Expanding CIP Security™ with the CIP Authorization Profile

Joakim Wiberg
Head of Technology
HMS Networks

David Smith
Cybersecurity Architect
Schneider Electric

Jack Visoky
Principal Engineer and Security Architect
Rockwell Automation

Presented at the ODVA
2022 Industry Conference & 21st Annual Meeting
March 9, 2022
San Diego, California, USA

Abstract

Cyber security within Industrial Ethernet has exhibited rapid growth, with CIP Security and EtherNet/IP™ emerging as a leader. End users seek to take advantage of the features provided by the CIP Security Profiles today and related open ecosystem. Benefits include data integrity and data confidentiality, device identity and authentication, and user authentication. These features are provided by Security Profiles as defined today and serve as a base for CIP Security devices. Over time CIP Security has been extended with new optional Security Profiles targeting different applications and functionality.

Within this paper the idea of a new optional profile named “CIP Authorization Profile” is explored and evaluated. The CIP Security Authorization Profile will enhance CIP to provide additional security properties such as general, flexible authorization where access policy can be based on any attribute of the user and/or system. Concepts and open systems that might serve as a base for the CIP Authorization Profile are explored.

This paper will provide advanced insights regarding technology and requirements for the CIP Authorization Profile that will eventually be added to CIP Security. As the CIP Authorization Profile is officially developed within ODVA it may deviate from the scenarios described in this paper. However, the general application of the CIP Authorization Profile can be understood from this paper.

Keywords

CIP Security, Cybersecurity, CIP Authorization Profile, Authentication, Authorization, Role-Based Access Control (RBAC), Attribute-Based Access Control (ABAC), Open Policy Agent (OPA), Rego, XACML, General Data Modeling Language

Introduction

Volume 8 of the CIP Specification currently defines CIP Security via several profiles. These profiles provide security features like secure transport, user authentication, automatic certificate enrollment, and others as well. However, there is currently no standard profile that defines features around configurable authorization. That is, there is no standard way to define what privilege levels are required to access what resources. Defining this functionality in a standard way gives the user a powerful mechanism to define authorization in a highly flexible and configurable way. However, providing this functionality is not a simple task, there are many ways in which this could be done technically, each with their own tradeoffs. Before defining a profile for authorization and access control policy an investigation must be done regarding technology options, as well as tradeoffs for the user. It is important to balance flexibility against complexity and to define this profile in such a way that it is user-friendly yet powerful enough to provide options to the user. This paper provides a starting point for this investigation to aid in the definition of this profile.

CIP Security and CIP Security Profiles overview

CIP Security is defined in Volume 8 of the CIP Networks Specification and includes the definition of security-related requirements and capabilities for CIP devices. Volume 8 at present is focused on EtherNet/IP, as EtherNet/IP-connected devices represent the largest risk due to enterprise network connectivity and provides a secure transport mechanism for EtherNet/IP devices.

CIP Security for EtherNet/IP devices makes use of the IETF-standard TLS (RFC 5246) and DTLS (RFC 6347) protocols in order to provide a secure transport for EtherNet/IP traffic. TLS is used for the TCP-based communications (including encapsulation layer, UCMM, transport class 3), and DTLS for the UDP-based transport class 0/1 communications. This approach is analogous to the way that HTTP uses TLS for HTTPS.

The secure EtherNet/IP transport provides the following security attributes:

- Authentication of the endpoints – ensuring that the target and originator are both trusted entities. Endpoint authentication is accomplished using X.509 certificates or pre-shared keys.

- Message integrity and authentication – ensuring that the message was sent by the trusted endpoint and was not modified in transit. Message integrity and authentication is accomplished via TLS message authentication code (HMAC).

- Message encryption – optional capability to encrypt the communications, provided by the encryption algorithm that is negotiated via the TLS handshake.

Figure 1 shows the protocol layering:

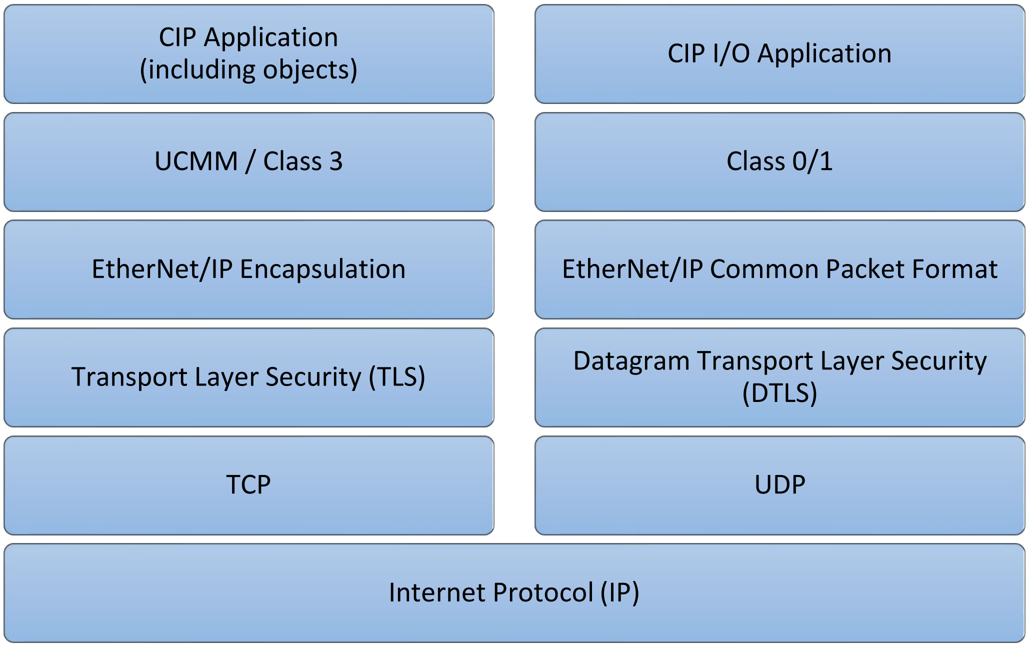
CIP Application
(including objects)
CIP I/O Application
UCMM / Class 3
Class 0/1
EtherNet/IP Encapsulation
EtherNet/IP Common Packet Format
Transport Layer Security (TLS)
Datagram Transport Layer Security
(DTLS)
TCP
UDP
Internet Protocol (IP)

Figure 1 EtherNet/IP over TLS and DTLS layering

The following example illustrates how the secure EtherNet/IP transport would mitigate a security threat.

Consider a simple end-user application that consists of an EtherNet/IP-connected programmable controller (PLC) and several EtherNet/IP-connected I/O devices. At initial configuration time, the user configures the PLC and each I/O module with a pre-shared key (PSK) and disables the non-secure EtherNet/IP TCP and UDP ports. Subsequent EtherNet/IP communications take place over TLS and DTLS, and require that each endpoint possess the PSK that has been configured.

Assume further that an employee has unknowingly downloaded malware that sends programming commands to the PLC's IP address via EtherNet/IP. If the malware attempts to connect to the PLC without using TLS, the PLC will not accept the connection. If the malware attempts to connect via TLS, but doesn't know the PSK, the TLS connection will not be established. In either case, the malicious programming commands will not be sent to the PLC.

Recognizing that every CIP device does not need to provide the same level of support for all defined security features, CIP Security defines the notion of a Security Profile see Table 1. A Security Profile is a set of well-defined capabilities to facilitate device interoperability and end-user selection of devices with the appropriate security capability. At present, three profiles are defined for EtherNet/IP devices, and two potential future profiles are identified for CIP-level security capability.

Security Profile	General Description
EtherNet/IP Integrity Profile (Obsoleted)	Provides secure communications between EtherNet/IP endpoints to assure data integrity and device authenticity.
EtherNet/IP Confidentially Profile	Provides secure communications between EtherNet/IP endpoints and ensures data confidentiality for transport class 0/1 traffic. Includes the EtherNet/IP Integrity profile as a subset.
CIP Authorization Profile (future)	Provides secure communications between CIP endpoints to ensure device and user authenticity.

CIP User Authentication Profile	Provides User-level authentication for CIP communication
Resource-Constrained CIP Security Profile	Provides a lightweight version of the protections afforded by other CIP Security Profiles specifically for highly Resource-Constrained devices

Table 1 CIP Security Profiles

Table 2 shows the security properties provided by each of the profiles:

Security Property	EtherNet/IP Confidentially Profile	CIP Authorization Profile (future)	CIP User Authentication Profile	Resource-Constrained CIP Security Profile
Device Authentication	√	√		√
Trust Domain	Broad – group of devices		Narrow - individual device/user	Broad – option to be Narrow via Gateway or Proxy
Device Identity	√		√ (Identity of User)	√
Data Integrity	√			√
Data Confidentiality	√			**Via Gateway or Proxy**
User Authentication			√	
Change Detection (Audit)		√		
Policy Enforcement (Authorization)		Flexible	Fixed	Via Gateway or Proxy

Table 2 Supported Security Properties

Authentication vs. Authorization

Access control is a security term that is used to reference a set of policies for restricting access, in broader terms this can be access to tools, functionality, or physical location. Though, in this paper, access control relates to restricting access to information, such as data and the software used to manipulate the data.

Software is used to access and grant authorization to users and devices that need to access the digital information. Authentication and authorization are integral components of digital information access control. Although the two terms might sound similar, they are distinct security concepts in the world of identity and access control management.

Authentication is the security practice to confirm that the user is who they claim to be. Authentication is the first step in security on logging in and gaining access to digital information. The process of authenticating a user could be accomplished by different means such as passwords, authentication apps, or biometrics. Any of these mechanisms could be used to prove that that the user is who they claim to be, thereby authenticating the user.

Once authenticated, a user can see the information that they are authorized to see and access information that they are authorized to access. Authorization in system security is the process of giving the user permission to access a specific resource or function. This could be granting access to folders on a server or starting certain applications.

Authorization strategies

After a user has been authenticated the user authorization can be determined in different ways, often referred to as authorization strategies. Over time many different authorization strategies have been developed for different purposes and use cases. Some common authorization strategies are Role-Based Access Control (RBAC), Attribute-Based Access Control (ABAC), Graph-Based Access Control (GBAC), and Discretionary Access Control (DAC). Within this paper the two most prominent ones, Role-Based Access Control (RBAC) and Attribute-Based Access Control (ABAC) will be discussed and further one of them will be used as an example of how to realize a CIP Authorization Profile.

Role-Based Access Control (RBAC)

RBAC treats authorization based on permissions associated with roles and not directly with the user. Users are assigned to roles bases on the permissions they are supposed to have. Often times a user will have multiple roles. A role can be seen as a collection of permissions. The “Principle of Least Privilege” states that a user should be assigned the role of least access necessary for the job to be done. This is an important principle when implementing RBAC or any other access control mechanism.

As an example, an “administrator” for a plant would have permissions to reflect this role. In this administrator role the user would be able to change any or most of the configuration, update firmware and read any data, simply put the administrator role would almost have no restrictions. On the other hand, a user possessing the “view” role would be limited to just view data.

The advantage of using RBAC is that managing authorization privileges becomes easier because system managers can deal with users and permissions in bulk instead of having to deal with them one by one.

The existing CIP Security User Authentication Profile provides for user authentication and assigning roles to specific users. The specification includes a minimum list of roles which all compliant products must support, with the option to add additional roles as needed for a user. This profile allows for user decision in terms of what users/devices to assign to a given role, as well as what authentication mechanism to use to authenticate the users/devices. However, the profile does not define an interoperable way to assign what a given role can do within a device; that is the role of the CIP Authorization Profile to be defined in the future (and the subject of this paper’s investigation).

Attribute-Based Access Control (ABAC)

ABAC provides access based on who the user is rather than what they do, for example in what organization they work and how they were hired. These attributes allow for easier control structures since permissions can be based on the user’s department, location and so on. Utilizing attributes from a user, information that already exist in the HR system, permits for a rich and flexible control structure.

Using the example above from RBAC, if a user is promoted from working on the floor to become a control engineer then the person would automatically gain administrator rights when HR updates the attributes in their system.

Requirements for the CIP Authorization Profile

A few high-level requirements are defined for the CIP Authorization Profile:

- CIP Endpoints must be policy enforcement points – The authorization policy will be enforced within CIP endpoints therefore the mechanism for communicating that policy must be well-suited to an embedded device. In other words, the technology cannot be prohibitively difficult to implement in an embedded device or rely on technologies rarely present within an embedded device.

- Reuse existing technologies if possible – CIP Security always maintains a preference for using existing security technologies rather than inventing new ones. This is not always possible due to the unique needs of industrial protocols, but if an existing technology can be used then it should be leveraged for this profile. Note that a highly ubiquitous authorization policy technology doesn't exist within the IT world.
- Support both RBAC and ABAC – Some users will prefer the simplicity of RBAC for their system, whereas others will require the flexibility of ABAC. The Authorization Profile should support both of these, although it is possible that some of the more complex features might be optional.
- Provide options for both simple access policy and advanced access policy – Similar to the RBAC and ABAC requirement the access policy should support simple declarative statements such as "resource x can only be accessed by an administrator" as well as more complex logical statements such as "resource x can be accessed by [administrators OR [engineers AND time of day == 8:00 – 15:00]]. Again, more complex logical statements may be optional.
- Integrate with IT systems – As much as possible the technology chosen should be integrated into existing IT systems. This is related to the idea of re-using existing technology if possible.

Existing authorization systems

There are many existing authorization schemes that allow a user to configure how authorization will be done in a system. However, it is worth noting that although some of these schemes have gained traction in particular industries or applications, none is deployed ubiquitously like TLS is for communication security. That is, although the CIP Authorization Profile could use one or more of these authorization technologies, there is no single technology that stands out as a clear market leader. The following section provides a list of options as well as a brief discussion of the technology. Note that this list is not at all exhaustive, many options are not discussed here. This list is meant to be illustrative of some of the more popular options available.

OPA (Open Policy Agent) and Rego

OPA is a policy engine that allows a user to define an access policy via Rego, the open language for defining access policy. OPA and Rego have proven successful in cloud-based software environments and with various compute functions "as a service" (referred to as XaaS) gaining traction it has grown in popularity and usage. Rego supports a wide variety of access control policies and is not limited to simple RBAC, although RBAC could certainly be supported. Rego uses expressions written against input to determine access criteria. It provides a lot of flexibility and power for writing complex expressions, although given this it may be overly-complex for the needs of a CIP device.

Although these two pieces are meant to work together, Rego could be used by itself in CIP endpoints or could be used with OPA running within a CIP endpoint. OPA would likely be an optional component as CIP conformance test would not be mandating a given implementation of Access Control, although OPA might be useful for highly resourced environments. Given that OPA usually runs in enterprise/cloud environments, it may not be suitable for the many embedded devices which implement CIP and EtherNet/IP. Despite this, Rego may still provide a useful technology for configuring access policy within a CIP device.

XACML (eXtensible Access Control Markup Language)

XACML is an XML based language for describing access policy. XACLM has been in use for more than 20 years and as such has a lot of runtime and support within commercial and open source libraries. XACML defines Rules as part of Policies and Policies as part of PolicySets. XACML was mainly designed with ABAC in mind, although can certainly also be used with RBAC. Rules contain conditions that are evaluated for a given access request, the result of the evaluation determines whether or not the access is permitted. Like OPA and Rego, XACML was not designed primarily for embedded systems, although it is likely a bit better structure for this use than Rego and OPA.

General Data Modeling Language
It is also possible to make use of a general data modeling/data encoding language to define a custom access policy format. For example, YANG (Yet Another Next Generation) could be used to define an access policy language. Encoding could be done in JSON or XML, or even a new format. This would of course require a significant design effort to define rules and structure of the access policy. However, it would allow for something far more custom and better suited to CIP devices, although at the expense of extra effort both in terms of design and implementation within a device. However, this is an option that should be seriously considered.

Mapping to CIP

Every CIP node is modeled as a collection of objects. An object provides an abstract representation of a particular component within a product. Anything not described in object form is not visible through CIP. CIP objects are structured into classes, instances and attributes.

A class is a set of objects that all represent the same kind of system component. An object instance is the actual representation of a particular object within a class. Each instance of a class has the same attributes, but also has its own particular set of attribute values. As Figure 2 illustrates, multiple object instances within a particular class can reside within a CIP node.

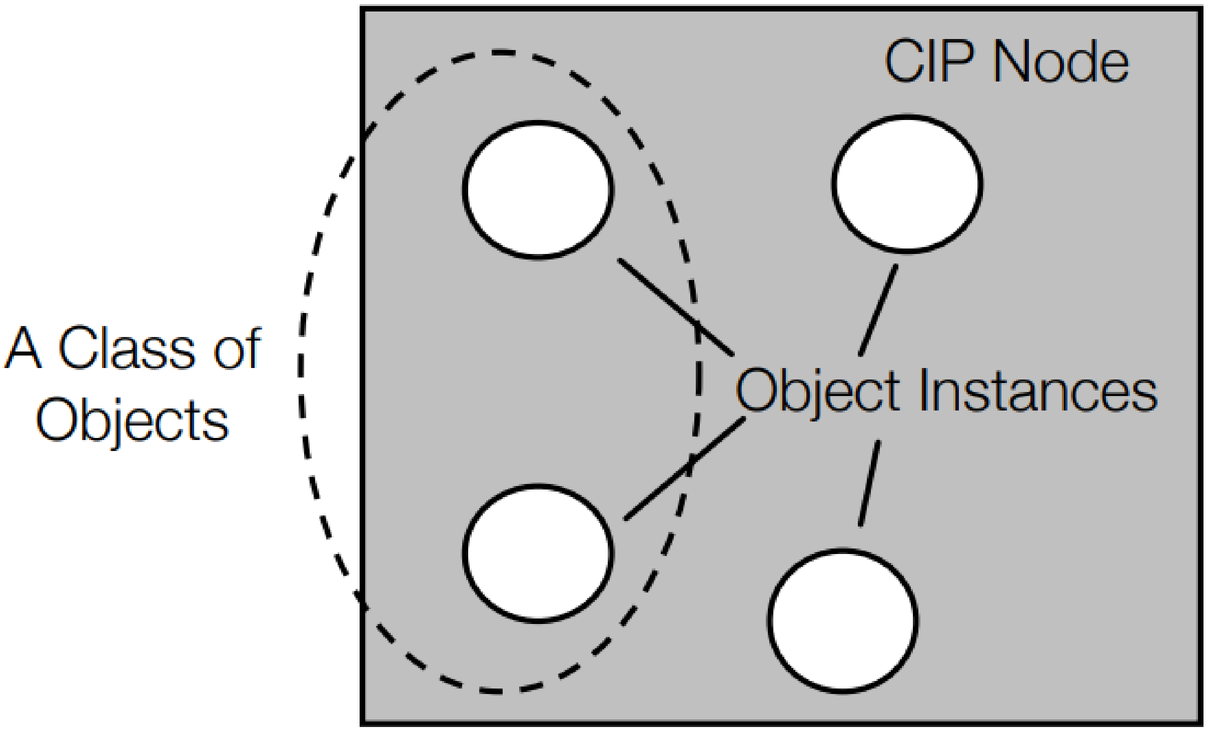

Figure 2 CIP Classes and Instances

The CIP family of protocols contains a large collection of commonly defined objects. Objects defined in the CIP Networks Library or may be Vendor specific and only used by one vendor. The objects group attributes, generally the data being represented.

In order to access the data and perform other actions with the data or behavior provided by the object, CIP Services are invoked. These CIP services are common in nature, meaning they may be used in all CIP Networks and they are useful for a variety of objects. Furthermore, there are object-specific service codes that may have a different meaning for the same code, depending on the class of object. Finally, defining vendor-specific services according to the requirements of the product developer is possible.

As a part of the CIP Security User Authentication Profile the idea of a RBAC scheme with six well-defined roles has been introduced. Volume 8 states that this scheme is defined as the minimum set of functionality to allow for interoperability between Originators and Targets. Furthermore, Volume 8 discusses the fact that there is no standardized way to make granular changes to access control policy stating that this is a future concept of the CIP Security specification. The six roles that have been defined are:

- Administrator
- Engineer
- Operator
- Auditor
- Viewer
- Anonymous

Each role has a general baseline description related to access policy. Some guidance is provided on what access levels each role enables, although specific access control policy is up to the product's vendor. As an example, Volume 8 states the following regarding the Administrator and Operator roles.

The Administrator role allows for any and all access to the product. That is, once an administrator has been properly authenticated, this role may access any protected resources on a Target. Specifically, the Administrator role is the only role that can configure User Authentication once the Target has been provisioned for User Authentication. Note that although the Administrator should have access to any resources, this does not prevent a product from limiting access due to overriding conditions such as functional safety. For example, a device can prevent changes that could put the device into an unsafe state.

The Operator role is meant to work with runtime access to equipment. An Operator role should be able to perform troubleshooting, set and interact with I/O, monitor operations, and perform limited configuration that might be necessary for device replacement. Therefore, the operator should be permitted to create or reconfigure I/O connections, although would not be permitted to perform significant configuration, such as downloading a program to a PLC or changing the screens available in an HMI.

It's suggested in Volume 8 that it is not feasible to enumerate what CIP objects, attributes, and services each role may access, although in cases where prescriptive guidance is warranted it is explicitly provided within the CIP Networks Library of Specifications. In this paper the concept of a flexible way to provide fine-grained access control to attributes (data) and permissions (services) is introduced. Two mechanisms for this are discussed, which may be implemented independently or combined for a joint mechanism for managing access control. The first of these is a CIP service-based mechanism, and the second is a document-based mechanism.

CIP Service-Based Access Policy Management

For the purpose of introducing the fine-grained access through CIP Services, a fictitious CIP object with just three instance attributes is used. The object's instance attributes are laid out as in Table 3, using the same notation as in the CIP Networks Library.

Attr ID	Need in Implem	Access Rule	NV	Name	Data Type	Description of Attribute
1	Optional	Get		AtReference	BOOL	0 = Drive hasn't reached SpeedRef 1 = Drive has reached SpeedRef
2	Required	Get		SpeedActual	INT	Actual speed
3	Required	Set		SpeedRef	INT	Commanded reference speed

Table 3 Example Instance Attributes

As already mentioned above and in Volume 8 it wouldn't be feasible to enumerate what CIP objects, attributes, and services each role may access, at least not in one new CIP object used for authorization management. The reason for this is that a CIP device contains multiple CIP objects and each CIP object can have many attributes. For some devices this can sum up to several hundred or even thousands of attributes. Having this in one singe authorization management object would likely end up being cumbersome and unmanageable.

An alternative way to look at this would be to allow each attribute to also carry permission information. This permission information would be additional information that would go together with the attribute data within the module. The permissions would carry the roles that would be allowed to get or set the attribute.

In order to interface with and manage permissions the already established notion of services would be used. In this case two new general services would be defined named, View_Permission_Information and Apply_Permission_Information. The former would be used to get the current permission information and the latter to apply new permission information. Both services would use a list of service parameters, View_Permission_Information would return the list of parameters and Apply_Permission_Information would receive the list of parameters. Table 4 shows the list of parameters.

Name	Data Type	Description of Parameter
Engineer access	ENUM	Access rights to Engineer
Operator access	ENUM	Access rights to Operator
Auditor access	ENUM	Access rights to Auditor
Viewer access	ENUM	Access rights to Viewer
Anonymous access	ENUM	Access rights to Anonymous

Table 4 Apply_Permission_Information parameters

Basically, this is just a list of roles and what access rights should be applied. Note that Administrator has been omitted since this role should have full access to all data in a CIP device, and it helps avoid potential risk of locking out the Administrator thus bricking the CIP device.

The ENUM data type would carry values indicating what kind of access that the specific role should have. I.e. no access at all, read the attribute, or read and write to the attribute as shown in Table 5.

Access	Value
No access	0
Read access	1
Write access	2

Table 5 Access rights

Summing this up using the example from Table 3 the attribute table with the permission information added would look something like Table 6. Here the instance has been configured to prevent anonymous users to have no access to any attribute. The viewer can only access the SpeedActual attribute and only read it. The operator and auditor can read all three attributes, but not modify SpeedRef which is the only settable attribute. The engineer has read access to all attributes and access to modify the SpeedRef attribute.

Attr ID	Need in Imp	Access Rule	NV	Name	Data Type	Permission information		Description of Attribute
1	Optional	Get		AtReference	BOOL	Engineer access	Read	0 = Drive hasn't reached SpeedRef 1 = Drive has reached SpeedRef
						Operator access	Read	
						Auditor access	Read	
						Viewer access	No	
						Anonymous access	No	
2	Required	Get		SpeedActual	INT	Engineer access	Read	Actual speed
						Operator access	Read	
						Auditor access	Read	
						Viewer access	Read	
						Anonymous access	No	
3	Required	Set		SpeedRef	INT	Engineer access	Write	Commanded reference speed
						Operator access	Read	
						Auditor access	Read	
						Viewer access	No	
						Anonymous access	No	

Table 6 Example Instance Attributes with permission information

With the potential number of attributes in a complex CIP device the database of permission information would be huge. One important thing to consider is that it needs to be reasonably easy to get an overview of all permission settings so an audit can be made making sure that the CIP device has been configured with the correct authorization settings. One way to solve this would be to provide an easy to use interface to gather all permission information from a CIP Device. There are several ways this could be done. Either to have a service that provides all permission information from the whole CIP device, this likely would return a large response that in the end would be a bit unmanageable. An alternative way would be a service, View_All_Permission_Information, that returns all permission information for one specific instance.

Document-Based Access Policy Management

Another potential mechanism for managing access policy is to use a more document-based structure. A document that enumerates access policy can be produced in a particular language and encoding, and then distributed to CIP endpoints that will then enforce that access policy. Documents can be signed and even have portions encrypted if necessary, and the signature provides authenticity assurances regardless of the transport mechanism (or in many cases in addition to the transport mechanism). Depending on the defined language, a document may have the ability to implement complex access policy rules that include logical operations (AND/OR/IF THEN) on various attributes and device states. Although the level of support for more complex policy statements may vary by device this would provide a powerful mechanism for defining highly flexible and customized access policy to protected CIP resources.

The mechanisms discussed in the “Existing Authorization Systems” section all use a document-based mechanism (Rego, XACML, and a YANG Model-based mechanism). This type of mechanism has the advantage of better integrating with many existing authorization definition technologies. However, downsides to this approach also exist. This approach would require devices to support a technology beyond CIP, which includes parsing of a given language and then translating that into policy enforcement on incoming CIP messages. These technologies may also provide more flexibility than is necessary for the average user of CIP Security, although that could also provide some opportunities for future expansion of the technology.

Conclusion

This paper has explored different approaches how authorization models could be applied to CIP Security. None of these are a “one-size-fits-all” solution, rather each offer advantages and disadvantages. These “models” could be implemented with using purpose-built CIP functionality or would require use of other external well-defined standards. As such, this paper provides guidance on some of the characteristics of each model described. The paper serves as a discussion expanding CIP Security with a CIP Authorization Profile.

References

[1] ODVA, Inc. The CIP Networks Library, Volume 8: CIP Security™, PUB00299
[2] OPA and REGO https://www.openpolicyagent.org/docs/latest/policy-language/
[3] XACML https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml
[4] YANG https://datatracker.ietf.org/doc/html/rfc7950
[5] NIST Access Control Website https://csrc.nist.gov/Projects/Role-Based-Access-Control
[6] RFC 5246 TLS 1.2 https://datatracker.ietf.org/doc/html/rfc5246
[7] RFC 6347 DTLS 1.2 https://datatracker.ietf.org/doc/html/rfc6347

****

The ideas, opinions, and recommendations expressed herein are intended to describe concepts of the author(s) for the possible use of ODVA technologies and do not reflect the ideas, opinions, and recommendation of ODVA per se. Because ODVA technologies may be applied in many diverse situations and in conjunction with products and systems from multiple vendors, the reader and those responsible for specifying ODVA networks must determine for themselves the suitability and the suitability of ideas, opinions, and recommendations expressed herein for intended use.