A Secured Communication Based On Knowledge Engineering Technique

M. W. Youssef
Head of Computing & Information Division
The Shoura Assembly, Cairo, Egypt

Hazem El-Gendy
CS Dept. Chair, Faculty of CS & IT,
Ahram Canadian University

***Abstract*— Communication security has become the keynote of the "e" world. Industries like eComm, eGov were built on the technology of computer networks. Those industries cannot afford security breaches. This paper presents a methodology of securing computer communication based on identifying typical communication behavior of each system user based on the dominant set of protocols utilized between the network nodes.**

Keywords- Computer Communications; Computer/ communications Protocols; Network Security; Authentication; Scrambling; Encryption; Standard Protocols; ISO Open System Interconnections (OSI) Model; object behavior analysis; knowledge engineering.

I. INTRODUCTION

Modern Industries such as *eCom* (electronic commerce) and *eGov* (electronic Government) [1] depend heavily on computer networks security [1-16]. Security has been managed in many ways such as: authentication [2], securing networks communication using encryption and scrambling [3], protocol modification [4] and others used firewalls [5].

The TCP/IP has become the industry standard for computer networks protocols. The TCP/IP depends on the OSI model which consists of seven layers covering from physical layer to application layer. Each one of those layers utilizes several protocols in order to perform its function. When computer nodes communicate over the network, they use those protocols relevant to each layer.

Accordingly, various types of protocols services can be utilized in a network environment. HTTP pages, HTTPS, SNMP, FTP, SMTP, POP3, IMAP, DNS, SSH, TELNET, SSL, TCP, ICMP, SIP, UDP, Media Streaming and a range of other ports with a variety of services [6]. Each one of the previous protocols is associated with what is called a port number [7].

A port number is a way to identify a specific process to which a network message is to be forwarded when it arrives at a server. For the Transmission Control Protocol and the User Datagram Protocol, a port number is a 16-bit integer that is put in the header appended to a message unit. This port number is passed logically between server transport layers and physically between the transport layer and the Internet Protocol layer and forwarded on. Some services or processes have conventionally assigned permanent port numbers. These are known as well-known port numbers. In other cases, a port number is assigned temporarily (for the duration of the request and its completion) from a range of assigned port numbers. This is called an ephemeral port number.

The main hypothesis of this research is, each network user has a network utilization pattern, from that pattern, a user behavior can be detected then a user profile can be deduced. For example, some users tend to use only HTTP protocol even for their emails; others use SMTP protocol for emails, FTP for file transfer and HTTP for WEB sessions.

In order to do that a knowledge base [8] is created containing users behavior models. The knowledge base is generated from captured protocols generated from each session. That requires a Packet filtering mechanism [9] that is able to capture the traffic, identify traded protocols, apply statistical analysis process on those protocols [10], generate statistics that represent utilization patterns and store deuced profiles in the system knowledge base. Eventually, behavioral analysis programs run against the stored information in the knowledge base to generate a user network utilization behavior model.

The system is self-learning and has the ability to tune itself with every additional acquired bit of information. Accordingly, the system starts with zero knowledge, but as users establish more sessions its knowledge increase and overtime tune itself until it reaches a stable user behavior model.

After having a stable user behavior model, the system watches for behavior anomalies, as with each user session the system keeps an eye on users' behavior and comparing those behaviors to the existing knowledge base, for any reason, if a user behavior changed (anomaly detected), the system gives alerts to the system administrator that a user is doing something that is different from its normal behavior. The system administrator can then investigate the problem.

In this system, users are identified by their MAC addresses and protocols are identified from protocol identifier in packet headers at one stage and from port number at another.

The rest of this paper is organized as follows. Section two presents the structure of packet headers. Section three presents the relationship between protocols and port numbers. Section four discusses packet filtering techniques. Section five presents the design of the knowledge engineering security system and finally, section six is the paper conclusion and proposed future work.

II. TCP/IP HEADER LAYOUTS

The TCP/IP header consists of several fields, each of which has a specific function in data communication. The layout of the TCP/IP header is presented in figure 1. An important field in TCP/IP header is protocol identifier field [29]. This field will be used to build the system knowledge base.

The TCP/IP header Layout

00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Version				IHL				Differentiated Services								Total length															
Identification																Flags			Fragment offset											
TTL								Protocol								Header checksum															
Source IP address																															
Destination IP address																															
Options and padding :::																															

Figure 1: The TCP/IP header Layout

In the TCP/IP, every packet header contains some important information. The main information is:

- IP source address.
- IP destination address.
- Protocol (whether the packet is a TCP, UDP, or ICMP packet).
- TCP or UDP source port.
- TCP or UDP destination port.
- ICMP message type.

In addition, packet headers contain additional information regarding packets that aren't reflected in the packet headers, such as:

- The interface the packet arrives on.
- The interface the packet will go out on.

In this research, packet headers will be intercepted and protocol identifiers will be captured to be used for building the behavioural model in the knowledge base.

III. PROTOCOLS AND PORT NUMBERS

For each application-specific or process-specific protocol there is a port number serving as a communications endpoint. It is used by the transport protocols of the Internet Protocol Suite, such as Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). Servers for each protocol service use that given port number, that port number allows the identification of running protocol and types of connections simply by specifying the appropriate port number. Port numbers are divided into three ranges: well-known ports, the registered ports, and the dynamic or private ports. The well-known ports are those from 0 through 1023. Examples include: :File Transfer Protocol (FTP) port 21; Secure Shell (SSH) port 22; Telnet remote login service port 23; Simple Mail Transfer Protocol (SMTP) port 25; Domain Name System (DNS) service port 53; Hypertext Transfer Protocol (HTTP) used in the World Wide Web port 80; and so on. The registered ports are those from 1024 through 49151. IANA maintains the official list.[1] The dynamic or private ports are those from 49152 through 65535. One common use is for ephemeral ports.

Some examples of the ways in which port numbers are to selectively route packets to or from your site:

- Identify all incoming SMTP connections.
- Identify email and FTP services
- Identify dangerous services like TFTP, the X Window System, RPC, and the "r" services (*rlogin*, *rsh*, *rcp*, etc.).
- Identify all connections to or from certain systems.

In this research, port numbers will be used to identify services and applications protocols to be used for building the behavioural model in the knowledge base.

IV. PACKET FILTERING TECHNIQUES

Packet filtering systems route packets between internal and external hosts, in some cases, they do it selectively. They allow or block certain types of packets in a way that reflects a site's own policy. There are several types of packet filters, the most common are:

1. A screening router.
2. Proxy Services.
3. Using a Combination of Techniques and Technologies.

A. *A Screening Router To Do Packet Filtering.*

This type of routers is used in a packet filtering firewall which is known as a *screening router*. A screening router looks at packets more closely and in addition to determining whether or not it *can* route a packet towards its destination, a screening router also determines whether or not it *should*. "Should" or "should not" are determined by the site's security policy, which the screening router has been configured to enforce. [2]

B. *Proxy Services.*

Proxy services are specialized application or server programs that run on a firewall host: either a dual-homed host with an interface on the internal network and one on the external network, or some other bastion host that has access to the Internet and is accessible from the internal machines. These programs take users' requests for Internet services (such as FTP and Telnet) and forward them, as appropriate according to the site's security policy, to the actual services. The proxies provide replacement connections and act as gateways to the services. For this reason, proxies are sometimes known as *application-level gateways*.[3]

C. Using A Combination Of Techniques And Technologies.

Some protocols such as Telnet and SMTP can be more effectively handled with packet filtering. Others such as FTP, Archie, Gopher, and WWW are more effectively handled with proxies.

On the other hand, Proxy services are effective only when they're used in conjunction with a mechanism that restricts direct communications between the internal and external hosts. Dual-homed hosts and packet filtering are two such mechanisms. If internal hosts are able to communicate directly with external hosts, there is no need to use proxy services.

Accordingly, this research has used a combination of techniques in order to collect the required packets.

V. The Design Of The Knowledge Engineering Security System

Knowledge engineering as a tool has been used in this research due to its nature of being data driven rather than instruction driven which is required in this research state where data is the main actor in the system. A similar technique was used in [11, 12]. The presented system consists of two parts those parts are:

1) *The Knowledge Engine.*
2) *Anomalies Detector.*

The two modules are working continuously in conjunction in. The main design of the knowledge engine is presented in figure 2.

A. Design of the Knowledge Engine

The Knowledge engine components are:

1. Packet capturing module
2. Collected Packets Base.
3. Packets Classifier.
4. Classified Packets Base.
5. Behaviour analyser module (Knowledge Engine).
6. Knowledge Base (Users Behaviour Models).

1) Packet Capturing Module.

This main function of this module is to capture packets coming and going through the network. This module is used for both the system knowledge engine and the anomalies detector. For the first it stores the collected packets in the knowledge engine collected packets base and for the second it stores the collected packets in the anomalies detector collected packets base.

2) Collected Packets Base.

It contains the knowledge engine collected packets.

3) Packets Classifier.

This module is responsible for analysing packets at network layer to extract certain key information according to a set of criteria that a packet must match to be accepted in a trace buffer. There are two things used to make that sub module work:

Packet offset: it is used to define a location based on the start of the packet. [13]

Protocol offset: it takes into account that the packet analyser may be applied on networks that use different frame types. Its offset values are based on the start of data after the MAC header. [13]

An example of data offsets is presented in table 1.

By the end of the process, it stores classified packets in the classified packets base.

4) Classified Packets Base.

It contain all the system classified packets, it represents the first step in the knowledge building process.

5) Behaviour analyser module (Knowledge Engine).

The main function of this module is modelling users' behaviours. That is done by applying two methods. The first method is used to model the typical behaviour using finite state machines. The second method is used to model the signature of behaviours.

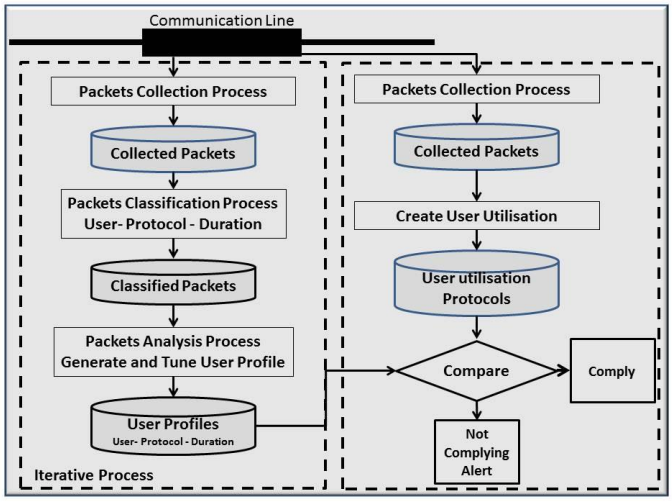

Figure 2: Design of the Knowledge Engineering Engine

Offset	Type	Value
0x00/0d	Packet (Ethernet)	Destination Hardware Address
0x06/6d	Packet (Ethernet)	Source Hardware Address
0x0C/12d	Protocol	Source IP Address
0x10/16d	Protocol	Destination IP Address
0x14/20d	Protocol	Source Port Number
0x16/22d	Protocol	Destination Port Number
0x21/33d	Protocol	Flags fields
0x1C/28d	Protocol	Data over UDP
0x28/40d	Protocol	Data over TCP

Table 1 Summarised data offset

Each module generates its specific instances to form user behaviour. Its attributes are:

- BehaviourNo: it contains an identifier value of a specific behaviour.
- BehaviourName: it contains name of the intended behaviour.
- BehaviourType: it contains type of the intended behaviour
- BehaviourPhase: it contains phase of the intended behaviour
- InitialCertainty: it contains the degree of certainty of the first instance of the intended behaviour.
- CertaintyIncrement: it contains the increment value added to the degree of certainty at each occurrence of the intended behaviour.

6) Knowledge Base (Users Behaviour Models).

All behavioural models are presented in the system knowledge base.

B. Design Anomalies Detector.

The anomalies detector components are:

1. Packet capturing module.
2. Collected Packets Base.
3. Session utilisation module:
    - Packets Classifier.
    - Classified Packets Base.
    - User utilisation profile generator (per session analyser module).
4. Anomalies detector.

1) Packet Capturing Module, Collected Packets Base and Session Utilisation Module.

The main objective of this phase is to build on the run spontaneous user behaviour to compare it the existing behaviours in the system knowledge base. Accordingly, in this phase, stages 1, 2 and 3 are very similar to what takes place in phase one, the knowledge engine phase, the difference is, it is done session by session or flow by flow based in the type of the captured protocol.

2) Anomalies Detector.

This module is responsible for detecting user behaviours anomalies and deducing actions towards detected behaviours. The anomalies detector module consists of two sub modules:

the rule base for behaviour comparison and the inference engine to decide actions against that behaviour. Both sub modules use a fact base of states of behaviour.

a) Inference engine rules base

This rule base is used to store already acquired knowledge in the domain of anomalies rather than the ordinary domain. This knowledge is represented through action (situation) and premise (conclusion) production rules.

b) Inference engine

This sub module makes forward-chaining (event driven) reasoning on events in the fact base and heuristics in the rule base to deduce consequences of detected behaviours and then their appropriate reactions to broadcast that to the system administrator. In addition, in ambiguous behaviour, the inference engine supports both of uncertainty and conflict resolution.

c) Fact base of States of Behaviours.

This fact base is used to store information on behaviours that are in progress or completely detected. Its attributes are:

- InstanceNo: it contains an identifier value to a specific behaviour instance.
- PacketNo: it contians an identifier value accompanied with each inspected packet.
- BehaviourNo: it contains an identifier value to the intended behaviour.
- CurrentCertainty: it is assigned with the degree of certainty of the intended behaviour instance.

VI. Conclusions And Future Work

This paper presented discussion of the design of a proposed network security technique that is based on knowledge engineering concepts. The system utilises a packet filtering system to build a behavioural model of users' behaviour and detects anomalies in those behaviours.

In order to do that the system utilised an inference engine that is based on the frequent reevaluation of the stored states of users' behaviors. The advantage of that approach stems from relaying on a dynamic control structure rather than a static control structure. Accordingly, the system works in data-driven way. The system rules can communicate with one another only by way of the data.

The system utilized a “stateful” packet filtering engine. That technique allowed the system to perform the four basic networks security processes: alert, protect, respond and manage.

An integral part of the system is its knowledge base. That knowledge base allowed the system to detect anomalies in users' behaviour and generate alerts. More importantly, the system has the ability to learn over time to tune and modify its models.

Finally, the system needs to be tested with large volume of data in order to measure its performance and time considerations. Also, the system needs further tests where the computation of the conflict set is a non-trivial problem.

REFERENCES

[1] H. Makhlouf, M. W. Youssef and S. Ismaeel, "التجارة الالكترونية و الحكومة الالكترونية", Ain Shams Press Inc, second edition, 2008

[2] M. W. Youssef and H. Elgendy, "Applying Open Networks Communication Authentication By Scrambling and Encrypting Layer Two Packet Content.", IJCSNS, June 2011.

[3] M. W. Youssef and H. Elgendy, "Applying Open Networks Communication Authentication"; IEEE conference Security and Applications, St. Petersburg, 2011.

[4] M. W. Youssef, "Securing Computer Networks Communication By Modifying Computer Network Communication Protocols"; IEEE conference Security and Applications, St. Petersburg, 2011.

[5] O. I. Sherif, "A Firewall Based Schema for Computer Network Security", M.Sc. thesis, Institute of statistical studies and research, 2001.

[6] Comer 2000, Sect. 11.2 - The Need For Multiple Protocols, p. 177, introduces the decomposition in layers and Sect. 11.3 - The Conceptual Layers Of Protocol Software, p. 178.

[7] "Port Numbers". The Internet Assigned Numbers Authority (IANA).

[8] Avelino J. Gonzaie, Douglas D. Donkel, "The Engineering of Knowledge-based Systems Theory and Practice", Prentice-Hall, Inc. 1993.

[9] Youssef M. W. and Roshdy K., "A Proposed Packet Filtering System To Protect Open Networks", ISCA, 2005.

[10] Ronald E. Walpole, Raymond H. Myers, "Probability and statistics for engineers and scientists", Macmillan publishing company, 2001.

[11] Youssef M. W. and G. Morris, "A Decision Support System for Forecasting Dynamic Objects Behaviour And it's Application on Lake Qaroun", Port Said Engineering Research Journal, March 2003.

[12] Youssef M. W. and G. Morris, "Using Multiple Regression Models To Simulate Dynamic Objects Behaviour In A Decision Making System", Port Said Engineering Research Journal, September 2003.

[13] J. Casad, "Teach yourself TCP/IP in 24 Hours", SAMS, Second Edition, 2001.

[14] [14] Nabil El Kadhi and Hazem El-Gendy, "An Intelligent Bidirectional Authentication Method", International Journal of Computer and Network Security, Vol. 2, No. 10, 2010, pp. 1-7.

[15] Mohamed Wagdi and Hazem El-Gendy, "Applying Open Networks Communications Authentication", Proceedings of the 11th International Conference on Intelligent Transport System – Telecommunications sponsored by IEEE and IEEE Communications Society, Saint. Petersburg, Russia, 23-25 Aug. 2011, pp. 650-657.

[16] Mohamed Wagdi Youssef and Hazem El-Gendy, "Scrambling and Encrypting-Based Authentication for Open Networks Communications", International Journal of Computer Science and Network Security, June 2011, pp. 24-29.