Channel Access Security

2006

kasemirk@ornl.gov

Channel Access Security

- The *IOC Application Developer's Guide* has a full chapter about AS, about 20 pages for the R3.14.8 release.
- The CA server decides what type of access is permitted. For the IOC, this is implemented as follows:
  - Each record is assigned to an Access Security Group by setting its ASG field
    - "DEFAULT", "VAC_ENGINEER", …
  - Each ASG is defined via Rules.
    Read and write permissions depend on the
    - Name of the user who runs the CA client
    - Name of the computer on which the CA client runs
    - Type of record field (two, fixed categories)
    - Optionally a CALC-record type computation

Pros and Cons

- **CA Security is very flexible**
  - **Prohibit "write" access to a list of PVs, unless accessed by**
    - **Specific user**
    - **At specific computer**
    - **Under specific circumstances (machine mode, …)**
- **Limitations**
  - **Designed for high performance in "friendly" environment.**
    - **Won't affect access via IOC shell.**
    - **User and computer names are sent by client, without authentication, in clear text.**
- ***Meant to aid operations, avoid mistakes, not to prevent malicious attacks.***

Default Setup

- Doing nothing is equivalent to this:
  - Create file "as.cfg":

```
ASG(DEFAULT)
{
        RULE(1, READ)
        RULE(1, WRITE)
}
```

  - Add this line to your st.cmd:

```
asSetFilename("path_to_the_file/as.cfg")
```

- Result:
  - Every record uses the "DEFAULT" ASG.
  - … which allows full read/write.
  - The 'asprules' and 'asdbdump' commands now show something (on vxWorks, use 'asDump')
- Caveat:
  - If the AS config file does not exist or contains an error, all access is prohibited!
  - Use 'ascheck' on the host before loading a file into the IOC.

Read-Only Example

- Group that allows read, but no write:

```
ASG(READONLY)
{
  RULE(1, READ)
}
```

- To have an effect, set the ASG field of at least one record to READONLY.
  - One can change ASG fields at runtime.
  - Via Channel Access, unless AS prohibits it…
- 'caput' will show that the old and new values stay the same
- EDM will change cursor when over read-only field.

Restricted Example

- Limit write access to
  - members of a user access group UAG,
  - while on a computer in the host access group HAG:

```
UAG(x_users) { ubuntu }
HAG(x_hosts) { ubuntu }
ASG(X_TEAM)
{
  RULE(1, READ)
  RULE(1, WRITE)
  {
     UAG(x_users)
     HAG(x_hosts)
  }
}
```

- Caveats:
  - The CA *client library* sends the user and host names to the server. Especially the host name can be tricky:
  - It's *not* the client's IP address!
  - It's the result of the 'hostname' command,
  - … which might differ from the DNS name
  - The 'casr' command on the IOC can sometimes help to show who and from where is connecting via CA, and the 'asdbdump' command shows who they pretend to be.

Mode-Based Example

- Limit write access to times where some variable meets some criteria
  -
```
ASG(MODE)
{
  INPA(tx:setpoint)
  RULE(1, READ)
  RULE(1, WRITE)
  {
    CALC(A < 50)
  }
}
```
- This is based on the same code as the 'CALC' record
  - One can assign inputs 'A' to 'L'.
  - The computation should result in 0 or 1, the latter allowing access.

RULE(<level>, <what>)

- <level> is 0 or 1.
  - The dbd file assigns each field to an access security level. Fields that are typically changed during operation are on level 0.
    - Example: For the AI record, VAL is level 0, the rest is level 1.
  - Rules for level 1 also grant access to level 0.
  - Example: Everybody can write 'VAL' (level 0), but restrict other fields:

```
ASG(WRITE_SOME)
{
  RULE(1, READ)
  RULE(0, WRITE)
  RULE(1, WRITE)
  {
     UAG(x_users)
     HAG(x_hosts)
  }
}
```

- <what> is NONE, READ, or WRITE
  - Plus an optional TRAPWRITE, which will cause invocation of a 'trap write listener', i.e. custom C code that might be added to the IOC. This can be used to log write access by user and host, it doesn't otherwise affect access security.

And that's all I have to say about that!

Acknowledgements

- Material and ideas have been copied from the IOC Application Developer's Guide
  - Marty Kraimer, Janet Anderson, Andrew Johnson (APS) and others