Authentication and Authorisation for Research and Collaboration

Attribute Authority Operations

A fond of knowledge worth protecting

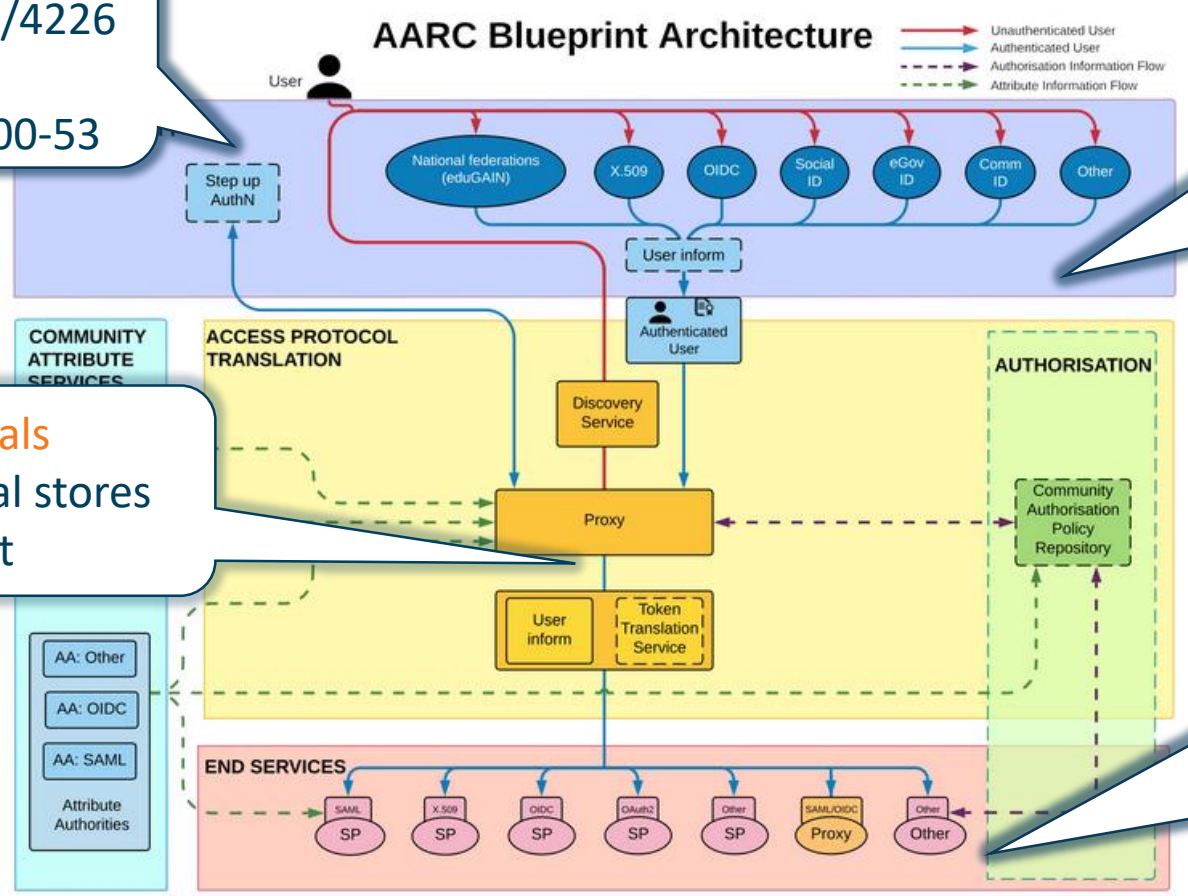
David Groep
AARC Community Policy WG

Nikhef

AEGIS meeting
2020-02-10

Operational guideline landscape for - proxy or source - AAI components

MFA
RFC6238/4226
FIPS140
NISTSP800-53



Authentication/identity sources
Sirtfi
(eduGAIN) baselining
IGTF AP Profiles
NIST SP800-63
eduGAIN sec. team workflow

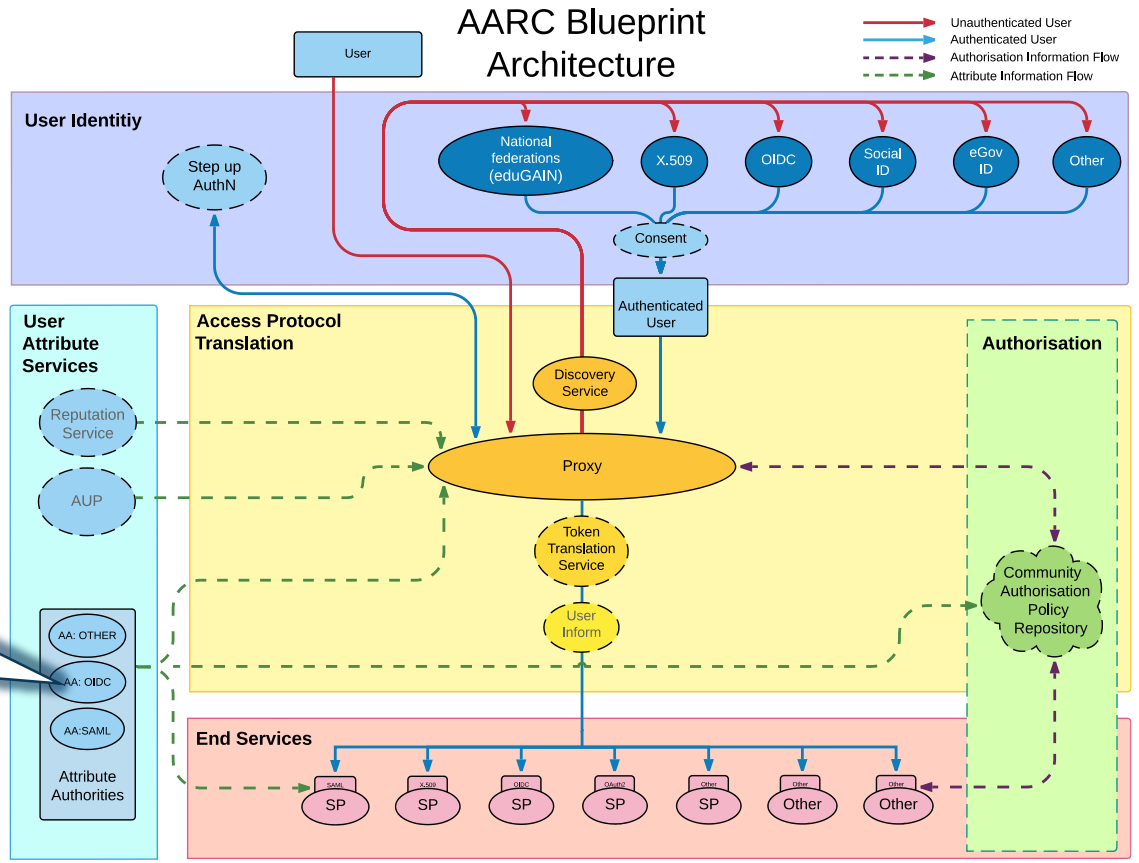
Ephemeral credentials
- trusted credential stores
- protection at rest

Service provider operations
ISO27k
Sirtfi
Infrastructure response plans

Operational security focus in the BPA: beyond just the IdPs

Community membership management directories and attribute authorities

- integrity of membership
- identification, naming and traceability
- site and service security
- protection on the network
- assertion integrity



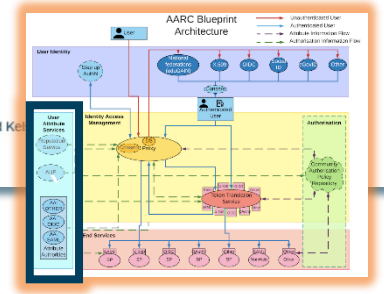
Guidelines for Secure Operation of Attribute Authorities and other issuers of access-granting statements
(AARC-I048, in collaboration with IGTF AAOPS)

AARC-G048: keeping users & communities protected, moving across models

trusted delegation of response from communities to operators, and from services to communities in recognizing their assertions

Structured around concept of "**AA Operators**",
operating "**Attribute Authorities**" (technological entities),
on behalf of, one or more, **Communities**

Guidelines for Secure Operation of Attribute Authorities and other issuers of access-granting statements



Publication Date 2018-11-22
Authors: David Groep;David Kel
Paetow;Maarten Kremers
Document Code: AARC-G048

3.3. Attribute Assertions

1. Assertions provided by an AA must be integrity-protected. They must be signed by the identified AA, or be transmitted over an integrity-protected channel where the server has been authenticated, and preferably both.

Push model
Where the protocol supports it, enable protection also of the messages conveyed over the established channel. Good examples: SAML Attribute Query should enable message signing and use TLS.

Pull model
As a good example: LDAP should enable TLS protection of the channel

3.4.1. Key Management

1. A key used to protect assertions should be dedicated to assertion protection functions.

Push model
If the AA both signs assertions and provides functionality over protected channels, the keys used to sign assertions shall be different from those protecting those channels.

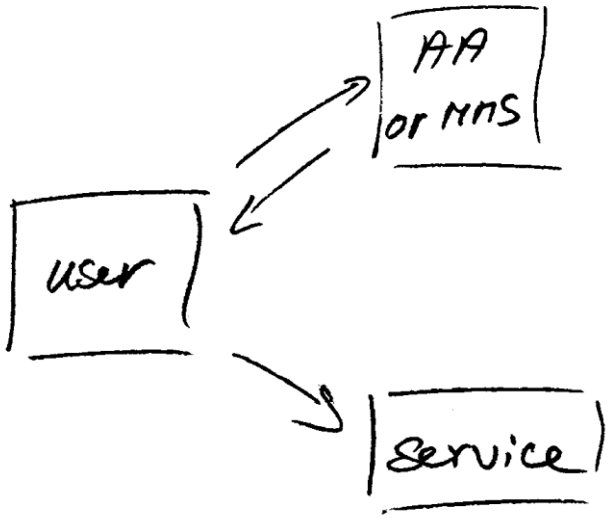
Pull model
The key of the AA must be used solely for protecting connections to its protocol endpoint and ensure an integrity protected and mutually authenticated channel.

https://www.igtf.net/guidelines/aaops/

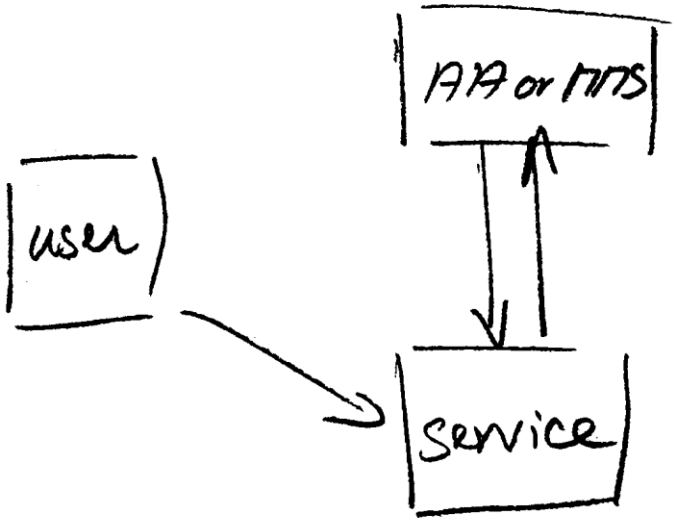
https://aarc-community.org/guidelines/aarc-g048/

Protecting the community membership data and its proxy

- Intentionally targeted broader than just BPA-style communities, since operational security spans data centres and infrastructures using other forms of AA membership management
- PRACE: ‘pull model’ directory-based communities
- BPA: encourages ‘push model’ attribute-carrying service requests



push model – the common BPA method
(e.g. SAML AttributeStatement, VOMS AC)



pull model – common when using directories
(e.g. LDAP in PRACE, GUMS in OSG)

push and pull model diagrams as per RFC2904 – the 3rd (agent) model is uncommon in research/collaboration scenarios except for provisioning

When the AA is in a managed (and in a data centre) …

Many of the recommendations are already implemented ‘implicitly’

- because common software implements it: e.g. signing SAML assertions and JWTs
- because a good data centre already has network monitoring and central logging in place
- because you signed up to Sirtfi (didn’t you?) – so you collaborate in incident response
- because you have trained IT operations personnel looking after the service

And some are intuitive best practice

- like assigning a unique and lasting name to a group
- because implemented controls follow ought to be those that have been documented

Forward looking and specific requirements

Some controls are specific to AA operations and protect against current and future threats:

- minimum signing key length so that the community is not broken in the next few years (at least 112-bit symmetric, i.e. >=2048 bit RSA keys)
- protect the key from data breaches, compromise, ransomware, and exfiltration by using HSM Hardware Security Modules or equivalent controls (and the HSMs you need are not that expensive, or you can even rent them in AWS…)

Or deal with commensurate incident response (you don't want just a big red button):

- re-issuance of attribute statement must be based on fresh data
- release them only in accordance with the community's policy and maximum life time
- require appropriate client authentication before releasing attributes to prevent data breaches
- for non-revocable tokens (like OAuth Access Tokens or PKIX 3820 proxies), limit life time <24hrs (for OIDC, these are anyway typically 15 minutes)

G048 AA Ops guidelines and AA hosting

Guideline was written with both physical and virtual deployment in mind

> "An AA may be run in a virtual environment that has security requirements the same or better than required for the AA, and for all services running in this environment, and it must not leave this security context. **Any virtualization techniques employed** (including the hosting environment) **must not degrade the context** as compared to any secured physical setup. Only AA Operator designated personnel should have control over the virtualisation and security context of the AA."

- if you can host it on-prem, the easiest solution is to host it on your security-service VM infrastructure (e.g. alongside your IdP, your AD, or your master LDAP servers) to limit guest compromise)
- If you run it in a cloud provider, select a provider that offers proper security and network controls, implement account role separation, and deploy the offered protections. E.g. in AWS you have *a lot* of controls available to do so. But Azure &co hve the same. – and rent a netHSM

Deployment guidance included …

1. Assertions provided by an AA must be integrity-protected. They must be signed by the identified AA, or be transmitted over an integrity-protected channel where the server has been authenticated, and preferably both.

Push model
Where the protocol supports it, enable protection also of the messages conveyed over the established channel. Good examples: SAML Attribute Query should enable message signing and use TLS.

Pull model
As a good example: LDAP should enable TLS protection of the channel

6. The network to which the AA system is connected must be highly protected and suitably monitored.

Service access should be protected by at least two distinct control layers not running the same software or operating system, and the AA system must not run any unnecessary services. The network should be monitored for anomalous events, such as detection of data exfiltration, credential probing, and brute-force attacks. It should preferably also be protected

Implementation of AA Operations Security guidelines

1. review what is now in AAOPS G048 and comment
2. the comments will feed into an FAQ and additional guidance
3. evaluate feasibility by adopting AAOPS G048 – volunteer welcome!
4. It will then evolve and likely
   be amended to include these lessons in FAQ or document
5. feel safe (or at least safer) when hosting attribute sources
   and offering them to users and communities!

https://aarc-project.eu/guidelines/aarc-g048/

Thank you

Any Questions?

https://aarc-community.org

The work leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme and other sources.