

Introducing the GBA Covert Channel in IP Multimedia Subsystem (IMS)

Ghader Ebrahimpour^(✉), Siavash Khorsandi, and Ali Piroozi

Computer Engineering and IT Department,
Amirkabir University of Technology, Tehran, Iran
{ebrahimpour.g, khorsand, falipiroozi}@aut.ac.ir

Abstract. IP Multimedia Subsystem (IMS) is an access-agnostic IP-based technology to provide a wide range of real-time communication and multimedia services for end-users. Security is one of the main challenges in large-scale deployment of IMS. In this paper, we have focused on a particular security hole in the authentication process in IMS. We introduce and discuss the application scenarios of a new covert channel in Generic Bootstrapping Architecture (GBA). GBA-based authentication is part of GAA standard based on shared-secret published by 3GPP. It is demonstrated that this vulnerability can be established to gain unauthorized access to unlawfully utilize the services.

Keywords: IMS · GBA · Covert channel · Authentication protocol · Bootstrapping procedure

1 Introduction

IP Multimedia Subsystem (IMS) is an international standard aimed to replace outgoing TDM-based technologies to provide multimedia communication and content on an access-agnostic basis. It is to unify digital communication infrastructures regardless of their access technologies (cellular, WiFi, PSTN and etc.) [2]. IMS is introduced in 3GPP release 5 to 8 in partnership with TISPAN as a realization of Next Generation Networks (NGN) concept [1]. IMS is defined on top of IP networks and is widely utilizing IP-based protocols published by IETF. Security is one of the main challenges facing IMS deployments. User authentication is one of the basic services needed to control user access to resources and services. A number of different approaches for authentication are proposed. GBA is the standard-based authentication approach based on shared secret used for authentication and key agreement.

Covert channel is a security threat where unauthorized information flow is established between two participating parties in a single system or a network of connected systems [5]. As the complexity of computer systems grows, various forms of covert channel have been proposed, such as covert channel in IP, SIP, and other network protocols [12, 13, 15].

Covert channel attacks can potentially create high security risk, because they create an illegal communication channel between processes that are not allowed to communicate by the security policy. Although covert channel implementation in general is

very difficult and sometimes quite complicated if implemented correctly it can lead to severe information leakage [14].

Research in covert channels can be placed in three categories: detection and explaining of covert channels, measuring the amount of information that is transferred through these channels and mitigating covert channels effects [11]. Since the security is one of the major challenges in IMS, therefore investigation of covert channels in these networks will also be important. To best of our knowledge, research on covert channels in IMS has so far been limited to IP and SIP protocols and other aspects of system operation and in particular the authentication process is not investigated.

In this paper we introduce a new covert channel that uses the GBA authentication process for transferring information. Unlike other covert channels, the GBA covert channel uses the user authentication protocol to transfer data. We will explain a scenario in which proposed covert channel has been used for impersonation, in order to gain unauthorized access to services.

The rest of this paper is organized as follows. The concept and various categories of covert channels are described in Sect. 2. In Sect. 3 we will take a glimpse to IMS and its architecture. Section 4 will discuss the GBA, a method for authentication in the IMS, components and reference points of it. We present the GBA covert channel, in Sect. 5. Finally, the last section concludes the paper.

2 Covert Channels

Covert channel refers to any communication channel that can be used by a process to illicitly transfer information that violates the system's security policy. In most cases resource sharing can be a potential vehicle to create a covert channel. There are many threats that should be considered. One of these threats that have not paid enough attention is covert channels [6]. The TCSEC defines two kinds of covert channels: covert timing channels and covert storage channels. These types of covert channels are further elaborated in the following sections.

2.1 Covert Storage Channels

Covert storage channel is a method of communication, in which one process writes in a storage location and another process reads it, these actions can be done directly or indirectly [4]. In other word, one process writes to a shared resource and another process reads from it. Storage channels can be established between processes that parties are in a single computer or networked computers [5]. Printer queue is an example of this type of channel. Two processes are running, when a process with a higher security level wants to sends 1 to a low level process fills up the printer queue, and when it wants to sends 0 leaves the queue. The other process pulls the printer queue to receive data [6].

2.2 Covert Timing Channels

Covert timing channels is a method of communication in which a process sends information to another process by timing modulation in its use of one resource, as such the second process would gain the information by observing the changes in the timings compared to the normal scenario [4].

3 IMS Architecture

In previous section we introduced covert channels. In this section we present IMS, its different layers and architecture. Figure 1 gives an architectural overview of the IMS. IMS has at least three different layers: transport, control and service. Each layer has a special purpose and interacts with its adjacent layer through abstract standard interfaces [2].

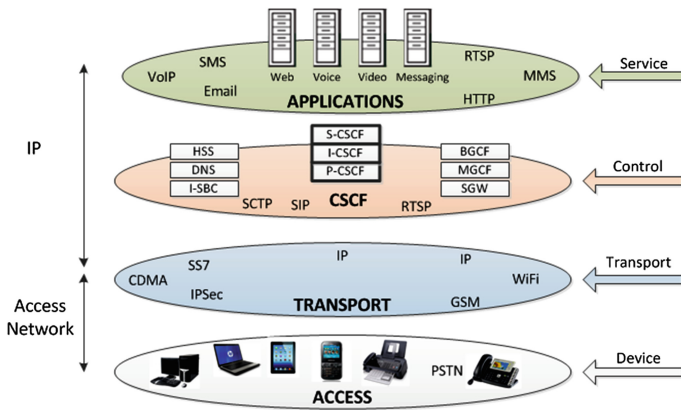


Fig. 1. IMS architecture [2]

The transport layer abstracts the actual access network's functions from the IMS architecture, i.e., it (transport layer) is the intersection point of access layer and the IP network layer. Every entity above the transport layer is based on IP, while the layer below this layer is not based on IP and can use every protocol. The control layer is responsible for authentication, routing and distribution of IMS traffic between the transport and the service layer. The majority of the traffic in control layer is Session Initiation Protocol (SIP)-based. The main component of this layer is the Call Session Control Function (CSCF) which simplifies the correct interaction among the application server, media servers and the Home Subscriber Service (HSS). The HSS is a database of subscriber and multimedia session control information. Application services are provided by the service layer. New IMS-based service and applications are in this layer, too. This layer is the final layer that gives the IMS architecture the power and flexibility to develop new services.

The power of IMS architecture is that, the transport and control layers can effectively separate the provided services from the access networks, so the services can be provided without concerning about the access network details or how the service reaches the device. The IMS devices don't need to know the service's place and the access network that currently is used, because, the service is easily accessible. This disassociation is a key characteristic that give the next wave of IMS services the ability to transfer to any device [2].

The focus of the IMS is S-CSCF control layer's component that controls registration process, routing, storing session status and service's profiles. When a UE sends a registration request to a S-CSCF, it takes UE's authentication information from HSS and based on this information, produces a question for UE. The UE sends its answer to the S-CSCF. After verification of the UE's response, the S-CSCF accepts the registration and monitors UE's registration status. Subsequently, the user can use the IMS services.

4 Authentication in IMS

Several authentication protocols have been introduced in IMS. The AKA protocol is the main mechanism for authentication and key establishment in this network [3]. The AKA mechanism is used in IMS known as IMS-AKA. IMS-AKA is performed through the SIP protocol. GBA is another authentication method approved by 3GPP. GBA is a mechanism that enables the authentication of a user if they own a valid identity in an HSS. GBA is a generalization of IMS-AKA and supports arbitrary application protocols (such as HTTP), while IMS-AKA only supports SIP. GBA authentication method was introduced for services that do not support SIP. Some services might be available over HTTP protocol. For example, it shall be possible to manage the data on the Presence Service over the HTTP. Other services such as conferencing might be available using HTTP. So access to services over HTTP can be done in a secure manner [7]. GAA is a 3GPP solution to the growing need for authentication and key agreement between the UE and the network and the GBA is one of possible ways to use the GAA [17].

4.1 GBA

GBA performs authentication by creating a challenge and confirming that the UE responses is same with the calculated values in the HSS. The GBA consists of four entities: UE, NAF, BSF, and HSS, which are shown in Fig. 2 [8].

Network Entities of GBA. In this section we describe the network entities of GBA.

User Equipment (UE): UE is a device used by user to access services that supports NAF-specific protocols. It is divided into two components: Mobile Entity (ME) and Universal Integrated Circuit Card (UICC) [16].

Bootstrapping Server Function (BSF): BSF is hosted in a network element under the control of a Mobile Network Operator (MNO). BSF, HSS, and UEs participate in GBA in which a shared secret is established between the network and a UE by

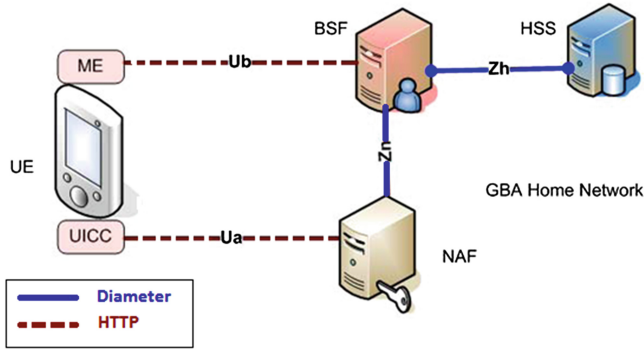


Fig. 2. Network entities of GBA

running the bootstrapping procedure. The shared secret can be used between NAFs and UEs, for example, for authentication purposes [8]. The BSF will restrict the applicability of the key material to a set of NAFs by using a suitable key derivation. The lifetime of the key material is set according to the local policy of the BSF [9]. Network Application Function (NAF): NAF is hosted in a network element. The UE requests a service by contacting a NAF. The NAF has the functionality to communicate with the subscriber's BSF. It should be able to acquire shared key material established between the UE and the BSF during the bootstrapping procedure [9]. After the bootstrapping procedure has been completed, the UE and the NAF can run some application specific protocol which the authentication of messages will be based on session keys generated during the bootstrapping procedure between the UE and the BSF [8].

Home Subscriber Server (HSS): HSS is a database that stores GUSSs (i.e. user authentication information).

GBA Reference Points. GBA's interfaces are as follows:

Ub: The reference point Ub resides in between the UE and the BSF and provides mutual authentication between them. It allows the UE to bootstrap the session keys based on 3GPP AKA infrastructure.

Ua: The Ua interface provides the application protocol and has been secured using the keys agreed between the UE and the BSF during the execution of bootstrapping procedure over reference point Ub.

Zh: The reference point Zh is used between the BSF and the HSS. It allows the BSF to fetch the required authentication information and all GUSSs from the HSS.

Zn: The reference point Zn is used by the NAF due to fetching the key material exchanges between the UE and the BSF during the bootstrapping procedure over the reference point Ub [8].

Zh and Zn Interfaces are based on the Diameter protocol and Ua and Ub Interfaces are based on the HTTP protocol.

5 Covert Channel in GBA

GBA can be divided into two procedures: Bootstrapping Authentication Procedure (BAP) and Bootstrapping Usage Procedure (BUP). The BAP consists of authenticating the client to the home network and deriving the key material. In the BUP, the UE informs the NAF what key to use and the NAF then fetches this key from the BSF [10].

In the GBA's covert channel, BUP is used in order to illegally transform information. Obtaining information by the attacker calls for two steps: the first one is running BAP, and the second one is using the values obtained during the BAP to get confident information from the NAF.

5.1 Bootstrapping Authentication Procedure (BAP)

Bootstrapping Authentication Procedure (BAP) is as follows:

Before the UE and the NAF start their communication, they should decide whether they want to use GBA or not. When the UE wants to interact with the NAF, starts a communication on the Ua interface with the NAF (without GBA parameters). If the NAF needs shared keys from GBA and UE's request does not have them, the NFA, responds with a bootstrapping initiation message.

When a UE wants to interact with a NAF and it knows a BAP procedure is required, it should first perform a bootstrapping procedure. Otherwise, the UE will perform the bootstrapping authentication, just if a bootstrapping initiation message has been received or if the lifetime of the key in UE has expired [8]. The UE sends a HTTP request to BSF and the BSF retrieves the complete user's GUSSs and an AV (Authentication Vector) from HSS (over the reference point Zh) [9].

After that, the BSF forwards RAND and AUTN to the UE in a 401 message (without the CK, IK and XRES) [9]. It should be mentioned that, every AV includes CK, IK, RAND, XRES and AUTN, as the following relation:

$$AV = RAND \parallel AUTN \parallel XRES \parallel CK \parallel IK$$

Where RAND is a random challenge, AUTH is a network authentication token, XRES is an expected authentication result, CK is a session key for encryption, and IK is a session key for integrity.

The UE checks the AUTN to verify that it comes from an authorized network. The UE calculates RES (user authentication result), IK and CK, too [17]. The result is two session keys, IK and CK in the BSF and the UE. The UE sends another HTTP request to the BSF containing the digest AKA response which is calculated using RES [9]. The BSF authenticates the UE by checking the digest AKA response. The BSF creates Ks by concatenation of CK and IK, it also produces B-TID which is used for binding subscriber identity to the key in reference points Ua, Ub and Zn. After, the BSF sends a 200 OK message to the UE which contains a B-TID to indicate the success of authentication and the key's lifetime. The UE produces Ks by concatenation of CK and IK. The BSF and the UE use Ks to create Ks-NAF which is used for securing the Ua reference point [9]. The Ks-NAF is computed as follows:

$$Ks_NAF = f_{KD}(Ks, "gba-me", RAND, IMPI, NAF-ID)$$

f_{KD} is the key derivation function and is implemented in the ME. Key's derivation parameters consist of the user's IMPI, NAF-ID, and RAND. NAF-ID consists of the full name of the NAF's DNS concatenated with the Ua security protocol identifier. The BSF and the UE should store Ks for future use until the Ks' lifetime has expired or it is updated [9].

5.2 Bootstrapping Usage Procedure (BUP)

BUP is the second procedure for using GBA and starts with TLS handshake between the UE and the NAF. When NAF and UE decide to use GBA, whenever the UE wants to interact with the NAF, it starts a communication with NAF on the reference point Ua and sends B-TID to it. So, the NAF can receive related keys from the BSF. The NAF starts a communication with BSF over the reference point Zn. The NAF requests the key corresponding to the B-TID that the UE has sent to it. With this request, the NAF also sends its public hostname to the BSF, and the BSF shall be able to verify that the NAF is authorized to use that hostname.

The BSF creates the key required to protect the protocol used over the reference point Ua, using Ks and key derivation parameters; Afterwards, the BSF sends the key to the NAF for the requested key Ks-NAF, as well as the bootstrapping time and the lifetime of the key. If the key identified by the B-TID is not available in BSF, the BSF will notify NAF with its response and the NAF will send a request to the UE for a bootstrapping renegotiation. NAF will continue its communication with UE by the protocol used over the reference point Ua. When the run of the protocol over the reference point Ua is complete, the purpose of bootstrapping is satisfied because it allows the UE and NAF to use the reference point Ua in a secure way [9].

5.3 The GBA Covert Channel Operation

In the GBA covert channel, NAF sends information to UE illegally using the bootstrapping usage procedure (BUP). This covert channel has two steps: First, a user performs the bootstrapping usage procedure (BUP) and receives a B-TID from BSF. Second, the user communicates with the NAF and receives information over covert channel. The second step is as follows: the user sends an HTTP request with required parameters for authentication, received in the first step, to the NAF. The NAF sends a bootstrapping renegotiation message if it wants to transfer bit 1, and a "200 OK", which is used for success authentication, if it wants to transfer bit 0. Thus, the user can send requests and receive responses to obtain information from the NAF.

5.4 A Simple Scenario for Using GBA Covert Channel

Suppose the following scenario shown in Fig. 3. User A is a typical user and has their B-TID and Ks_NAF. To use services, if the user A has not been authenticated, they

should perform the BAP and then send a request to the NAF to be authenticated. We suppose that the user A has already performed BAP, and now wants to use services. As shown in Fig. 3, in BUP, the user A sends a request to the NAF containing B-TID. The NAF receives this request and communicates with subscriber's BSF to acquire shared key material (such as Ks_NAF) established between the UE and the BSF during the bootstrapping procedure. At this point, the NAF has both A's B-TID and Ks_NAF . In the next step, the NAF sends an HTTP response with a "401 Unauthorized" status code to the user A and then receives and verifies the A's response. If the A's authentication succeeds, the NAF provides the services to the user A and stores A's B-TID and Ks_NAF for future connections.

Suppose M is an attacker that communicates with the NAF through the covert channel. The user M can use the covert channel to receive the A's authentication values from the NAF (Fig. 4). When the user M obtains the A's B-TID and Ks_NAF through the covert channel, they can use them for authentication and use of services. Now the user M can use services which has been provided for the user A as the identity of user A.

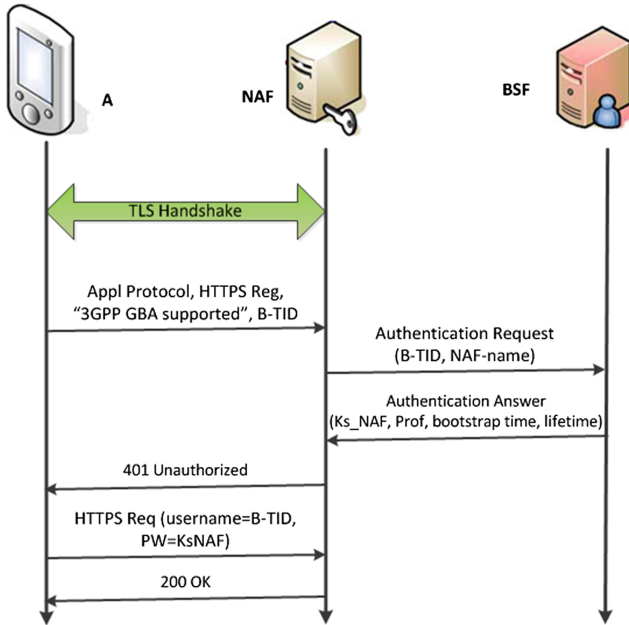


Fig. 3. Obtaining A's B-TID and Ks_NAF by NAF

5.5 Other Form of Covert Channel

The covert channel introduced in Sect. 5.1 can be done in other forms. In this subsection, another form of the GBA covert channel is introduced below:

The attacker (M) has two B-TIDs and their corresponding Ks_NAFs . One B-TID has expired, and the other one has not. The attacker establishes two distinct connections with NAF and sends the expired B-TID through the first connection and the other

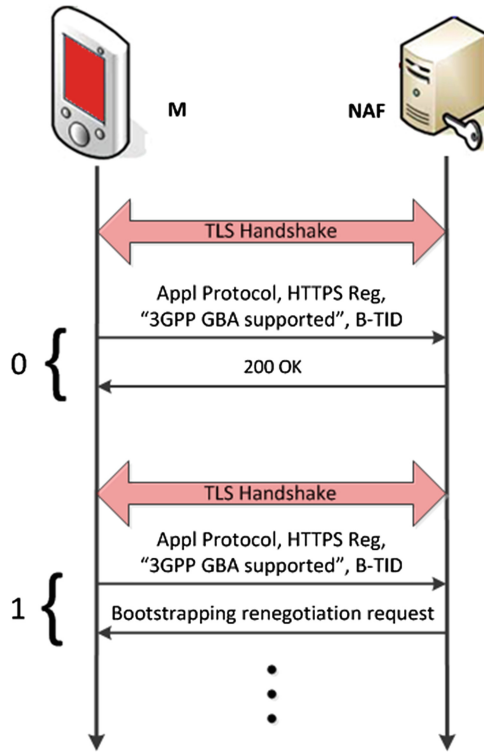


Fig. 4. Information transferring by covert channel

through the second. If the NAF wants to transfer bit 1, it sends an HTTP response with a “401 Unauthorized” status code to the M through the first connection and nothing through the other; and if the NAF wants to send bit 0 to the M, it will respond to the second connection and discards the first one. Thus, NAF will send information to M through the covert channel.

5.6 Elimination

One of the reasons that led to this covert channel is the lack of BSF in the user’s authentication process between NAF and UE. On the other hand, for user authentication, both B-TID and Ks_NAF are provided for NAF. A user that has these two values can authenticate themselves to the NAF, so if the NAF sends B-TID and Ks_NAF to the attacker in an unintended manner, the attacker can impersonate themselves as a typical user. Therefore, if one of these values, which is probably Ks_NAF, is not provided for the NAF, the NAF cannot send those values to an unauthorized user.

6 Conclusion

In this paper we described one of the authentication procedures in IMS and introduced a new covert channel in the authentication process. This covert channel misuses the

authentication procedure in GBA and sends information from one network's entity to another which can be done in different forms. We also designed a scenario that led to an impersonation attack. There can be some other scenarios resulting in severe information leakage which in turn causes other forms of attacks. Therefore, in order for the information leakage to be prevented, some prevention methods should be considered.

References

1. 3GPP, Overview of 3GPP Release 5 - Summary of all Release 5 Features. 3GPP - ETSI Mobile Competence Centre, Technical report (2003)
2. Salchow, K.: Introduction to the IP Multimedia Subsystem (IMS): IMS Basic Concepts and Terminology. Whitepaper F5 Netw. Inc. (2007)
3. 3GPP TS 33.102 V11.5.0., Technical Specification Group Services and System Aspects; 3G Security; Security architecture (2012)
4. Department of Defense. Department of Defense Trusted Computer System Evaluation Criteria, DOD 5200.28-STD (The Orange Book) edition (December 1985)
5. McFail, M.: Covert storage channels: A brief overview. In: PACISE Conference, Bloomsburg, PA (2005)
6. Pennington, E., Oblitey, W., Ezekiel, S., Wolfe, J.: An Overview of Covert Channels. Covert Channels Research Group Computer Science Department IUP, Indiana
7. 3GPP TS 33.222: Generic Authentication Architecture (GAA); Access to network application functions using Hypertext Transfer Protocol over Transport Layer Security (HTTPS), (Release 12) (2013)
8. 3GPP TS 33.220: Generic Authentication Architecture (GAA); Generic bootstrapping architecture, (Release 12) (2013)
9. Ilyas, M., Ahson, S.A.: IP Multimedia Subsystem (IMS) Handbook. CRC Press, Boca Raton (2008)
10. Olkkonen, T.: Generic Authentication Architecture. In: Security and Privacy in Pervasive Computing, Seminar on Network Security, Espoo 2006
11. Millen, J.: 20 years of covert channel modeling and analysis. In: Proceedings of the 1999 IEEE Symposium on Security and Privacy, pp. 113–114 (1999)
12. Mazurczyk, W., Szczypiorski, K.: Covert channels in SIP for VoIP signalling. In: Jahankhani, H., Revett, K., Palmer-Brown, D. (eds.) Global E-Security, pp. 65–72. Springer, Heidelberg (2008)
13. Cabuk, S., Brodley, C.E., Shields, C.: IP covert channel detection. ACM Trans. Inf. Syst. Secur. TISSEC **12**(4), 22 (2009)
14. Gallagher, P.R.: A guide to understanding covert channel analysis of trusted systems, National Computer Security Center, USA (1993). <http://fas.org/irp/nsa/rainbow/tg030.htm>
15. Zander, S., Armitage, G., Branch, P.: Covert channels in the IP time to live field. In: Proceedings of Australian Telecommunication Networks and Applications Conference (ATNAC) (2006)
16. Kasera, S., Narang, N.: 3G Networks. Tata McGraw-Hill Education, New York (2004)
17. 3GPP TS 33.919; Generic Authentication Architecture (GAA); System description, (Release 11) (2012)

Computer Networks and Distributed Systems
International Symposium, CNDS 2013, Tehran, Iran,
December 25-26, 2013, Revised Selected Papers
Jahangir, A.H.; Movaghar, A.; Asadi, H. (Eds.)
2014, X, 169 p. 64 illus., Softcover
ISBN: 978-3-319-10902-2