

Security Threats and Solutions for Application Server of IP Multimedia Subsystem (IMS-AS)

Muhammad Sher, Shaoke Wu
TU Berlin/Fokus Fraunhofer
Kaiserin-Augusta-Allee 31
10589 Berlin, Germany
Email: sher@fokus.fraunhofer.de

Thomas Magedanz
TU Berlin/Fokus Fraunhofer
Kaiserin-Augusta-Allee 31
10589 Berlin, Germany
Email: magedanz@fokus.fraunhofer.de

Abstract—In this paper we will explore security threats and attacks possibility and security solution for Application Server of IP Multimedia Subsystem(IMS-AS). The SIP Application Server is an important entity of IP Multimedia Subsystem (IMS) because applications providing value added services are deployed on the Application Server. The SIP Application Server is triggered by Serving Call State Control Function (S-CSCF) which will redirect certain messages to IMS-AS based on internal filters and criteria or by requesting filter information from Home Subscriber Server (HSS). The critical security attacks on IMS-AS include flooding attacks, messages flows attacks etc. We will propose two tiers security mechanism based on Security Architecture for the TLS (Transport Layer Security) and Intrusion Detection System (IDS) against these attacks to secure IMS Application Server. This work is part of Secure Service Provisioning (SSP) Framework for IP Multimedia Subsystem (IMS) 3G beyond Testbed of Fokus Fraunhofer.

I. INTRODUCTION

Web application servers are ubiquitous, remotely accessible and open based architecture. The attackers can launch attacks at different protocols level against a web server easily e.g. the TCP SYN Flood attack at transport layer and the Smurf attack at network layer. The custom web-based application may introduce additional vulnerabilities, because the applications might be designed without considering security and privacy. The programming errors in web applications often translate directly into vulnerabilities. The vulnerabilities can be utilized by the attacker to launch attacks against the web applications e.g. SQL Injection attacks. All the attacks can cause a loss of service to users, typically the loss of network connectivity and services by consuming the bandwidth of the victim network or overloading the computational resources e.g. CPU and memory of the victim web server. In the worst case, the attacks can cause the loss of data or disruption of the web servers.

Attacks against web servers and web-based applications account for a substantial portion of security incidents on the Internet. The large amount of web servers and the continuous disclosure of vulnerabilities associated with web-based applications make web servers a popular target for malicious hackers. It was reported by SECURITYTRACKER [1] that in the period between April 2001 and March 2002 web-related vulnerabilities accounted for 23% of the total number of vulnerabilities disclosed to the public.

Our objective is to protect IMS Application Server which is based on SIP Servlet Container merges with existing HTTP Servlet Container Jetty [2]. The SIP Servlet API [3] is developed to standardize the platform for development and deployment of SIP based services. It is one of the several possible technologies suggested by the third Generation Partnership Project (3GPP) [4] to build a SIP Application Server (AS) which is an important part of IP Multimedia Subsystem (IMS) [5] because applications providing value added services are deployed on the application server. The well defined APIs enable any developer to develop new applications which are able to be deployed on any SIP application server.

The paper is organized as the next section briefly explains IMS 3Gb Fokus Testbed and section III is about the architecture of IMS Application Server. Sections IV and V describe IMS-AS security requirements and security attacks scenarios respectively. Sections VI and VII are about the security solution and propose design of Intrusion Detection System (IDS) respectively. The last section concludes the paper.

II. IMS 3Gb FOKUS FRAUNHOFER TESTBED

In face of the current challenges within the telecommunications market are mainly a consequence of insufficient early access to new enabling technologies by all market players, the Fraunhofer Institute Fokus, known as a leading research institute in the field of open communication systems, has established with support from the German Ministry of Education and Research (BMBF) a 3G beyond Testbed, known as "National Host for 3Gb Applications" [6]. This Testbed provides technologies and related know-how in the field of fixed and wireless next generation network technologies and related service delivery platforms. As part of 3Gb Testbed, the FOKUS Open IMS playground [7] is deployed as an open technology test field with the target to validate existing and emerging IMS standards and to extend the IMS appropriately to be used on top of new access networks as well as to provide new seamless multimedia applications. All major IMS core components, i.e., x-CSCF, HSS, MG, MRF, Application Servers, Application Server Simulators, service creation toolkits, and demo applications are integrated into one single environment and can be used and extended for R&D activities

by academic and industrial partners [6]. All these components can be used locally on top of all available access technologies or can be used over a IP tunnels remotely.

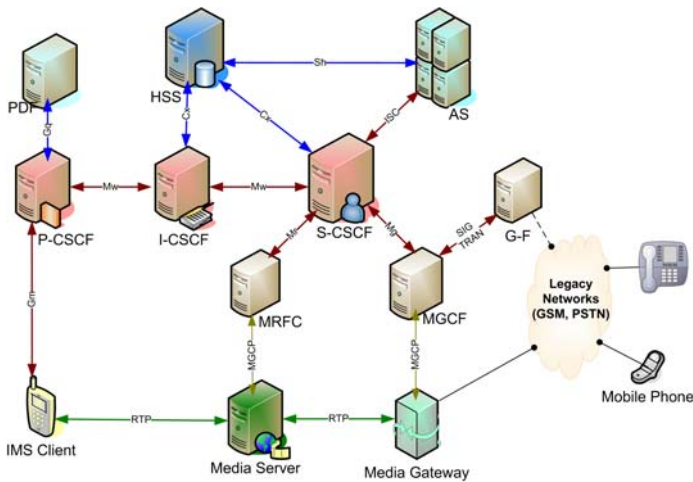


Fig. 1. IMS 3Gb Fokus Testbed Architecture

Users of the Open IMS playground can test their components performing interoperability tests. The SIP Express Router (SER) [8], one of the fastest existing SIP Proxies can be used as a reference implementation and to proof interoperability with other SIP components. Yet, the major focal point of IMS playground is to put on Application Server aside. A variety of platforms enable rapid development of innovative services. An evaluation of strengths and weaknesses of different platform options has been accomplished and can be adopted on customer's needs.

III. IMS APPLICATION SERVER ARCHITECTURE

The IMS Application Server as shown in figure 2 consists of SIP Servlet Execution, SIP Message Handling, Application Deploy Environment, SIP Servlet API Compliance and Bridge components. IMS/SIP Application Server is triggered by the Serving Call State Control Function (S-CSCF) which will redirect certain sessions to the SIP AS based on internal filters and criteria or by requesting filter information from the Home Subscriber Server (HSS). The SIP AS itself comprises filter rules to decide which of the possible many applications deployed on the server should be selected for handling the session. During execution of service logic it is also possible for the SIP AS to communicate with the HSS to get additional information about a subscriber or to be notified about changes in the profile of the subscriber. In SIP Servlet Execution, the queue of the servlets being executed will be observed by a thread which creates separate threads for execution. SIP Message Handling component receives the SIP messages and converts them into the SIPServletRequest (resp. SIPServletResponse), then dispatches the SIPServletRequest or SIPServletResponse to the corresponding SipServlet. Application Deploy Environment provides approach for the deployed applications to make known to AS. Using the deployment

descriptors of applications, the AS decides which servlet of specific application should handle a SIP message. The SIP Servlet API Compliance provides the possibility for application developer to develop the applications independent of the SIP Servlet container. The SIP Servlet API for itself is rather exciting basis for creating of communication services, but more benefit will be provided to clients with applications which have more rich feature by including the web into the communication process. As already mentioned that AS merges with Jetty [2], a HTTP Servlet container and Bridge component is responsible for bridging SIP servlet execution environment to the HTTP Servlet execution environment. As every specification contains its own session instance, the SIP Servlet specification introduces the concept of a spanning session, the SipApplicationSession. The converged applications should be able to access a common SipApplicationSession from the HttpSession as well from the SipSession.

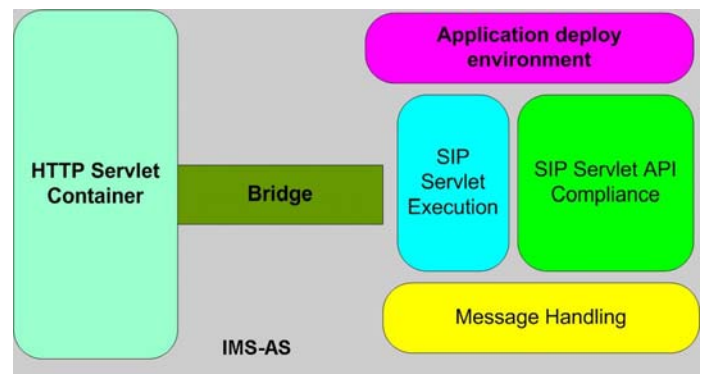


Fig. 2. IMS-AS Architecture

IV. IMS APPLICATION SERVER SECURITY REQUIREMENTS

The third Generation Partnership Project (3GPP) aimed to merge two of the most successful paradigms in communications i.e. cellular networks and the Internet. Within the IP Multimedia Subsystem (IMS) - a key component in the evolution of the 3G network - 3GPP specified in release 5 a comprehensive and service oriented architecture that includes session based quality of service in IP, charging mechanisms and standardized interfaces for application service integration and seamless interactions with legacy services. Under this technical premise a variety of promising value added services suppose to attend our entire communication life and AS is one of the implemented service containers. The IMS is gaining its popularity as the technology for transmitting multimedia traffic over IP networks. As the popularity of IMS increases, the AS within IMS is being subjected to different kinds of security threats and intrusions. First, the AS can suffer from the HTTP-based threats as we know that it integrates the web application container. Second, the text-based nature of SIP messages faces attacks like spoofing, hijacking and message tampering because the AS employ SIP for signaling. Finally, the Denial of Service (DoS) attack can be launched against the

AS. In addition, these attacks are simple to mount and with cheap flat rate Internet access is motivation to launch attacks.

V. IMS-AS SECURITY ATTACKS SCENARIOS

In this section we will try to explore possible attacks on the IMS-AS which are classified under time-dependent and time-independent attacks as shown in figure 3. The time-dependent attack means that a time interval is required to effect or damage the victim e.g. flood attack, but time-independent attack mean that it effect instantly on the target as soon as a data packet arrives which is on behalf of the attack e.g. a SQL-Injection attack.

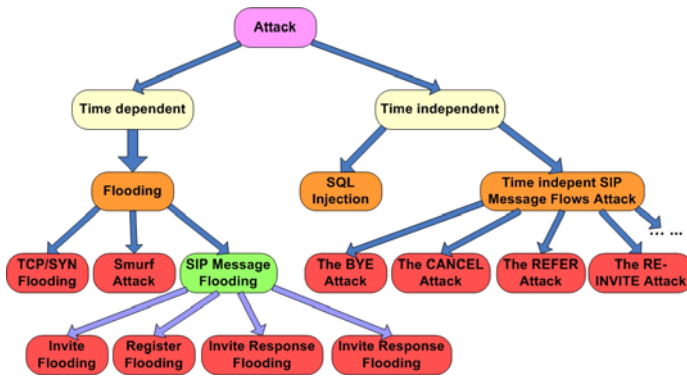


Fig. 3. Security Attack Categories

A. Time Dependent Attacks

From intrusion detection point of view the attacks that can be detected after a particular time instead of being detected immediately are time dependent. The primary feature of this category is that an attack is composed of large amount of data packets like flooding attack which is the most serious and popular threat for IMS Application Server. The attacker sends the victim large amount of network traffic to overload the network resources. As a result the resources are not available to the actual users. In the case of IMS-AS, the SIP servlet server can be overwhelmed by the flooding attack. There will not be available resources to handle the legitimate Sip and HTTP messages. The attack can use a range of protocols, including Internet Control Message Protocol (ICMP), User Datagram Protocol (UDP), TCP, and SIP.

1) *TCP SYN Flooding Attack*: The TCP/SYN attack is a common example for the flooding attack that works by creating scores of half-open connections. A half-open connection occurs when server sends a SYN-ACK message, but never receives an ACK message from the client. This is achieved when attacking system sends SYN messages to a target server with a return address other than its own (known as IP-spoofing). The server then sends a SYN-ACK message to the machine specified in the SYN message, which is, of course, not the IP address of the attacking machine. Thus server never receives final ACK and connection is never completed. These uncompleted connections are called "pending connections", and are written in a buffer of limited size. Eventually, as

the attacking machine creates an ever increasing number of pending connections causes buffer to overflow. As explained in the example given in figure 4, at first the attacker sends a faked SYN packet whose source is unreachable IP address, then the victim will respond with a SYN-ACK message. The network don't know how to route this SYN-ACK message. At last the victim shall never receive an ACK message responding to the SYN-ACK message. The memory occupied by the connection can be released only after the TCP connection will have been timeout. The similar attack is the TCP/ACKs flood attack

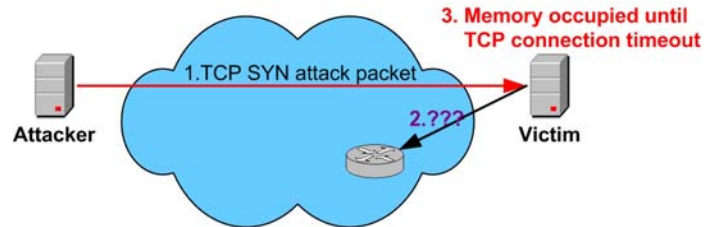


Fig. 4. TCP/SYN Flood Attack

which launched in the reverse direction by exploiting the answer packets. Using this technique an attacker sends packets to randomly chosen destination IP addresses and forges the source address of the packets to the victim's address. To amplify the attack the attacker can make use of joint power of multiple systems when launching TCP/SYN flood attack called Distributed TCP/SYN flood attack.

2) *SIP Message Flooding Attack*: There are several types of SIP Message Flooding Attack. We list here four attacks i.e. Invite Flooding, Register Flooding, Invite Response Flooding and Register Response Flooding.

a) *Invite Flooding and Register Flooding*: Attacker sends a large amount of SIP Invite messages (source IP address can be spoofed) to the victim and makes it busy processing the messages. The Register flooding attack is similar to the Invite flooding. The Register method instead of Invite method is utilized to launch the Register flooding.

b) *Invite Response Flooding and Register Response Flooding*: The Invite response flooding and Register response flooding somewhat differ from the flooding attacks above which always intend to overwhelm the victim. The goal of the Invite response flooding is to get the authentication using exhausted search. A lot of Invite messages are sent in order to crack the password for the authentication. In the case of the Register response flooding the attacker tries to send REGISTER messages with wrong credentials to SIP proxy which requires the authorization of registration requests.

B. Time Independent Attacks

1) *SQL injection*: The text-based nature of SIP messages provides opportunity for message tampering attacks in SIP applications, similarly to HTTP messages. SQL injection is kind of message tampering attack which already exploited successfully on Internet environment. The concept of SQL injection seems to be quite simple and can be launched in any

application that creates and executes SQL statements on-the-fly. This attack is not only targeting in data modification, but also in the downfall of database services to cause a DoS. The utilization of Web interfaces for the provision of value-added services in IMS-AS makes this attack more attractive to the potential perpetrators. SQL injection in SIP can be triggered every time a SIP network entity (e.g. SIP UA, SIP Proxy) is asking for authentication. So, in case a SIP network element requests authentication, the User Agent (UA) on behalf of the authorized user computers the appropriated credentials based on the HTTP Digest mechanism. The result of this computation (credentials) is included in the message's authorization header. Then the message is forwarded to the proxy server, which has to authenticate the received message. Thus, it recalculates user's credentials using the user's password stored in the "Subscriber" table. To accomplish this task, it generates an SQL statement of the following syntax: "SELECT password FROM subscriber WHERE username='userA' AND realm='130.10.0.23'". In case a malicious user tries to launch an attack in the SIP architecture, exploiting SQL injection, he spoofs the SIP message and inserts the malicious SQL code in its Authorization header. This message can be any SIP message requiring authentication by a SIP server. The code can be embodied in the username or in realm fields in the Authorization header. As soon as the proxy receives a SIP message with an infected Authorization header, it will generate and execute the dangerous SQL statement which may delete or modify data in the database. As we know that IMS-AS integrates the HTTP Servlet container, the attacker can also utilize the HTTP message to launch the SQL injection attacks [9].

2) *Time Independent Message Flows Attack* : The IMS Application Server employs SIP for signaling and the SIP protocol specification describes methods to end or terminate session, cancel an invitation, redirect a call and update session parameters. But SIP specification does not include any specific security mechanisms. It is very likely that attacker will try to exploit any security vulnerability in the SIP methods and cause DoS to the provided service. For example faked BYE message can be used by the attacker to tear down established session immediately. Moreover, attacker will try to discover possible security flaws in the applications or take advantage of existing protocol holes similar to attacks launched against Internet applications and services exploiting vulnerabilities at the signaling application level.

a) *The BYE Attack*: The BYE request is used to terminate an established session. An attacker possibly can utilize the BYE request to tear down a session. In figure 5 the blue broken line represents the established signaling session between UA1 and UA2 via a SIP proxy, the other blue line stands for the bidirectional media transmission. Later, the attacker sends a faked BYE message, which will be proxied to UA1, to the SIP proxy. UA1 will believe that it is UA2 who wants to tear down the connection by sending the BYE message. UA1 will stop its outward RTP flow immediately, while UA2 will continue to send RTP packets to UA1, since UA2 has no notion

that the connection should be terminated. The unidirectional media transmission caused by the attack is described in the figure 5 using the red line from UA2 to UA1. To launch this kind of attack, the attacker needs to learn all necessary session parameters. This can be accomplished either by sniffing the network or performing a man-in-the-middle attack to insert a BYE request into the session.

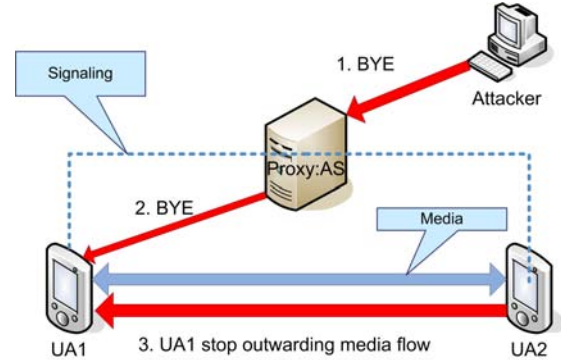


Fig. 5. The BYE attack

b) *The RE-INVETE Attack*: A successful INVITE request established both a dialog between two user agents and a session using the offer-answer model. The goal of RE-INVITE method is to modify the actual session. The modification can involve changing addresses or ports, adding a media stream, deleting a media stream, and so on. Therefore the attacker can launch a DoS attack via sending a forged RE-INVITE message to enforce any unauthorized modification.

There are still some SIP methods which can be employed to launch attacks e.g Update, and more attack possibilities. But we will focus only the above attacks to secure IMS Application Server.

VI. IMS-AS SECURITY MECHANISMS

The proposed security mechanisms focus on protecting the IMS Application Server from attacks contained in SIP messages. Intruder uses two approaches to launch attacks relying on SIP messages. The first one is to intercept and fake the SIP messages exchanged between legitimate UAs and the Application Server. The other approach is to send malicious SIP messages directly to AS e.g. the intruder transmits a SIP message, which contains SQL-Injection. Therefore, we introduce a two tiers security mechanism shown in figure 6 to safeguard IMS-AS.

The first tier is to utilize the TLS (Transport Layer Security) [10] mechanism to secure the communication channel. TLS mechanism can exclude the intruder from intercepting and forging the exchanged SIP messages. It should be noted that the SIP signaling path is hop-by-hop and from security point of view this means that the whole signaling path between the UA and the AS must be secured by the TLS [10]. TLS has many advantages over IPsec and successful introduction of the protocol in the Internet has proved its usability and effectiveness. But the biggest difficulty with TLS is that it does

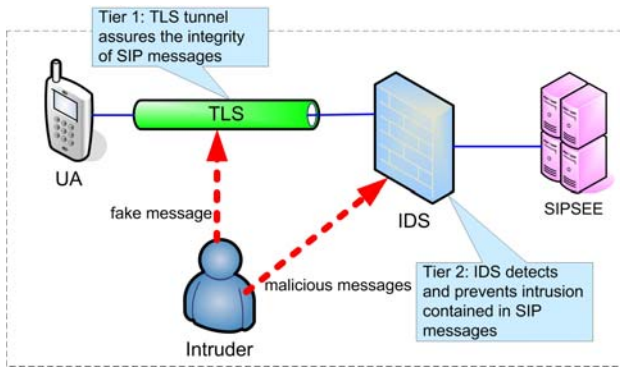


Fig. 6. Two Tiers Security Mechanism

not run over UDP which is usually used by SIP entities. In the future, it's possible that we will introduce other mechanisms as supplement or substitution to the TLS.

The second tier is to deploy an Intrusion Detection System for IMS Application Server. The task of the IDS is to detect and prevent attacks which can not be barred by the first tier technique e.g. Bob is a legitimate as well as malicious user. He is qualified to use the TLS communication channels to send SIP messages to the AS. As a malicious user, Bob intends to launch SQL-Injection attack to drop a table in the database of AS. SIP provides a challenge-based mechanism for authentication that is based on authentication in HTTP and simple challenge-based authentication as illustrated in figure 7.

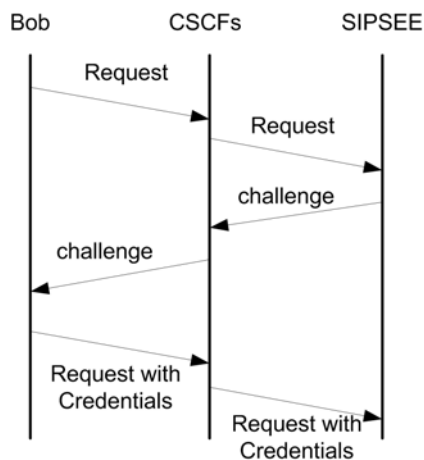


Fig. 7. Example of Challenge-based Authentication

At the end of authentication, Bob can inject SQL statement into the Request with Credentials. The authorization Header of the injected Request may look like:

```
Authorization:
  Digest username="Bob';drop table films;' ",
  realm="example.com",
  ... ..
```

If the Application Server is not equipped with IDS, the above Request can cause loss of data, namely drop table "films" in AS. In the next we will explain the architecture,

methodology and performance of our IDS.

VII. INTRUSION DETECTION SYSTEM FOR IMS-AS

Intrusion detection is performed by analyzing one or more input event streams, looking for the manifestation of an attack. Historically, detection has been achieved by two different approaches: anomaly detection or misuse detection. Anomaly detection relies on models of the normal behavior of a computer system. These models can focus on users, applications, or the network. Behavior profiles are built by performing a statistical analysis on historical data or by using rule-based approaches to specify behavior patterns. An anomaly detector compares actual usage patterns against established profiles to identify abnormal patterns of activity. Misuse detection systems take a complementary approach. Misuse detection systems are equipped with a number of attack descriptions. These descriptions are matched against a stream of audit data to find evidence that the modeled attack is occurring. Our intrusion detection system is a misuse detection system.

A. Architecture of IMS-AS IDS

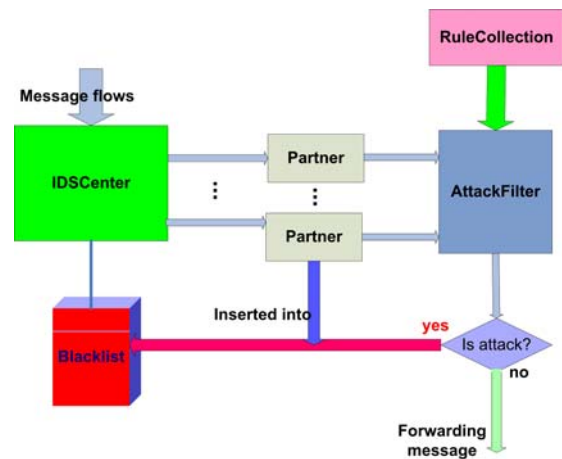


Fig. 8. IDS Architecture

The architecture of IDS is shown in figure 8. From design point of view IDS is illustrated in figure 6 which should be deployed before the AS as application firewall. The AS can be roughly divided in two components: SIPStack and SIPServer. The prototype of the IDS is integrated with AS and is placed between these two components. In the architecture of IDS, the IDSCenter has interface for exchanging SIPServletRequest and SIPServletResponse with the SIPStack and the SIPServer. The incoming SIP messages forwarded by the SIPStack and the outgoing messages generated by the SIPServer first pass through the IDSCenter, which maintains a list of Partner. Each Partner represents a UA which is communicating with AS. The Partner is identified with the corresponding UA's SIP-URI or SIPs-URI. The life cycle of a Partner begins at receiving the first SIP message from a new UA and ends with the termination of the dialog. Partner is a stateful object. On receipt of a SIP message passing through the IDSCenter,

it updates the state of the corresponding Partner, or creates a new Partner if the corresponding Partner did not exist. The IDS also contains a Blacklist storing the SIP-URI of the detected malicious user, which is prevented by the IDS from sending messages to and receiving messages from AS. The IDFilter carries out the comparison. The component "RulesCollection" is responsible for reading the attack patterns from the description file for attacks and loading them into a list in runtime.

B. Attack Detection Methodology

IDS can distinguish two types of detection based on classification of attacks mentioned in the previous section. The detection behavior is illustrated with a control flow graph in figure 9. First the IDS will create or update a Partner corresponding to arrived SIP message. Then the message should be compared with the defined rules of time-independent attacks. If message matches with the defined rule, it means that it is an attack and an intruder is detected. Each Partner has a timer that enables the Partner to carry out a periodic self-checking. By this self-checking, the Partner is compared with the defined rules of time-dependent attacks.

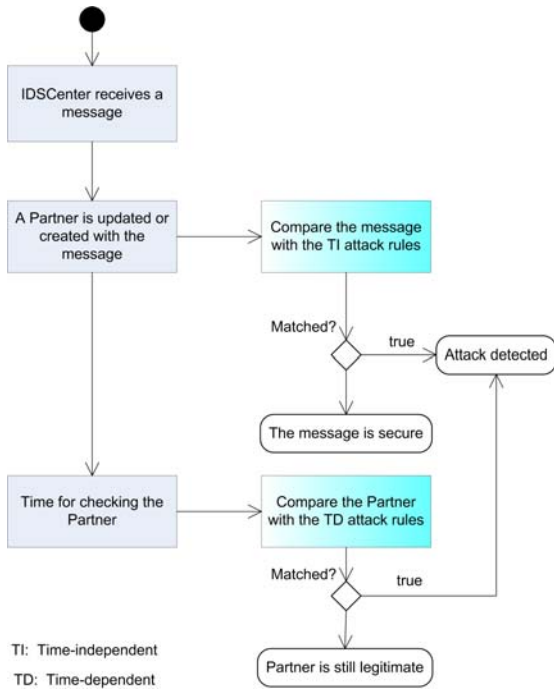


Fig. 9. Control Flow Graph for Intrusion Detection

C. Attack Patterns

We have utilized XML to build these attack scenarios. The description model consist of a two-level hierarchy. At the basic level we present XML elements to describe the type and name of an attack. The sub elements are introduced to describe the parameters of an attack. For instance, an INVITE flooding attack illustrated in figure 10.

```
<rule name="INVITE flooding" type="flood">
  <sip-method method="INVITE" />
  <number>100</number>
  <interval>60</interval>
  <alert content="INVITE message flood" From="">
  <white-list>158.88.168.168</white-list>
</rule>
```

Fig. 10. Example of Attack Pattern

D. IDS Performance

We have utilized SIP Forum Test Framework (SFTF) [11] to perform the functionality and performance tests. The test environment is a PC with P4 2GHz CPU, 512 MB memory. The test tool and the IMS AS are installed in the same PC. Through the test, it is verified that the prototype fulfills all the functionality of designed requirements. The single performance metric is the delay introduced by the IDS. In our experiments, the delay refers to the total delay (D) for a transaction introduced by IDS. The total delay (D) as shown in figure 11 consists of the delay for incoming request (D1) and the delay for outgoing response (D2). In each performance

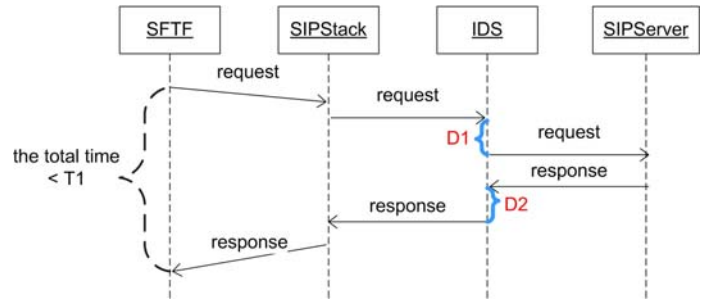


Fig. 11. The sequence diagram for supposed transaction in testing

testing, the test tool (SFTF) sends a couple of INVITE requests to the IMS AS. The number of INVITE requests starts from 10 and the amount is increased of 10 in the following tests. D1 and D2 are the average delays needed by IDS to process the incoming request and outgoing response. They are calculated with the following function:

$$(Averagedelay) = \frac{\sum_{k=1}^n T_k}{n}$$

In each test, for each incoming request we record the time interval when it arrives to the IDS and then is forwarded to the SipSeeServer. For each outgoing response we record the time interval when it arrives to the IDS and then it is forwarded to the SIPStack. Both for incoming request and outgoing response, each time interval T_k between the two time points is the individual detection delay. k is the index of message and n is the sum of messages which the IDS receives. Theoretically, greater the n is, the more average delay will be. But it should be also taken into account, that each message occupies some resources of computer and too much coexisting messages may make computer slow down for some

possible additional tasks. In the case of our tests, n is the amount of requests in each test. From performance test, we have obtained the chart shown in figure 12. The blue curve marked (\diamond) represents the total delay (D) varies in the range 1.6 ms to 9.36 ms. The default estimate round-trip time ($T1$) is 500ms as referred in SIP RFC3261 [12]. Usually the IDS does not cause the retransmission of SIP messages because the average delay introduced by the IDS is very shorter than the $T1$. The broken line describes trend of the total delay. This trend shows the more incoming requests are, the greater the delay introduced by the IDS is.

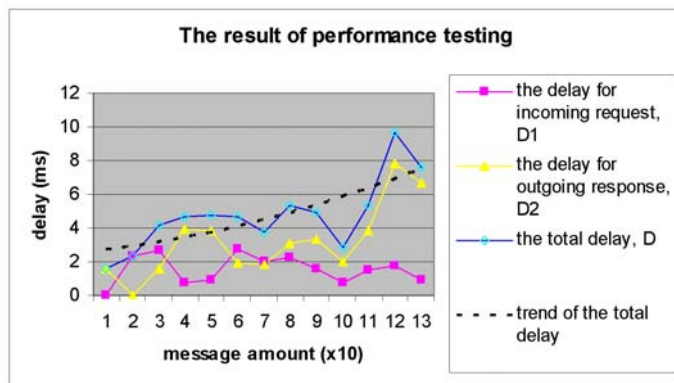


Fig. 12. Delay and Performance Test

VIII. CONCLUSION

In this article we have explored different attacks possibilities for IP Multimedia Subsystem (IMS) Application Server as a part of Secure Service Provisioning (SSP) Framework [13] of IMS playground within 3Gb Testbed at Fokus Fraunhofer. The architecture of IMS Application Server is SIP based and facing many security threats because SIP is text-based, not a mature protocol and standardized recently. We have also proposed security solutions and mechanisms to secure and protect the IMS-AS.

ACKNOWLEDGMENT

This research work is funded and supported by BMBF (German Federal Ministry of Education and Research) under project MAMS (Multi-Access, Modular-Services Framework) in article AP560 which describes "Security in Network Abstraction and Open IMS".

REFERENCES

- [1] Securitytracker, "Tracker of the latest vulnerabilities", <http://securitytracker.com/learn/statistics.html>
- [2] Jetty, "Java HTTP Server and Servlet Container", <http://jetty.mortbay.org/jetty/index.html>.
- [3] Anders Kristensen, et al, "JSR 116: Sip Servlet API", March 2003.
- [4] 3GPP, "The 3rd Generation Partnership Project", <http://www.3gpp.org/>
- [5] Miikka Poikselkae, Georg Mayer, Hisham Khartabil and Aki Niemi, "IP Multimedia Concepts and services in the Mobile Domain", John Wiley & Sons, Ltd, 2004.

- [6] K. Knttel, T.Magedanz, D. Witszek, The IMS Playground @ Fokus an Open Testbed for Next Generation Network Multimedia Services, 1st Int. IFIP Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (Tridentcom), Trento, Italian, February 23 - 25, 2005, Proceedings pp. 2 11, ISBN 0-7695-2219-x, IEEE Computer Society Press, Los Alamitos, California.
- [7] IMS Playground, <http://www.fokus.fraunhofer.de/ims/>
- [8] SIP Express Router (SER), "a high-performance, configurable, free SIP (RFC3261) server", <http://www.iptel.org/ser/>
- [9] Low Cost Tools for Secure and Highly Available VoIP Communication Services (SNOCER), "an research project supported within the Sixth Framework Programme of the EU Commission", <http://www.snocer.org/>
- [10] T.Dierks, C.Allen, "The TLS Protocol", Version 1.0, January 1999, RFC 2246, <http://rfc.net/rfc2246.txt>
- [11] J.Peterson, "S/MIME Advanced Encryption Standard (AES) Requirement for the Session Initiation Protocol (SIP)", July 2004, RFC 3853, <http://rfc.net/rfc3853.html>.
- [12] SIP Forum Test Framework, "A test software for SIP", <http://www.sipfoundry.org/sftf/>
- [13] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler, "SIP: Session Initiation Protocol", RFC3261, June 2002.
- [14] M.Sher, T.Magedanz, "Secure Service Provisioning Framework(SSPF) for IP Multimedia System and Next Generation Mobile Networks", in Proceeding 3rd International Workshop in Wireless Security Technologies(IWWST05), London, U.K., April 2005,pp. 101-106, Available: <http://www.iwwst.org.uk>