UNICORE/X security basics

Krzysztof Benedyczak
UNICORE Security PT
Short outline

• Short UNICORE architecture tour
• UNICORE environment
• Configuration
• Logging
• Authorization
• What was not covered, where to search for help?
UNICORE Architecture

- **Client layer software**
  - UNICORE

- **Gateway**
  - Registry
  - Broker

- **UNICORE/X**
  - IDB
  - TSI
    - Batch system

- **UVOS Service**
  - VO

- **Remote data**
  - USpace
  - XUUDB
UNICORE Architecture

- Client layer software UNICORE
- Gateway
- Registry
- Broker

UNICORE Atomic Services
- XACML
- XNJS

Gateway
- UNICORE/X
- IDB
- TSI
- Batch system

UNICORE/X
- IDB

UNICORE Atomic Services
- XNJS
- IDB

UAS container
- UNICORE Gateway

Remote data
- USpace

XUUDB
- Gateway

UVOS VO Service
- Gateway
UAS container

• Most of the UNICORE servers (UNICORE/X, Registry, Workflow Service, Service Orchestrator, ...) use the same UAS container.
  – UNICORE Gateway and TSI are exceptions.

• Security is therefore configured in the same way for all UAS services.

• UNICORE Gateway is special but its configuration is simple.

• TSI is rarely reconfigured/maintained.
Network environment

- Only one port must (and should) be opened on a site's firewall: the Gateway's external port.
- Only the Gateway's machine address must be public.
- It is strongly suggested to block access to other ports used by UNICORE services.
- There must be network communication possibility between Gateway and all other services except TSI.
  - And between UNICORE/X and TSI.
Basic operations

• Starting & stopping:
  – INSTALLDIR/start.sh INSTALLDIR/stop.sh
    • All services that were installed.
    • From the script contents you can judge dependencies.
  – INSTALLDIR/COMPONENT/bin/{start|stop}.sh
    • For the specified component only.
    • For TSI it is bit different: INSTALLDIR/tsi/bin/{start|kill}_tsi
  – There are RedHat-ish init scripts available in extras/redhat-init.d and SUSE template.
Security related configuration files

• **Gateway (in conf/):**
  - `gateway.properties`
    - main config (public address; tuning; advanced options)
  - `connections.properties`
    - list of accessible services behind gateway
  - `security.properties`
    - keystore and truststore setting
  - `crlcheck.properties`
    - CRL sources
Security related configuration files (2)

- **UAS-based servers (in conf/):**
  - `wsrflite.xml`
    - Basic network&WS configuration: keystore and truststore, available web services.
  - `uas.config`
    - General UNICORE configuration: authorization (and much more)
  - `security_policy.xml`
    - XACML policy. Usually not touched.
  - (only in UNICORE/X) `xnjs_legacy.xml`
    - Configuration of job processing engine + TSI connection.
Log files

• UAS servers and Gateway log through log4j.
  – TSI logs are nearly useless (you can check whether TSI has been started correctly).
• Log files are in the service log/ subdirectory.
• By default rolled everyday.
  – Old log files are not automatically removed.
• Configuration in component's conf/logging.properties file.
Logging

• To debug security problems (e.g. user can't access the grid for unknown reason) set log4j.logger.unicore.security to:
  – DEBUG - the most useful, lot of diagnostics is printed.
  – TRACE - as DEBUG and much more - mostly useful when reporting bugs to developers.
Logging (2)

- To set what is actually written modify:
  
  ```
  log4j.appender.A1.layout.ConversionPattern=\n  %d [%t] %-5p %c %x - %m%n
  ```
  
  • `%c` will print the whole context for which you can set level.

- You can set level of log4j.logger.CONTEXT to get messages only from the one module. This is hierarchical. E.g.:
  
  ```
  log4j.logger.unicore.services=INFO
  log4j.logger.unicore.services.RegistryEntryUpdater=DEBUG
  ```

- Always check your log files for messages with ERROR or FATAL level. WARN is also suspected.
Authorization

This information is valid for upcoming 6.3.2 and mostly valid for 6.3.1.

• Attribute based - in UNICORE authZ policy is usually unchanged, but user attributes are modified.

• Attributes provide information used for:
  – authorization (whether access is granted)
  – incarnation, i.e. mapping a user to the local system

• Available attribute sources:
  – XUUDB - default. A very simple standalone server.
  – UVOS - advanced, flexible with groups support, SAML service.
  – File - simple per-host attributes list read from a file.
Authorization (2)

• Multiple attribute sources can be used together with a configurable combing policy:
  – MERGE_LAST_OVERRIDES - all sources are evaluated, subsequent sources override.
  – FIRST_APPLICABLE - attributes from the first source that has any attributes for the user are used.
  – FIRST_ACCESSIBLE - attributes from the first source that is accessible are used.
  – MERGE - union of attributes is returned.

• You can use meta-attribute source to form subchains (with different policy).
Example

# The main chain configuration:
  uas.security.attributes.order=UVOS_CLUSTER FILE
  uas.security.attributes.combiningPolicy=MERGE_LAST_OVERRIDES

# The FILE source cfg:
  uas.security.attributes.FILE.class=
  eu.unicore.uas.security.file.FileAttributeSource
  uas.security.attributes.FILE.file=conf/localAttributes.xml

# The UVOS_CLUSTER is a sub chain:
  uas.security.attributes.UVOS_CLUSTER.class=
  de.fzj.unicore.uas.security.util.AttributeSourcesChain
  uas.security.attributes.UVOS_CLUSTER.order=UVOS1 UVOS2
  uas.security.attributes.UVOS_CLUSTER.combiningPolicy=FIRST_ACCESSIBLE

# And configuration of the two real sources used in the sub chain:
  uas.security.attributes.UVOS1.class=
  eu.unicore.uas.security.vo.SAMLPullAuthoriser
  uas.security.attributes.UVOS1.configurationFile=conf/vo.config1
  uas.security.attributes.UVOS2.class=
  eu.unicore.uas.security.vo.SAMLPullAuthoriser
  uas.security.attributes.UVOS2.configurationFile=conf/vo.config2
Banning users

• With default authZ policy a user must possess the attribute role with user value to gain access.

• Remove the attribute (or set to 'banned') to ban the user.
  – You can do this in your XUUDB, file or UVOS.
  – If you don't have access to the global users DB (usually UVOS) then override the role in your local DB (usually XUUDB).
  – When using UVOS you can assign attributes for groups too.
Advanced topics

• UNICORE without gateway.
  – To improve file transfer speed.
  – Supported, but bit risky.

• Securing UNICORE/X<->TSI connection.
  – Requires extra work (you need special Perl modules for TSI); suggested when not trusted users have access to UNICORE/X machine.

• Restricting Registry access.
  – Suggested.

• Changing the XACML policy.
  – For advanced access tuning, though XACML is hard.
Reporting problems

- http://sourceforge.net/projects/unicore/develop
  - Bugs and Feature Request
  - Please log in to SF first
  - Check open bugs if your problem isn't already reported.

- Soon it will be possible to use GGUS.
Documentation links

• Main entry point for official documentation:
• Wiki (slowly growing)
• Distribution docs/ directory.
• Config files are well commented.
• If there is no information there ask at unicore-support [at] lists.sourceforge.net
Thank you!

EMI is partially funded by the European Commission under Grant Agreement RI-261611