XMCO

The Portal

API documentation

XMCO - Security consulting company

www.xmco.fr

info@xmco.fr

Tel : +33 (0)1 79 35 29 30

SUMMARY

Authentication

To authenticate, make a HTTP request `POST` towards `https://leportail.xmco.fr/api/authenticate` .

The request should have the following structure :

```
POST /api/authenticate HTTP/1.1
Host: leportail.xmco.fr
Content-Type: application/json
Content-Length: 76
Connection: close

{"username":"value","password":"password"}
```

Request(s) example(s)
cUrl

```
curl -i -d '{"username":"value", "password":"password"}' -H "Content-Type: application/json" -X POST 'https://leportail.xmco.fr/api/authenticate'
```

The response to this request will include a session cookie, identified as `XMSESSION` . **It must be sent to the API for each request.**
To do this, the request must include a header `Cookie` of the form `XMSESSION=xxxxx` .

Case of 2 factor authentication

If 2 factor authentication is enabled, an additional request to `https://leportail.xmco.fr/api/validate_otp` is necessary.

The session cookie retrieved from the request to `https://leportail.xmco.fr/api/authenticate` will have to be included in this request, which will have the following structure :

```
POST /api/validate_otp HTTP/1.1
Host: leportail.xmco.fr
Content-Type: application/json
Content-Length: 76
Connection: close
Cookie: XMSESSION=xxxxxxxxxx

{"otp":"the code received by SMS"}
```

Request(s) example(s)
cUrl

```
curl -i -d '{"otp":"le code reçu par SMS"}' -H "Content-Type: application/json" -H "Cookie: XMSESSION=xxxxxxxxxx" -X POST 'https://leportail.xmco.fr/api/validate_otp'
```

The response to this request will include the session cookie that must be sent to the API for each request.
The session cookie retrieved from the request to `https://leportail.xmco.fr/api/authenticate` will no longer be used.

Action plans

Access to the list of tickets assigned to me

Prerequisites: Obtain your user ID (`user_id`)

This identifier will allow you to filter the tickets assigned to you. It is contained in the field `user._id` obtained by a request `GET` to the url `https://leportail.xmco.fr/api/user/current`.

Request(s) example(s)
cUrl

```
curl -H "Content-Type: application/json" -H "Cookie: XMSESSION=xxxxxxxxxx" 'https://leportail.xmco.fr/api/user/current'
```

Response format :


```
{
"user": {
"_id": "55ca2585f2c3425976c28abf",
"first_name": "Jean-Yves",
"company": "xxxxxxxxxx",
"...": "..."
},
"...": "..."
}
```


Tip: This ID can be retrieved via the My Account section on `https://leportail.xmco.fr`. It can be found in the URL

Get the list of tickets assigned to me

This list can be obtained via a GET request to the url `https://leportail.xmco.fr/api/action_plan/list` specifying the argument `assignee_id` with the identifier obtained previously.

Request(s) example(s)
cUrl

```
curl -G -g -H "Content-Type: application/json" -H "Cookie: XMSESSION=xxxxxxxxxx" 'https://leportail.xmco.fr/api/action_plan/list' --data-urlencode 'assignee_id=55ca2585f2c3425976c28abf'
```

Response format :


```
{
"_items": [
{
"_id": "5d3bfe1ad234ff01d3a35e2c",
"_created": "Mon, 06 May 2019 15:52:14 GMT",
"_deleted": false,
"_etag": "e8b7960beb33e23c7e771f91e4dd68142f852f9a",
"_latest_version": 2,
"_updated": "Thu, 01 Aug 2019 13:29:40 GMT",
"_version": 3,
"assignee": "...",
"certified": false,
"company": "...",
"custom_fields": {
"category": "disclosure",
"subcategory": "disclosure_other",
"identification": "automatic",
"scope": "passive"
},
"description": "...",
"description_format": "html",
"impact": 0.0,
"is_open": false,
"journals": [
"..."
],
"owner": "55ca2585f2c3425976c28ac0",
"quickwin": false,
"ref_id": 331750,
"revision": 0,
"scope": "5d3bfe0cd234ff01d3a35d9a",
"severity": "low",
"status": "new",
"title": "Une adresse IP est référencée dans la liste noire Spamhaus",
"tracker": "5ca6019ff44a43003f808ffd",
"type": "serenety_action_ticket",
"users": [
"55ca2585f2c3425976c28ac0"
],
"closed_on": "Thu, 01 Aug 2019 13:29:40 GMT",
"assets": [],
```

```
"cpe_names": []
}
],
"_meta": {
"total": 1
}
}
```


Access to the list of company tickets

Prerequisites: Obtain your company ID (`company_id`)

This identifier is contained in the field `user.company` obtained by a request `GET` to the url `https://leportail.xmco.fr/api/user/current`.

Request(s) example(s)
cUrl

```
curl -H "Content-Type: application/json" -H "Cookie: XMSESSION=xxxxxxxxxx" 'https://leportail.xmco.fr/api/user/current'
```

Response format :


```
{
"user": {
"_id": "55ca2585f2c3425976c28abf",
"first_name": "Jean-Yves",
"company": "xxxxxxxxxx",
"...": "..."
},
"...": "..."
}
```


Get the list of the company's tickets

This list can be obtained via a request `GET` to the url `https://leportail.xmco.fr/api/action_plan/list` specifying the argument `company_id` with the identifier obtained previously. The results returned are limited according to your access rights! If no filter is applied, the result can therefore be voluminous.

Request(s) example(s)
cUrl

```
curl -G -g -H "Content-Type: application/json" -H "Cookie: XMSESSION=xxxxxxxxxx" 'https://leportail.xmco.fr/api/action_plan/list' --data-urlencode 'company_id=55ca2584f2c3425976c28aa4'
```

Access tickets linked to a particular module (Audit, Yuno, Serenety)

Prerequisite: Obtain your company ID (`company_id`)

See above.

Prerequisites: Obtain tracker IDs

This list can evolve over time, we propose you to recover it dynamically. This list can be obtained via a request `POST` to the url `https://leportail.xmco.fr/api/tracker/by_company` by specifying the argument `company_id` with the identifier obtained previously and with one of the following values for the parameter `module` : `watch`, `serenety` or `pentest`. We can thus extract the list of tracker identifiers, contained in the field `_id` of each document on the list returned by the API.

The parameter `queryExtras` supports the MongoDB syntax.

Various filters are available, including :

- open tickets : `"is_open":{"$eq":true}`
- the "New" or "In progress": `"status":{"$in":["new","in_progress"]}`
- the creation date : `"_created":{"$gt":"2019-07-31T22:00:00.000Z"}`

Request(s) example(s)
cUrl

```
curl -g -XPOST -H "Content-Type: application/json" -H "Cookie: XMSESSION=xxxxxxxxxx" 'https://leportail.xmco.fr/api/tracker/by_company' --data '{"module":"serenety","company_id":"55ca2584f2c3425976c28aa4"}'
```

Response format :


```
{
"_items": [
{
"_id": "5ca6019ff44a43003f808ffd",
"_created": "Thu, 04 Apr 2019 13:07:43 GMT",
```

```
"_updated": "Thu, 04 Apr 2019 13:07:43 GMT",
"_deleted": false,
"_etag": "16e0ecc7-3b47-4dc1-b2ed-4324d3d9ca57",
"_version": 1,
"_latest_version": 1,
"name": {
"fr": "Alertes Serenety",
"en": "Serenety alerts"
},
"module": "serenety",
"fields": [
"..."
]
},
"..."
]
}
```


Get the list of tickets associated with a module

This list can be obtained via a GET request to the url `https://leportail.xmco.fr/api/action_plan/list` specifying the argument `company_id` with the identifier obtained previously and by setting the queryExtras argument with the desired filter. In our specific case, we will use the syntax `'tracker':{'$in': ['list of tracker IDs previously retrieved']}`

Request(s) example(s)

cUrl

```
curl -G -g -H "Content-Type: application/json" -H "Cookie: XMSESSION=xxxxxxxxxx" 'https://leportail.xmco.fr/api/action_plan/list' --data-urlencode 'queryExtras={"tracker":{"$in":["5ca6019ff44a43003f808ffd","5ca6019ff44a43003f808ffe"]}}' --data-urlencode 'company_id=55ca2584f2c3425976c28aa4'
```

Response format :


```
{
"_items": [
{
"_id": "5d3bfe1ad234ff01d3a35e2c",
"_created": "Mon, 06 May 2019 15:52:14 GMT",
"_deleted": false,
"_etag": "e8b7960beb33e23c7e771f91e4dd68142f852f9a",
"_latest_version": 2,
"_updated": "Thu, 01 Aug 2019 13:29:40 GMT",
"_version": 3,
"assignee": "...",
"certified": false,
"company": "...",
"custom_fields": {
"category": "disclosure",
"subcategory": "disclosure_other",
"identification": "automatic",
"scope": "passive"
},
"description": "...",
"description_format": "html",
"impact": 0.0,
"is_open": false,
"journals": [
"..."
],
"owner": "55ca2585f2c3425976c28ac0",
"quickwin": false,
"ref_id": 331750,
"revision": 0,
"scope": "5d3bfe0cd234ff01d3a35d9a",
"severity": "low",
"status": "new",
"title": "Une adresse IP est référencée dans la liste noire Spamhaus",
"tracker": "5ca6019ff44a43003f808ffd",
"type": "serenety_action_ticket",
"users": [
"55ca2585f2c3425976c28ac0"
],
"closed_on": "Thu, 01 Aug 2019 13:29:40 GMT",
"assets": [],
"cpe_names": []
}
],
"_meta": {
"total": 1
}
}
```

Update an action ticket

Prerequisites: Obtain the identifier and the `etag` of the ticket to be modified

This information can be found in the fields `_id` and `_etag`, obtained via the queries previously described.

Update the ticket

An update consists of a request `PATCH` to the url `https://leportail.xmco.fr/api/action_plan/xxxxxxxxx`. This request must contain the `etag` in its header "If-Match" as well as the modifications in the form of a dictionary.

Request(s) example(s)

cUrl

```
curl -g -X PATCH -H "Content-Type: application/json" -H "Cookie: XMSESSION=xxxxxxxxxx" -H "If-Match: etag" 'https://leportail.xmco.fr/api/action_plan/xxxxxxxxx' --data '{"severity":"low","status":"in_progress"}'
```

Response format :


```
{
"_created": "Tue, 06 Aug 2019 08:27:32 GMT",
"_deleted": false,
"_etag": "aad40814de0e81ca0420273d05b906a59c3b6976",
"_id": "xxxxxxxxx",
"_latest_version": 2,
"_links": {
"self": {
"href": "action_plan/xxxxxxxxx",
"title": "Actionticket"
}
},
"_status": "OK",
"_updated": "Thu, 08 Aug 2019 14:03:57 GMT",
"_version": 2
}
```

Recovery of Yuno bulletins

Data format

The YUNO advisories can be retrieved by making a request `GET` towards `https://leportail.xmco.fr/api/advisory` . The call will return a document in JSON format, which will have the following structure :

Request(s) example(s)
cUrl

```
curl -H "Content-Type: application/json" -H "Cookie: XMSESSION=xxxxxxxxxx" 'https://leportail.xmco.fr/api/advisory'
```

Example of an INFO bulletin


```
{
"_items": [
{
"_id": "5b893bf6503bc6000e93d13e",
"exploitation_vector": "none",
"platforms": [],
"references": "https://www.bankinfosecurity.com/air-canada-attack-exposed-data-on-20000-mobile-app-users-a-11441",
"nonvulnerable": "",
"vulnerable": "",
"content_fr": {
"description": "La compagnie aérienne Air Canada a forcé ses 1,7 million d'utilisateurs à changer leur mot de passe après avoir détecté une activité anormale. Air Canada a annoncé que 20 000 comptes auraient été exposés. Les comptes contiendraient des détails sur les passeports des clients. La plupart des informations dépendent de ce que l'utilisateur souhaite partager. En règle général, un profile d'application contient au moins le nom, le prénom, une adresse postale et un email. Mais d'autres utilisateurs enregistrent également leur passeport en incluant : date de naissance, nationalité, date d'expiration du document, pays de délivrance et pays de résidence. Air Canada ajoute que la faille n'a exposé aucun moyen de payement. La compagnie assure que ces données sont chiffrées et en accord avec les règles du PCI-DSS.",
"title": "Une cyber attaque a exposé les données de 20 000 utilisateurs mobiles d'Air Canada",
"description_format": "text",
"mitigation": "",
"remediation": "",
"expert_advice": ""
},
"advisory_type": "INFO",
"severity": "high",
"cve_refs": [],
"exploit_code": "",
"damage": [],
"languages": [
"fr"
],
"ref": "2018-3543",
"files": [],
"impact": 0,
"vendor": "",
"content_en": {
"description": "",
"title": "Air Canada: Attack Exposed 20,000 Mobile App Users' Data",
"description_format": "text",
"mitigation": "",
"remediation": "",
"expert_advice": ""
},
"tags": [
"Attaque",
"Vie privée"
],
"ioc": "",
"official_title": "Air Canada: Attack Exposed 20,000 Mobile App Users' Data",
"exploit_code_lang": "none",
"_created": "Fri, 31 Aug 2018 07:55:33 GMT",
"_updated": "Fri, 31 Aug 2018 13:00:38 GMT",
"_deleted": false,
"_etag": "71ac4bf26b5aae02a6ab7bf9854cc7cbed344c3c",
"cpe_names": [],
"_links": {
"parent": {
"title": "home",
"href": "/"
},
"self": {
"title": "Advisory",
"href": "advisory/5b893bf6503bc6000e93d13e"
},
```

```
"collection": {
"title": "advisory",
"href": "advisory"
}
},
"_version": 1,
"_latest_version": 1
},
{
"_id": "abcdefabcdefabcdefabcdef",
"...": "..."
}
],
"_links": {
"parent": {
"title": "home",
"href": "/"
},
"self": {
"title": "advisory",
"href": "advisory?max_results=25"
},
"next": {
"title": "next page",
"href": "advisory?max_results=25&page=2"
},
"last": {
"title": "last page",
"href": "advisory?max_results=25&page=36700"
}
},
"_meta": {
"page": 1,
"max_results": 25,
"total": 36700
}
}
```


Example of a PATCH bulletin


```
{
"_items": [
{
"_id": "5b893bc5503bc6000e93d0e0",
"exploitation_vector": "remote",
"platforms": [
"55ca2585f2c3425976c28ac1"
],
"references": "https://www.synology.com/fr-fr/support/security/Synology_SA_18_51",
"nonvulnerable": "",
"vulnerable": "* DSM 6.2 * DSM 6.1 * DSM 5.2",
"content_fr": {
"description": "Une vulnérabilité a été corrigée au sein des produits de Synology. Son exploitation permettait à un attaquant de voler des informations et de manipuler des données. La faille de sécurité non référencée provenait d'une erreur non spécifiée au sein de Synology DiskStation Manager (DSM). En exploitant cette vulnérabilité un attaquant était en mesure de voler des informations et d'injecter du code JavaScript et HTML dans les pages accédées par les clients.",
"title": "Manipulation de données et vol d'informations via une vulnérabilité au sein des produits Synology (Synology-SA-18:51 DSM)",
"description_format": "text",
"mitigation": "",
"remediation": "Le CERT-XMCO recommande l'installation de la version 6.2.1-23824 des produits DSM 6.2, DSM 6.1 et DSM 5.2.",
"expert_advice": ""
},
"advisory_type": "PATCH",
"severity": "medium",
"cve_refs": [],
"exploit_code": "",
"damage": [
"data_tampering",
"data_theft"
],
"languages": [
"fr"
],
"ref": "2018-3540",
"files": [],
"impact": 0,
"vendor": "SYNOLOGY",
"content_en": {
"description": "",
"title": "Data manipulation and information disclosure via a vulnerability in Synology products (Synology-SA-18:51 DSM)",
"description_format": "text",
"mitigation": "",
```

```
"remediation": "The CERT-XMCO recommends installing the version 6.2.1-23824 of the products DSM 6.2, DSM 6.1 and DSM 5.2.",
"expert_advice": ""
},
"tags": [],
"ioc": "",
"official_title": "Synology-SA-18:51 DSM",
"exploit_code_lang": "none",
"_created": "Fri, 31 Aug 2018 07:08:39 GMT",
"_updated": "Fri, 31 Aug 2018 12:59:49 GMT",
"_deleted": false,
"_etag": "2060ecb85285eea4774f35251df1b93fc0dbfd60",
"cpe_names": [
"59a3ccc5f2c3427b958db52d"
],
"_links": {
"parent": {
"title": "home",
"href": "/"
},
"self": {
"title": "Advisory",
"href": "advisory/5b893bc5503bc6000e93d0e0"
},
"collection": {
"title": "advisory",
"href": "advisory"
}
},
"_version": 1,
"_latest_version": 1
},
{
"_id": "abcdefabcdefabcdefabcdef",
"...": "..."
}
],
"_links": {
"parent": {
"title": "home",
"href": "/"
},
"self": {
"title": "advisory",
"href": "advisory?max_results=25"
},
"next": {
"title": "next page",
"href": "advisory?max_results=25&page=2"
},
"last": {
"title": "last page",
"href": "advisory?max_results=25&page=36700"
}
},
"_meta": {
"page": 1,
"max_results": 25,
"total": 36700
}
}
```


Search filters

It is possible to apply search filters using the parameter `where`. The format to be applied is that of MongoDB requests. All MongoDB operations are supported, with the exception of operators `$regex` and `$where` .

Request(s) example(s)

Return only INFO advisories

```
curl -g -H "Content-Type: application/json" -H "Cookie: XMSESSION=xxxxxxxxxx" 'https://leportail.xmco.fr/api/advisory' --data-urlencode 'where={"advisory_type":"INFO"}'
```

Return only advisories published between 8 and 22 February 2018 (dates follow the RFC2822 format)

```
curl -g -H "Content-Type: application/json" -H "Cookie: XMSESSION=xxxxxxxxxx" 'https://leportail.xmco.fr/api/advisory' --data-urlencode 'where={"_created":{"$gte":"Thu, 8 Feb 2018 00:00:00 GMT","$lte":"Thu, 22 Feb 2018 00:00:00 GMT"}}'
```

Return only High severity advisories

```
curl -g -H "Content-Type: application/json" -H "Cookie: XMSESSION=xxxxxxxxxx" 'https://leportail.xmco.fr/api/advisory' --data-urlencode 'where={"severity":"high"}'
```

Return only advisories about Debian

```
curl -g -H "Content-Type: application/json" -H "Cookie: XMSESSION=xxxxxxxxxx" 'https://leportail.xmco.fr/api/advisory' --data-urlencode 'where={"cpe_names":"55ca2587f2c3425976c28d8d"}'
```

Return only advisories about Wordpress, with severity greater or equal to "Medium" and published since 8 February 2018

```
curl -g -H "Content-Type: application/json" -H "Cookie: XMSESSION=xxxxxxxxxx" 'https://leportail.xmco.fr/api/advisory' --data-urlencode 'where={"cpe_names":"55ca2586f2c3425976c28d43","severity":{"$in":["urgent","high","medium"]},"_created":{"$gte":"Thu, 8 Feb 2018 00:00:00 GMT"}}'
```

Retrieving the identifier of a technology

As illustrated in the sample requests, it is possible to apply filters to retrieve advisories for a particular technology. For this purpose, it is necessary to know the identifier of the technology in question.
It is possible to search for a technology from its name by making a query `GET` to the endpoint `cpename` .

Request(s) example(s)
If you wish to obtain information on the technology `Wordpress`

```
curl -H "Content-Type: application/json" -H "Cookie: XMSESSION=xxxxxxxxxx" 'https://leportail.xmco.fr/api/cpename' --data-urlencode 'where={"name":"Wordpress"}'
```

It will return the following document :

Example of a search result on a technology


```
{
"_items": [
{
"_id": "55ca2586f2c3425976c28d43",
"edition": "*",
"uri_23": "cpe:/a:wordpress:wordpress",
"meta_ref": null,
"is_meta": true,
"target_hw": "*",
"is_custom": true,
"sw_edition": "*",
"version": "*",
"product": "wordpress",
"fs": "cpe:2.3:a:wordpress:wordpress:*:*:*:*:*:*:*:*",
"vendor": "wordpress",
"update": "*",
"target_sw": "*",
"language": "*",
"part": "a",
"name": "Wordpress",
"cpe_names": [],
"other": "*",
"_created": "Tue, 11 Aug 2015 16:40:38 GMT",
"_updated": "Tue, 11 Aug 2015 16:40:38 GMT",
"_deleted": false,
"_etag": "b3db0e4513c71206eb97cc2bf6dd288e9627eaa1",
"_links": {
"self": {
"title": "Cpename",
"href": "cpename/55ca2586f2c3425976c28d43"
}
},
"_version": 1,
"_latest_version": 1
}
],
"_links": {
"parent": {
"title": "home",
"href": "/"
},
"self": {
"title": "cpename",
"href": "cpename?where={\"name\":\"Wordpress\"}"
}
},
"_meta": {
"page": 1,
"max_results": 25,
"total": 1
}
}
```


The list of documents validating the search filters is in the list `_items` . Each of these documents has a field `_id` which contains the identifier of the element.
In the previous example, only one technology with the name `Wordpress` has been reassembled. Its identifier is `55ca2586f2c3425976c28d43` .
Technology bulletins `Wordpress` can therefore be recovered with the filter `/advisory?where={"cpe_names": "55ca2586f2c3425976c28d43"}`

Paging

The returned results by the API are paginated.
By default, each page contains 25 results (up to 50 can be obtained using the parameter `max_results`).
It is possible to navigate from page to page using the parameter `page` : `/advisory?page=2` , `/advisory?page=3` , etc.

Asset recovery

It is possible to retrieve the assets processed by Serenety by making a request `GET` towards `https://leportail.xmco.fr/api/serenety/asset` .

Request(s) example(s)

Retrieve assets

```
curl -H "Content-Type: application/json" -H "Cookie: XMSESSION=xxxxxxxxxx" 'https://leportail.xmco.fr/api/serenety/asset'
```

Return only the IP addresses

```
curl -g -H "Content-Type: application/json" -H "Cookie: XMSESSION=xxxxxxxxxx" 'https://leportail.xmco.fr/api/serenety/asset' --data-urlencode 'where={"type":"ipv4"}'
```

The returned documents will have the following structure :

Note: The data structure of these properties may change. The content of each property depends on the information available about the asset.

Serenety asset example


```
{
"_items": [
{
"_latest_version": 1216,
"_updated": "Mon, 20 Aug 2018 07:44:49 GMT",
"_links": {
"self": {
"href": "asset/19ca3a48-67de-475e-8a6a-f4368cfaf114",
"title": "Asset"
}
},
"_id": "19ca3a48-67de-475e-8a6a-f4368cfaf114",
"_created": "Mon, 12 Dec 2016 16:36:32 GMT",
"type": "hostname",
"related_assets": [
"00bbee3e-7fd2-4917-bf5a-ce4e3d068763"
],
"tags": [
"intrusive"
],
"_version": 1216,
"value": "leportail.xmco.fr",
"_deleted": false,
"company": "14c05f9d-6b3f-4c30-8cfd-1450d7028924",
"properties": {
"ssl_protos": [
{
"protos": [
"tlsv1_1",
"tlsv1_2"
],
"port": 443,
"run": "f01a2a0f-3a55-4fc1-a82e-93cba2f33665"
}
],
"ssl_ciphers": [
{
"ciphers": {
"HIGH": [
"AES128-GCM-SHA256",
"AES128-SHA",
"AES128-SHA256",
"AES256-GCM-SHA384",
"AES256-SHA",
"AES256-SHA256",
"CAMELLIA128-SHA",
"CAMELLIA256-SHA",
"DHE-RSA-AES128-GCM-SHA256",
"DHE-RSA-AES128-SHA",
"DHE-RSA-AES128-SHA256",
"DHE-RSA-AES256-GCM-SHA384",
"DHE-RSA-AES256-SHA",
"DHE-RSA-AES256-SHA256",
"DHE-RSA-CAMELLIA128-SHA",
"DHE-RSA-CAMELLIA256-SHA",
"ECDHE-RSA-AES128-GCM-SHA256",
```

```
"ECDHE-RSA-AES128-SHA",
"ECDHE-RSA-AES128-SHA256",
"ECDHE-RSA-AES256-GCM-SHA384",
"ECDHE-RSA-AES256-SHA",
"ECDHE-RSA-AES256-SHA384"
]
},
"port": 443,
"run": "f01a2a0f-3a55-4fc1-a82e-93cba2f33665"
}
],
"ports": [
{
"banner": "Microsoft IIS httpd 7.5",
"service": "http",
"type": "tcp",
"run": "93d5abfe-966a-4c8e-8d94-6d961153256d",
"no": 80
},
{
"banner": "Le serveur",
"service": "https",
"type": "tcp",
"run": "93d5abfe-966a-4c8e-8d94-6d961153256d",
"no": 443
}
],
"ssllabs": [
{
"hpkp": false,
"cert_trusted": true,
"cert_issues": [],
"cert_transparency": true,
"mark": "A+",
"run": "f01a2a0f-3a55-4fc1-a82e-93cba2f33665",
"vulns": {
"logjam": false,
"poodleSSL": false,
"drown": false,
"crime": false,
"freak": false,
"ticketbleed": false,
"heartbleed": false,
"poodleTLS": false,
"CVE-2014-0224": false,
"CVE-2016-2107": false,
"ROBOT": false,
"beast": false
},
"cert_chain_issues": [],
"forward_secrecy": true,
"hsts": false
}
],
"ssl_cert": [
{
"valid_to": "Mon, 12 Oct 2020 00:00:00 GMT",
"wildcard": true,
"alt_names": [
"*.xmco.fr",
"xmco.fr"
],
"fingerprint": "D4:11:B2:D7:A1:C5:61:61:06:84:18:A1:16:7F:C5:1D:49:E4:6E:B5",
"cert": "-----BEGIN CERTIFICATE-----CERTIFICATE_DATA-----END CERTIFICATE-----\n",
"sig_algo": "sha256WithRSAEncryption",
"expired": false,
"symantec_untrust": false,
"run": "f01a2a0f-3a55-4fc1-a82e-93cba2f33665",
"key": "RSA 2048",
"ca_name": "Thawte TLS RSA CA G1",
"cert_name": "*.xmco.fr",
"valid": true,
"issuer_name": "C=US, O=DigiCert Inc, OU=www.digicert.com, CN=Thawte TLS RSA CA G1",
"valid_from": "Tue, 10 Jul 2018 00:00:00 GMT",
"port": 443,
"sig_is_weak": false,
"self_signed": false,
"serial": "72058909014212716801862329242211204 34"
}
]
},
"_etag": "d57a66508494ae45cda60d76b0a5211dd96c2703"
```

```
},
{
"_id": "abcdefabcdefabcdefabcdef",
"...": "..."
}
],
"_links": {
"self": {
"href": "asset?where={\"value\": \"leportail.xmco.fr\", \"company\": \"14c05f9d-6b3f-4c30-8cfd-1450d7028924\"}",
"title": "asset"
},
"parent": {
"href": "/",
"title": "home"
}
},
"_meta": {
"max_results": 25,
"page": 1,
"total": 1
}
}
```


Search filters

- Return only the IP addresses : `/serenety/asset' --data-urlencode 'where={"type": "ipv4"}'`
- Return only the asset 'xmco.fr'(it will be always contained in the _items list) : `/serenety/asset' --data-urlencode 'where={"value":"xmco.fr"}'`
- Return only assets of which port 80 is exposed : `/serenety/asset' --data-urlencode 'where={"properties.ports.no": 80}'`
- Return only the assets vulnerable to HeartBleed : `/serenety/asset' --data-urlencode 'where={"properties.ssllabs.vulns.heartbleed": true}'`

List of available properties

- **ssl_cert** : SSL certificate information
- **ssl_ciphers** : Information on supported encryption algorithms
- **ssl_protos** : Information on supported SSL protocols
- **ssllabs** : SSLLabs Analysis Information
- **ports** : Information on the services on display
- **axfr** : Information on the configuration of the DNS zone transfer
- **http_controls** : HTTP configuration information
- **dns_records** : DNS configuration information
- **vulns** : Information on detected vulnerabilities
- **domain_expiration** : Domain Expiration Information