Fault-tolerant Distributed AAA Architecture Supporting Connectivity Disruption

Karri Huhtanen*, Antti Kolehmainen†
*Radiator Software Oy, Finland
Email: karri.huhtanen@radiatorsoftware.com
†Tampere University, Finland
Email: antti.kolehmainen@tuni.fi

***Abstract*—Authentication, Authorisation and Accounting (AAA) infrastructure provides a vital cog in the authentication and authorisation security frameworks of both small and large organisations. In many scenarios, organisations have come to rely on the high-quality Internet connectivity in order to deploy AAA services to the cloud, or become part of a larger AAA federation. However, many scenarios, such as with industrial networks deployed in challenging environments, constant high-bandwidth connectivity is not guaranteed. We therefore propose a novel distributed AAA architecture capable of service delivery even during connection disruption or connectivity intermittence. The architecture relies on co-operating AAA servers to provide replication and fault-tolerance residing on-premise and in cloud/core-based deployments.**

***Index Terms*—AAA, RADIUS, TACACS+, distributed systems, network security, availability**

I. Introduction

The development and growing adoption of the cloud services in organisations have led to a centralisation of organisational Authentication, Authorisation and Accounting (AAA) infrastructure and increased dependency of a stable, continuous Internet connectivity. While many organisations cannot function without Internet connectivity coupled with AAA availability, there are organisations, infrastructures, and services for which the AAA functionality is required to operate without the Internet connectivity and survive with its integrity intact regardless of the length and frequency of the disruptions.

Multi-Factor Authentication (MFA) is a subset of the AAA infrastructure which is being increasingly used for secure user authentication and authorisation of operations critical network infrastructure devices. Centralisation of the AAA and MFA increases the organisation's dependency of a continuously working connectivity between organisational infrastructure and the AAA/MFA service provider, especially when using verification methods requiring Internet/network connectivity. This kind of dependency is not acceptable for some organisations. Neither is performing AAA/MFA communication over third party infrastructure and networks.

In this short paper, we examine initial design, implementation and deployment issues related to our proposed hybrid architecture model for fault-tolerant distributed AAA architecture with the specific use case of ensuring the multi-factor authentication functionality over network disruptions.

II. Related Work

The research work, architecture, and functionality rely heavily into utilising and combining standardised Internet Protocols for AAA such as RADIUS [1] and TACACS+ [2] as well as the additional functionality provided to them by standards like RadSec [3] and EAP-TLS [4].

For multi-factor authentication the Internet standards such as HOTP [5] and TOTP [6] provide suitable open standardised basis for interoperable token authentication. EAP-TLS [4] and its newest update EAP-TLS v1.3 [7] make it possible to extend the local authentication functionality to client certificate based network authentication even for end user devices.

As our proposed architecture combines centralised AAA with local onsite AAA, synchronisation of the credential databases and log information is required. For these we will rely on database and file system synchronisation as well as log transfer to the existing implementations already available by the software components we are using. Additional attention will be given to configuring these components securely.

III. Aims and Objectives

Our aim is to validate the proposed architecture and technologies by designing and implementing proof-of-concept deployment first utilising virtual hosts and networking in the testbeds and facilities offered by the information as well as automation cybersecurity laboratories at Tampere University. In addition, we plan to incorporate the research into other cybersecurity research testbeds available at educational organisations in the Tampere region in Finland.

The first objective is to design and implement the first proof-of-concept setup utilising virtual hosts and networking to validate the design, find and solve possible issues, and refine the design accordingly.

The second objective is to design and build a model to deploy and distribute the solution not only to other sites, but also to other virtualisation options, such as containers running within industrial and enterprise network switches.

The third objective is to encourage and increase the use of architecture and solution and gather feedback first from internal partners and when possible from industry partners to develop the architecture and concept further.

IV. Design

The initial architecture of the solution is depicted in the Figure 1. The architecture consists of a centralised AAA service representing primary AAA cloud service and onsite AAA services representing secondary or backup AAA service running on an independent remote site. In a normal situation, remote sites are assumed to be using centralised AAA either through onsite AAA service or directly as their operational AAA service. Disruptions in the connectivity may then result in AAA service changes such as switching from onsite AAA service to centralised AAA or vice versa.

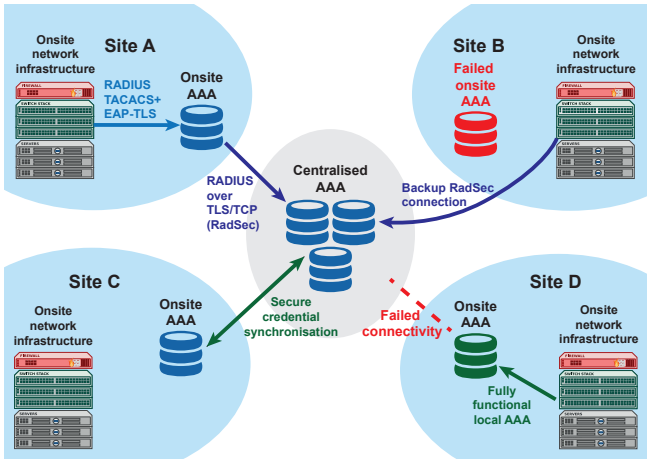

Figure 1. Fault-tolerant AAA architecture

As for the AAA protocols, RADIUS [1], RADIUS over TLS [3] and TACACS+ [2] were chosen since they are most commonly used in authenticating, authorising and accounting network and network device access. These protocols also support the use of one-time-token based authentication algorithms such as HOTP [5] and TOTP [6] as well as EAP-TLS [4] certificate based authentication.

A. Centralised AAA

A production level centralised AAA consists, in an optimal case, of a redundant database cluster, RADIUS and RADIUS over TLS proxies and servers as well as possible user interfaces for managing the AAA service (Figure 1). This most likely requires the use of multiple virtual hosts and/or containers for service delivery.

Our proof-of-concept setup will use one database server and two RADIUS servers with proxy and authentication capabilities. This setup allows us to verify our RADIUS design and functionality as well as develop the database configuration further. We may also have to introduce a VPN service between the centralised AAA and the onsite AAA if the database security features such as access control are not considered secure and flexible enough especially for database contents synchronisation.

B. Fully operational remote site (Site A and Site C)

Site A in Figure 1 is an example of the fully functional remote site. The site has its own AAA service, to which the site network infrastructure is connected with RADIUS and TACACS+ protocols. The onsite AAA can then proxy the requests over TLS-secured RADIUS to the centralised AAA service and thus translate between the unencrypted and encrypted protocols.

The fully functional remote site also has AAA credential synchronisation connection active and is synchronising the credentials from primary centralised AAA to the local AAA. This is planned to be implemented with database management system synchronisation features and will be authenticated and secured at a minimum with client and server certificates, possibly using even VPN if the previously mentioned are considered insecure. Different synchronisation methods will be evaluated during the proof-of-concept work.

C. Remote site with failed onsite AAA (Site B)

If a remote site's onsite AAA fails (Site B, Figure 1), the network equipment can be configured to switch over to centralised AAA. This kind of a failover however requires that the centralised AAA is willing to serve unencrypted RADIUS or TACACS+ or that the network equipment support RADIUS over TLS (RFC 6614) [3]. A better solution may be to add a redundant onsite AAA capable of proxying the traffic over TLS-secured RADIUS.

D. Remote site with failed external connectivity (Site D)

The connectivity failure is a failure we are specifically focusing when designing, implementing, and piloting the architecture. As discussed earlier, the centralised AAA does not work well if the Internet connectivity between site and centralised service is down. Our proposed solution is to constantly synchronise AAA credential information between the centralised AAA and onsite AAA so that in an event of a site connectivity failure, the site is able to function independently until the connectivity is restored. It is one of our objectives to research suitable options and processes for a disconnected site AAA connecting back to the centralised AAA and how synchronisation in these cases should be handled.

E. Secure credential synchronisation

Secure synchronisation of credentials is the most challenging part of the proposed solution. First, the synchronisation connection must have a secure authentication. This requires at a minimum the utilisation of certificate authentication for the centralised AAA and onsite AAA endpoints as usernames and passwords may be trivially compromised. In addition, careful consideration for exposing the database management system directly to the clients connecting over the Internet is in order. If the security is not deemed to be at a suitable level an additional layer should be added utilising VPN connection between the centralised and onsite AAA.

Second, preservation of the AAA credential data integrity over network disruptions requires careful consideration of technical solutions combined with development of recovery processes. Figures 2, 3, 4 and 5 illustrate the problem step by step in more detail.

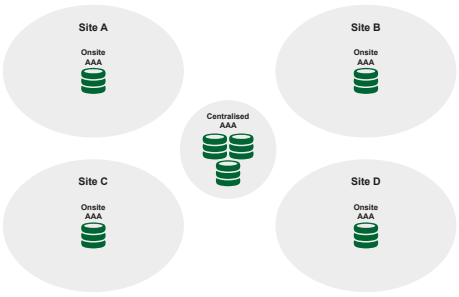

Figure 2. Phase 1: Normal situation: Centralised AAA is in sync with site AAAs.

In the Figure 2 the situation is normal and the centralised AAA is accessible by all the onsite AAA replicas.

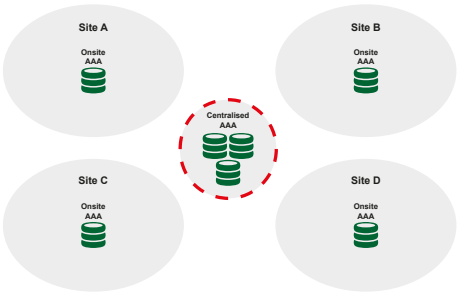

Figure 3. Phase 2: Connectivity loss: Onsite AAAs start to work independently

Then in the Figure 3 the connectivity between the centralised AAA and onsite AAAs is lost. The onsite AAAs start to work independently. If the network connectivity disruption is a short one, there may not be many changes and updates in the AAA credentials, but for example log data, which otherwise would be collected in the centralised location, is now stored or buffered locally.

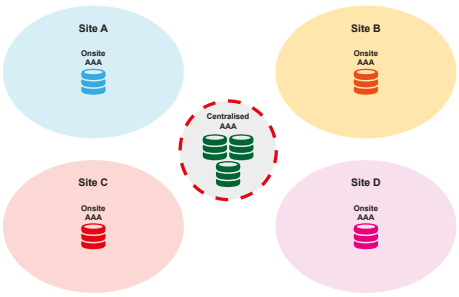

Figure 4. Phase 3: Over time the onsite AAA data diverges from Centralised AAA

If the disruption continues longer, there will be increased need to be make changes and adjustments also to the local onsite AAA database. These changes diverge the onsite replica (Figure 4) further from the originally replicated database. Even more log data is stored and buffered locally. The AAA data between centralised AAA and onsite AAA drifts even further apart.

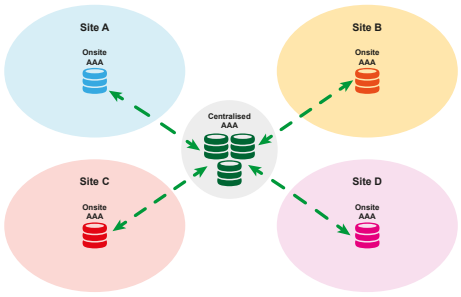

Figure 5. Phase 4: Connectivity restored: Re-Synchronization of data

When the connectivity between the sites and centralised AAA is restored (Figure 5), depending on length of the network disruption and the amount of sites affected, the re-synchronisation of the AAA data needs careful consideration. One potential policy decision to consider is to let the centralised AAA data take the priority over the onsite AAAs, but trust the locally generated data to be imported into the centralised log management. Depending on the amount of changes and updated data in the onsite AAAs this may mean that the changes and adjustments done in the onsite AAA replicas are then lost.

V. Conclusion

In this short paper, we propose a novel distributed AAA architecture capable of service delivery even during connection disruption or connectivity intermittence. The design and implementation described will be undertaken and evaluated across several cybersecurity testbeds capable of mimicking the scenarios described. The iterative design based on the empirical research ensures that the solutions are improved and validated during the process producing better results and feedback, which will also drive standards activity in this area.

VI. Acknowledgements

Funding was provided by S^{2}ERC Appia and ERDF KyLÄ

References

[1] A. Rubens, C. Rigney, S. Willens, and W. A. Simpson, "Remote Authentication Dial In User Service (RADIUS)," RFC 2865, Jun. 2000. [Online]. Available: https://www.rfc-editor.org/info/rfc2865

[2] T. Dahm, A. Ota, dcmgash@cisco.com, D. Carrel, and L. Grant, "The Terminal Access Controller Access-Control System Plus (TACACS+) Protocol," RFC 8907, Sep. 2020. [Online]. Available: https://www.rfc-editor.org/info/rfc8907

[3] K. Wierenga, M. McCauley, S. Winter, and S. Venaas, "Transport Layer Security (TLS) Encryption for RADIUS," RFC 6614, May 2012. [Online]. Available: https://www.rfc-editor.org/info/rfc6614

[4] D. Simon, R. Hurst, and D. B. D. Aboba, "The EAP-TLS Authentication Protocol," RFC 5216, Mar. 2008. [Online]. Available: https://www.rfc-editor.org/info/rfc5216

[5] M. View, D. M'Raihi, F. Hoornaert, D. Naccache, M. Bellare, and O. Ranen, "HOTP: An HMAC-Based One-Time Password Algorithm," RFC 4226, Dec. 2005. [Online]. Available: https://www.rfc-editor.org/info/rfc4226

[6] M. View, J. Rydell, M. Pei, and S. Machani, "TOTP: Time-Based One-Time Password Algorithm," RFC 6238, May 2011. [Online]. Available: https://www.rfc-editor.org/info/rfc6238

[7] J. P. Mattsson and M. Sethi, "EAP-TLS 1.3: Using the Extensible Authentication Protocol with TLS 1.3," RFC 9190, Feb. 2022. [Online]. Available: https://www.rfc-editor.org/info/rfc9190