

14. Module - Pentesting APIs & Cloud Applications

Intro to API

We will focus on web APIs:

- Web services (SOAP/XML)
- REST APIs (JSON)

SOAP

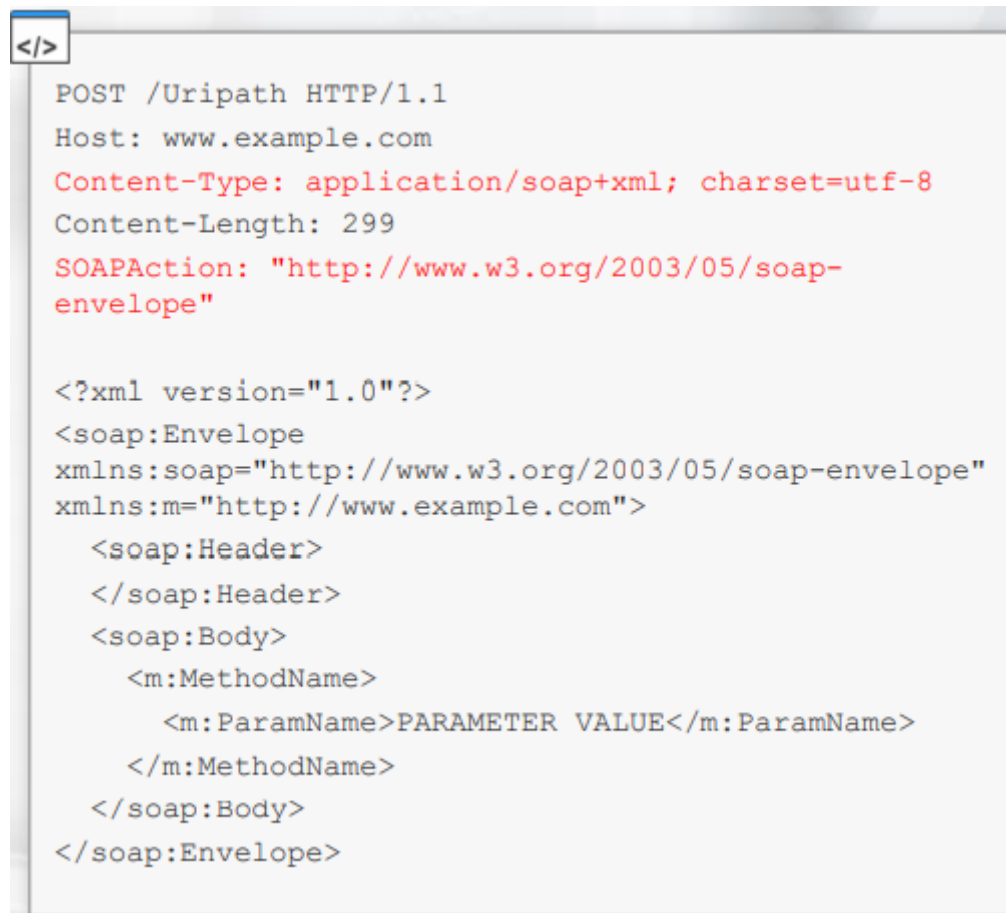
-> SOAP Messages (HTTP Requests) are an XML type

-> It must contain

1.Content type text/xml

2.SOAPAction is sometimes used just for the standard

REQUEST:



```
</>
POST /Uripath HTTP/1.1
Host: www.example.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: 299
SOAPAction: "http://www.w3.org/2003/05/soap-envelope"

<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:m="http://www.example.com">
  <soap:Header>
  </soap:Header>
  <soap:Body>
    <m:MethodName>
      <m:ParamName>PARAMETER VALUE</m:ParamName>
    </m:MethodName>
  </soap:Body>
</soap:Envelope>
```

RESPONSE:



```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope
/">
  <soap:Body>
    <MethodResult xmlns="http://tempuri.org/">
      <ResultValue>TheValue</ResultValue>
    </MethodResult>
  </soap:Body>
</soap:Envelope>
```

-> API contains both human and machine-readable documentation.

-> For SOAP-based APIs, the documentation is stored in WSDL files. Usually, these files are stored under the „?wsdl” path, for example, <https://api.example.com/api/?wsdl>.

-> REST API

GET /api/v2/methodName

-> REST API has documentation standard called the WADL file. A sample WADL can be viewed here: <https://www.w3.org/Submission/wadl/>

-> Except for GET requests, API methods parameters are passed in the request body

An exemplary REST API request can be seen to the right:

- Path often contains the API version
- Content-Type application/json header is required
- Parameters are passed as JSON array

It is also often possible to pass the REST API parameters as XML, so the equivalent of the request from the previous slide would look like the listing to the right.

</>

```
POST /api/2.2/auth/signin HTTP/1.1
HOST: my-server
Content-Type:application/json
Accept:application/json
```

```
{
  "credentials": {
    "name": "administrator",
    "password": "passw0rd",
    "site": {
      "contentUrl": ""
    }
  }
}
```

```
POST /api/2.2/auth/signin HTTP/1.1
HOST: my-server
Content-Type:text/xml
```

```
<tsRequest>
  <credentials name="administrator"
password="passw0rd">
    <site contentUrl="" />
  </credentials>
</tsRequest>
```

More Resources:

<https://swagger.io/>

<https://www.w3.org/TR/wsdl.html>

<https://www.w3.org/Submission/wadl/>

<https://www.w3.org/TR/soap/>

API Testing and Attacking

While testing an API, you should answer:

- > What is the API name and version?
- > Is it a custom implementation or, for example, an open-source product?
- > Is there any online documentation available? Are there any interesting methods?
- > Does the documentation exist on the target server (?wsdl, ?wadl, or similar)?
- > Does the API require authentication, or is publicly available?
- > If there is both local and public documentation for an API, do they match? Maybe some methods were hidden from local users (typically ones that allow insecure operations).

So your purpose is to gather as many API endpoints as possible and to be able to speak to them. You should also be able to get the WSDL/WADL file for further testing.

Reconstructing API calls from a raw WSDL/WADL file would be time-consuming, so we use tools like **Postman**, **SOAPUI**, **WSDLer**