Security Policy Monitoring System

Slim Toueiti#1, Amir Souissi*2, Karim Hassan#3

High Institute of Technology Studies of Jendouba
Campus universitaire - 8189 Jendouba du nord, Tunisia
[1] Toueiti@gmail.fr
[2] Amir.souissi@yahoo.fr
[3] Karim.hassen@hotmail.fr

Abstract — *The security of a computer system is the set of technical resources needed to prevent the unauthorized use, misuse, or alteration of that computer system. Several techniques and tools have been developed to ensure the security of a computer system, but a software tool is missing to monitor and follow the security policy. The idea behind this article is to set up a security policy monitoring system that uses existing hardware and software resources to ensure the proper functioning of a machine connected to the global network (Internet).*

Keywords — *Security; Security Policy; Monitoring.*

I. INTRODUCTION

The security of a computer systems has become the stake of companies. New technologies influence our personal and professional habits. As a result, security has become an essential aspect that must be taken into account.

A. Risks

The risk is the potential loss of what requires protection. If there is no risk, there is no need for security. Risk is the combination of threat and vulnerability. In the absence of vulnerability, threats pose no risk. Similarly, in the absence of threats, vulnerability poses no risk. Measuring a risk involves attempting to identify the probability of an adverse event occurring.

A risk can be [1]:

- **Minor**: Vulnerability exposes a risk to the user, but it is unlikely that anything will happen.
- **Medium**: Vulnerability poses a significant risk to privacy, integrity, availability.
- **Major**: Vulnerability poses a real threat to privacy, integrity, availability.

The multiplicity and the decentralization of the networks increase the risks of intrusion, it is necessary to take consciousness, by measuring the possible consequences and to know the causes, in order to take the appropriate measures. The majority of the risks come from the outside, in the form of an attack which is a deliberate hindrance to the computer resources and resulting from a human activity.

There are two types of attacks:

- Passive attacks: It is the attacks that do not change the state of the information. They are based on mechanisms such as [2]:
  - Observation using probe
  - Traffic analysis
- Active attacks: These are dangerous attacks, based on mechanisms such as [3]:
  - Fraudulent connection to computer or network equipment.
  - Altering messages in transit over a network.
  - Denial of service due to server or network saturation.

B. Countermeasures (Tools and Methods)

To ensure the security of a computer system there are many techniques and tools (software and / or hardware). The most famous techniques are briefly cited bellow:

- **Cryptography**: Data Encryption was invented to ensure the confidentiality and integrity of the data, it consists in making a message unintelligible. There are two types:
  - *Symmetric cryptography* using the same key during encryption or decryption. We distinguish DES, 3DES, AES, etc.
  - *Asymmetric cryptography*, with two keys, one used for encryption and the other private for decryption the message.
- **Identification and authentication**: Identifies the person (user) who is trying to perform a task and if this user has the privileges to do.
- **Firewall**: A firewall is used to protect the local network from external problems. It

processes packets at the physical level, at the IP (network layer) level and up to the transport layer.

- **Anti-virus**: To protect against viruses, it is very necessary to use anti-virus software. The main features of this software are to prevent the presence of viruses, to diagnose the type of infection and to repair the contaminated files.
- **Anti-spyware**: Many tools exist for detecting spyware on the computer and for deleting it, scanning the computer's central memory, the registry database, and the hard disks in a system to search these parasites
- **Data Recovery Tools**: Certain data kept on the information system of an enterprise are vital for its operation, and must therefore under no circumstances be lost or unavailable.

There are other techniques as required, but not all of them can be cited.

II. Related Works

There are two reasons why cannot have complete security of a computer systems.

The first reason that systems are poorly protected is that security is expensive. While the second reason is due to a lack of dynamic control of the application of the security policy.

Most organizations spend significant amounts on high-tech protection such as firewalls, anti-virus software, intrusion detection systems, and biometric locking devices as part of their IT security efforts.

But even the most advanced technologies, hardware or software, cannot resist the human element.

Existing solutions are usually called UTM (Unified Threat Management). Invented in 2004 by Charles Kolodgy. These are network firewalls that provide functionality as needed. These features include:

- Firewall
- Anti-spam
- Network traffic management
- Anti-virus
- Proxy
- Intrusion detection or prevention system (IDS or IPS),

All these features are grouped in the same box (alliance) equipped with a Linux operating system. The box is managed using an application (web application) accessible on the network as an intranet or extranet. The UTM can therefore manage the security tools mentioned above but they do not incorporate a security policy. So, the administrator uses the UTM through the web interface to enable or disable a service or feature. It can also add or remove rules in the firewall system. But the UTM does not monitor a security policy. Example of UTM: SOPHOS [4] and FORTINET [5].

The security tools that currently exist are performing and respond well to the needs, but the problem is how to use these tools? How to minimize the human factor? For example, if the administrator forgets to shut down a vulnerable service, or leaves a critical folder with type 777 permission, how do you resolve these problems?

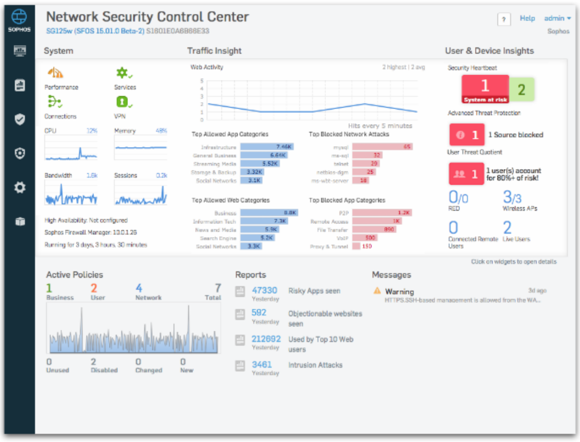


Fig.1: SOPHOS - Example of UTM

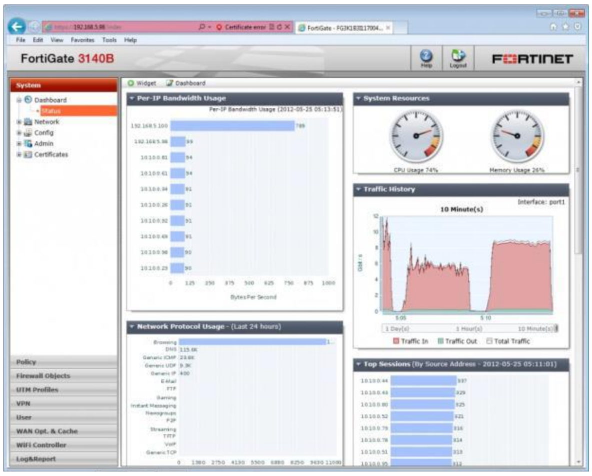


Fig.2: FORTINET - Example of UTM

Our solution Security Policy Monitoring System (SPMS) to this problem consists of three steps:

1. Design a security policy as required,
2. Fill in the rules in a database,
3. Monitor the application of the policy with scanning and scanning routines.

Our tool is not a firewall or antivirus, it is a system that allows to follow the security measures used on the different servers of a company. It monitors whether a rule or procedure defined by the security policy and well put into practice.

If an administrator forgets to activate the firewall to protect against a risk, the system will make a permanent check to ensure the activation of this command.

The methods used by SPMS are:

- A scan of the open ports on the different servers (nmap)
- A comparison between the rules of the firewall and those applied on a server
- Followed up the list of users of all installed servers:
    - FTP server
    - Mail server
    - Database server
    - Ext.
- An analysis of log files:
    - Log connexions
    - Log of a proxy (ex: squid)
    - Log of an FTP server
    - Ext.
- A check on the access rights defined on all the files in the server.

Above the architecture of the solution:

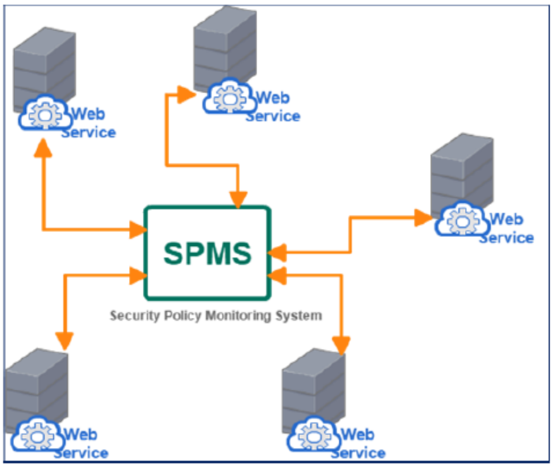


Fig.3: Architecture of the SPMS solution

The realization of our tool requires:

- A design to define the security policy in a database
- A remote server scan procedure
- Tools installed on each server to:
    - Analyze log files
    - Check the firewall rules
    - Harvest information related to the use of different services.
    - A web service on each server that returns the analysis and monitoring results to our system.

Each server must contain a web service that allows to return:

- The current state of the server (cpu, memory, ext.)
- Allows you to detect anomalies related to the application of the security policy.
- Check open ports,
    - Check the access rights of folders and critical files,
    - Follow the rules of the firewall,
    - Ext.

III. Contribution

The development of this system and the experimental are carried out on virtual machines on the Public Cloud OVH. Ubuntu 14 is the operating system used on different VMs.

The application is installed on an Ubuntu 14.04.2 VM. All machines are connected to the Internet and each has a public IPv4 IP address.

A. Definition of security policy (Design)

Our system consists in monitoring the security policy of the different servers of a company, in a distributed architecture, for this we must design a database that safeguards the rules and procedures to be applied.

The architecture consists of a set of servers, each server is characterized by:

- A name (host name),
- A public IP address (accessible directly over the network),
- A short description of its role (ftp server, mail or other).

A server can host one or more services, each service is described by:

- A name,
- A port number,
- The name of the transport protocol (TCP, UDP, ...)
- A description.

The security policy allows you to set rules and procedures for securing servers. We have identified the rules that will be implemented in our system:

1. Services Rules:
   - Running Service (open port)
   - Stopped Service (port closed)
2. Firewall Rules: Each server activates the acceptance or non-acceptance of IP packets according to a given filter. The rules will be saved in a table (firewall),
3. User rules (accounts):
   - Users list,
   - Connected users.
4. Rules that define users of a service such as FTP or others:
   - Connected users,
   - Users disabled.
5. Rules for server directories and files:
   - Refers to directories and critical files,
   - Access rights,
   - Update detection

B. *Follow-up of the security policy*

The policy monitoring procedures use the data stored in the database and scan data on the servers to identify anomalies of violation of one or more rules. Our system consists of the main application that communicates with routines that run on a server.

The main application makes it possible to manage the servers and the security policy of each one. You can then add, modify, delete and list the servers. A space to describe each server:

- Services,
- Users,
- Firewall rules,
- Critical folders and files,

Once the database is filled then we can make analyzes and scans using routines and shell scripts that run on each VM and retrieve the results through a web service.

Each service is defined by the transport protocol and a port number, the famous "NMAP" port scanner can detect the open ports of a remote server. Simply pass in parameter the IP address of the server. Below is an example of the most basic use:

By comparing the values entered in the database for the server (Figure 5) and the scan carried out (Figure 4) it is possible to validate the correct operation of the services of each machine. The figure below shows the scan results on the PSMS application.

The firewall rules for each machine are stored in the database. The application can communicate with the web service which returns the actual rules of the firewall on the machine.

The web service running on the remote machine allows, by executing a shell script, to return the actual firewall rules in progress. The following command returns the iptables firewall rules (Figure 6).

```
$ iptables -L --line-numbers
```

The application checks for correct operation on the server by comparing the result with the rules of the security policy stored in the database.
The list of users must be saved for each VM, this allows us to detect orphaned users. This ensure that only the users set by the PS that may exist on the VM.

Some services are shared across multiple users such as FTP, so you can track and control the list of users for each service. A script on a VM allows you to retrieve the actual users and compare them with those recorded to ensure the correct use of the service.

Protecting critical directories and files is an assurance of system integrity. A simple solution allows us to monitor the access rights on each resource, the permissions must match the rules of the policy.

The system also tracks the updating of these disk resources, one solution is to use "checksum" functions such as MD5SUM. The calculated value is then recorded in the database and is compared each time with the value returned by the script executing on the relevant VM.

The list of files of the selected server corresponds to such table.

The system shows us the defaults that concern the files or critical files of the system (Figure 7) according to the security policy established beforehand.

IV. Conclusion and Future Works

The solution achieved the majority of the goal of securing a server accessible from the Net. However, it is possible to spread the application for other functionalities such as the analysis and filtering of the log files of the services or of the system itself. This analysis makes it possible to detect possible intrusions.

The solution is limited to version 4 of the IP protocol, a new version can easily integrate the IPV6 protocol.

Another limit of the solution is the use only the Linux operating system (Ubuntu), we have to create new scripts to include other Linux distributions such as OpenSUSE or even for other operating systems such as Windows and MAC OS.

The current system has been successful detecting short cuts to take precautions stipulated in the security policy. An improvement allows us to perform actions to start or stop a service directly from the VM that hosts the solution.

References

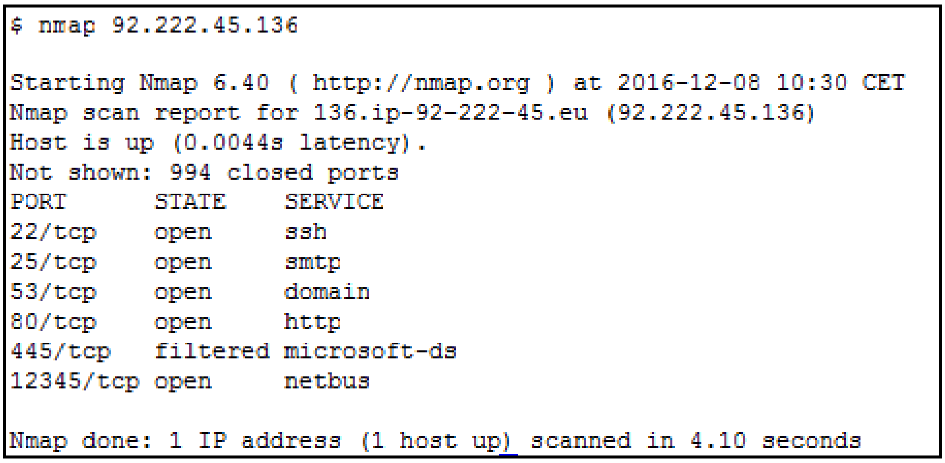
[1]. JR Bettman, JW Payne, R Staelin, Cognitive considerations in designing effective labels for presenting risk information, Journal of Public Policy & Marketing – JSTOR

[2]. Peris-Lopez, P., Hernandez-Castro, J., Estevez-Tapiador, J.M., Ribagorda, A.: M2AP: A Minimalist Mutual-Authentication Protocol for Low-cost RFID Tags. In: Ma, J., Jin, H., Yang, L.T., Tsai, J.J.-P. (eds.) UIC 2006. LNCS, vol. 4159, Springer, Heidelberg (2006)

[3]. O Berthold, H Federrath, S Köpsell, Web MIXes: A system for anonymous and unobservable Internet access, Designing Privacy Enhancing Technologies: International Workshop on Design Issues in Anonymity and Unobservability, Berkeley, CA, USA

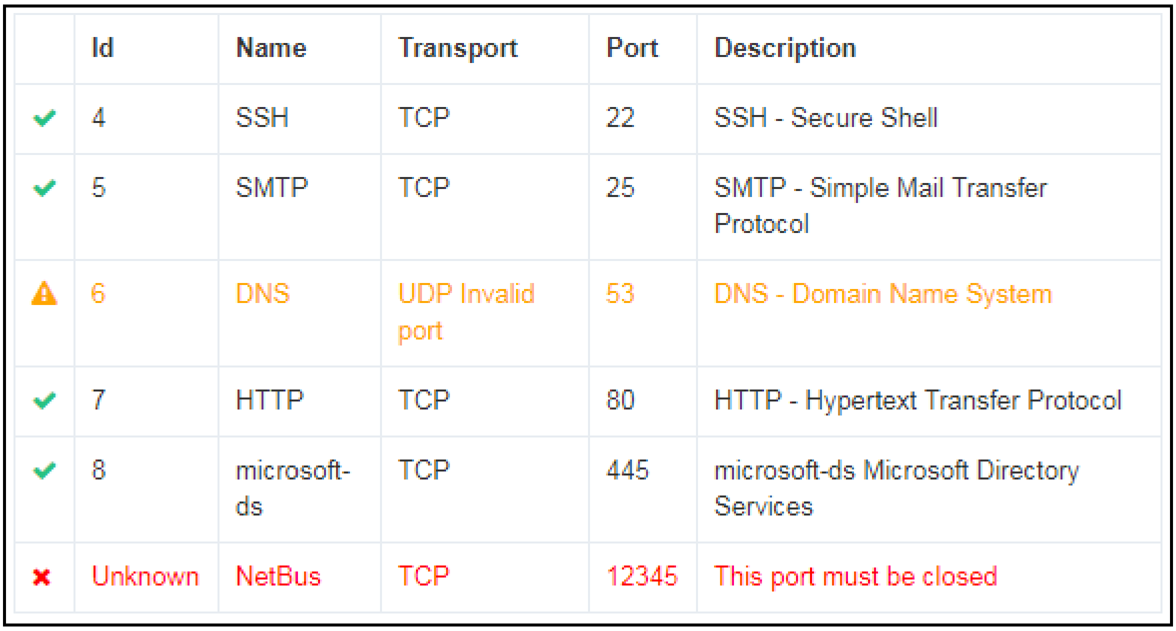
[4]. SOPHOS: https://utm.trysophos.com/

[5]. FORTINET: https://www.fortinet.com/



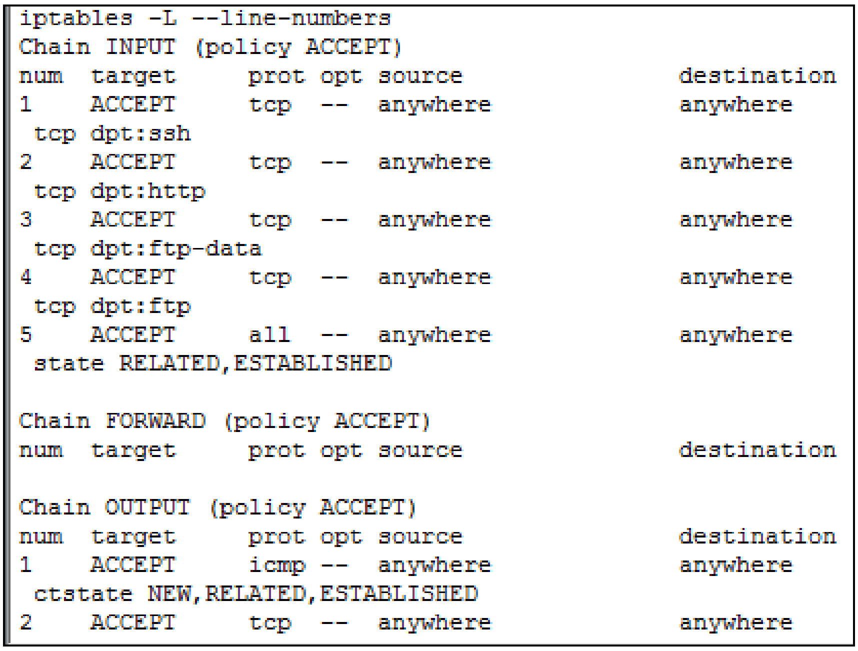
```
$ nmap 92.222.45.136

Starting Nmap 6.40 ( http://nmap.org ) at 2016-12-08 10:30 CET
Nmap scan report for 136.ip-92-222-45.eu (92.222.45.136)
Host is up (0.0044s latency).
Not shown: 994 closed ports
PORT      STATE    SERVICE
22/tcp    open     ssh
25/tcp    open     smtp
53/tcp    open     domain
80/tcp    open     http
445/tcp   filtered microsoft-ds
12345/tcp open     netbus

Nmap done: 1 IP address (1 host up) scanned in 4.10 seconds
```

Fig.4: Ports Scan on a remote server with NMAP



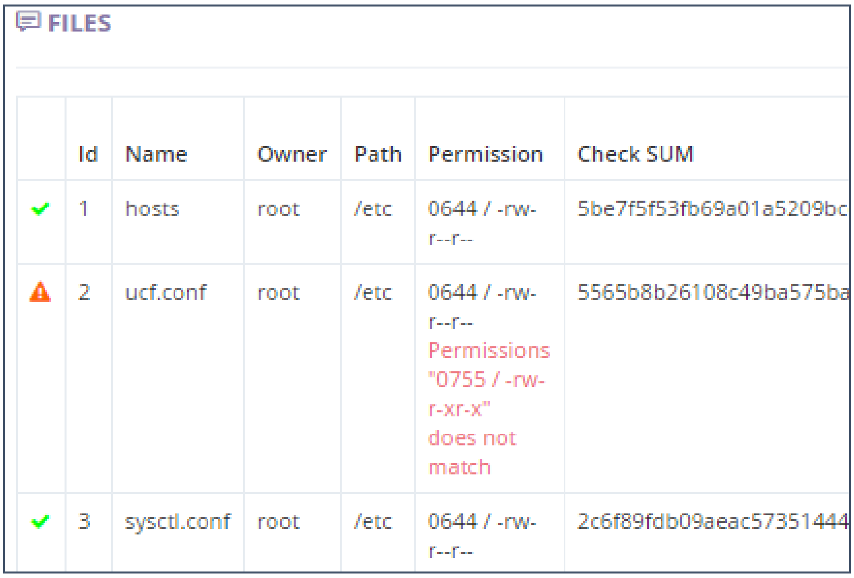
	Id	Name	Transport	Port	Description
✓	4	SSH	TCP	22	SSH - Secure Shell
✓	5	SMTP	TCP	25	SMTP - Simple Mail Transfer Protocol
⚠	6	DNS	UDP Invalid port	53	DNS - Domain Name System
✓	7	HTTP	TCP	80	HTTP - Hypertext Transfer Protocol
✓	8	microsoft-ds	TCP	445	microsoft-ds Microsoft Directory Services
✖	Unknown	NetBus	TCP	12345	This port must be closed

Fig.5: Scan results on PSMS



```
iptables -L --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1    ACCEPT     tcp  --  anywhere             anywhere
 tcp dpt:ssh
2    ACCEPT     tcp  --  anywhere             anywhere
 tcp dpt:http
3    ACCEPT     tcp  --  anywhere             anywhere
 tcp dpt:ftp-data
4    ACCEPT     tcp  --  anywhere             anywhere
 tcp dpt:ftp
5    ACCEPT     all  --  anywhere             anywhere
 state RELATED,ESTABLISHED

Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination
1    ACCEPT     icmp --  anywhere             anywhere
 ctstate NEW,RELATED,ESTABLISHED
2    ACCEPT     tcp  --  anywhere             anywhere
```

Fig.6: List of firewall rules on a VM.



FILES

	Id	Name	Owner	Path	Permission	Check SUM
✓	1	hosts	root	/etc	0644 / -rw-r--r--	5be7f5f53fb69a01a5209bc
⚠	2	ucf.conf	root	/etc	0644 / -rw-r--r-- Permissions "0755 / -rw-r-xr-x" does not match	5565b8b26108c49ba575ba
✓	3	sysctl.conf	root	/etc	0644 / -rw-r--r--	2c6f89fdb09aeac57351444

Fig.7: Validation of critical files in a VM.