Part II - An Intrusion Detection and Prevention System for Internet Protocol Multimedia Subsystems

1. Introduction

The described Intrusion Detection and Prevention System (IDS/IPS) is based on the paper "A Cross Layer Spoofing Detection Mechanism for Multimedia Communications Services" by Nikolaos Vrakas and Constantinos Lambrinoudakis. The mechanism offers a greater protection level to a multimedia service by collecting data from multiple networking layers and compares them in order to verify that a communication is not spoofed or in any way compromised. These values are extracted from the Data Link Layer (MAC Address), the Network Layer (IP Address) and the Application Layer (SIP ID, Call-ID). This multi-level mechanism ensures greater security levels without demanding security meters such as heavy encryption algorithms that will slow down the performance of the servers. It demands, however, detailed socket programming, packet recognition algorithms, parsing functions, data structures and great linux operating system knowledge in order to deploy, install and run properly.

2. Location

The location of the mechanism is to be considered of great importance, as it can be critical to the performance of the service. It was decided that the mechanism will be installed on the Call Session Control Function (CSCF) and more specifically on the Proxy Call Session Control Function (PCSCF). The PCSCF is the first entity the UE will interact with. This way we can ensure that spoofed and malformed messages will reach the PCSCF and shall not go any further to other IMS components. This is the default usage of the PCSCF in the IMS architecture. As a result, the service will not be loaded with unnecessary messages that may slow down its performance or even worse bring it down (Denial of Service, Memory Overflow, etc.) and the threat will be eliminated immediately.

3. Components

The built mechanism consists of two parts: the firewall filter and the sniffer. Below follows a general description of the functionality of those two and after that a more technical description.

3.1 Firewall Filter

The firewall filter is basically a set of rules that ensure no unauthorized user will use the service. It filters all messages that reach the PCSCF and allows only those containing the

REGISTER method. It is designed in a way in order not to affect any other services that may run on the server, but only this of the PCSCF. This means only REGISTER methods are allowed at first, and when a user is authenticated then all methods are allowed. Any other message that is directed to the PCSCF will be rejected. This requires filtering commands with port recognition since the PCSCF will run on a set port, as well as string comparison in the packet, in order to identify which method a packet contains.

3.2 Sniffer

This is the main program of the IDS/IPS. What it does is monitoring all incoming packets of the PCSCF, which can only be REGISTER methods as no other method is allowed to cross the filter. From every incoming packet, it gathers specific information and stores them in a table. When a user is authenticated by the service, the sniffer matches the data from the table and then will allow the user to use all available SIP methods (INVITE, MESSAGE etc.).

If the message contains a REGISTER method, it will extract the MAC, the IP, the SIP ID and the Call-ID of the packet. All this data will be stored in an array and will serve as a cross check database within the program, where the Call-ID will be the primary key. If the sniffer detects a 200 OK message, then it will extract only the Call-ID from the packet. It will search in the array for that very same Call-ID. If it is found, it means that the user has already tried to register with the service and the latter responded positively, allowing him to authenticate. Therefore, the user should be allowed all methods. This means there will be a firewall exception rule which allows the user to send any method he wants, as long as it is originated from the same MAC Address, the same IP Address and contains the same SIP ID as the ones he registered with.

For example, if Bob registered with MAC aa:aa:aa:aa and IP 1.2.3.4 and the service receives a message from Bob but from MAC bb:bb:bb:bb, then this message is considered an identity theft attempt and will be ignored.

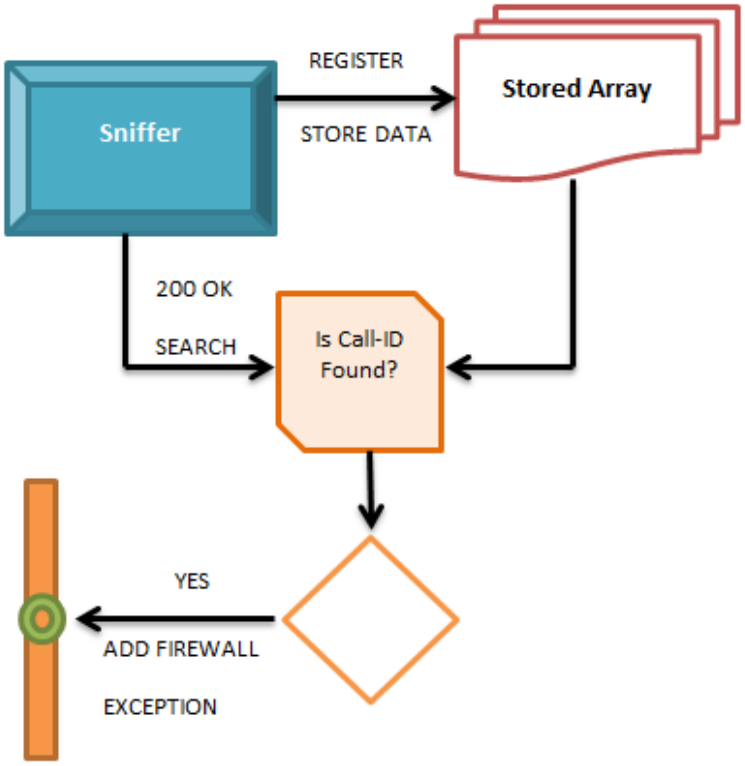

Figure 8: IMS Sniffer Archtecture

4. Technical Information

This section will provide technical information about the mechanism components and describe the tools that were used to create them.

4.1 Firewall Filter

The firewall filter is, as mentioned before, a set of firewall rules that will filter all incoming messages to the PCSCF and will allow only those containing the REGISTER method. Due to the fact that the Open IMS platform was the one used to conduct testing and since this specific platform can only run on a Linux operating system, the firewall used in this case was the embedded Linux firewall, called iptables. The commands used by the filter are described below:

```
iptables –F
```

This rule will flush (delete) any existing iptables rules.

```
iptables -A INPUT -i lo -j ACCEPT
```

This rule will accept any INPUT (incoming) packets, as long as they use the lo (local) interface

```
iptables -A INPUT -p udp --dport 4060 -m string --string
"REGISTER" -j ACCEPT --algo bm
```

This is the filtering rule and is very important. What this rule does is accept packets that:

- Are incoming messages
- Are using UDP protocol
- Are destined to port 4060
- Contain the string “REGISTER”

If any packet does not agree with all the above conditions, then it is passed to the next rule below. If it agrees, it will be accepted by the system and will be forwarded to the PCSCF.

```
iptables -A INPUT -p udp --dport 4060 -j DROP
```

Any message using udp protocol and is destined to port 4060 shall be rejected, unless it was forwarded by the 3rd rule.

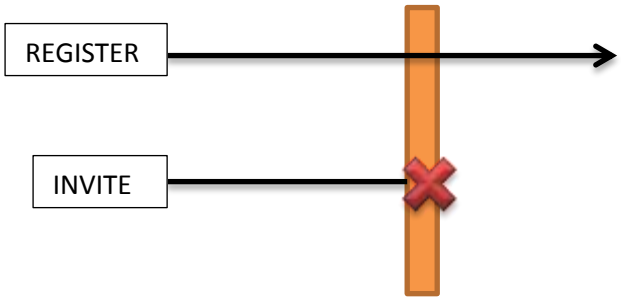

Figure 9: Example case of filter in use

4.2 Sniffer

The sniffer program is the one to monitor the messages, store the needed data and add exceptions to the firewall filter. It is programmed in C and it is using the libpcap library, which is used by other well-known and powerful sniffers as well, e.g. wireshark. This library is extremely helpful, since it allows us to capture a great amount of information about the packet, which will be used from the mechanism for cross layer intrusion detection (mac address, ip).

One of the functions of the sniffer is to detect messages containing the REGISTER method and extract useful data to an array (as described previously). This array will keep growing and having data added until it reaches a manually set limit and removes all data from its cells, becoming empty and ready to have data inserted again.

Another function the sniffer has is to detect messages that have the 200 OK header included. After that, it performs a search in the array for the Call-ID. If it is found, then the user will be granted access to the IMS service. This will happen with a built in command in the sniffer, which adds a new rule to the firewall filter via a system command.

Both of these functions use a custom parser made from scratch in order to fit the needs of this assignment precisely. Some functions used by the parser are:

- Strstr (compares if a string is found in another string)
- Strcmp (compares if two strings are the same)

- Strncmp (compares a number of letters of two strings to see if they match, e.g. the first 6 digits)
- Strtok (fragments a string into segments, helping parse useful data from the packet)

The sniffer was modified in order to sniff only UDP packets, since the Open IMS platform used for testing used only UDP protocol. Additionally, the sniffer will work on the Ethernet interface of a server.

5. How it Works

As mentioned above, at first only REGISTER methods are allowed. Therefore, the user must first register with the service and authenticate themselves. A REGISTER method is sent to the IMS service and the first to receive this packet is the PCSCF. The sniffer extracts some information from this packet that will later be used to match the authorized user and grant him full access to the IMS service. This information contains:

- Ethernet (MAC) Address
- IP Address
- User Identity
- Unique Call-ID

By this, the program helps us hold information about a user that tried to register. That way we know that a user with *MAC: aa:aa:aa*, *IP: 1.1.1.1*, *Identity: alice@newims.org* tried to register with a *Call-ID: abcd1234*. This information is stored in an array and the sniffer now waits for the server response. That means that the server will send a challenge to the UE with a *401 UNAUTHORIZED* header, the UE will respond with another REGISTER method, this time containing the credentials of the user (secret password). The user will then be authenticated and will receive a *200 OK* message which terminates the authentication procedure and says that the user is indeed authenticated. This last message is of great importance, since it is the message that the sniffer will cross reference with the existing data of the stored table. It will cross check the Call-ID of the *200 OK* message and see if the same Call-ID exists in the stored table. If so, this means that a user that tried to register before is indeed authorized to use the service and we must allow him access at once.

For example, if the *200 OK* message has *a Call-ID: abcd1234* then we know that user *alice@newims.org* is authenticated (since the Call-ID is found in the array) and the program adds a new rule to the firewall filter that allows all incoming messages to be accepted, as long as they are originated from the same MAC, IP and come from the same user.

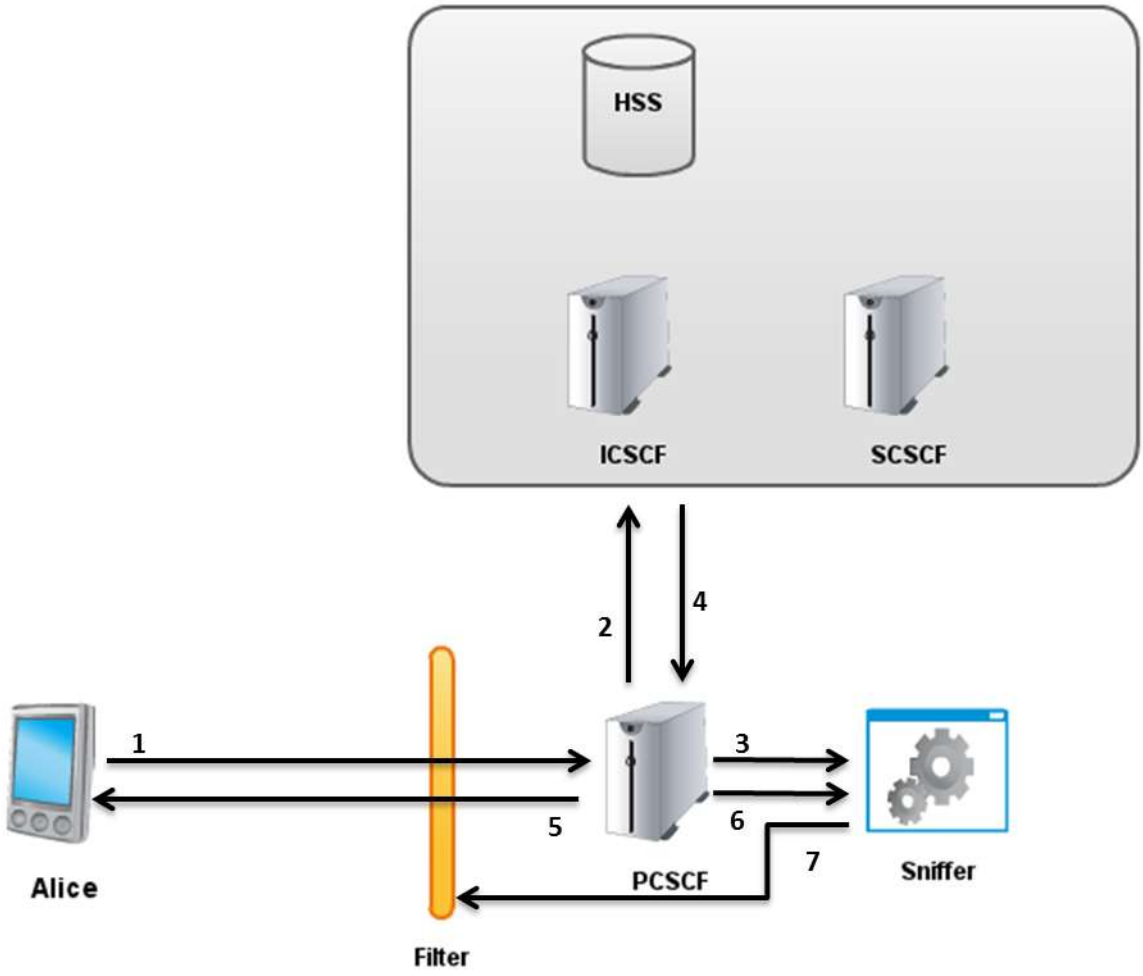

Figure 10: Description of the mechanism

1. The UE sends a REGISTER method to the PCSCF
2. The PCSCF forwards the message to the rest of the IMS network
3. The sniffer stores the specific information in a table
4. The PCSCF receives a 200 OK message from the network
5. The 200 OK message is being forwarded to the UE
6. The sniffer detects the 200 OK message and cross checks the Call-ID with the table
7. The sniffer adds an exception to the filter in order to allow the authenticated user

6. Conclusions

It becomes more and more important to ensure authentication in telecommunication services, such as IMS and while encryption algorithms may offer that, it is a drawback as it introduces more delays in the service. With the deployed mechanism we can ensure a great authentication level with lightweight means, which will not affect client devices and UE and we can provide a multi-level authentication mechanism by gathering data from multiple layers in the OSI model.

The avoidance of spoofed messages and DoS/Buffer Overflow attacks is also to be considered of great importance, since it improves the performance of the service and does not require additional sources to process unauthorized messages.

References

Jane Dudman, *Voice over IP: what it is, why people want it, and where it is going*, **JISC Technology and Standards Watch**, September 2006

Ricky M. Magalhaes*, Session Initiation Protocol (SIP) and It's Functions*, February 2005

Dorgham Sisalem, John Floroiu, Jiri Kuthan, Ulrich Abend, Henning Schulzrinne, *SIP Security,* **Wiley Publications,** 2009

Miikka Poikselkä, Georg Mayer, Hisham Khartabil, Aki Niemi, *The IMS: IP Multimedia Concepts and Services, Second Edition,* **Wiley Publications,** 2006

Nikos Vrakas, Costas Lambrinoudakis, *A Cross Layer Spoofing Detectiuon Mechanism for Multimedia Communications Services*