ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-WESH2002

EPICS pva ACCESS CONTROL AT ESS

G. Weiss, ESS, Lund, Sweden

Abstract

At ESS (European Spallation Source), pva (PV Access) has been selected as the default EPICS protocol. However, initial releases of EPICS version 7 do not implement any access control of client requests in pva. In order to control write operations that may cause harm to the system, access control is highly desirable. This paper details how the ca (Channel Access) access security concept is reused and extended for pva access control. It also explains how ESS intends to deploy and manage access control in terms of infrastructure, tools and responsibilities. Limitations of the access control mechanism are also discussed.

SCOPE

The scope of pva access control is to provide means to restrict write operations on selected/critical PVs to a predefined group of users, sending requests from a predefined list of hosts. Access control of read requests is **not** in scope.

ca ACCESS SECURITY RECAP

Before going into the details of pva access control, it may be of value to recap the features of the ca access control.

When a ca client sends a read or write request on a PV, it may optionally send a client and host identity. These identities are arbitrary strings set by the client. In practice however, the client identity is set to the user name of the process running the client software, and the host identity is set the host name running the client software. This is the case with the EPICS ca command line clients *caget* and *caput*.

On the receiving side, the IOC server may use the client and host identity to grant or deny access to a PV. The rules governing the access control are defined in ACFs (Access Control File) read by the IOC on boot, or when an explicit ACF reload is requested. If the IOC does not load an ACF on boot, all requests are allowed from all users and hosts.

Rules defined in the ACF are based on matching identities to the client and host identities sent by the clients. As an example, consider the following ACF:

```
UAG(uag) {user1, user2}
HAG(hag) {host1, host2}
ASG(DEFAULT) {
    RULE(1, READ)
    RULE(1, WRITE) {
        UAG(uag)
        HAG(hag)
    }
}
```

Here the user access group (UAG) named **uag** lists client (user) identities **user1** and **user2**, and the host access group (HAG) named **hag** lists hosts **host1** and **host2**. The DEFAULT rule – which applies if no other rule is defined – states that all clients and hosts are granted read access, while write access is granted only for **user1** and **user2** on **host1** or **host2**.

By default, all database records in the IOC are associated with the DEFAULT ASG rule, but individual records may use the ASG field to point to some other rule in the ACF.

A detailed description of ca access control is found in reference [1].

pva ACCESS CONTROL

Starting from EPICS version 7.x (TBD), IOCs and PV Access gateways support access control along the same lines as ca access control. There are however some important differences to consider:

- Only write requests (e.g. through pvput) are subject to access control. All read requests are granted to all user identities on all hosts, irrespective of explicit READ rules in the ACF.
- UAG rules are more flexible as definitions may list user groups in addition to user names.
- HAG definitions may list either IP addresses or host names, or both.

To expand on the previous example ACF example, consider the following content of an ACF:

```
UAG(uag) {role/group1, role/group2, user1}
HAG(hag) {10.20.30.40, host2}
ASG(DEFAULT) {
    RULE(1, WRITE) {
        UAG(uag)
        HAG(hag)
    }
}
```

User groups are identified using the syntax **role/<group>**, where the prefix **role/** is fixed.

For the IOC to grant write access using the above ACF, the user identity provided in the client request must be member of either **group1** or **group2**, or the user identity may be **user1**.

The list of groups of a user identity is determined by the operating system of the IOC's host. In many cases such a list would be a union of local groups and groups managed by some authentication service like LDAP or Active Directory. The pva access control implementation does **not** directly query external services for group information, it only depends on data provided by the host operating system via a system call.

The above ACF also limits write requests to clients on a host with IP address **10.20.30.40** and a host named **host2**. In contrast to ca access control, HAG rules are based on the IP address associated with the incoming write request. Host name lookup is performed when the ACF is parsed. When

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-WESH2002

the above HAG rule is applied, a call from a client on **host2** must match the IP address retrieved by this lookup.

ACF BACKWARDS COMPATIBILITY

An ACF prepared for ca access control is valid for pva access control as long as the host name lookup successfully matches the clients IP address to the host name entries in the HAG rules.

Conversely, an ACF prepared for pva access control is valid as long as the ca client specifies the user and host identities to match UAG and HAG definitions. However, a user name like **role/foobar** is of course invalid on most operating systems.

SECURITY CONSIDERATIONS

The ca access control mechanism cannot be regarded as anything else but a very basic form of access control. A client application may freely set the user and host identity to any value listed in UAG and HAG definitions in the ACF and thus acquire access to PVs subject to access control. In order to send request that is granted access by the IOC, the user in control of the client need only be familiar with the contents of the ACF.

In pva access control a client may still spoof the user identity or specify any user name belonging to a user group listed in the UAG rules of the ACF. However, since pva access control uses the IP address associated with the client request, a client sending a request from an unauthorized host would need to use IP address spoofing in order to be approved by a HAG rule.

Using host names in HAG rules for pva acces control does improve security as the IP address will be matched against the result of a DNS query.

Given the limited protection offered by pva access control, ESS will use it solely to reduce the risk of PV modifications that ultimately may cause harm to the system or personnel.

PERFORMANCE

The list of user groups for a user identity is provided by the operating system via a system call (e.g. getgroups() on Linux). If the host running the IOC server is configured to use an external authentication service like Open LDAP or Active directory (e.g. through the sssd daemon on Linux), the system call in question incurs additional execution time, typically a few milliseconds. However, the additional execution time is not a factor for each client request if the underlying authentication service is configured to use a cache.

Further, the hostname lookup performed by the IOC server when the ACF is parsed will also incur some overhead. This overhead may be non-negligible if the DNS service is unavailable, and it will then prolong the IOC boot time by timeout period of the host's DNS query. DNS redundancy on the network is therefore advocated.

ACF MANAGEMENT AT ESS

Since ACFs are regular files that need be available on the file system for IOCs at boot time, management and deployment can easily become cumbersome in a complex environment composed of a large number of sub-systems and IOCs. After all, the ACF contains data from multiple sources; information from the network environment (HAG rules) is combined with information from the user authentication services (UAG rules). PV identities must also be known when composing ACFs restricting access on PV level.

To start with, ACFs at ESS will be created manually and deployed in a limited scope. However, the ESS eco system does provide sources of information that may be used to facilitate and automate management of ACFs. This requires careful analysis as a ACF management tool would need to integrate services and tools that are still under development. High-level requirements for an ACF management tool would roughly be:

- Integration with authentication service, which is Open LDAP at ESS.
- Integration with IOC information services. This is under development at ESS.
- Integration with PV identities service, i.e. Channel Finder
- ACF syntax check.
- Easy deployment of ACF.
- Control of ACF management permissions.

A possible enhance for pva access control is the ability to deploy the ACF contents as a PV that may be monitored by the IOCs that wish use access control. This would of course facilitate deployment compared to the current implementation where an updated ACF must be copied to the correct location and then re-read by affected IOCs.

DELEGATION OF ACF MANAGEMENT

As stated above, pva access control will be used at ESS to restrict write access of a selected set of critical PVs to a limited number of users. Responsibility to identify such PVs – and the users that may modify them – will be delegated to the organizational units that possess the required knowledge. These units will also assume responsibility to manage ACFs, thus offloading software development and system administration personnel.

ACKNOWLEDGEMENTS

ESS wishes to acknowledge that pva access control was realized by Michael Davidsaver of Osprey DCS [2].

REFERENCES

[1] Chapter 6: Access Security, https://epics.anl.gov/EpicsDocumentation/AppDevManuals/AppDevGuide/3.13BookFiles/accessSecurity.html

[2] Osprey Distributed Control Systems, https://ospreydcs.com