Threat Modeling an Identity Management System for Mobile Internet

Cristina K. Dominicini[1], Marcos A. Simplício Jr.[1], Rony R. M. Sakuragui[1], Tereza C. M. B. Carvalho[1],
Mats Näslund[2], and Makan Pourzandi[3]
[1]Laboratory of Computer Architecture and Networks, PCS-EP, University of São Paulo – São Paulo, Brazil
{cdominic, mjunior, rony, carvalho}@larc.usp.br
[2]Ericsson Research - Stockholm, Sweden - mats.naslund@ericsson.com
[3]Ericsson Research, Open Systems Laboratory - Ville Mont-Royal, Canada - makan.pourzandi@ericsson.com

***Abstract*- The advent of Mobile Internet and Web 2.0 raised the need for identity-oriented and user-centric services. In recent years, many Identity Management Systems (IdMS) have been developed to allow users to safely control and reuse their identity attributes. Service providers and users rely on the trust that the mechanisms provided by the IdMS are secure. However, if an attacker succeeds in exploiting some vulnerability of an IdMS, all the services that rely on it will be compromised. Therefore, it is crucial to perform an extensive threat analysis to ensure a deep understanding of the security issues when designing, implementing and operating such systems. In this paper, we tackle this issue by presenting a threat model of an IdMS for Mobile Internet that is composed of two enabling technologies: GAA/GBA and OpenID.**

I. Introduction

The advent of Mobile Internet and Web 2.0 raised the need for identity-oriented and user-centric services. In recent years, this new range of services has received much attention, and many Identity Management Systems (IdMS) have been developed to allow users to safely reuse and control their identity attributes in different contexts [1]. An IdMS authenticates its users and, then, provides tokens that can be verified by service providers, acting as a trusted entity for authentication and authorization [2].

In Mobile Networks, each user is uniquely identified by the credential (often a smartcard) contained in his/her mobile phone and provided by his/her Mobile Network Operator. So, when considering the design of an IdMS for Mobile Internet, it is interesting to take benefit of already existing Mobile Network Operators' robust authentication mechanisms by integrating them with internet-based identity management standards and technologies.

An IdMS can enable secure services to Mobile Internet users, because it transfers the responsibility of performing identity management from several service providers and centralizes it in a secure and trustful system. Thus, service providers and users rely on the trust that the mechanisms for user authentication and identity management provided by the IdMS are secure. If an attacker succeeds in exploiting some vulnerability of an IdMS, all the services that rely on it will be compromised. Moreover, ensuring the security of such systems in the long run is an involved task, because security threats and identity frauds are becoming more common and complex. Therefore, it is crucial to perform an extensive threat analysis to ensure that security architects and developers have a deep understanding of the security issues when designing, implementing and operating an IdMS.

This paper presents a threat model of an Identity Management System (IdMS) for Mobile Internet that is basically composed of two enabling technologies: GAA/GBA (Generic Authentication Architecture/Generic Bootstrapping Architecture) [3] and OpenID [4]. For this purpose, we follow a methodology inspired by studies such as [5] and [6]: we first describe the system and its functionalities; then, identify assets that require protection; and, finally, determine the potential threats and corresponding vulnerabilities for each of the system components.

The paper is organized as follows. Section II provides a general description of related studies in the area. Section III gives an overview of GAA/GBA and OpenID technologies, and outlines how to integrate them in order to enable the main functionalities of the analyzed IdMS. Section IV introduces the threat analysis methodology adopted. In Section V, the threat modeling of the target system is performed. Section VI describes the corresponding vulnerabilities and provides suggestions on countermeasures. Finally, Section VII presents the conclusions and indicates some ideas for future work.

II. Related Works

There are relatively few studies in the literature focused on formal methodologies to threat modeling complex and networked systems. Among the most relevant works, we can mention [5] and [6]: in [5], the authors propose a generic methodology for threat analysis in Personal Networks; in [6], the authors investigate how threat modeling can be used as foundation for specifying security requirements, and outline an approach for identifying threats in networked systems. In comparison, it is more common to find works that provide a threat model for some specific system. Examples include the threat analysis of Web Services and Grids [7] [8], of an Identity Federation Protocol [9], and of the Secure Session Protocol for Web Services [10]. Regarding the target system, there are couple of works that analyze the vulnerabilities of the OpenID protocol [11][12][13][14]. However, these works normally focus on specific vulnerabilities and do not perform a comprehensive threat modeling. Furthermore, to the best of our knowledge, there is no initiative that focuses on the threat modeling of the interworking of GAA/GBA and OpenID technologies.

III. Background and Enabling Technologies

The target system integrates GAA/GBA and OpenID technologies to enable identity management services for Mobile Internet. This section describes these enabling technologies and their role in the system operation, thus providing the basis for the threat analysis of the subsequent section.

A. GAA/GBA

GAA (Generic Authentication Architecture) is a generic architecture for mutual authentication and key agreement (AKA) used in Mobile Network Operators' infrastructure. Its fundamental building block is the Generic Bootstrapping Architecture (GBA), which is specified by 3GPP (3rd Generation Partnership Project) in TS 33.220 [3]. GBA provides mechanisms that mobile applications can rely upon for authentication between servers and clients. The user authentication is possible if the user has a valid identity on the Mobile Network Operator (e.g., a USIM card).

The main components of GBA architecture are shown in Figure 1 and described as follows [15]:

- User Equipment (UE): the end user mobile phone.
- Network Application Function (NAF): the application server.
- Home Subscriber Server (HSS): the subscribers' database that contains the long-term keys for each subscriber.
- Bootstrapping Server Function (BSF): a trusted entity at the Mobile Network Operator which facilitates the authentication and the key agreement between the UE and the NAF.

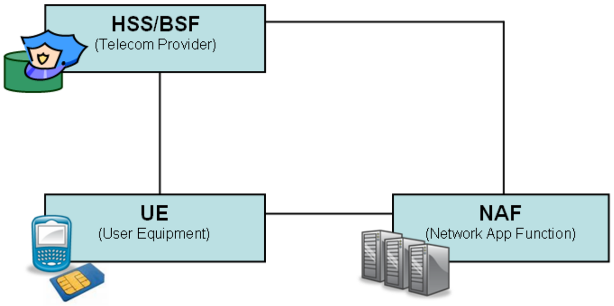

Fig. 1. GBA Network Model

B. OpenID

OpenID is an open, lightweight, decentralized, and HTTP-based single sign-on protocol, which aims at providing a single digital identity to log on to different websites while using only a web browser on the client side [16]. The user identifier is defined by an URI (Uniform Resource Identifier) or an XRI (Extensive Resource Identifier) and must be unique; for example, the identifier bob.idprovider.com defines the unique identity owned by "bob" at "IDProvider.com". The OpenID protocol is specified by the OpenID Foundation [4], which includes major companies in the digital identity industry such as VeriSign, Microsoft, Yahoo, and Google. Contrary to GAA, OpenID does not assume the presence of any specific (hard) credential such as a USIM.

The main components of OpenID architecture are shown in Figure 2 and described as follows [17]:

- User-Agent: the end user's Web browser which implements HTTP/1.1.
- OP (OpenID Identity Provider): an Authentication server which provides assertions allowing an end user to prove that he/she controls an Identifier.
- RP (Relying Party or Service Provider): a Web application that wants proof that the end user controls an Identifier.

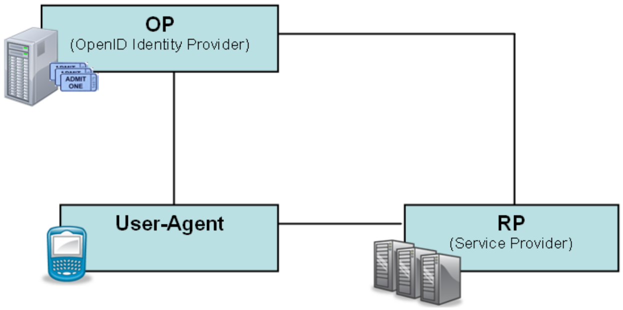

Fig. 2. OpenID Network Model

C. Interworking between GAA/GBA and OpenID

The interworking between GBA and OpenID has been studied in the literature [18][19]. The benefits of this approach relate to the possibility of combining the popular and flexible OpenID protocol for identity management with the strong two-factor authentication of GBA [18]. The architecture combines the GBA and the OpenID by joining the NAF functionality of GBA (see Figure 1) with the IdP functionality of OpenID (see Figure 2), as shown in Figure 3.

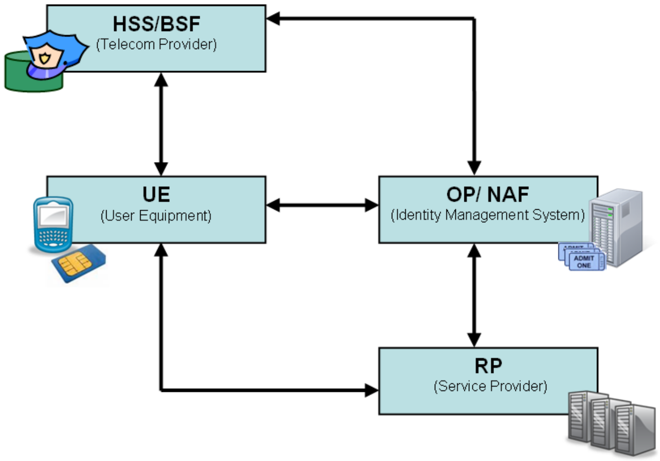

Fig. 3. GBA/OpenID Interworking.

The operation flow for a typical authentication is displayed in Figure 4, and described as follows [18][19].

1) The UE provides its OpenID identifier.
2) The RP determines the OP endpoint from the identifier provided. Then, the RP generates and caches a shared secret with the OP (if not previously available).
3) The RP sends an authentication request to the OP by redirecting the user's web browser to the OP's URL.
4) If no valid key is available (i.e., in the first use), the OP redirects the UE to the GBA server (HSS/BSF). Then, the UE starts the bootstrapping process, which will result in the creation of a Ks shared key (light-gray key in Figure 4). Using the Ks, the UE computes a NAF-specific key called Ks_NAF (dark-gray key in Figure 4).
5) The UE generates an application request to the NAF. The request carries an authorization header containing a transaction identifier and a challenge response (both received from the BSF).
6) Using the transaction identifier and its own ID, the NAF sends an authentication request to the BSF.
7) The NAF retrieves the shared key Ks_NAF (dark-gray key in Figure 4) from the BSF, and uses this key to check the challenge response from the UE.
8) The NAF redirects the browser to the return RP URL. The response header carries a signed authentication assertion.
9) The RP checks the assertion using previously defined shared secret (step 2).
10) If the check is successful, the UE is logged in the RP.

IV. Threat analysis Methodology

The following basic definitions come from RFC 2828 [20]:

- Vulnerability: a flaw or weakness in a system's design, implementation, or operation and management that could be exploited to violate the system's security policy.
- Threat: a potential for violation of security, which exists when there is a circumstance, capability, action, or event that could breach security and cause harm. That is, a threat is a possible danger that might exploit a vulnerability.

The threat analysis is a formal process of identifying, documenting and mitigating the security threats of a system [5]. The objective of a threat analysis is to provide the following abilities [21]:

- specify, design, and implement a computer system without vulnerabilities;
- analyze a computer system to detect vulnerabilities;
- address any vulnerability that may be present during the operation of the computer system;
- detect attempted exploitations of vulnerabilities.

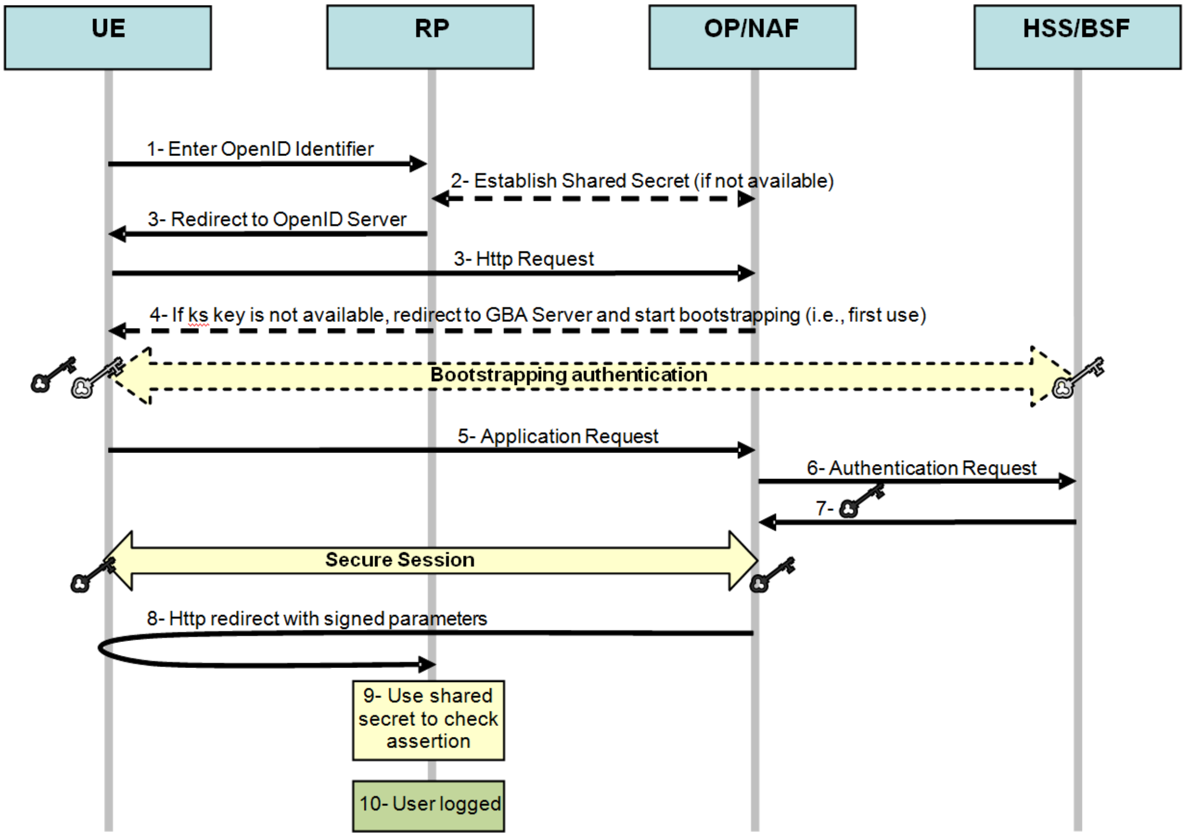
UE
RP
OP/NAF
HSS/BSF
1- Enter OpenID Identifier
2- Establish Shared Secret (if not available)
3- Redirect to OpenID Server
3- Http Request
4- If ks key is not available, redirect to GBA Server and start bootstrapping (i.e., first use)
Bootstrapping authentication
5- Application Request
6- Authentication Request
7-
Secure Session
8- Http redirect with signed parameters
9- Use shared secret to check assertion
10- User logged

Fig. 4. GBA/OpenID interworking: flow of operations

Common Vulnerabilities Enumerations are often useful to figure out a set of threats and vulnerabilities found in similar systems. For instance, the Common Weakness Enumeration (CWE) [22] is a well-known list of software systems weaknesses, created to serve as a common baseline standard for weakness identification, mitigation, and prevention efforts. Moreover, the Web Application Security Consortium (WASC) released a document called Threat Classification, which is an effort to classify weaknesses and attacks that can lead to the compromise of a website, its data, or its users [23]. However, these standards merely provide general guidance, being unable to take into account all the unique characteristics of a particular system for a deeper analysis [6]. Thus, it is important to follow a systematic process for threat analysis that is able to pinpoint system-specific weaknesses.

In this work, we use a methodology for threat analysis that combines the threat analysis processes from two related studies [5][6]. The resultant process consists of three main phases – threat modeling, risk management, and mitigation plan –, as shown in Figure 5. The focus of this paper is on the phase of threat modeling, which involves four steps: characterizing the system, identifying assets, determining threats and determining vulnerabilities.

Following the threat modeling, the next phase is the risk management, where the identified threats and vulnerabilities are ranked based on their impact on the assets, and the likelihood that the threat will happen. Finally, the last phase is the mitigation plan, in which countermeasures are selected and decision is made whether to mitigate, remove, transfer, or accept related risks. Although this paper does not cover these two last phases, they can be carried out by using the outcome of the threat modeling phase presented here.

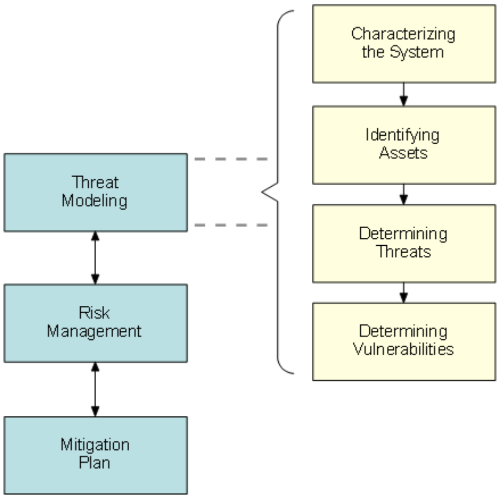

Fig. 5. Process of Threat Analysis.

The system vulnerabilities can be classified from various perspectives, such as the techniques used to exploit them, their nature, or the software and hardware components that make up the vulnerability [21]. In this work, we use the taxonomy of actors to group the identified threats and vulnerabilities, similarly to the model described in [7]. Firstly, it is necessary to identify the actors (i.e., the principal system components) and their corresponding responsibilities in the system operation. Then, the threats are identified, grouped and classified according to the system component (or actor) they affect.

V. Threat Modeling

A. *Characterizing the System*

From the system's operation described in Section III (see Figures 3 and 4), it is possible to outline the following actors, shown in Figure 6.

- User: an end user endowed with a valid identity on the Telecom Provider, who accesses the system via a web browser that implements HTTP/1.1.
- Identity Provider (IdP): a server that enables identity management and single sign-on for participating users and Service Providers, using OpenID protocol.
- Service Provider (SP): a web application that relies on an assertion provided by the IdP to supply services to users.
- Telecom Provider (TP): an authentication server that uses the long-term key of its subscribers (e.g., a USIM) for enabling authentication and key exchange between users and the IdP.

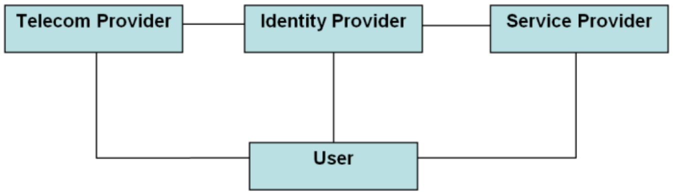

Fig. 6. System Actors.

The analysis presented here applies to any framework where a TP acts as an authentication server (using some arbitrary authentication method) inside the OpenID protocol. When dependence on the actual authentication method is important, we will analyze the case when the TP uses GAA/GBA technology.

B. *Identifying Assets*

Identifying assets that can be damaged or violated in the system is important in order to determine what effectively needs to be protected [5][6]. The main assets in the considered IdMS are shown in Table I, and categorized according to the taxonomy of actors.

C. *Determining Threats*

Using the information gathered in previous steps and analyzing the flow of operations outlined in Figure 4, it is possible to identify a set of system's threats. A convenient method for threat enumeration is to go through each of the system's assets and create threat hypotheses that violate confidentiality, integrity, or availability of the assets [6]. Table II shows the identified threats, classified by the taxonomy of actors.

TABLE I
IDENTIFYING ASSETS

Category	Asset	Code
User	- User's identity: it must not be stolen or disclosed to unauthorized parties. - User's activity: it must not be possible for an attacker to track the sites where the user logs into using the services provided by the IdP.	-AU01 -AU02
Service Provider	- Services provided by the SP: whose availability for (and only for) authorized users must be assured.	-AS01
Identity Provider	- Services provided by the IdP: whose availability for (and only for) authorized users must be assured.	-AI01
Telecom Provider	- Services provided by the TP: whose availability for (and only for) authorized users must be assured.	-AT01

AU: User Asset; AS: Service Provider Asset; AI: Identity Provider Asset; AT: Telecom Provider Asset.

TABLE II
DETERMINING THREATS

Category	Threat	Asset	Code
User	- Privacy Violation - Credentials theft and/or compromise	-AU02 -AU01	- TU01 - TU02
Service Provider	- Malicious URL - Session Swapping	- AS01 - AU02	- TS01 - TS02
Identity Provider	- Denial of Service (DoS) - Replay - Cross-Site Request Forgery	-AI01 -AU01, AS01 -AU01, AS01	- TI01 - TI02 - TI03
Telecom Provider	- Identity Information Exposure - UE Impersonation	-AU01 -AU01	- TT01 - TT02

TU: User Threat; TS: Service Provider Threat; TI: Identity Provider Threat; TT:Telecom Provider Threat.

VI. DETERMINING VULNERABILITIES

In this section, we determine what are the system's vulnerabilities related to the assets and the threats identified in Tables I and II, respectively. Besides, there is an effort to map these vulnerabilities into the Common Weakness Enumeration (CWE) [22], aiming to facilitate comparison across computer vulnerability databases. Furthermore, we outline some countermeasures to address the identified vulnerabilities. Afterwards, these countermeasures can be used in conjunction with a risk management analysis as inputs to a mitigation plan.

A. *User Vulnerabilities*

VU01	
Related Threat	TU01: Privacy Violation
Affected Asset	AU02: User's activity
CWE Mapping	CWE-359: Privacy Violation
Description	Since the IdP becomes a central place for all login activities across all sites, a malicious IdP can easily follow user's activity on the internet. The SPs are similarly able to track users between sessions, as long as the user always uses the same OpenID identifier [13].
Countermeasure	To protect user's privacy with respect to the SP, the IdP should establish a unique and persistent pseudonymous identifier for each user-SP pair [24]. On the other hand, the OpenID protocol does not specify any mechanism for hiding this information from the IdP. Apart from using multiple OpenID logins, the users cannot do much to avoid disclosing their web browsing patterns to IdPs [11].

VU02	
Related Threat	TU02: Credentials theft and/or compromise
Affected Asset	AU01: User's Identity
CWE Mapping	CWE-522: Insufficiently Protected Credentials
Description	It refers to the insufficient protection of user credentials or to the deficient authentication and authorization mechanisms [10]. An attacker could explore this by compromising the user system or by intercepting user-service communication.
Countermeasure	This vulnerability is substantially mitigated due to the strong authentication provided by GBA. Using GBA, it was not identified any way to exploit this vulnerability apart from physical theft/damage of the mobile phone used in the authentication process.

B. *Service Provider Vulnerabilities*

VS02	
Related Threat	TS02 : Session Swapping
Affected Asset	AU02: User's activity
CWE Mapping	CWE-718: OWASP Top Ten 2007 Category A7 – Broken Authentication and Session Management CWE-345: Insufficient Verification of Data Authenticity
Description	An attacker can force the victim to log into a SP using an account controlled by the attacker. The user may not realize that he/she is logged in as the attacker, allowing the attacker to monitor his/her activities on the SP. This happens because the OpenID protocol does not have a mechanism to bind the OpenID Positive assertions to the user's browser. Example: Alice authenticates at her IdP, and instead of submitting the positive assertion to the SP, she fools Bob and pass the assertion to him, who then authenticates at the SP using Alice's assertion.
Countermeasure	From the SP's side, the victim is sending an unsolicited positive assertion to the SP. So, SPs could reduce their exposure by not allowing unsolicited positive assertions [25]. This attack can also be prevented if the SP generates a fresh nonce at the start of the protocol, store the nonce in the browser's cookie and include the nonce in the "return_to" parameter of the OpenID protocol. Upon receiving a positive assertion from the user's IdP, the SP can validate if the nonce in the "return_to" URL matches the nonce stored in the cookie [26].

VS01	
Related Threat	TS01 : Malicious URL
Affected Asset	AS01: SP's service
CWE Mapping	CWE-20: Improper Input Validation
Description	When a user logs into a SP's site, he/she provides a URL as his/her login information. Thus, the SP needs to download the URL and extract the IdP address to continue with the protocol. In this context, an attacker can enter a malicious URL and lead to harmful attacks [11]. Examples: exploit of internal scripts: https://192.168.1.15/internal/auth?ip=1.1.1.1; third site scan: http://www.target.gov:1/, http://www.target.gov:2/; and DoS attacks: http://www.somesite.com/largemovie.flv.
Countermeasure	SPs must assume all input may be malicious and perform proper validation. SPs should use a whitelist filter of acceptable inputs and reject any input that does not strictly conform to specifications. Besides, blacklists can be useful for detecting potential attacks.

C. *Identity Provider Vulnerabilities*

VI01	
Related Threat	TI01 : Denial of Service (DoS)
Affected Asset	AI01: IdP's service
CWE Mapping	CWE-730: OWASP Top Ten 2004 Category A9 - Denial of Service
Description	A rogue SP could launch a DoS attack against an IdP by repeatedly requesting associations, authentication, or verification of a signature. The potentially most severe attack is during the association phase as each message requires the IdP to execute a discrete exponentiation [17]. This threat is particularly harmful because the user relies on the IdP for gaining access to all SP sites, causing an availability disruption.
Countermeasure	The IdP should limit the resources allocated to any SP to a bare minimum [27]. The IdP can use banning techniques and establish quotas based on some parameters, such as the IP address or the URI of the SP ("openid.realm" parameter of OpenID protocol).

VI02	
Related Threat	TI02 : Replay
Affected Asset	AU01: User's Identity, AS01: SP's service
CWE Mapping	CWE-294: Authentication Bypass by Capture-replay
Description	Once the user is authenticated at the IdP, he/she is redirected to the SP. The problem is that an eavesdropper can intercept a successful authentication assertion and replay it to log in as the victim user.
Countermeasure	This attack can be prevented if the IdP provides a nonce which ensures that only one person can log in, failing all subsequent attempts. However, a fast attacker who is sniffing the wire could obtain the URL and immediately reset the user's TCP connection, and then execute the attack [11]. Thus, a more effective approach is to use transport layer encryption for these connections.

VI03	
Related Threat	TI03 : Cross-Site Request Forgery
Affected Asset	AU01: User's Identity, AS01: SP's service
CWE Mapping	CWE-352: Cross-Site Request Forgery (CSRF)
Description	The IdP does not sufficiently verify whether a well-formed, valid, consistent request was intentionally provided by the user who submitted the request [22]. Example: Once the user is logged into his/her OpenID account (and hence have enabled single sign on), a malicious web page could use hidden iframes to silently contact a SP site (that the user has an OpenID account with) and perform some request on behalf of the user [13].
Countermeasure	The IdP must make sure that the form was served for the user. One method is to ensure the IdP puts a hidden form element containing something based on a secret and on data in the user's session object. In this manner, only a form served for a particular user will generate a valid submission for that user [28].

D. *Telecom Provider Vulnerabilities*

VT01	
Related Threat	TT01 : Identity Information Exposure
Affected Asset	AU01: User's Identity
CWE Mapping	CWE-200: Information Exposure CWE-300: Channel Accessible by Non-Endpoint ('Man-in-the-Middle')
Description	It refers to insufficient protection of the user's identity and the data about their movements from unauthorized third parties. There are two places in GAA where this kind of sensitive information may be revealed to an eavesdropper [15]: (a) during bootstrapping via the Ub interface, the private user identity (IMPI) is sent in the initial request from UE to BSF; (b) during application usage via the Ua interface, the same identifier (B-TID) is sent in every initial Application Request during the time that a certain master session key is valid, enabling onlookers to link two different sessions used by the same UE.
Countermeasure	The first threat can be easily addressed by performing bootstrapping via a secure channel. Similarly, the second threat can be addressed by running the application protocol inside a server-authenticated tunnel [15].

VT01	
Related Threat	TT02 : UE Impersonation
Affected Asset	AU01: User's Identity
CWE Mapping	CWE-266: Incorrect Privilege Assignment CWE-724: OWASP Top Ten 2004 Category A3 – Broken Authentication and Session Management
Description	GBA was designed to 3G networks but it also needs to support 2G networks, which have lower security level. This vulnerability refers to the SIM cloning threat in 2G GBA and was originated from the weakness of the COMP128 algorithm [15]. In this threat, the attacker

	obtains the cryptographic GSM key of a genuine subscriber and is able to impersonate the victim.
Countermeasure	This threat is not specific of 2G GBA, and operators are aware of the risk posed by using A3/A8 variants that utilize the COMP128 [15]. To protect against this threat it is recommended to move towards more secure variants of A3/A8.

E. Consideration regarding Phishing

Phishing is the most well known attack against OpenID protocol [11]. A rogue Service Provider can launch this attack by redirecting the user to a fake Identity Provider website where the user is tricked into entering his credentials, normally a pair username/password. In the case of the target system, the GBA-based authentication mechanism is able to protect users against phishing by preventing the attacker from seizing the user's credentials, as a mobile device is used as a physical key to gain access to services [15]. Besides, the keys shared between the IdP and the UE may be used for mutual authentication.

VII. CONCLUSION

This paper provides a threat model of an IdMS for Mobile Internet that integrates GAA/GBA and OpenID technologies. The results obtained indicate the strengths and weaknesses of the target system, taking into account vulnerabilities such as privacy violation, credential theft and/or compromise, session swapping, provision of malicious URLs, denial of service, cross-site request forgery, replay attacks, exposure of mobile identity information, and impersonation of mobile user. Afterwards, this model can be used as a basis for the development of more detailed models and the creation of a set of security best practices that security architects and developers can rely on when designing, implementing and operating such systems. Finally, this work reinforces the security benefits of using a robust authentication method, such as GBA, to enforce security in OpenID protocol.

In the near future, we intend to extend this work with a privacy study, as the problem of guaranteeing privacy in OpenID remains unsolved. Other possible future research directions include the formal analysis of some of the identified vulnerabilities using an approach similar to [9] and [10].

ACKNOWLEDGMENT

This work was supported by the Research and Development Centre, Ericsson Telecomunicações S.A., Brazil. The authors would like to thank the valuable comments and suggestions made by Pekka Nikander and Luis Barriga from Ericsson Research.

REFERENCES

[1] P. Weik and S. Wahle, "Towards a Generic Identity Enabler for Telco Networks," Proc. 12th Internat. Conf. on Intelligence in Networks (ICIN '08), Bordeaux, 20 – 23, October 2008.

[2] C. Steel, R. Nagappan, and R. Lai, "Core Security Patterns: Best Strategies for J2EE, Web Services, and Identity Management", Prentice Hall, Upper Saddle River, New Jersey, 2005.

[3] 3GPP, "TS 33.220 – Generic Authentication Architecture (GAA)," 3GPP, Tech. Rep., June 2006.

[4] OpenID Foundation, http://openid.net/foundation/

[5] A. Stango, N. Prasad, and D. Kyriazanos, "A threat analysis methodology for security evaluation and enhancement planning," in SECURWARE'09: Proc. of the International Conference on Emerging Security Information, Systems and Technologies. IEEE Computer Society, 2009, pp. 262–267.

[6] S. Myagmar, A. Lee, and W. Yurcik, "Threat modeling as a basis for security requirements," in StorageSS'05: Proc. of the ACM workshop on Storage security and survivability. ACM, 2005, pp. 94–102.

[7] N. Jiancheng, L. Zhishu, G. Zhonghe, and S. Jirong, "Threats analysis and prevention for grid and web service security," ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, vol. 3, pp. 526–531, 2007.

[8] Y. Demchenko, L. Gommans, C. de Laat, and B. Oudenaarde."Web services and grid security vulnerabilities and threats analysis and model." Grid Computing, 2005. The 6th IEEE/ACM InternationalWorkshop on, pp. 262–267, November 2005.

[9] M.H. ter Beek, C. Moiso, and M. Petrocchi, "Towards Security Analyses of an Identity Federation Protocol for Web Services in Convergent Networks," Telecommunications, AICT 2007. The Third Advanced International Conference on , pp.31-31, 13-19, May 2007.

[10] Y. Xiaolie and L. Lejian, "Verifying a Secure Session Protocol for Web Services," Networks Security, Wireless Communications and Trusted Computing, NSWCTC '09. International Conference on, vol.2, pp.301-304, 25-26, April 2009.

[11] E. Tsyrklevich and V. Tsyrklevich, "Single sign-on for the internet: A security story," BlackHat USA, Tech. Rep., 2007.

[12] H. Oh and S.Jin, "The Security Limitations of SSO in OpenID", ICACT'08 , 2008.

[13] D. Chadwick and S. Shaw, "JISC Final report – OpenID study," EDINA, University of Edinburgh, Tech. Rep., 2008.

[14] H. Lee, I. Jeun, K. Chun, and J. Song, "A new anti-phishing method in openid," In The Second International Conference on Emerging Security Information, Systems and Technologies, pp. 243–247, 2008.

[15] S. Holtmanns, V. Niemi, P. Ginzboorg, P. Laitinen, and N. Asokan, Cellular Authentication for Mobile and Internet Services. Wiley, 2008.

[16] D. Recordon and D. Reed, "OpenID 2.0: A platform for user-centric identity management," Proceedings of the Second ACM Workshop on Digital Identity Management, pp.11-16, 2006.

[17] OpenID Foundation, "OpenID Authentication 2.0". Available at http://openid.net/.

[18] 3G Americas, "Identity Management—Overview of Standards and Technologies for Mobile and Fixed Internet", January 2009.

[19] 3GPP, "TR 33.924 – Identity management and 3GPP security interworking," 3GPP, Tech. Rep., December 2009.

[20] R. Shirey, "Internet Security Glossary," RFC 2828, May 2000. Available at http://tools.ietf.org/html/rfc2828

[21] M. Bishop, Computer Security: Art and Science. Addison-Wesley, 2009.

[22] MITRE Corporation, Common Weakness Enumeration, http://cwe.mitre.org/

[23] Web Application Security Consortium, "Threat Classification", Version: 2.00, 2010. Available at http://www.webappsec.org/projects/threat/

[24] T. McBride, D. Silver, M. Tebo, C. Louden, and J. Bradley, "Openid 2.0 profile, identity, credential, & access management," ICAM, Tech. Rep., 2009.

[25] OpenID Foundation, http://wiki.openid.net/SecurityIssues

[26] A. Barth, C. Jackson, and J. Mitchell, "Robust defenses for cross-site request forgery," in CCS'08: Proc. of the 15th ACM conference on Computer and communications security, pp. 75–88, 2008.

[27] OWASP; "Top ten most critical web application security vulnerabilities. Whitepaper", January 2004.

[28] OpenID Foundation, http://wiki.openid.net/Security