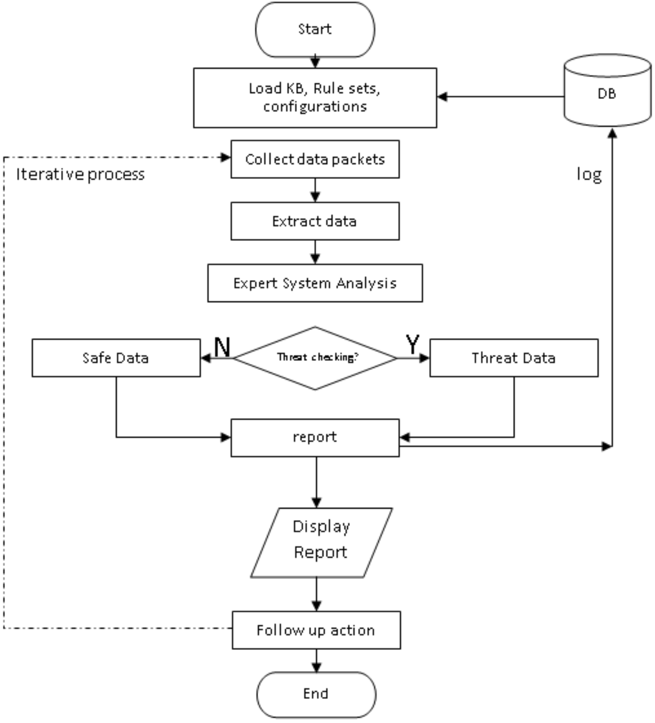


Figure 3.1 NIDS using Expert System

Figure 3.1 shows process that are performed in the server, when it started, the application will initially load all necessary information from database. The information including Knowledge Base, Rule sets, and other configurations related to run the Expert System to detect Network Intrusion. The knowledge base contains specific feature of different network instrusion behaviour, which taken from MySQL Database’s record about knowledge base and stored as variables of the web application, meanwhile the rule set is the rules that must be passsed by realtime data packets. The rule sets are also taken from MySQL Database’s record about rule sets and stored to structure the application. These information are kept in Memory as variables or class’ objects with its properties, in order to minimize delay during runtime process. After initialization has completed, now server mode is in runtime where it continuously collects data packets, then extracts its features. The extracted features is then analyzed by using Expert system that resides in the memory during the initial process. And any result of data packet process is reported as threats or safe data. Reports are stored in the database for future analysis, display to user, and follow ups, either to block similar data packets or not.

When a threat can not be identified by the expert system, the application will automatically put the type of packet data into the label "warning" with the type of intrusion is “blank”.

3.2 User Design Interface

User interface need to be developed to ensure that system can be clearly understand the usage and appearance.

3.2.1 Login System

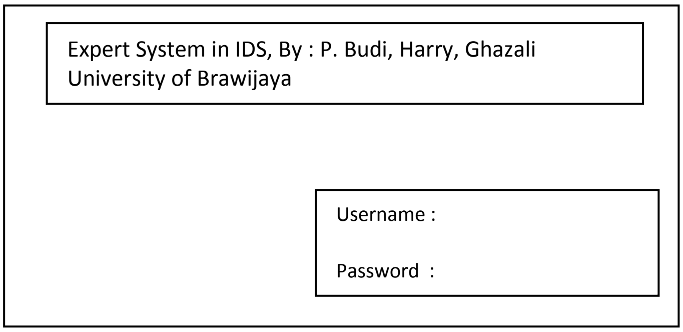


Figure 3.2 Login System

3.2.2 View Packet Data

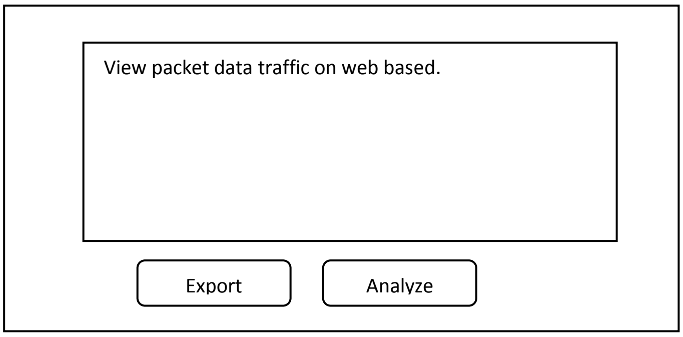


Figure 3.3 Packet Data Viewer

3.3 Database Design

In this study, used 1 database and 2 tables for storing information associated with the log results in real-time network packet capture and store data about patterns of information from each feature IDS. Information stored in the database can be changed later if needed.

This study uses a MySQL database created by the organization opensource and free for both individual and commercial use. Figure 3.4 is a view of the MySQL database that already has a database with the name "ids".

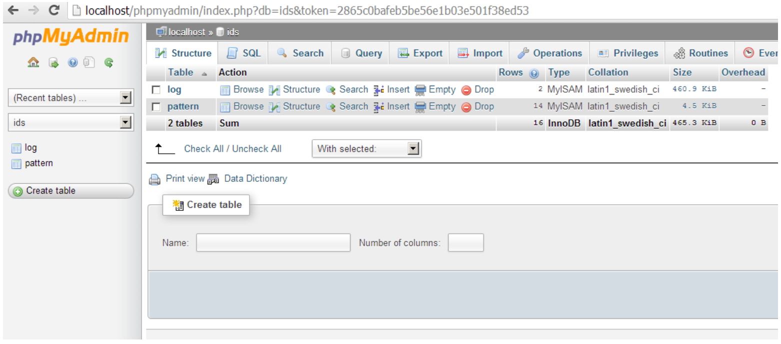


Table	Action	Rows	Type	Collation	Size	Overhead
log	Browse Structure Search Insert Empty Drop	2	MyISAM	latin1_swedish_ci	460.9 KiB	-
pattern	Browse Structure Search Insert Empty Drop	14	MyISAM	latin1_swedish_ci	4.5 KiB	-
2 tables	Sum	16	InnoDB	latin1_swedish_ci	465.3 KiB	0 B

Figure 3.4 Display MySql database containing ids database

From Figure 3.4 shows that the database "ids" has 2 tables, namely the log table and the table pattern. The following will explain in detail the contents of the log table and chart pattern.

Table 3.1 The structure of table "log"

Field Name	Field Type	Size	Description
Id_packet	Integer	11	Primary Key
packet	Text	11	Value of pattern
time	Text		Time of packet arrived and saved into table

Log table has several fields, namely id_packet which serves to store the unique identity of each row of data, field data from the packet containing the captured packets by application, and field time is the time when the packet is received in the database.

To create a design table like Table 3.1, done at the command prompt windows system, as for the creation of application code table can be seen in Figure 3.5.

```
CREATE TABLE `log` (
`id_packet` INT( 3 ) NOT NULL AUTO_INCREMENT ,
`packet` LONGBLOB NOT NULL ,
`time` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP ,
PRIMARY KEY ( `id_packet` )
) ENGINE = MYISAM DEFAULT CHARSET = latin1;
```

Figure 3.5 Source code of Creating table “log”

Table 3.2 Index of table “log”

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	id_pattern	14	A	No	

Space usage:

Data	2,536	B
Index	2,048	B
Total	4,584	B

Row Statistics:

Format	dynamic
Rows	14
Row length ø	181
Row size ø	327 B
Next autoindex	19
Creation	Dec 27, 2013 at 03:58 PM
Last update	Dec 28, 2013 at 10:30 AM

As for storing the information about the list and the type of intrusion, made table pattern. This table contains information about the type and pattern of the intrusion. table pattern can be seen in the table below 3.3.

Table 3.3 The structure of table “pattern”

Field	Type	Length/Values	Description
Id_pattern	Integer	11	Primary Key
category	Text		Category of intrusion Name of intrusion
name	Text		Information about pattern
pattern	Text		Filename of Application
status	Text		

To create a design table like Table 3.3, done at the command prompt windows system, as for the creation of application code table can be seen in Figure 3.6.

```
CREATE TABLE `pattern` (
`id_pattern` INT( 11 ) NOT NULL AUTO_INCREMENT ,
`category` TEXT NOT NULL ,
`name` TEXT NOT NULL ,
`pattern` TEXT NOT NULL ,
`status` VARCHAR( 100 ) NOT NULL ,
PRIMARY KEY ( `id_pattern` )
) ENGINE = MYISAM DEFAULT CHARSET = latin1;
```

Figure 3.6 Source code of Creating table “pattern”

Table pattern consists of several fields, namely: id_pattern that serves as a unique identifier of each row of data, category field which will store data about the categories of a intrusion which does include a probe or a denial of service or other. name field contains information about the name instrusion, filed pattern is complete information about the pattern of intrusion which according to its category, and the last field is the actual status of containing the file name of the application that corresponds to the existing categories.

Table 3.4 Index of table “pattern”

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	id_pattern	14	A	No	

Space usage:

Data	2,536	B
Index	2,048	B
Total	4,584	B

Row Statistics:

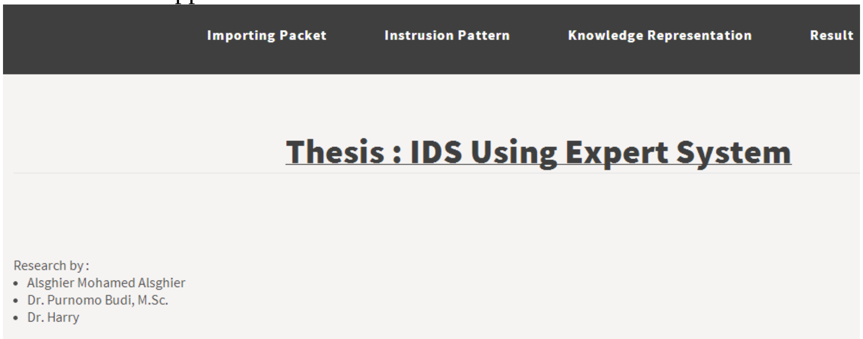
Format	dynamic
Rows	14
Row length ø	181
Row size ø	327 B
Next autoindex	19
Creation	Dec 27, 2013 at 03:58 PM
Last update	Dec 28, 2013 at 10:30 AM

IV. IMPLEMENTATION AND RESULT

The tools used in this implementation are php designer and Xampp. Php designer is an IDE for php developer to arrange coding of programming. Xampp is a application bundling that consists of php, mysql, and mail server. MySql is a database engine that stored knowledge base and inference engine logic. These are implementation results of design interface :

4.1 Top menu / home menu

The main menu contains some information, including title of research, researcher name, and displays hyperlinks to other menus like the menu for editing the knowledge base, running applications and the menu contains information about the application.



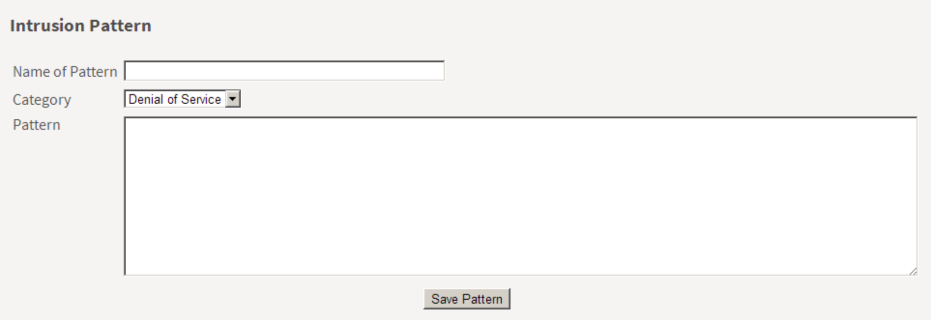
Thesis : IDS Using Expert System

Research by :
- Alsghier Mohamed Alsghier
- Dr. Purnomo Budi, M.Sc.
- Dr. Harry

Figure 4.1 Main Menu

4.2 Maintaining the Rule

This menu provides an easy way to manage rule base on the application. Admin can provide additional information network pattern behaviour, information and also the category that are used.


Intrusion Pattern

Name of Pattern

Category: Denial of Service

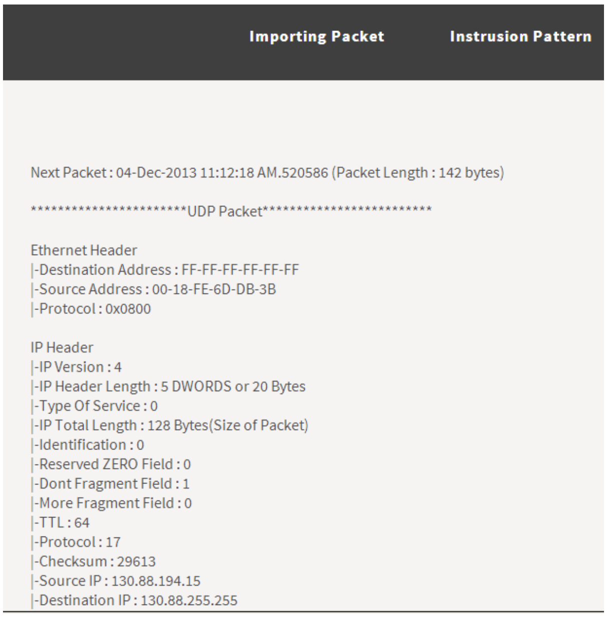
Pattern

Figure 4.2 Rule Base Configurations

4.3 Inputting the Facts

4.3.1 Realtime Packet Data

In this section, application provides an easy way to the admin to import realtime packet data.

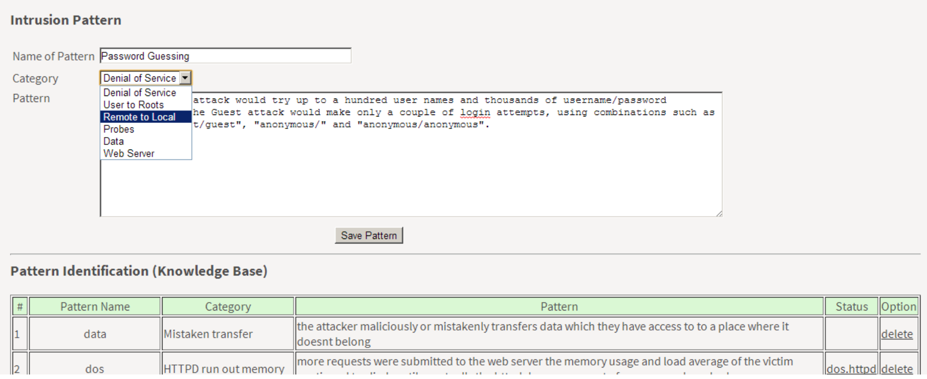


```
Next Packet : 04-Dec-2013 11:12:18 AM.520586 (Packet Length : 142 bytes)

**********************UDP Packet************************

Ethernet Header
|-Destination Address : FF-FF-FF-FF-FF-FF
|-Source Address : 00-18-FE-6D-DB-3B
|-Protocol : 0x0800

IP Header
|-IP Version : 4
|-IP Header Length : 5 DWORDS or 20 Bytes
|-Type Of Service : 0
|-IP Total Length : 128 Bytes(Size of Packet)
|-Identification : 0
|-Reserved ZERO Field : 0
|-Dont Fragment Field : 1
|-More Fragment Field : 0
|-TTL : 64
|-Protocol : 17
|-Checksum : 29613
|-Source IP : 130.88.194.15
|-Destination IP : 130.88.255.255
```

Figure 4.3 Realtime Packet Captured

4.3.2 Pattern Management

In this section, application provides an easy way to the admin to make changes to the pattern , category of pattern, and information about related pattern.



Intrusion Pattern

Name of Pattern: Password Guessing

Category: Denial of Service

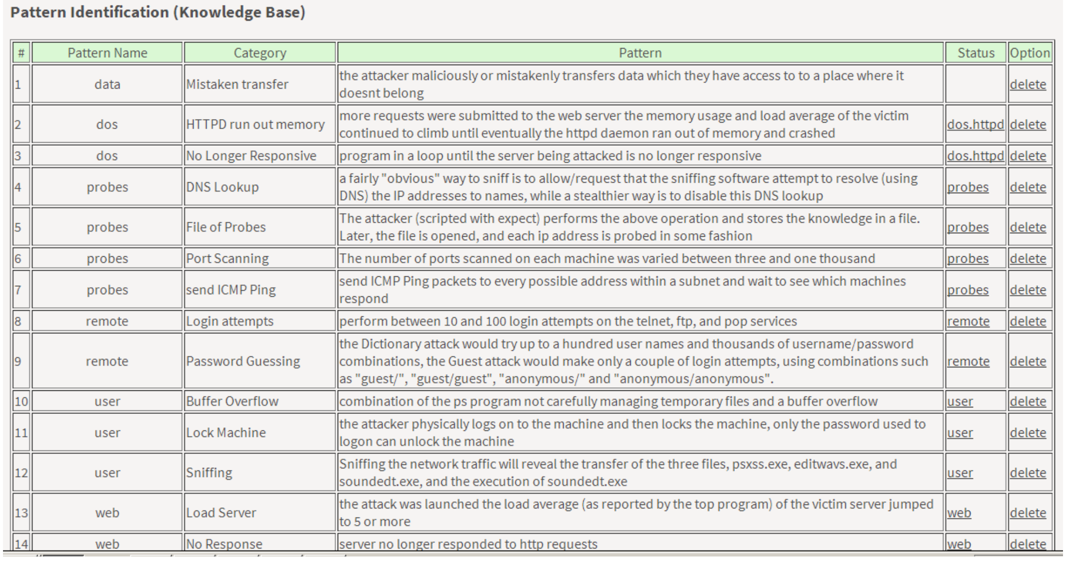
Pattern:

```
attack would try up to a hundred user names and thousands of username/password
he Guest attack would make only a couple of login attempts, using combinations such as
t/guest", "anonymous/" and "anonymous/anonymous".
```

Pattern Identification (Knowledge Base)

#	Pattern Name	Category	Pattern	Status	Option
1	data	Mistaken transfer	the attacker maliciously or mistakenly transfers data which they have access to to a place where it doesnt belong		delete
2	dos	HTTPD run out memory	more requests were submitted to the web server the memory usage and load average of the victim	dos.httpd	delete

Figure 4.4 Pattern Management



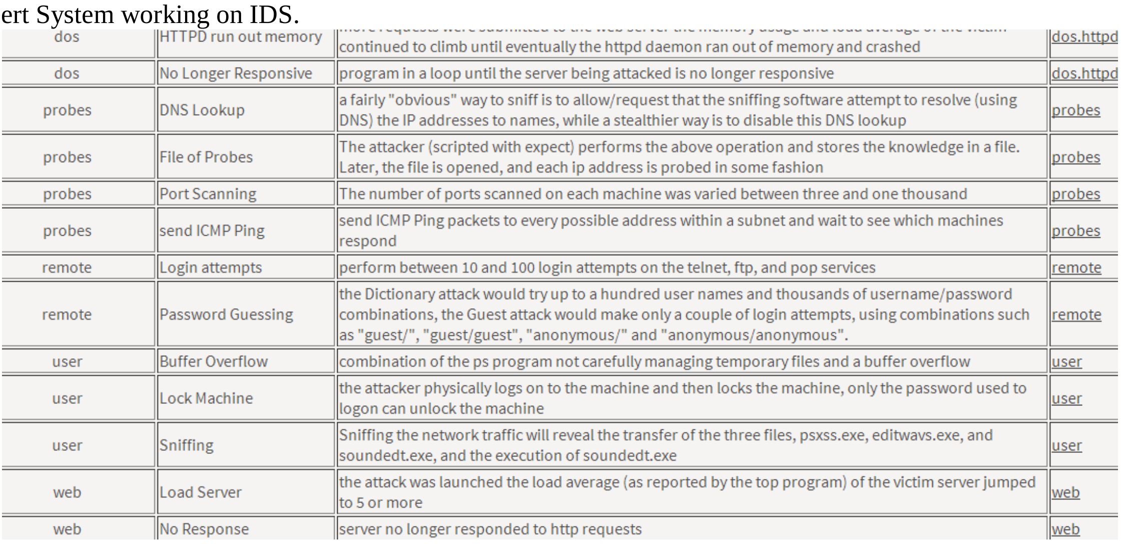
Pattern Identification (Knowledge Base)

#	Pattern Name	Category	Pattern	Status	Option
1	data	Mistaken transfer	the attacker maliciously or mistakenly transfers data which they have access to to a place where it doesnt belong		delete
2	dos	HTTPD run out memory	more requests were submitted to the web server the memory usage and load average of the victim continued to climb until eventually the httpd daemon ran out of memory and crashed	dos.httpd	delete
3	dos	No Longer Responsive	program in a loop until the server being attacked is no longer responsive	dos.httpd	delete
4	probes	DNS Lookup	a fairly "obvious" way to sniff is to allow/request that the sniffing software attempt to resolve (using DNS) the IP addresses to names, while a stealthier way is to disable this DNS lookup	probes	delete
5	probes	File of Probes	The attacker (scripted with expect) performs the above operation and stores the knowledge in a file. Later, the file is opened, and each ip address is probed in some fashion	probes	delete
6	probes	Port Scanning	The number of ports scanned on each machine was varied between three and one thousand	probes	delete
7	probes	send ICMP Ping	send ICMP Ping packets to every possible address within a subnet and wait to see which machines respond	probes	delete
8	remote	Login attempts	perform between 10 and 100 login attempts on the telnet, ftp, and pop services	remote	delete
9	remote	Password Guessing	the Dictionary attack would try up to a hundred user names and thousands of username/password combinations, the Guest attack would make only a couple of login attempts, using combinations such as "guest/", "guest/guest", "anonymous/" and "anonymous/anonymous".	remote	delete
10	user	Buffer Overflow	combination of the ps program not carefully managing temporary files and a buffer overflow	user	delete
11	user	Lock Machine	the attacker physically logs on to the machine and then locks the machine, only the password used to logon can unlock the machine	user	delete
12	user	Sniffing	Sniffing the network traffic will reveal the transfer of the three files, psxss.exe, editwavs.exe, and soundedt.exe, and the execution of soundedt.exe	user	delete
13	web	Load Server	the attack was launched the load average (as reported by the top program) of the victim server jumped to 5 or more	web	delete
14	web	No Response	server no longer responded to http requests	web	delete

Figure 4.5 List of Pattern

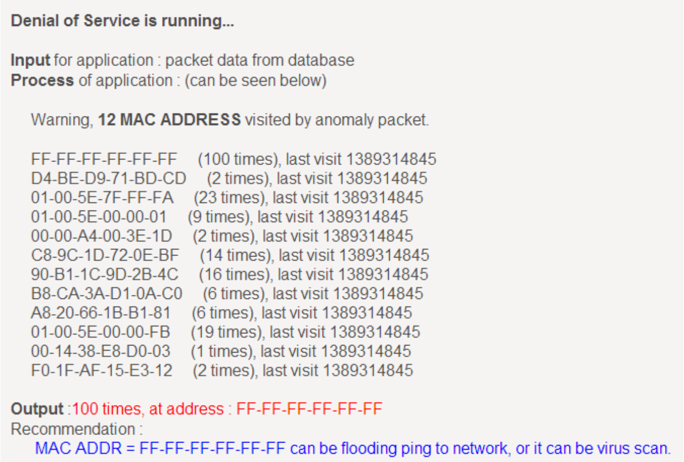
4.4 Running the Expert System

This menu is the center of this research, namely to make the system running and try the application of Expert System working on IDS.



dos	HTTPD run out memory	more requests were submitted to the web server the memory usage and load average of the victim continued to climb until eventually the httpd daemon ran out of memory and crashed	dos.httpd
dos	No Longer Responsive	program in a loop until the server being attacked is no longer responsive	dos.httpd
probes	DNS Lookup	a fairly "obvious" way to sniff is to allow/request that the sniffing software attempt to resolve (using DNS) the IP addresses to names, while a stealthier way is to disable this DNS lookup	probes
probes	File of Probes	The attacker (scripted with expect) performs the above operation and stores the knowledge in a file. Later, the file is opened, and each ip address is probed in some fashion	probes
probes	Port Scanning	The number of ports scanned on each machine was varied between three and one thousand	probes
probes	send ICMP Ping	send ICMP Ping packets to every possible address within a subnet and wait to see which machines respond	probes
remote	Login attempts	perform between 10 and 100 login attempts on the telnet, ftp, and pop services	remote
remote	Password Guessing	the Dictionary attack would try up to a hundred user names and thousands of username/password combinations, the Guest attack would make only a couple of login attempts, using combinations such as "guest/", "guest/guest", "anonymous/" and "anonymous/anonymous".	remote
user	Buffer Overflow	combination of the ps program not carefully managing temporary files and a buffer overflow	user
user	Lock Machine	the attacker physically logs on to the machine and then locks the machine, only the password used to logon can unlock the machine	user
user	Sniffing	Sniffing the network traffic will reveal the transfer of the three files, psxss.exe, editwavs.exe, and soundedt.exe, and the execution of soundedt.exe	user
web	Load Server	the attack was launched the load average (as reported by the top program) of the victim server jumped to 5 or more	web
web	No Response	server no longer responded to http requests	web

Figure 4.5 Running the Expert System Home



```
Denial of Service is running...

Input for application : packet data from database
Process of application : (can be seen below)

    Warning, 12 MAC ADDRESS visited by anomaly packet.

    FF-FF-FF-FF-FF-FF    (100 times), last visit 1389314845
    D4-BE-D9-71-BD-CD    (2 times), last visit 1389314845
    01-00-5E-7F-FF-FA    (23 times), last visit 1389314845
    01-00-5E-00-00-01    (9 times), last visit 1389314845
    00-00-A4-00-3E-1D    (2 times), last visit 1389314845
    C8-9C-1D-72-0E-BF    (14 times), last visit 1389314845
    90-B1-1C-9D-2B-4C    (16 times), last visit 1389314845
    B8-CA-3A-D1-0A-C0    (6 times), last visit 1389314845
    A8-20-66-1B-B1-81    (6 times), last visit 1389314845
    01-00-5E-00-00-FB    (19 times), last visit 1389314845
    00-14-38-E8-D0-03    (1 times), last visit 1389314845
    F0-1F-AF-15-E3-12    (2 times), last visit 1389314845

Output :100 times, at address : FF-FF-FF-FF-FF-FF
Recommendation :
    MAC ADDR = FF-FF-FF-FF-FF-FF can be flooding ping to network, or it can be virus scan.
```

Figure 4.6 Running of dos.httpd.php file

V. Conclusion

The use of expert systems in the field of intrusion detection systems are still not widely used in the field of computer networks, but some researchers have tried but it still does not give good results. Design expert system has been created and validated by experts in the field of computer networks. After the design has been created and feasible for use, the next activity is to implement the design into a computer program and generate an expert system application in accordance with the purpose of research. Testing and validation conducted by researchers and accompanied by experts, experts from the field of computer networking is the center of information technology. The results obtained do not vary much with the results of calculations performed by the computer.

References

[1]. Abraham, Ajith. 2005. Rule-based Expert System. Oklahoma State University, OK, USA.

[2]. Donald, W.A. 1986. A Guide to Expert Systems, Addison- Wesley, Boston, MA.

[3]. Faloutsos, M; Faloutsos, P; and Faloutsos, C. On power-law relationships of the internet topology. In Proceedings of ACM SIGCOMM, 1999.

[4]. Goodson, James. 1990. The Development of An Expert System for Software Cost Estimation. Air Force Institute of Technology, USA.

[5]. Kelly, James. 2006. Master Thesis : An Examination of Pattern Matching Algorithms for Intrusion Detection System. Carleton University, Canada.

[6]. Kozushko, H. 2003. Intrusion Detection: Host-Based and Network-Based Intrusion Detection Systems. Independent Study.

[7]. Krishnan, Sharaynee. 2008. Master Thesis : Development of An Expert System For Welding Process Verification Using MS Visual Basic. Universiti Teknikal Malaysia Melaka, Malaysia.

[8]. Niemela, Antti. 2010. Master Thesis : Traffic Analysis for Instrusion Detection in Telecommunications Networks. Tampere University of Technology.

[9]. Ortiz, Agustin. 1988. Master Thesis : Development and Implementation of an Expert System for remotely Accessing a Relational Database. University of Florida, USA.

[10]. Pfenning, Frank. 2006. Logic Programming : Forward Chaining. Lecture.

[11]. R. M. Colomb. 2007. Ontology and the Semantic Web. Amsterdam, NLD: IOS Press, 2007.

[12]. Rouse, Margaret. 2005. http://whatis.techtarget.com/definition/framework. Accessed at 23 March 2013.

[13]. School of Computer Science. 2007. Object-Oriented Programming. University of Kwazulu-Natal.

[14]. Strittmatter, Marcel. 2003. Network Troubleshooting Expert System. Swiss Federal Institute of Technology Zurich, Swiss.

[15]. Symantec Website. Network intrusion Detection Signatures, part 1, 2, and 3. Accessed at 15 July 2013 from http://www.symantec.com/connect/articles/network-intrusion-detection-signatures-part-one, http://www.symantec.com/connect/articles/network-intrusion-detection-signatures-part-two, http://www.symantec.com/connect/articles/network-intrusion-detection-signatures-part-three.

[16]. Ullrich, J. MSSQL worm (sqlsnake) on the rise. Accessed from website: www.incidents.org/diary/diary:php?id=156, 2002.

[17]. Verhodubs, Oleg. 2011. Towards the Semantic Web Expert System. Scientific Journal of Riga Technical University.

[18]. Yegneswaran, V; Barford, P; Ulrich, J. 2003. Internet Intrusion: Global Characteristics and Prevalence. SIGMETRIC'03. ACM:San Diego, CA, USA.