CIP Security Pull Policy

Jack Visoky, Joakim Wiberg, David Smith
Rockwell Automation, HMS Networks, Schneider Electric

Introduction

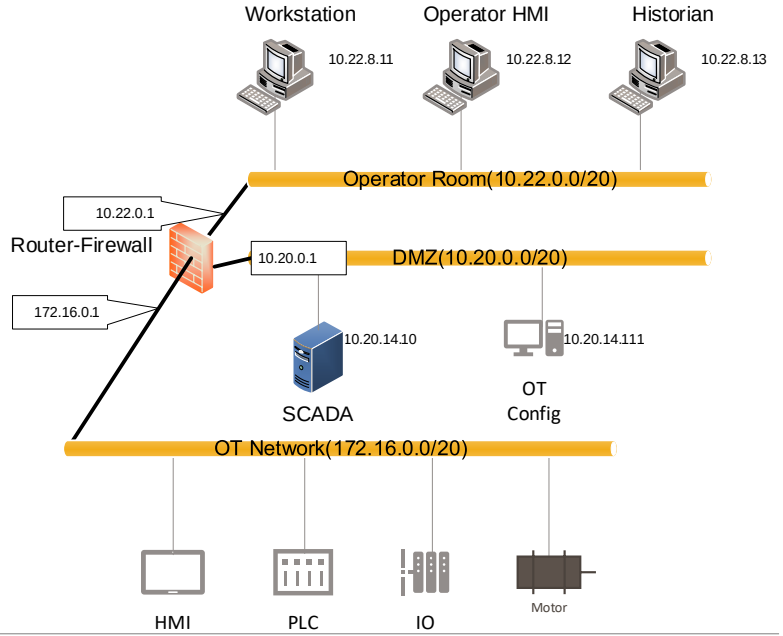
- We already have the CIP Security Pull Model, where a device is able to automatically request/receive a certificate
- There is still a lot of other configuration needed for CIP Security to work
  - Allowed Cipher Suites
  - Additional Trust Anchors (certificates)
  - Send full certificate chain or not
  - Etc.
- It would be ideal if a device could also automatically request/receive this information
  - And it was delivered in a secure manner!

© 2023 ODVA, Inc.
All rights reserved.

Use Cases – What is the Motivation

- Deliver CIP Security config to a device behind a NAT
  - Configuration tool on the public network cannot push config to the private network
- Client-only software
  - There is no CIP Target to push to!
- Automatic device replacement with full CIP Security deployment (not just certificate)
  - Everything is delivered automatically

Ethernet LAN Diagram



© 2023 ODVA, Inc.

All rights reserved.

Could this be used to deliver other CIP configuration?

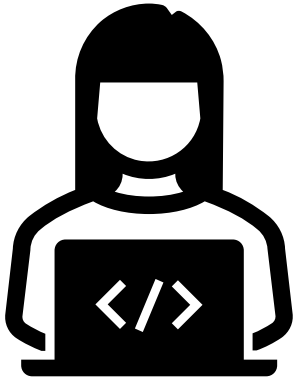
- Sure, but that is not our focus
- That said, we are defining a generic way to set CIP attributes and call services of various objects, which is essentially what CIP configuration is
  - At the same time other technologies (e.g. Safety, Motion) may present corner cases that need to be dealt with

>

© 2023 ODVA, Inc.
All rights reserved.

Requirements

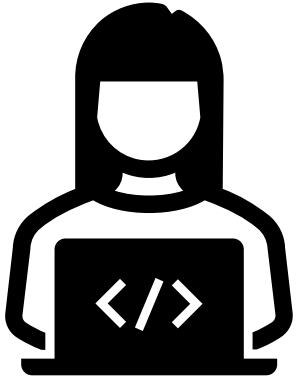
- **Document format –** we need to deliver the data as a document so that it is independent of transport
- **Authenticity –** fundamental requirement for verifying the config was not tampered and came from an authorized source
- **Confidentiality –** not nearly as important as authenticity but it could be useful, especially for PSKs or for an environment where there is a lot of concern regarding data confidentiality in general
- **Versioning –** important to ensure that consumers of the document are using the same version, also prevents downgrade/replay style attacks



© 2023 ODVA, Inc.
All rights reserved.

Requirements

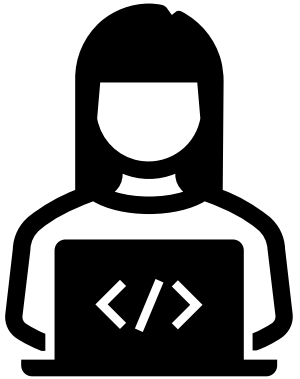
- **Automatic discovery –** existing Pull Model for certificates provides this, need to continue to support automatic discovery of the server
- **Configuring retry –** if the server is unavailable for any reason then the CIP device needs to be able to retry (could happen if too many nodes attempt communication at once)
- **Trigger a reconfiguration –** after initial policy is deployed there are likely going to be changes, so the device needs to “check back” for updates, this needs to be configurable somehow



© 2023 ODVA, Inc.
All rights reserved.

Requirements

- **Ease of use with CIP Configuration –** it's the whole point of this endeavor, to support CIP configuration!
- **Suitable for embedded computing environments –** most (although not all) CIP endpoints are devices running in an embedded compute environment; technologies not well suited to this are not ideal (e.g. parsing takes up too many resources, etc.)
- **Human Readable –** nice to have; primary purpose is for machines to consume this data, not humans, but still it would be ideal if a human could read this data



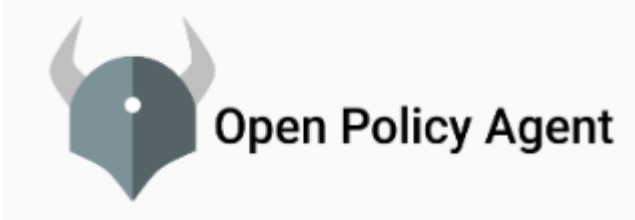
© 2023 ODVA, Inc.
All rights reserved.

Technologies – Some Examples

- The following are some example technologies that were evaluated
- More could of course be looked at, but these four give a reasonable representative set of the options
- AutomationML
  - Powerful but optimized for a different use case (data exchange amongst engineering tools)
- JSON
  - Lightweight and compact, schema would need to be defined
  - Would be well suited to the task as it would be custom-built for it

© 2023 ODVA, Inc.
All rights reserved.

- Existing policy language
  - Reusable but not well suited to embedded environments
  - Example of this would be Rego, which is used with the Open Policy Agent (OPA)
    - Others exist but clearly none are purpose built for this task
- Encoded CIP services
  - Devices already understand this
  - Services could simply be encoded in a document
  - High reuse but not human readable or independently extensible



© 2023 ODVA, Inc.
All rights reserved.

How do they stack up?

	AutomationML	Custom JSON	Existing Policy Language like Rego	Encoded CIP Commands	Explanation
Document Format	10	10	10	10	All options provide data in a document format
Authenticity	10	10	10	8	AutomationML, JSON, and most existing policy languages already have mechanisms for applying a digital signature. A custom file of encoded CIP commands would need to define a mechanism or choose from one of the many file signing formats.
Confidentiality	10	10	10	8	Essentially the same scoring and same reasoning as for authenticity
Versioning	10	10	10	8	Once again, the same reasoning holds; existing languages can use existing versioning

© 2023 ODVA, Inc.
All rights reserved.

How do they stack up?

	AutomationML	Custom JSON	Existing Policy Language like Rego	Encoded CIP Commands	Explanation
Automatic Discovery	n/a	n/a	n/a	n/a	None of these technologies provide this, it would need to be added through another means like DNS-SD
Configuration Retry	9	9	9	7	AutomationML, JSON and existing policy languages can easily encode this via a name-value pair, or encoded CIP commands as those don't have a seamless way to do this
Trigger a Reconfiguration	9	9	9	7	Same reasoning as for Configuration Retry

© 2023 ODVA, Inc.
All rights reserved.

How do they stack up?

	AutomationML	Custom JSON	Existing Policy Language like Rego	Encoded CIP Commands	Explanation
Suitable for an embedded environment	6	10	2	10	JSON is a very lightweight technology and would be specifically tailored to this use case, therefore it is highly efficient. Many of the existing languages are not well suited to an embedded space. AutomationML is used in some embedded applications, but is feature rich and built on XML, which is not very lightweight. CIP commands are already used in the embedded space, so this option is also very well suited.
Human Readable	8	9	9	3	JSON and many existing languages are very human readable, AutomationML is a bit more complex but still fits here. CIP commands however are not generally human readable.

© 2023 ODVA, Inc.
All rights reserved.

	AutomationML	Custom JSON	Existing Policy Language like Rego	Encoded CIP Commands	Explanation
Optimized for CIP	6	9	1	10	A custom JSON for CIP Security policy is well suited to delivering CIP, and of course CIP commands are perfectly suited to this task. AutomationML is not, and many existing policy languages are not possible to use for this purpose.
Totals	78	86	70	71	

© 2023 ODVA, Inc.
All rights reserved.

Conclusions

- This is a big step forward to enable security on more use cases, especially for IIoT
- JSON with a custom schema seems like the best option
- Spec enhancement to follow
  - Being worked on in SIG right now

© 2023 ODVA, Inc.
All rights reserved.


2023
ODVA
Industry Conference and 22nd Annual Meeting