SA Forum Security Service (SEC): An use case study

M. Pourzandi
Makan.Pourzandi@ericsson.com, Ericsson, 8400 Decarie, Montreal, QC H4P 2N2, Canada.

C. Tindel
candin@redhat.com, Red Hat, 140 Broadway, New York, NY 10005, USA.

Abstract

Security has become an important requirement for HA systems. To address this, the SA Forum Security service (SEC) Specifications have been recently published [6]. This paper presents an use case of using SEC to enforce and improve the security of a GGSN node. The goal of the paper is not a detailed study of security in GGSN applications but rather show through a simple use case the place of SEC in SA Forum eco-system and clarify different concepts behind the SEC specifications for SEC users and implementers. The case study starts with presenting some attack scenarios, then the security requirements deduced from the attacks are explained and how they lead to the adequate security policy. Finally, we describe how SEC mitigate the attacks and enforces the security for the GGSN system.

Keywords: Service Availability Forum, Security, Highly Available systems, SA Forum Security Service.

1 Introduction

The Service Availability Forum (SA Forum) is a consortium of computer and communications companies working together to develop open standards for supporting hardware, application high availability, and local and remote systems management for such systems [4].

The SAF-based clusters by nature are exposed to public networks. Till recently, server clusters not used for web servers were considered not to be a major target for the hackers, though the external attacks become a new reality after the Spring 2004 attacks on High Performance Computing infrastructures worldwide [1, 2, 3]. In the same way, the SAF-based clusters can be target of attackers who may try to steal SAF services, eavesdrop on SAF messages and sell the confidential data to organized crime, disrupt cluster operations for vandalism or black mailing, etc. In addition, cluster operations can be disrupted using external denial-of-service attacks on cluster nodes, or by attacks against communications between remote users and clusters. It is in this perspective that the SA Forum SEcurity service (SEC) was developed [6]. Many middleware frameworks dealing with distributed system integrate security services as an essential part of their functionality: CORBA
Security Service [8], Grid Security Service [9], Web Service Security [10]... Even though SEC in functionality is similar to these security services, SEC in addition concentrates on the security aspects specific to highly available systems. For example, there has been special attention paid to support run-time updates of the security policy or to avoid any service availability disruption due to incoherences in the security contexts of active and standby processes.

In this paper, we present SEC through detailing different steps to secure a simplified HA application, i.e. a Gateway GPRS Support Node (GGSN). Our goal is not to present a detailed study of GGSN security, or presenting SEC in details as SEC has already been presented in [5] and the detailed specifications are available from [4]. We rather intend to help future SEC users and implementers to better understand the current SEC specifications through this simple use case study.

We base this paper on so far approved functionality for SEC: authentication and authorization. It is clear that as the SEC functionality is extended, we expect more benefits from deploying SEC.

The paper is organized as follow: In sections 2 and 3, we briefly present SEC and the target test GGSN. Then in the section 4, different attack scenarios are discussed, we finally present how SEC can mitigate those attacks in the section 5.

## 2 Security service overview

We briefly present SEC here as a detailed study can be found in [6]. The basic entity used for authentication, authorization and accounting for SEC is the user application process. We do not consider covering all processes running in the SAF system but those only using SAF service interfaces. Furthermore, SEC API is only accessed by SAF AIS services (see Figure 1). It is assumed that platform security mechanisms are used to enforce the right processes to access SEC API. It is assumed that the HA system is protected by external security mechanisms like firewalls and intrusion detection systems.

### 2.1 Security context

The security context defines the set of different privileges for SAF AIS client application processes. Different security contexts are assigned to the processes based on their authentication. All processes in the same security context have the same permissions. The same is valid for SAF resources: All resources with the same SEC Context are submitted to the same access rules. By default, processes can access resources with the same security context.

Some SAF resources are pre-configured to have determined security contexts through security configuration files or defined by SAF system management. Dynamically created SAF resources at run-time are assigned the security context of their owner (creator).
2.2 Authentication

The AIS services will support a new secure initialization interface which allow AIS service client applications to pass their security credentials to the AIS Service. The AIS services in turn authenticate the process with the help of SEC. SEC will return the security context associated with client’s credentials to the AIS service, which should use this security context for further authorization requests related to that process. It is the AIS service’s responsibility to keep track of security context associated with the AIS service client application. The typical case is that a process has the same security context for all AIS Services.

For now, the only authentication mechanism recognized by SEC is UNIX UIDs. The SEC specifications already mention other authentication mechanisms, we expect that the list of authentication mechanisms will increase in the future releases of SEC to include shared secret and public/private key based authentication mechanisms.

2.3 Authorization

In SAF security model, the SAF AIS services are in charge of protecting access to their resources (they have the role of Policy Enforcement Point, PEP). When a SAF AIS client application wants to access some SAF resources, the AIS service forms a request, sends it to SEC which finds the corresponding security policy (based on the client’s security context, resource security context and the operation) and sends back an answer to grant or deny the access to the resource (SEC has the role of Policy Decision Point, PDP). For example, before granting access to a checkpoint, the SAF Checkpoint service will need to
check the permissions of the client application to read the checkpoint.

3 GGSN node

Our use case target application is a Gateway GPRS Support Node (GGSN). A GGSN is part of the network architecture as proposed in Third Generation Partnership Projects (3GPP) release 5 specifications. A GGSN node among other functionality provides access to the Internet from operator’s access network. Therefore, its security is paramount to the overall security of the operator’s network. Previously in [12], Kamalvanshi and Jokiaho have studied a SAF based implementation of GGSN application. We base our use case on their study. Therefore, we do not discuss the GGSN functionality here as a detailed study can be found in [12].

Due to the lack of space, we only present a limited set of functionality for the GGSN application, i.e. session management, Internet connection management, user profile management and charging interface (see Figure 3). We assume that these components are implemented as different software components with 2N redundancy model [7]. Though simple, the use case developed here can be easily extended to other components of GGSN or with different redundancy models.

We assume that GGSN application is running on top of a cluster of Linux systems. For the remaining of the paper,
whenever we refer to GGSN system, we mean the cluster running the GGSN application. By GGSN nodes, we mean different Linux systems part of the cluster running the GGSN application.

4 GGSN attack scenarios

The digital security has gone through fundamental changes during these last years: from highly advertised vandalism acts mainly motivated for fame and general recognition, the attacks now targets more and more high profile sensitive applications. These attacks are performed by highly skilled attackers and motivated for criminal goals. It is the latter case which interests us in the case of GGSN security use case. In the following, we describe different attack scenarios on GGSN and categorize them according to the attacker's skills\(^1\). Our goal here is not to cover all possible attack scenarios but rather discuss SEC usage through these attack scenarios.

4.1 GGSN security assumptions

It is assumed that the GGSN application is protected by firewalls on all connections (specially on Gi connections which are directly connected to the Internet). All GGSN nodes have valid and cohesive firewalling rules enforced. Furthermore, the inter node connections are secured, e.g. using IPSec. Furthermore, all GGSN nodes are security hardened: all useless services turned off, no compiler on the nodes, correct file system permissions are set, etc.

4.2 Attacks by poorly skilled attackers

In this scenario, we assume that the attacker is poorly skilled. S/he uses basic attack tools to attack the GGSN node. This kind of attack is well known and can be mitigated by the external security mechanisms to the GGSN node, i.e. firewalls.

To illustrate this, we take the case where the attacker launches a Denial of Service attack (e.g. ping flood) against the GGSN from few IP addresses in the Internet. Such an attack can be mitigated by the firewalls protecting GGSN Internet connections (Gi connections) before reaching the GGSN application.

4.3 Attacks by average skilled attacker

In the following scenario, we consider an average skilled attacker. S/he manages to gain access to the Intranet (through some malware, virus, back door software...). S/he then exploits some zero-day exploit\(^2\) to launch an attack from the Intranet

\(^1\)Bear in mind that the adjectives “average skilled”, “poorly skilled” used in the following are subjective. They are used to compare the attacker skills and not to scientifically qualify its skills.

\(^2\)A zero-day exploit is an attack which uses an undisclosed vulnerability in an application.
and gains access to one of the nodes in the GGSN system. The attacker can then exploit this security breach in one or several nodes to access, steal, modify sensitive data etc. Such an attack can be detected by a host based intrusion detection system or through some other security mechanisms.

To better illustrate this scenario, we take the case which the attacker gains access to the Internet connection manager on node 2 (see Figure 3). S/he tries to read different check points from the billing component or sensitive files on the nodes, but fails as the security is tightened up by the different security mechanisms (see §4.1). After several attempts, the host based intrusion detection system detects the unusual activities on that node, these anomalies are reported to the administrator who investigates and removes the malware.

4.4 Attacks by highly skilled attacker

In this case, the attacker is highly skilled and targets a specific GGSN node. S/he is probably employed by a criminal organization, in order to steal or modify the sensitive data from the GGSN application. We assume that the attacker is skilled enough to gain access to some software components on one of the GGSN nodes. The attacker then installs its own Trojan downloader application and analyzes the node to detect the applications running on the compromised node. Based on this info, s/he uses a hacked or stolen version (open source or perhaps legitimately purchased) of SAF libraries to develop his own application outside of the hacked GGSN node. S/he downloads the malicious SAF application using its Trojan downloader on the breached GGSN node. This Trojan SAF application can access the SAF resources or take control of the GGSN application.

To illustrate this, we take the example where the attacker gains access to the Internet connection manager or a possible internal collusion. S/he then tries to access check points files, but fails as the security is tightened up by different security mechanisms (see §4.1). S/he then looks around in the library directory to realize different SAF service libraries. S/he downloads its Trojan SAF application and accesses the list of check points available in the system through IMM OM API. S/he reads the check points and send them over to distant/hostile system for detailed analysis. Furthermore, s/he uses IMM OM API to re-configure or take control of the GGSN node.

This scenario shows an important security breach as SAF resources can be stolen and sold, or further the security breach can be used to take control of the HA-application and compromise the availability of the application. Therefore, it is important to put in place security mechanisms at SAF service level which can either mitigate these attacks or control damages on the HA-application by providing containment and compartmentalization.

---

3Remember we assume that there is no compiler on any GGSN node and that the attacker is not going to download his own compiler on that node.
5 SAF Security service usage

In the section 4, we illustrated why security mechanisms are needed at SAF service level. In this section, instead of a general study on how SEC can mitigate the attacks and enhance overall security of HA systems, we rather discuss SEC usage through an example: how SEC can mitigate the attack described in §4.4. We describe a simplified scenario in which the SAF system manager from a simple security requirement, i.e. deny access to checkpoints for unauthorized users, write the appropriate security policy (c.f. §5.1), then explain how the SAF check point service [13], and SEC enforce this security policy (c.f. §5.3).

5.1 Security policy

The security policy describes the general behavior of the entities from a security point of view. It describes how the entities are authenticated, what are the access rules for them, the security attributes for communications between different entities and their accountability, etc. The security policy is defined by the SAF system management based on the application and deployment characteristics. It is then interpreted by SEC and propagated to different SA Forum Services which enforce the security rules defined in the security policy. At the time of writing, SA Forum does not consider standardizing the security policy language syntax. There are several possible standardized languages like SAML or XACML [14, 15]. Further work is needed in order to decide on an appropriate security policy language for SA Forum systems.
Figure 4. The mapping between different user applications identified by their authentication and security context IDs, and different resources identified by their SA Forum name and the security context IDs. The mapping is expressed in XML language. Note that we simplified the rule by removing XML language details.

For our example, we chose XACML language to express the security policy, i.e. denying access to checkpoints for unauthorized users. Using XACML, the simplified security policy for our GGSN system is a single rule represented in Figure 3. This rule named 'BackEndPrivateRule' has the effect of permitting the operation SAF_CHK_READ (reading check points) on the resources with the SEC context ID //saf/BackEndPrivate by the processes with SEC context ID BackEndComps@ids.saf.

The different security operations (e.g. SAF_CHK_READ) are defined in SEC specifications (Appendix A, [6]). Though, SEC context IDs are arbitrary identifiers which need to be assigned by SAF system management to different SAF resources and SAF client applications in the SAF eco-system. This mapping is considered for now to be SEC implementation dependent. IT must be defined by the SEC implementors and provided to the administrators of the SAF application.

For the need of our simple example, we chose to use a simple XML based mapping. In our mapping language, the SAF resources (e.g. checkpoints, event channels, log streams etc) are identified using their SAF AIS object name as defined in §4.3.9, [17]. In Figure 4, the resource //saf/CHK/users-cache has been mapped into the //saf/BackEndPrivate SEC context ID. The processes are not directly mapped to SEC context IDs. SEC uses a double indirection assignment schema. First, the process is classified by the authentication token which they use. Second level of indirection is then used to map the authentication tokens to different SEC context IDs (a detailed explanation can be found in §5, [6]). In Figure 4, the SAF system management maps the UNIX users safapp1 at node 3 and node 4 to SEC context ID 'BackEndComps@ids.saf.

4Given the large number of processes in the SAF eco-system, this approach is unrealistic.
We used OpenXACML, an open source implementation of XACML [16], to implement the above security policy and mapping and verified that they enforce the desired behavior.

5.2 Security policy provisioning

Once the security policy has been developed by the SAF system managers, there is a need to provision SEC with the security policy as the authentication and authorization decisions are based on it. SA Forum IMM service (IMM) is used as main entry point for provisioning SEC with the security policy, security configuration and other security relevant information.

Different SAF system management agents (SMA) read the security policy described in any SEC implementation dependant language, interpret it and stores it in IMM using OM API under well-known configuration object names. Upon initialization, SEC reads these configuration objects in IMM and initializes its security policy. The algorithms and logics used in the decision engine in SEC are implementation dependant. This approach provides the necessary flexibility to support different security models and security deployments used in various HA applications.

5.3 Access enforcement through different SAF AIS service calls

As explained in the section 2, the access to different SAF resources is enforced by AIS SAF services in collaboration with SEC. In this model, SEC is the policy decision point and different SAF services are policy enforcement points, i.e. SAF services before granting or denying access to their resources must request SEC to make decision on granting or denying the access and enforce that decision. The sequence diagram in Figure 5 presents the interactions between SAF service client application, SAF Service in occurrence SAF Checkpointing service and SEC in order to grant access SAF resources, i.e. checkpoints. The client upon initialization will pass its credentials to the SAF Checkpoint service which in turn authenticates the client and receives its security context from SEC (actions 1-5 in Figure 5). This SEC context ID is then used in subsequent calls to authenticate the SAF service application client process (actions 7 and 11 in Figure 5). When the SAF Checkpoint client application requests to open the checkpoint 'users-cache', the checkpoint services passes its security context to SEC. SEC decides to grant or deny the access to the 'users-cache' checkpoint based on client’s security context and the security rules defined in the security policy (actions 6-13 in Figure 5). Based on this decision, SAF Checkpoint service will grant or deny access to the 'users-cache' checkpoint for the client’s application.

Note that for ease of comprehension, we presented a simple authorization use case. In SEC specifications, there are dispositions to provide the AIS services with all security permissions for a client application therefore eliminating need for multiple authorization request calls. In addition, the AIS services can always cache the access authorizations for client
applications as SEC will inform the AIS services as soon as the security policy changes for the client application (c.f. §5.4).

Therefore with SEC enabled, to access SAF resources, there is need for valid SAF security credentials. Therefore, for the attack scenario described in the section 4.4, the attacker needs to acquire valid security credentials to access ‘users-cache’ checkpoint. Less than it, the attack fails (this case should be the normal secure case).

### 5.4 Run time security policy updates

In HA systems, it is paramount that security policy can be modified and enforced at run-time without any need for service interruption. The current SEC approach supports the run-time security policy updates and enforce applied changes without any need for service disruption. The SMAs modify or update the SEC configuration objects in IMM through the IMM API (these objects represent the security policy and SEC configuration attributes). SEC registers as an object implementer for these security objects, i.e. SEC is called when these objects are created, changed or deleted. Therefore, when security policy is modified by the SMAs in IMM, SEC is automatically informed. Then, based on these modifications, SEC updates its authorization and authentication decision engine. Furthermore, SEC informs AIS services of these changes through SEC callbacks so they can enforce them.

For example, if the user safapp1 at node 4 is removed from the security context type ‘BackEndComps@ids.saf’, from then on SEC will deny access to the checkpoints of SEC context ID ‘//saf/BackEndPrivate’. In addition, SEC will inform the SAF Checkpoint service about the change in the security context of safapp1. The SAF Checkpoint service then re-evaluates all authorization related to the processes of this UNIX user account and removes their access to the resources with SEC context ID ‘//saf/BackEndPrivate’.

### 6 Conclusions

In this paper, a simplified case study was used to describe SEC functionality in the context of an HA application, in occurrence a GGSN system. The case study started with presenting some attack scenarios, then security requirements deduced, which lead to the adequate security policy, for finally describing how SEC can be used to mitigate the attacks and enforce the security for the GGSN system. Even though a simple use case, it clarifies concepts behind the SEC specifications and its use in the context of HA systems.

### References

Figure 5. The access control is enforced by SAF Checkpoint service in collaboration with SEC.


[17] Service Availability Forum, Service Availability Interface, Overview SAI-Overview-B.04.01.