

# Channel Access / PV Access Security

Kay Kasemir

July 2026

Material copied from IOC Application Developers Guide  
by Marty Kraimer, Janet Anderson, Andrew Johnson (APS)  
and others

ORNL is managed by UT-Battelle, LLC for the US Department of Energy



U.S. DEPARTMENT OF  
**ENERGY**

# “Security”?

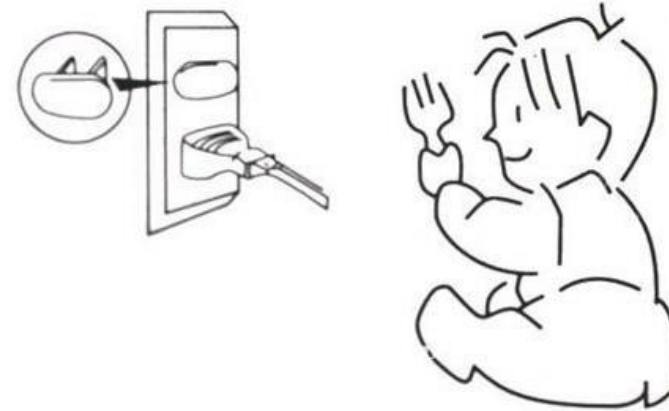
Not like this

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



More like this

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



*With PV Access, encryption and authentication are under development*

# Function and Scope

Control reading and/or **writing** of EPICS records via Channel/PV Access

- Almost never used to limit reading

## Criteria:

- Who, which user?
  - Control system engineer may always access everything
  - Beam Line Staff may always access most things
  - Beam Line Users cannot write certain things
- From where, which machine?
  - Full access from Beam Line Control Room OPIs
  - No write access from anywhere else
- When, in which system state?
  - Read-only while experiment is running, while automation is enabled, ...
  - Writable when experiment idle, manual control enabled, ...

# Limitations

## ... Via Channel Access

- Nothing is encrypted

## Who?

- \$USER

## From Where?

- Host name, easy to fake

## ... Via PV Access

- Encryption is available

## Who?

- "Principal" identified by certificate

IOC console (*dbpf*, ...) not affected

# Specification Summary

- A record belongs to one access security group (ASG)
- Security config file defines ASG:
  - Multiple rules (read or write)
    - Groups of users (which user)
    - Groups of hostnames (which machine)
    - Optionally qualified by the value of PVs (which state)
  - Rules give statements like:
    - Operators may write any property of PVs in this group from any OPI in the control room in any system state
    - Maintenance personnel may write values of PVs in this group from any maintenance OPI when the system state is *maintenance*

# EPICS DB

- Record
  - Assigned to access security group
  - `field(ASG, "LIMITED")`
  - Default ASG is “*DEFAULT*”
- Fields have *Access Security Level* property
  - Most in ASL1
  - Some are ASL0
  - Nobody can remember. See \*.dbd

# Access Security File

```
UAG(<name>) { <user> [, <user> ...] }
```

```
...
```

```
HAG(<name>) { <host> [, <host> ...] }
```

```
...
```

```
ASG(<name>)
```

```
{
```

```
    [INP<index>(<pvname>) ...]
```

```
    RULE(<level>,NONE |READ|WRITE [,NOTRAPWRITE | TRAPWRITE])
```

```
    {
```

```
        [UAG(<name> [,<name> ...])]
```

```
        [HAG(<name> [,<name> ...])]
```

```
        CALC(<calculation>)
```

```
    }
```

```
    ...
```

```
}
```

```
...
```

# RULE(<level>, <what>, [<trap option>])

- <level> is 0 or 1.
  - The dbd file assigns each field an access security level. Level 1 fields are typically related to record behavior and configuration. Level 0 fields are related to value.
    - Example: For the AI record, VAL is level 0, all the rest are 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 may be added to the IOC. This can be used to log write access by user and host, it doesn't otherwise affect access security.

# Default Implicit Behavior

- If no access security file is loaded, all users from anywhere may read and write all fields of all records anytime

# Explicit Configuration Equivalent to Default

- Create file *simple.acf* with the following content:

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

- Add the following line to your *st.cmd*:  
`asSetFilename("path_to_the_file/simple.acf")`

## Result:

- ✓ Since, by default, records belong to the ASG named *DEFAULT*  
full *read/write* to all records is allowed
- ✓ Functionally equivalent to doing nothing
- ✓ But now, the *asrules* and *asdbdump* commands show something
- Caveat:
  - If the AS config file does not exist or contains a syntax error,  
*all access is prohibited!*
  - Use the *ascheck* utility on the host before loading a file into the IOC

# Read-Only

- Group that allows read, but no write:

```
ASG (READONLY)
{
  RULE (1, READ)
  # Nothing in here about WRITE...
}
```

- To have any 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*, *CS-Studio*, ..) will indicate read-only access via cursor or disabled widgets

# Limit Write to 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) { epics-dev }
HAG(x_hosts) { some_host_name }
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 be *myhost* or *myhost.some.site*, might differ from 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

# Changes for PV Access

By default, the “HAG(...)” entries must be *numeric* IP addresses, not *host names*, to avoid confusion about plain and fully qualified host names

UAG(“epics-dev”, “**role/operators**”) will match

- User with name “epics-dev”
- Any user who is in the “operators” **group**

Based on Linux user and group info, see “id”

```
$ id
uid=1000(epics-dev) groups=1000(epics-dev),10(wheel)
```

# Limit Access by System State

- Limit write access to times where some set of variables meets some criteria
  - ASG (MODE)

```
{
  INPA (Accelerator:Mode)  # "Accelerator:Mode" is normal PV
  RULE (1, READ)
  RULE (1, WRITE)
  {
    CALC (A < 5)
  }
}
```
- This is based on the same code as the *CALC* record
  - PVs may be assigned to inputs *A* through *L*
  - The computation should result in 0 or 1, the latter allowing access

# Two Basic Approches

## Restrict

- DEFAULT is all writable
  - As would be the case without doing anything
- Other groups then limit access to certain users and hosts

## Open up

- DEFAULT is mostly read-only
- Other groups then open access to certain users and hosts

→ Safer and  
may be easier to maintain

# SNS Beamline Example

- **DEFAULT**
  - Anybody can read
  - 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

# Additional Security Measures

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