Implementation of C-BAS: Certificate-based AAA for SDN Experimental Facilities

Umar Toseef and Kostas Pentikousis
EICT GmbH, Berlin, Germany
Email: [umar.toseef, k.pentikousis]@eict.de

***Abstract*—Recent work in software-defined networking experimental facilities has been shifting towards large scale deployments through federation of resources that span across continents and make it possible to perform experiments at a global scale. The success of such deployments very much depends on the design and implementation of essential, secure mechanisms for authentication, authorization, and accounting (AAA) that not only ensure the robustness of such facilities against intrusions and unauthorized use but also ease experimentation and system administration in such complex environments. C-BAS is an initiative in this direction that uses a secure and flexible certificate-based AAA architecture for SDN experimental facilities. Advanced certificate-based authentication and authorization makes C-BAS inherently resilient against attacks specific to traditional AAA mechanisms, increases flexibility and autonomy in experimental facility system administration, and facilitates federation. This article introduces the implementation details of C-BAS, explains its features through use cases, and evaluates its computational performance.**

I. Introduction

A comprehensive authentication and authorization mechanism is of vital importance for any SDN experimental facility (SEF) that allows the use of finite computational and network resources for designated R&D purposes. An access control mechanism for a SEF serves a dual purpose. First, it enforces the application of policies: Access control to SEFs is required, on one hand, to enforce the usage policies on legitimate users and restrict access for others. Second, it ensures that SEF remains operational and experiment results are not affected by misbehaves. Currently, SEFs are mostly relying on legacy AAA mechanisms such as LDAP (Lightweight Directory Access Protocol) [1] and Kerberos [2] which were not designed specifically for SEF use and therefore lack the features considered essential for modern SEFs [3]. On the contrary, certificate-based AAA, used by C-BAS, not only fulfills the basic needs of SEFs, i.e., enabling fine-grained access controls upon critical resources, but also overcomes inherent shortcomings of LDAP and Kerberos such as single authoritative source of trust, inflexible system of authorization, cumbersome process of synchronization of AAA entities to realize federations, and so on. In addition, the evolved architecture of C-BAS makes it secure against disruptions and interference from attackers and enables the support of different member roles and permissions. Furthermore, C-BAS is highly scalable and suitable for large experimental facilities and their federation. C-BAS is extensible through plugins, autonomous to minimize system administration efforts, and modular to ease software maintenance and further developments.

C-BAS has gone through many design and development cycles, employed by few EU projects (e.g., FELIX [4]] and has received contributions from a number of researchers. To the best of our knowledge, C-BAS is the only open source AAA solution for SEFs that is actively being maintained and developed and freely accessible to the whole research community. There are few other clearinghouse implementations that have been developed for particular SEFs and are often restricted to consortium members. For example, Expedient [5] is a GENI control framework with a tightly coupled clearinghouse and, therefore, lacks compared to C-BAS in terms of distribution and flexibility. Similarly, the ProtoGENI [6] clearinghouse has been based on a particular Slice Facility Architecture (SFA) [7] and designed to support only small sized federations. Likewise, the GENI clearinghouse [8] with restricted dissemination policy implements its own clearinghouse API which is followed within the GENI federation. In contrast, C-BAS access interface supports the Common Federation API version 2 (FAPIv2) [9] set of standard APIs that any GENI-compatible Federation should offer. Moreover, C-BAS is flexible enough to support other APIs through plugins.

This paper introduces the implementation details of C-BAS and its software composition. It explains the mutual interaction of software components and also provides insights regarding the communication of C-BAS with other entities in SEFs. Finally, the paper presents measurements results from stress testing the C-BAS implementation.

II. Software Architecture

This section introduces the software architecture of C-BAS and its software components.

A. EiSoil

The implementation of C-BAS is based on EiSoil [`www.eict.de/eisoil`], an open source light-weight framework originally developed for creating Aggregate Managers (AM) in SEFs. However its plugin capabilities and helper functions for common tasks make it a perfect candidate to base C-BAS development on it. EiSoil maintains a service registry or plugin-manager to host its plugin modules. This plugin-manager has been especially designed so that functionality implemented in one plugin module can be easily leveraged by other plugins. This architectural design is a main feature of EiSoil that offers service abstraction, modularity, extensibility as well as a clear separation of concerns. This way, the use of EiSoil not only expedites the software development cycle but also makes software maintenance easier.

EiSoil, developed in Python, builds most of its functionality through plugins. From an implementation viewpoint, an EiSoil

plugin has a simple predefined code structure, i.e., source code files are organized under an individual folder along with two special files; (i) a JSON-style menifest document that describes the service provided by the plugin along with its dependencies on other modules and services, (ii) a Python coded initialization script that is responsible for bootstrapping as well as registering services of the plugin with the plugin-manager.

Out of the box EiSoil comes with a number of plugin modules covering a variety of helper functions. These plugin modules, termed vendor modules, are maintained by the EiSoil development team. For example, `configdb` is one of the vendor plugins which configures the underlying database that adds persistence to EiSoil. Similarly, `configrpc` provides means to perform configurations related to Remote Procedure Call (RPC). Another vendor plugin `geniv3rpc` implements an RPC request handler for the method calls of the GENI AM API [10]. Likewise, helper functions related to basic authentication and authorization are supplied through a plugin named `geniutils`. Moreover, plugins `mailer` and `scheduler` implement the necessary support to send emails and perform scheduling to manage reservations.

Other prominent features of EiSoil include logging of error and debug information, scheduling and dispatching of asynchronous jobs, as well as, the ability to communicate with the remote entities across the network.

B. C-BAS Software Components

C-BAS [`www.eict.de/c-bas`] capitalizes upon the EiSoil framework and, therefore, implements all of its functionality through plugins. Following the EiSoil design philosophy, C-BAS decouples its access interface from the clearinghouse management API as depicted in Fig. 1. This decoupling allows CBAS to support multiple APIs (e.g., FAPI [9], SFA [7], FELIX [11], [4] etc.) through code reuse without modifying the overall architecture of the clearinghouse. The `delegate` module in Fig. 1 acts as a translator between service access interface (e.g., FAPI) and the core clearinghouse methods. In addition, `delegate` performs error handling by catching all clearinghouse errors and exceptions and re-throwing them after mapping onto access API specific counterparts. The `delegate` is also responsible for all kind of namespace translations such as from resource URN to UUID, etc. Most importantly, it is `delegate` which ensures that an API method call satisfies all requirements of authentication and authorization before forwarding it to the clearinghouse.

In the following a list of C-BAS plugins is provided along with the short description.

`fedrpc` has its name derived from 'Federation RPC API'. This plugin module builds the service access interface of C-BAS and is in charge of handling received XML RPC calls of FAPIv2.

`ofed` is an important plugin module which implements the `delegate` as shown in Fig. 1. Owing to the fact that C-BAS supports multiple services like, Slice Authority (SA), Member Authority (MA), etc., for each service a dedicated delegate is implemented. This delegate not only maps FAPIv2 calls onto clearinghouse methods but also takes care of authentication

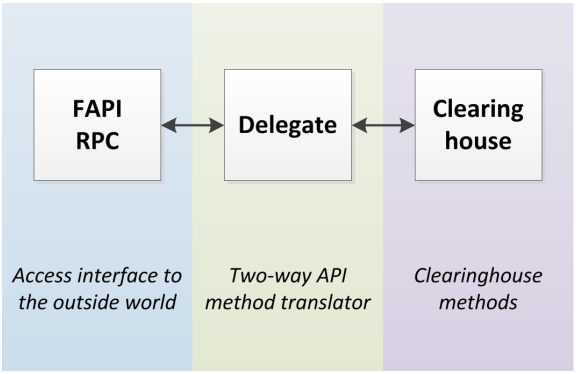

Fig. 1. C-BAS service interfaces

and authorization. This is achieved through an extensive use of `geniutils` plugin which is based on GENI provided library for SFA [7] styled certificate-based authentication and authorization. However, for use within C-BAS the aforementioned library has been extended to support the rich functionality and member roles advocated by C-BAS.

As implied from its name **`fedtools`** is a plugin that is comprised of helper functions to be exploited by all plugins. These helper functions include conformation and validation checks for arguments passed in RPC method calls, credentials verification, and authorization checks for requested information. Moreover, they are used to get a mapping between member roles and their default set of privileges (see Table. I) as well as to obtain the set of privileges needed to authorize a service access.

`omemberauthorityrm` implements the Member Authority (MA) services of the clearinghouse to offer management of member information and SSH keys. It maintains a database of registered members, their credentials, certificates, and public SSH keys. Private keys corresponding to member certificates and private SSH keys are always handed over to their owners instead of making them part of the database. Moreover, it is `omemberauthorityrm` that registers new members and issues them certificates and credentials based on their assigned roles. It should be noted that MA maintains only system credentials which are not associated with any particular project or slice. These system credentials are mainly employed in tasks related to member information management and SSH key management by members themselves or by C-BAS system administrators. Being the anchor point for members information, `omemberauthorityrm` is consulted to update, lookup, and delete the member information and SSH key records. It is also often contacted by user-agents to retrieve member credentials and public SSH keys. The certificates and credentials issued to members bear a unique serial number for identification purpose. They have usually a short validity time period and must be renewed in time. It is possible to invalidate a member certificate before its expiration through certificate revocation process. For this purpose, `omemberauthorityrm` maintains the Certificate Revocation List (CRL) which is periodically updated and disseminated to other system entities involved in authentication and authorization. It should be noted that member registration and revocation are part of the FELIX API that extends FAPIv2.

osliceauthorityrm plugin realizes Slice Authority (SA) services of clearinghouse. These services include management of projects, slices, and slivers. In addition, it also maintains user credentials for projects and slices. Project credentials are often used internally by SA to manage user membership for projects. On the other hand, slice credentials which represent user membership for a slice and his privileges on that slice, are mainly required to authorize GENI AM API calls. `osliceauthorityrm` implements a number of functions to facilitate the creation, update, lookup, and removal of projects, slices and their memberships, e.g. lookup of all projects or slices for a given member, members list of a project/slice, credentials update when a member role is changed, etc. `osliceauthorityrm` also ensures that there is always exactly one Lead member for each slice/project who is the principal contact point for all activities of that slice/project. In contrast to MA, SA does not support membership revocation for slice/project. However this can be achieved indirectly either through the revocation of member certificate at MA which would automatically invalidate any issued slice credentials or by removing user membership for the slice/project so that slice credentials would not be renewed for that user. By default slice credentials have a configurable lifetime of one month after which they must be renewed.

`osliceauthorityrm` also supports delegation of slice credentials that allows a member to delegate some or all of her privileges to another member. Delegated credentials can be generated through SA provided API method. Once generated these credentials are returned to the user without storing them in the local database. Delegated credentials get invalidated if members certificates of involved parties expire or revoked.

oregistryrm implements the Federation Registry (FR), sometimes also referred to as Service Registry, which serves as primary contact point for C-BAS and its associated SEF. It keeps pointers to all C-BAS services (e.g., SA, MA) and AMs in the facility. Moreover, `oregistryrm` serves lookup requests for public/trusted certificate of authorities like SA and MA. All information available at `oregistryrm` is statically configured in a file named 'registry.json' during C-BAS setup. Owing to the fact that information distributed by FR is public, it serves all requests without requiring authentication and authorization. Pointer or URL of FR has to be shared with other federations or user-agents out-of-band.

mongodb realizes persistence for C-BAS by implementing a lightweight layer between clearinghouse and noSQL MongoDB [12] database. Any plugin which needs persistence leverages the helper functions offered by `mongodb`. It facilitates execution of database queries like create, update, lookup, and delete of entries in a collection.

registration plugin handles new member registration requests sent by the C-BAS registration client. It performs the necessary sanity checks on such requests, processes them and returns member certificate and credentials back to the user through the C-BAS registration client. It should be noted that `registration` is accessible only to C-BAS registration client while the privileged user-agents perform member registration through MA API method.

Fig. 2 is UML package diagram that lists constituent plugins of C-BAS and depicts their inter-dependency.

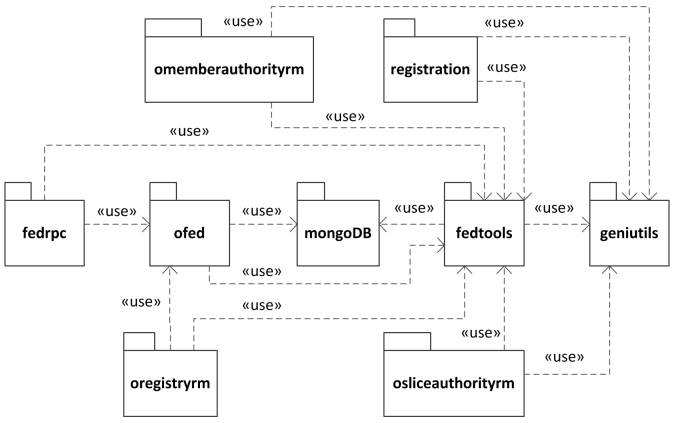

Fig. 2. UML package diagram of C-BAS

C. Use Case: Slice Creation

In order to provide a clear insight into the C-BAS request handling Fig. 3 illustrates a message sequence diagram of a use case where a slice is created through a user-agent. In order to create a slice in a project, the user has to first acquire his credentials for that project. For this purpose, a lookup is performed to fetch the user's project membership information which includes his credentials. To authorize this lookup, the user's system credentials are presented which must pass through two checks; first check verifies if these credentials are valid and trusted while the second check ensures that the user has sufficient privileges to access the requested information. If these checks are passed, project membership information is looked up and returned. At this point the user-agent can request for slice creation and present the project credentials retrieved in the previous step as means for authorization. These credentials are verified in the same way as described earlier and, upon success, the requested slice is created and the associated slice credentials are returned to the user-agent for use in GENI AM API calls. The creator of a slice is, by default, entitled with the Lead role of that slice. However a Lead member can willingfully transfer this role to any other member of the slice. A similar procedure is followed for project creation.

III. Member Roles and Privileges

In [3] a set of member roles was proposed for C-BAS based on the FAPIv2 specifications. However during the implementation this set has been extended for practical reasons. For example, an auditor role has been defined at project level to allow monitoring of all slices in a project. Similarly, a "root" member role has been introduced to facilitate the execution of system administrative tasks. This is useful in logging administrative actions for accountability purposes. In addition, another role has been defined for user-agents hosted by SEFs to enable them perform administrative actions, such as, registration of new members, etc. The user-agents that run in user premises like OMNI [13] do not need credentials as they typically speak-for their users.

Table. I shows the default privileges assigned to different member roles at their creation time. In addition to defaults, a member can be assigned with additional privileges through two ways: (i) through a membership update where usually

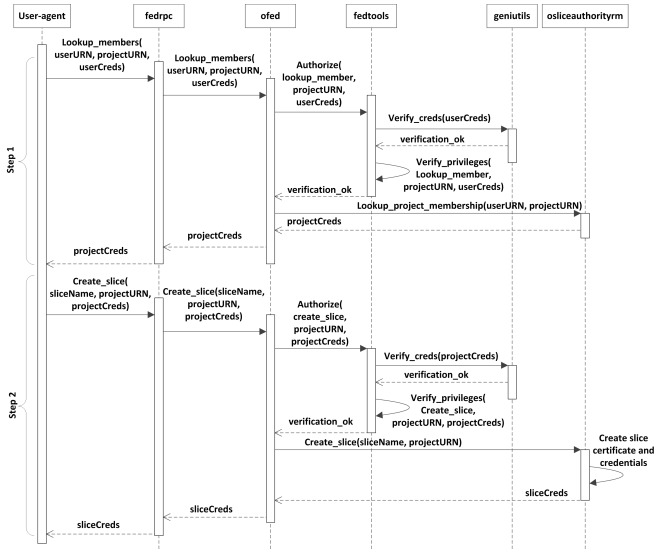

Fig. 3. Project credentials lookup and slice creation processes

an Admin or Lead updates a member's privileges in C-BAS database, or (ii) through the credential delegation process. The former is a long-term assignment which leads to regeneration of credentials while the latter is for short-term assignment where delegated credentials are generated with a limited time period validity. By policy, all default privileges can be delegated however a delagator may restrict further delegation of his delegated privileges.

TABLE I. MEMBER ROLES AND ASSOCIATED PRIVILEGES

Role	Context	Privileges
Auditor	Project	View, Monitor
	Global	View, Monitor
Member	Project	View
	Slice	View, Start, Stop
Admin	Project	View, Monitor, Update, SetAdminRole, AddMember, RemoveMember, ViewMember, UpdateMember, SetMonitorRole, CreateSlice, SlicesWildcard
	Slice	View, Update, SetAdminRole, AddMember, Start, RemoveMember, ViewMember, UpdateMember, Stop
Lead	Project	<Privileges of project admin>, SetLeadRole, Remove
	Slice	<Privileges of slice admin>, SetLeadRole
Root	Global	MembersWildcard, SlicesWildcard, ProjectsWildcard, RegisterMember, RenewMembership, RevokeMembership, RegisterService, ViewService, RemoveService,
User-agent	N/A	RegisterMember, RenewMembership, RevokeMembership, RegisterService, ViewService, RemoveService

IV. MEMBERSHIP REVOCATION

C-BAS supports certificate revocation to withdraw certificates affected by incidents, such as, when the private key of a certificate is lost or compromised, a certificate is mistakenly issued, or a member registration has to be confiscated. The Certificate Revocation List (CRL) maintained by C-BAS enumerates revoked certificate (using serial number) along with the reason for revocation and the revocation timestamp. CRL always bears its time of generation, next update time and the digital signatures of the issuing authority. CRLs distributed by C-BAS are of type X.509 version 2 encoded in PEM [14] format and fully conform to IETF standards [15]. In future releases, C-BAS is planned to get support for Online Certificate Status Protocol (OCSP) [16] that is considered a more reliable way of certificate status checking in real time.

V. EXPEDIENT AS A USER-AGENT

By design any user-agent compliant with FAPIv2 is compatible with C-BAS. For example, OMNI [13] is a command line user-agent from GENI that can interact with C-BAS. Although researchers favor command line based user-agents for large-scale experiments, GUI-based user-agents are preferred in classroom settings and for demonstration purposes. To the best of our knowledge there is currently no FAPIv2 compliant GUI-based user-agent available to the research community. This need has been addressed by extending Expedient [5]. In practice, Expedient's own integrated clearinghouse has been plugged out and its communication with C-BAS as a clearinghouse has been realized. It is advantageous that Expedient has built-in support for GAPI [10] which allows users to perform a variety of tasks using its web-based graphical interface including the management of projects, slices, and members as well as reservation of resources from different AMs. Expedient is hosted at the SEF and, therefore, can be trusted to perform certain administrative tasks like, member registration, service registration or de-registration. Authorization for such tasks is performed through user-agent credentials issued to Expedient as shown in Table I. However when performing user-driven tasks (e.g., project or slice creation) Expedient acts like a speaks-as user-agent and presents user credentials for authorization purposes.

A. Certificate-based Authentication

Expedient comes with the traditional password-based login mechanism which is not encouraged by C-BAS as explained in [3]. Instead, C-BAS promotes a more secure certificate and private key based authentication. The private key associated with the certificate consists of randomized text having a hard to guess relationship with the public key embedded in the certificate. Its unique cryptographic properties eliminate a number of vulnerabilities inherent to password based authentication mechanism. This work realized certificate-based user authentication for Expedient through an extension. Fig. 4 illustrates the process of user logging onto Expedient using the certificate and private key. At the login webpage that belongs to Expedient frontend, the user is prompted for her certificate and private key. Although the private key is required in the login process to verify the user's ownership of the presented certificate, its transmission all the way to the backend server is not desirable. Therefore, possession of private key should be proved indirectly, for example, through encrypting a challenge token with the private key in user browser and decrypting it using the user public key at the backend. For this purpose, we exploited the unique token that comes embedded in the login page to prevent Cross-Site Request Forgery (CREF) attacks. CSRF is an attack that lures the victim to submit a malicious request to a website to which the victim is currently authenticated. The goal of such an attack is to perform actions on behalf of the victim through inheriting her identity and privileges. The use of CSRF tokens enables the backend server to distinguish between forged and legitimate requests.

In conjunction with certificate based login, the CSRF token is treated as a challenge token from the backend which must be digitally signed by the user's private key. This task is performed in the login page so that the private key never leaves the user's machine. The digitally signed token along

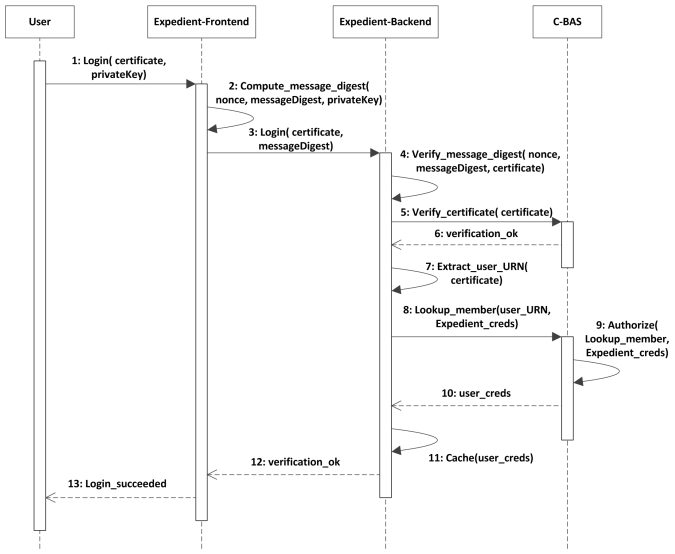

Fig. 4. Certificate-based user authentication with Expedient and C-BAS

with the certificate is transmitted to the backend server that verifies the digital signature using the public key embedded in the certificate. If verified it implies user's possession of private key associated with the presented certificate. Next, it is checked if certificate is valid, non-revoked, and issued by clearinghouse of C-BAS. This is performed by sending the user certificate to MA for verification. If passed, user authentication succeeds otherwise an error message is shown to the user. In the final step, the user's system credentials are fetched from C-BAS and cached at Expedient. For this purpose, Expedient performs a member lookup at C-BAS by providing its own credentials for authorization. As result, user credentials are retrieved and used in communication with C-BAS as shown in Fig. 3 and 5.

Certificate-based authentication enables the "friend-of-a-friend" feature which is highly desirable in scenarios where multiple SEFs have to federate. The federating SEFs would just need to exchange root certificates and mark them as trusted. By the virtue of this exchange, not only users of one SEF get authenticated at the other SEF but also the slice credentials issued by one SEF get recognized at the other.

B. Use Case: Resource Allocation

Fig. 5 demonstrates Expedient's interaction with C-BAS and an AM as a use case where an experimenter performs resource allocation. In first step, the user requests a list of available resources using Expedient. Next, in order to reserve resources valid slice credentials must be presented at AM. Though slice credentials can be acquired from C-BAS but this request should be authorized through project credential. Therefore, project a membership lookup is performed as an intermediary step to obtain project credentials for inclusion in lookup request for slice credentials. Onward, slice credentials serve the purpose of authorization for all GENI AM API calls, like "allocate", "provision" etc. It is encouraged that AMs should be equipped with the necessary mechanism to authorize API calls locally, however, C-BAS API also offers a method to do this task for an AM.

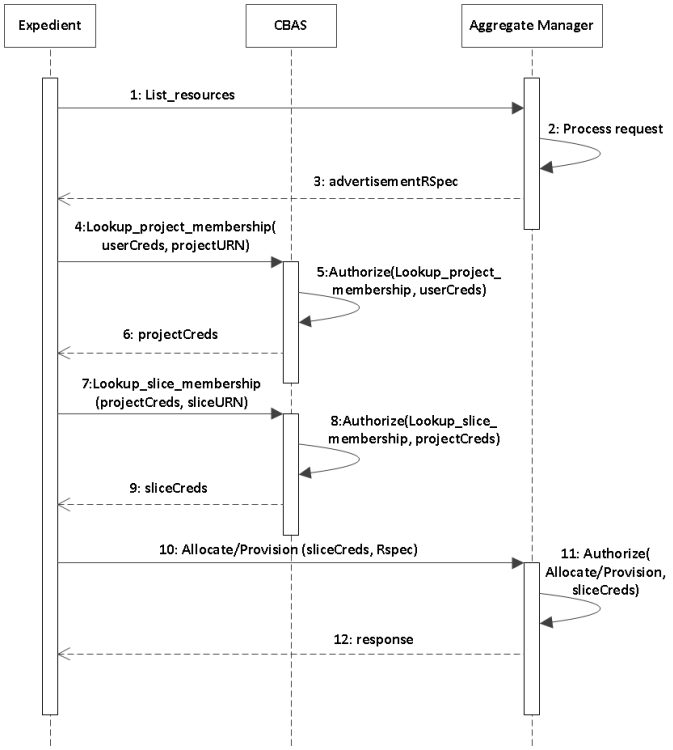

Fig. 5. Expedient's interaction with C-BAS and AM to request resource allocation

VI. Performance Evaluation

The performance of this C-BAS reference implementation has been evaluated through stress testing to judge its stability, scalability, and responsiveness. For this purpose, a set of operations was selected that encompasses all common functionality expected from C-BAS in typical SEFs. The performance evaluation tests were executed on a virtual machine running Linux (namely, Ubuntu 12.04.5 LTS) with a mere 2 GB of RAM. The server that hosts this virtual machine has a five-year old Intel® Xeon® E5507 2.27GHz CPU and employs KVM to host another six active virtual machines. The rationale behind not opting for state-of-the-art hardware was to show that C-BAS imposes no stringent requirements for computational resources to support even a sizeable SEF. Although C-BAS supports multithreading to process multiple requests in parallel and, therefore, has the capability to fully exploit today's multi-core CPUs for this set of stress tests a configuration with a single thread was chosen. In the following, a description and analysis is provided for four performance evaluation tests.

The user registration operation is performed once for each new experimenter accepted to use the experimental facility. When received at C-BAS such requests are first put through an authorization process involving certificate verification, credentials validation, and privileges check of the requesting entity. Fig. 6 shows the results of stress tests where user registration requests are sent to C-BAS in batches of different sizes. The distributions of request processing times have been depicted in the form of boxplots with whiskers representing maximum and minimum values of the sample space. The test scripts and C-BAS are running on the same virtual machine therefore negligible network delays are involved as all requests get

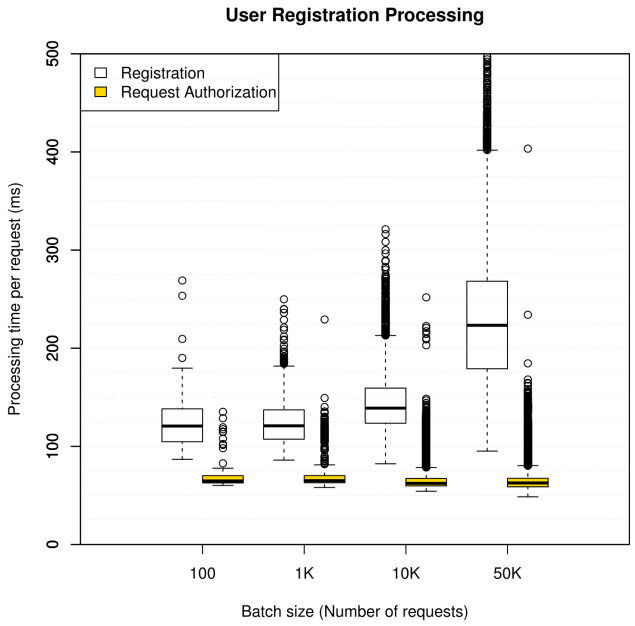

Fig. 6. Boxplots of request processing times for user registration at C-BAS.

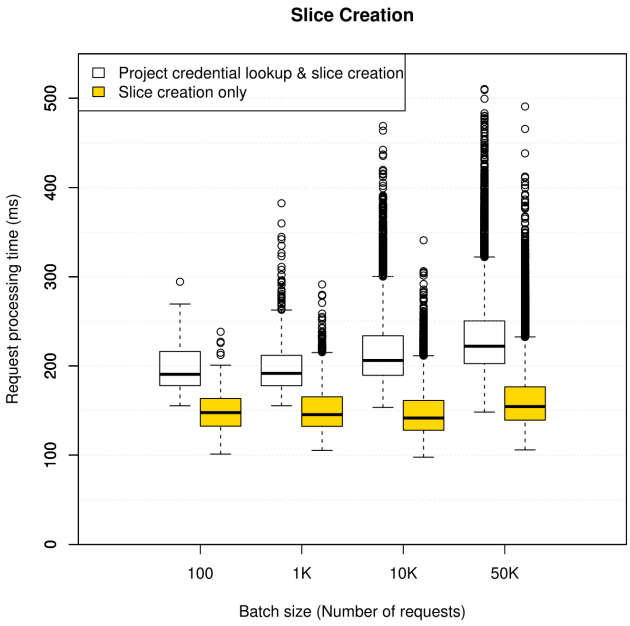

Fig. 7. Boxplots of request processing times for project credentials lookup and slice creation.

routed through the local loopback interface. It can be seen that the authorization process takes a well-bounded amount of time in the range of 50–75 ms for the set of hardware and software used in our stress tests, and remains unaffected by the size of the database. On the contrary, the total registration processing time increases as the database size of registered users grows. This is mainly because of operations related to database writing and indexing performed by the noSQL MongoDB daemon. In addition to database writing, the registration process also involves request authorization, as well as, the creation of user certificate (1024 bit RSA key pair) and signed credential. The processing time for these operations is not influenced by database growth.

For most SEFs, the number of registered users would fall in the range 1–10K and hence would experience median registration delays in the range of 120–140 ms. In short, although user registration is a one-time process performed for each new experimenter and is not a time-critical operation, it is handled efficiently by C-BAS.

Slice creation, illustrated in detail in Fig. 3, is an operation that is performed more frequently than the user registration. The project credentials, required to authorize slice creation request, are looked up as shown in Fig. 3:Step1. The request to create a slice is then sent by the user-agent and processed by C-BAS in Fig. 3:Step2. The results of a stress test that encompasses these operations are shown in Fig. 7. C-BAS serves such requests quite efficiently as evident from the near uniform distribution of response times with a median value of 150 ms for slice creation and 200 ms for a combined operation of project credentials lookup and slice creation. Moreover, the performance of C-BAS scales well as the number of created slices grows.

When a user logs onto Expedient, C-BAS has to provide its services for user authentication and credentials lookup as shown in Fig. 4:5–10. The performance of these operations has been evaluated using a setup where another virtual machine sends 1,000 requests to C-BAS in a sequential manner. In 90% of these requests, the credentials of a random registered user are demanded while in rest of the cases the requested user credential does exist not in the database. This makes such test more realistic by covering the cases where incorrect credentials are provided for user login at Expedient. Moreover, the evaluation test is performed for different sizes of the C-BAS database to assess their impact on performance. In Fig. 8 boxplots of request processing times for user authentication and credentials lookup are presented. The authentication process (Fig. 4:5–6) involves user certificate verification process including check against certificate revocation list. It can be noticed that user authentication process is mostly executed within 50–60 ms while authentication and credential lookup collectively (Fig. 4:5–10) takes 65–80 ms for completion. In addition, the request processing time is marginally affected by the database growth.

Project and slice credentials lookups are among the most frequent requests to be processed by C-BAS. For example, slice credentials are included in almost all GAPI calls as means for authorization (see Fig. 5). Performance evaluation of project and slice lookup is carried out in a similar manner as explained for the previous evaluation test; the resulting measurement distributions are illustrated in Fig. 9 as boxplots. It is evident that C-BAS can lookup credentials of a random slice (Fig. 5:4–6) in a database of up to 50K slices in less than 75 ms. Similarly, project and slice credentials lookups (Fig. 5:4–9) are collectively processed with a median value well below 150 ms. Furthermore, it is apparent that the credential lookup processing times are marginally affected by database size. With such performance C-BAS can carry out approximately 15 lookups per second in a sizeable database of 50K registered

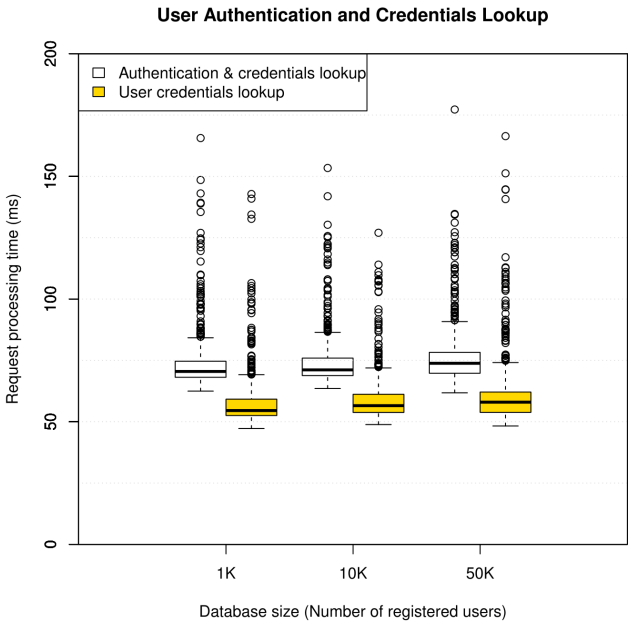

Fig. 8. Boxplots of request processing times for user authentication through certificate verification and user credentials lookup.

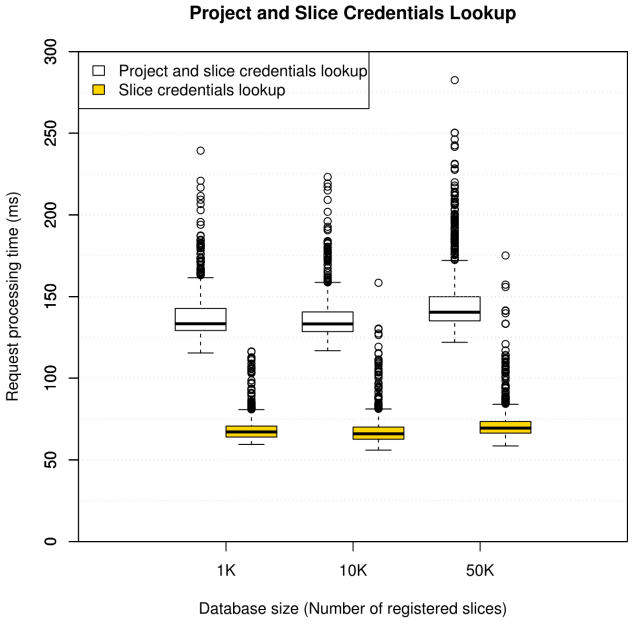

Fig. 9. Boxplots of request processing times for project and slice credentials lookup at C-BAS.

slices without issues arising from the (purposefully chosen) limited computational resource execution environment.

VII. Conclusion and Future Work

This paper discusses the importance of a comprehensive authentication and authorization mechanism for modern SDN experimental facilities (SEFs) and their federations. Traditional password-based authentication methods are not suitable for SEFs due to their design limitations and inherent vulnerability to common attacks. C-BAS is an advanced certificate-based AAA architecture that was proposed as an alternative in our previous published work [3]. Based on those design specifications, this follow-up paper reports the implementation-level details of C-BAS. We briefly introduced EiSoil, an open source lightweight framework which served as a fundamental building block for the C-BAS implementation. C-BAS services have been implemented through several software plugin modules, described in detail. In addition, request processing steps for a number of use cases have been elaborated with the help of UML sequence diagrams. Finally, the performance of this open source implementation has been evaluated through stress testing. The obtained results provided evidence of C-BAS's computational efficiency, scalability, and fitness for deployment in large scale SEFs.

Future work will consider support for dynamic policies to further enhance SEF autonomy, integration with monitoring infrastructure of SEFs, support for advanced credentials formats like ABAC [17], and feasibility check for deployments of C-BAS in certain parts of service providers' networks.

Acknowledgment

This work was conducted within the framework of the FP7 FELIX project, which is partially funded by the Commission of the European Union.

References

[1] "Lightweight Directory Access Protocol (LDAP): The Protocol," IETF RFC 4511, Jun. 2006.

[2] B.C. Neuman, et al., "Kerberos: an authentication service for computer networks," *IEEE Communications Magazine*, Sept. 1994.

[3] U. Toseef, et al., "C-BAS: Certificate-based AAA for SDN Experimental Facilities," in *Proc. EWSDN*, Sept. 2014.

[4] G. Carrozzo, et al., "Large-scale SDN experiments in federated environments," in *Proc. SACONET WOSDN*, March 2014.

[5] J. Naous, et al., "Expedient: A centralized pluggable clearinghouse to manage geni experiments," Jan. 2010.

[6] "Protogeni website," http://www.protogeni.net.

[7] "Slice Facility Architecture 2.0," http://groups.geni.net/geni/attachment /wiki/SliceFedArch/SFA2.0.pdf, July 2010.

[8] "GENI Clearinghouse," http://groups.geni.net/geni/wiki/GeniClearinghouse.

[9] "Common Federation API," http://groups.geni.net/geni/wiki/ CommonFederationAPIv2, Nov. 2013.

[10] "The GENI Aggregate Manager API," http://groups.geni.net/geni/wiki/GeniApi, 2013.

[11] R. Krzywania, et al., "Experiment Use Cases and Requirements," http://www.ict-felix.eu, FELIX Deliverable D2.1, Sept. 2013.

[12] "MongoDB database," http://www.mongodb.org/.

[13] "The Omni client," http://trac.gpolab.bbn.com/gcf/wiki/Omni.

[14] "Privacy Enhancement for Internet Electronic Mail," IETF RFC 1421–1424, 1993.

[15] "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," IETF RFC 5280, May 2008.

[16] "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP," IETF RFC 6960, June 2013.

[17] N. Li, et al., "Design of a role-based trust-management framework," in *IEEE Symposium on Security and Privacy*, 2002.