Security Incidents and Countermeasures: Blocking compromised credentials

Oscar Koeroo

Goal

Sites want to prevent one or more users to access their resources, irrespectively of the underlying middleware services or AuthZ frameworks

The requirement

Such controls should be simple to apply

The wish list

1. **Both the sites AND the infrastructure need to have a central point of control for denying access**

2. **It should not require middleware/service experts**

How to deny access

What controls do we have…?

How to deny access (0)

- **Certificate revocation by the CA**
  - You certificate can only be revoked if:
    - the private key was **compromised or exposed**
    - On request by the End-Entities, RAs and the CA itself
    - On request of other parties
      - *In this situation, the CA will need some sort of proof of the compromise or exposure*
  - The CRLs must be fetched by the relying parties
  - Is a relative slow process

- **Certificate revocation is not blacklisting!**
  - It's revoking somebody's (claimed) identity
  - OSCT generally use certificate revocation as a cleanup at the end of the incident resolution

How to deny access (1)

- **Blacklisting based on the DN in the MW**
  - The quickest solution is to blacklist a user based upon the DN
    - Everything is logged with at least the DN as a key
    - Evidently based upon that DN the access can be denied in a blacklist
  - By the sys-admins local decision a user can be banned
    - Possibly encouraged by the Operational Security Coordination Team (OSCT)

- **LCAS, gJAF and SAZ can perform this task**
  - Not all MW employs these frameworks/interfaces or offers a similar functionality

How to deny access (2)

- **Blacklisting based upon VOMS attributes**
  - Blacklisting (sub)groups or users with a specialized role
  - By using the FQAN pattern matching mechanism you can exclude certain groups of getting access
    - Having a BL-ed FQAN in the proxy would give you a deny
- **SAZ can blacklist FQANs**
- **LCAS can only do this when using the GACL (=minimalized XACML)**
  - Simple FQAN blacklisting will be made soon

How to deny access (3)

- **(Temporary) disablement or removal at the VOMS service**
  - Prevents VOMS credential creation and renewal
  - Stops the user from accessing the Grid based on VO credentials*
  - Prevents VOMS-only resource access

- **Issue:**
  - VOMS credentials were intended to be valid are valid for 24h normally, but are ‘upgraded’ to a new default of 72h
  - * Mkgridmap tool is too **nicely** configured
    - If any VO sync fails in the grid-mapfile creation, the behavior is to only allow adding DNs. This will not remove users from the grid-mapfile

How to deny access (*)

- **Other options**
  - SAZ
    - Exclude a CA
    - Deny access based upon the serial number of a user's certificate

Practical problems

- **Not all Middleware implements such a mechanism or makes use of AuthZ frameworks**
- **Non-uniform approach in denying access in MW**
  - Different formats
    - in a flat file
    - X(AC)ML (policy)
    - others formats
  - Different location
    - file
    - database
  - Different editors required
    - Vi / emacs
    - MW specific editor
    - Database tool
- **The blacklisting mechanism should have wildcard support**

Required solutions

- **All Grid MW that provide access to computing or storage resources MUST implement an Authorization mechanism or use an existing Authorization framework to prohibit access to its resources**

- **All Authorization software MUST be able to handle a simple flat file for DN based and FQAN based blacklisting**
  - The file SHOULD be read directly, but its contents MAY be copied to an other format or location
  - All MW MUST be able to read the one file on the system(s)
  - Wildcard support MUST be a supported feature