W3C Web of Things Security Metadata Proposal

Michael McCool

W3C WoT WG Security and Privacy TF

Prague, March 2018

Outline

- Issues
- Goals and Requirements
- Examples
- For details…
  - See https://github.com/w3c/wot-security/pull/86 for details
  - Eventually will be merged into https://github.com/w3c/wot-security/blob/working/wot-security-metadata.md
  - Longer term, should be merged into TD deliverable

Issues

- Many different security authorization schemes
- Interactions may have multiple forms with different protocols
  - And different protocols may support different security schemes
- Different interactions may require different security configurations
  - This can also be supported with separate Things, but this is awkward
  - Fine-grained per-interaction control is useful to support roles, scopes (eg for OAuth), etc.
- We would like to have overall defaults and local per-form overrides
  - But this may or may not be easy to do with JSON-LD…

Goals and Requirements

- Support shared configurations
- Support per-form security configuration
- Support OR-AND combinations of schemes
- Avoid redundancy
- Avoid verbosity
- Be as consistent as possible with related standards
  - Such as OpenAPI, RAML, OCF, etc.
- Support a range of security configurations used in practice
  - But conversely, keep the scope narrow enough that attack surface is minimized and testing is practical
- Describe, don’t prescribe…

Examples

- Basic Example
  - Basic Authentication
  - API Keys
  - OCF Security
- Additional Examples
  - Bearer Tokens using JWT
  - Proxies
  - OAuth

Basic Example

```
{
  "@context": ["https://w3c.github.io/wot/w3c-wot-td-context.jsonld"],
  "@type": ["Thing"],
  "name": "MyLampThing",
  "@id": "urn:dev:wot:my-lamp-thing",
  "security": [
    {
      "@id": "basic-config",
      "scheme": "basic"
    },
    {
      "@id": "ocf-config",
      "scheme": "ocf"
    },
    {
      "@id": "apikey-config",
      "scheme": "apikey",
      "in": "header"
    }
  ],
```

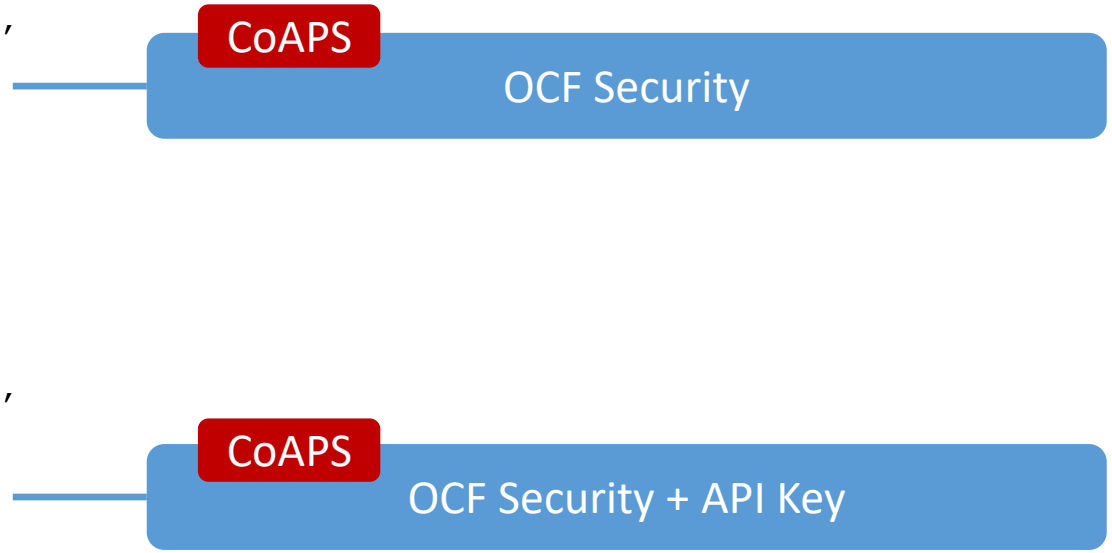
Basic Authentication Configuration

OCF Security Configuration

API Key Configuration

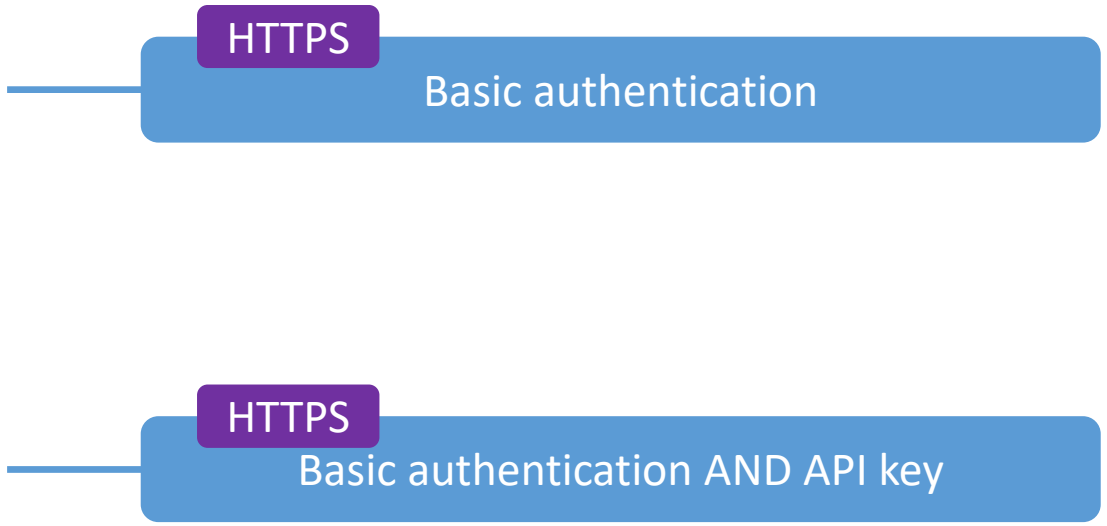
Basic Example

```
"interaction": [
    {
      "@type": ["Property"],
      "name": "status",
      "schema": {"type": "string"},
      "writable": true,
      "observable": true,
      "form": [
        {
          "href": "coaps://mylamp.example.com:5683/status",
          "mediaType": "application/json",
          "method": "coap:get",
          "security": "ocf-config"
        },
        {
          "href": "coaps://mylamp.example.com:5683/status",
          "mediaType": "application/json",
          "method": "coap:post",
          "security": ["ocf-config","apikey-config"]
        },
```



Basic Example

```
    {
        "href": "https://mylamp.example.com/status",
        "mediaType": "application/json",
        "method": "http:get",
        "security": "basic-config"
      },
      {
        "href": "https://mylamp.example.com/status",
        "mediaType": "application/json",
        "method": "http:post",
        "security": ["basic-config","apikey-config"]
      },
    ]
},
```



Basic Example

```
{
   "@type": ["Action"],
   "name": "toggle",
   "form": [
     {
       "href": "coaps://mylamp.example.com:5683/toggle",
       "mediaType": "application/json"
       "security": ["ocf-config","apikey-config"]
     },
     {
       "href": "https://mylamp.example.com/toggle",
       "mediaType": "application/json"
       "security": ["basic-config","apikey-config"]
     }
   ]
 },
```

CoAPS

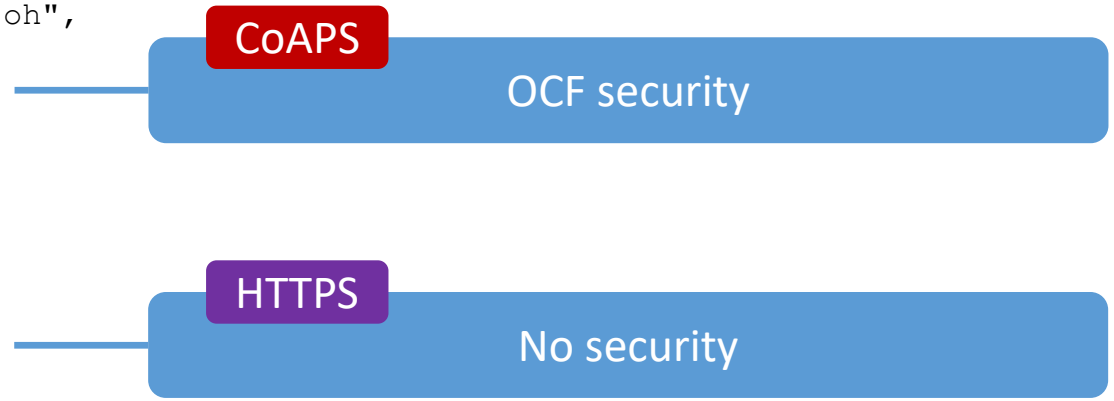
OCF security AND API key

HTTPS

Basic authentication AND API key

Basic Example

```
{
    "@type": ["Event"],
    "name": "overheating",
    "schema": {"type": "string"},
    "form": [
      {
        "href": "coaps://mylamp.example.com:5683/oh",
        "mediaType": "application/json"
        "security": "ocf-config"
      },
      {
        "href": "https://mylamp.example.com/oh",
        "mediaType": "application/json"
      }
    ]
  }
]
}
```



Bearer Tokens

```
{
  "@context": ["https://w3c.github.io/wot/w3c-wot-td-context.jsonld"],
  "@type": ["Thing"],
  "name": "FujitsuBeacon",
  "@id": "urn:dev:wot:fujitsu-beacon",
  "security": [
    {
      "@id": "bearer-token-config",
      "scheme": "token",
      "format": "jwt",
      "alg": "ES256",
      "as": "https://plugfest.thingweb.io:8443/"
    },
    ... // other security configurations, if needed
  ],
  ...
```

Bearer Token Configuration

Bearer Tokens

```
"interaction": [
    {
      "form": [
        {
          ...
          "security": "bearer-token-config"
        },
        ... // other forms
    },
    ... // other interactions
  ]
}
```

Bearer token authentication

Proxies

```
{
  "@context": ["https://w3c.github.io/wot/w3c-wot-td-context.jsonld"],
  "@type": ["Thing"],
  "name": "Festo",
  "@id": "urn:dev:wot:festo",
  "security": [
    {
      "@id": "proxy-config",
      "scheme": "basic",
      "proxy": "http://plugfest.thingweb.io:8087"
    },
    {
      "@id": "endpoint-config",
      ... // details omitted; independent of proxy configuration
    },
    ... // other security configurations, if needed
  ],
  ...
```

Proxy Security Configuration

Endpoint Security Configuration

Proxies

```
"interaction": [
    {
      "form": [
        {
          ...
          "security": ["proxy-config","endpoint-config"]
        },
        ... // other forms
    },
    ... // other interactions
  ]
}
```

Combined Proxy and Endpoint Security

OAuth

```
{
  "@context": ["https://w3c.github.io/wot/w3c-wot-td-context.jsonld"],
  "@type": ["Thing"],
  "name": "Camera",
  "@id": "urn:dev:wot:camera",
  "security": [
    {
      "@id": "oauth-config",
      "scheme": "oauth2",
      "flow": "code",
      "as": "https://example.com/api/oauth/dialog",
      "ts": "https://example.com/api/oauth/token",
      "rs": "https://example.com/api/oauth/refresh",
      "scope": [
        { "name": "read:frame" },
        ... // other scopes
       ]
    }
    ... // other security configurations, if needed
  ],
  ...
```

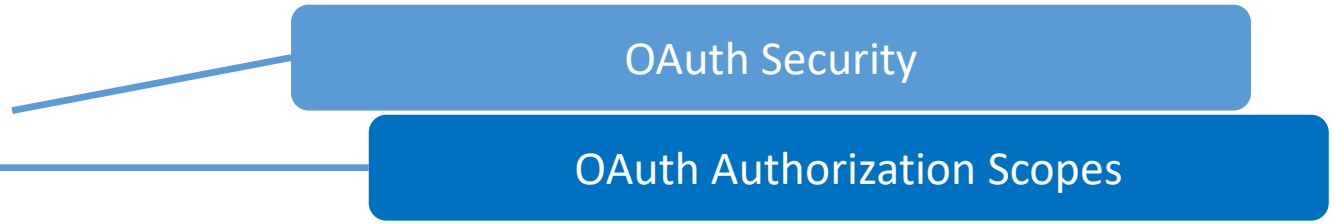
OAuth2 Security Configuration

OAuth2 “Code” Flow Configuration

OAuth2 Named Authentication Scopes

OAuth

```
"interaction": [
    {
      "@type": ["Property"],
      "name": "frame",
      ...
      "writable": false,
      "observable": true,
      "form": [
        {
          "href": "https://example.com/api/frame",
          "mediaType": "image/jpeg",
          "method": "http:get",
          "security": ["oauth-config"],
          "auth": ["read:frame"]
        },
        ... // other forms
    },
    ... // other interactions
  ]
}
```



Details

- List of security schemes supported
- Detailed definitions of tags
  - https://github.com/w3c/wot-security/pull/86
  - https://github.com/w3c/wot-security/blob/working/wot-security-metadata.md
- To Discuss:
  - Compatibility with JSON-LD
    - Use of defaults, arrays, @id, etc.
  - List of supported security mechanisms
  - Validation issues
    - There are only certain valid combinations of scheme and protocol…
  - Bikeshed issues: names (“as” vs. “authenticationUrl”, etc.)