Objective of the project:

The objective of this project is to provide cross layer mechanism that acts like shield against real-time spoofing attacks in VoIP and IMS. It also provided the lowest possible delay, while the Quality of Service (QoS) remains as the top priority for users and providers. This cross layer mechanism does not require much computational calculation thereby it will reduce the resource demands. It also provides high detection rates with false alarms that approach zero (even when the filter is inadequately trained and configured), a parameter that is very crucial for such mechanisms.

Methodology:

The Voice over IP (VoIP) environments and the most contemporary ones such as the IP Multimedia Subsystem (IMS) are deployed in order to provide cheap and at the same time high quality services to their users. Toward this objective, these infrastructures utilize the Session Initiation Protocol (SIP) for signaling handshakes since it is the most flexible and lightweight protocol available. But, it happens to be vulnerable to many attacks that threaten system’s security and availability. The proposed approach introduce a cross-layer mechanism, that is able to mitigate in real-time spoofing attacks such as SIP signaling, identity theft, masquerading, and Man in the middle, and also single and distributed source
flooding. It consists of three components: the policy enforcer which acts as a black list, and the spoofing and flooding modules. It also introduces a classification of SIP flooding attacks for better representation of the detection coverage. Concerning its performance, it does not require computational expensive calculations or resource demanding security protocols, thus being a lightweight mechanism.

Threats in SIP/IMS environments may originate from different layers, since the attacker attempts to exploit more than one of the protocol’s vulnerabilities. It is therefore important for the IDPS to be able to detect such behaviors before they become a threat for the higher layers and affect multimedia services.

The proposed mechanism is not only to prevent spoofing attacks, but also attacks against system’s availability known as flooding attacks.

Mechanism

The proposed mechanism handles every incoming message, takes decisions about its originality, and decides whether it will be routed it to its destination or not. The main concept is based on cross-layer binding between six values that can be gathered from layers 2, 3, and 5 of the network protocol stack. These values correlate a specific UE with a session, a set of IP addresses, and the identities of the subscribers. For instance, the frames located at the data link layer (layer 2) bear the network or MAC address of the utilized UE(user equipment).

For every incoming message, a tuple $E_i$, $\forall i \in (0, \ldots, n)$ is generated, where $n$ is the number of incoming messages and $n \in \mathbb{N}$. Every $E_i$ passes through the spoof checking module which decides whether the message is legitimate. Policy Enforcer (PE) will drop the message which is malicious. The latter holds a blacklist of known malicious $E_i$.

Now, the legitimate $E_i$s fed to the second module which consists of two tables: the registration table for holding registration messages’ data and the request table that holds the data of all the other requests. Every $E_i$s stored to one of these tables, depending on the type of the request. The position of the table where a tuple must be stored is calculated according to the bloom filters theory. A bloom filter is a data structure that can be utilized for testing the existence of an element $x_i$ in a set $X$. Every $x_i$ is hashed through $\lambda$ different hash functions. The result of every hash function points to a specific position of a vector of $m$ bits. The vector’s length is equal in bits as the output of the hash functions.
This model is being employed in order to avoid searching and sorting through the second module’s tables. The first column in both tables is a counting bloom filter. The input to the hash functions is the tuple Fi = {MACi, IMPIi, IPi}, FI \subseteq Ei. These three values denote a unique combination that is extracted from every Ei.

Therefore, the precise position for every user per UE/IP is the value H(Fi). C is the counting bloom filter that provides the mechanism with information about the number of messages per subscriber/UE and eliminates the time needed for detecting a specific tuple’s position. The values MAC, IP, SIP-IP denote the corresponding address at layer 3, 4, and 5 of the network protocol stack that have been involved in a specific request. The IMPI/IMPU holds the private/public id of the corresponding incoming message. The TS holds a timestamp and T.Dist the time distance between the last two timestamps that has been calculated by: T.Dist= TS_i - TS_{i-1}, where i is the number of messages that were stored in a specific row. Init.D_Avg and Curr.D_Avg denote the initial and current T.Dist value, respectively. Finally, the Trs value is a threshold for alarm triggering in single source flooding attacks.

**Conclusions:**

The detection covers most of the spoofed SIP message attacks and also flooding attacks when originated either from single or distributed sources. Its design poses a lightweight mechanism, free of complex and resource demanding calculations, a very crucial factor in such environments where QoS is of top priority. It also provides high detection rates with false alarms that approach zero (even when the filter is inadequately trained and configured), a parameter that is very crucial for such mechanisms.

**Software Requirements**

1. Operating system : Windows XP/7
2. Coding Language : J2EE
3. Data Base : HEIDESQL