Channel Access Security

Kay Kasemir

ORNL/SNS

kasemirk@ornl.gov

July 2017

Material copied from the IOC Application Developer's Guide

Marty Kraimer, Janet Anderson, Andrew Johnson (APS) and others

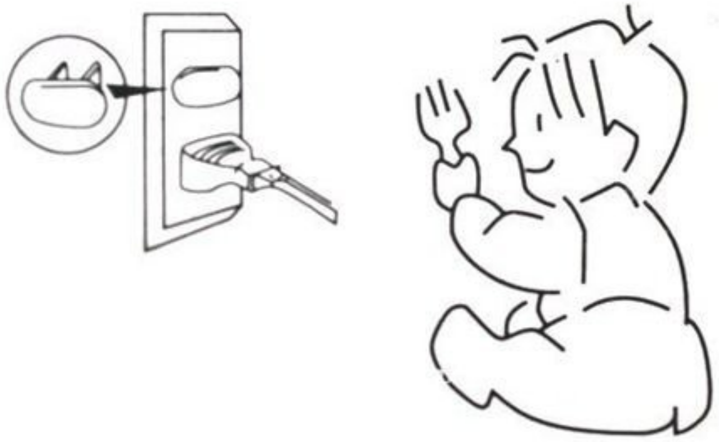
“Security”?

Not like this

- Fend off malicious hackers, evildoers, long-haired troublemakers?

More like this

- Prevent casual users from making mistakes!
- Help operators follow procedures!



http://asset3.cbsistatic.com/cnwk.1d/i/ne/p/2008/FortKnox_550x338.jpg
http://i01.i.aliimg.com/img/pb/216/247/491/491247216_377.jpg

Idea

Control reading and/or writing via Channel Access

- Almost never used to limit reading

Criteria:

- **Who?**
  - Control system engineer may always access everything
  - Beam Line Staff may always access most things
  - Beam Line Users cannot write certain things
- **From Where?**
  - Full access from Beam Line Control Room
  - No write access from anywhere else
- **When**
  - Read-only while experiment is running, while automation is enabled, …
  - Writable when experiment idle, manual control enabled, …

Limitations

… Via Channel Access

- Nothing is encrypted
- IOC console (dbpf, …) not affected

Who?

- $USER

From Where?

- Host name, easy to fake

Records...

- Assigned to Access Security Group
  - `field(ASG, "LIMITED")`
  - Default is “DEFAULT”
- Fields have Acc. Sec. Level
  - Most in ASL1
  - Some are ASL0
  - Nobody can remember. See *.dbd

Configuration

- Doing nothing is equivalent to this:
  - Create file ”simple.acf":
    ```
    ASG(DEFAULT)
    {
            RULE(1, READ)
            RULE(1, WRITE)
    }
    ```
  - Add this line to your st.cmd:
    ```
    asSetFilename("path_to_the_file/simple.acf")
    ```
- Result:
  - ✓ By default, records use the "DEFAULT" ASG.
  - ✓ … which allows full read/write.
  - ✓ The 'asprules' and 'asdbdump' commands now show something
- 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)
  # Nothing in here about WRITE…
}
```

- **To have an effect, set the ASG field of at least one record to READONLY.**
  - You 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**
- **Display tools (edm, CSS BOY, ..) will indicate read-only access via cursor or 'disabled' widgets**

List Specific Users and Hosts

- 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

- 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.

SNS Beamline Example

- DEFAULT
  - Anybody can read
  - Special list of experts can always write
  - Normal users cannot write in certain modes
- ALWAYS
  - Anybody can always read and write
  - Use for “STOP”, “ABORT” type PVs
- EXPERT
  - Anybody can read
  - Only special list of experts can write

Better “Security”

- Place IOCs in private network
  - No ‘telnet’ to their console
  - No Channel Access from malicious clients
  - Outside access (ssh, NXClient, …) controlled the usual way
- Add Channel Access Gateway to other networks
  - Gateway also has access security
  - Make it read-only

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