WSDL

Challenges

Notes

A web service is a server-side program that provides functionalities that can be invoked and used programmatically.
The main purpose of using web services is to have an interoperable architecture that is able to connect different devices and different pieces of software together

They are usually intended to facilitate:
- Integration between applications: application 'A' uses features implemented in application 'B'
- Separation within an application: front-end scripts that use web services functionalities to update the content

You can implement web services in many different ways. The most commonly used and popular ones are:

- XML-RPC: remote procedure call (RPC) protocol that uses XML (usually over HTTP) to invoke functionalities.
  - The first web service protocol. It works by sending HTTP requests that call a single method implemented on the remote system.

- JSON-RPC: remote procedure call protocol that uses JSON
  - JSON-RPC is very similar to XML-RPC. However, it provides more human-readable messages and takes less space to send the same message XML-RPC sends.

- SOAP: messaging protocol that uses XML and provides
  more functionalities of XML-RPC
  - SOAP (Simple Object Access Protocol) is somehow the successor of XML-RPC since it provides more functionalities such as encryption, digital signature, and routing of SOAP messages.
  - Note that SOAP web services may also provide a Web Services Definition Language (WSDL) declaration that specifies how they may be used.

- RESTful: set or principles to build a web service.
  - REST (Representational State Transfer) is not a protocol, but rather a set of principle to build web services that focus on system's resources.
  - RESTful APIs are generally based on HTTP verbs to determine the action:
    - GET : Retrieve a resource on the server, list a collection of records
    - Put : Change the state of a resource, replace or create it if it does not exist.
    - Post :Create a new resource or record
    - Delete : Delete a resource or record

WSDL :

(**W**eb **S**ervices **D**escription **L**anguage)

A web service is characterized by:
1 - Methods : Each reflects a service provided by the server application

2- Protocol that defines :

- The structure of each message used to request a service
- The structure of a message sent by the web service in response
- The transport method used to transmit the messages

WSDL is an XML-based interface description language. The description contains different objects depending on the WSDL version; the objects describe the messages, the services, how they can be requested, how they can be transported.

Elements :

The binding element describes **how** to access the service and defines the message format and protocol details for operations, messages (v. 1.1.) and interfaces (v. 2.0).

```
<wsdl:binding name="HelloServiceSoap11Binding" type="ns:HelloServicePortType">
    <soap:binding transport="http://schemas.xml soap.org/soap/http"
style="document"/>
    < . . [WSDL OPERATIONS] . . />
</wsdl:binding>
```

The PortType (v. 1.1) element defines the web service, the operations a client is allowed to request, the messages passed to each operation and the returned messages.

```
<wsdl:portType name="HelloServicePortType">
    <wsdl:operation name="sayHello">
        <wsdl:input message="ns:sayHelloRequest"/>
        <wsdl:output message="ns:sayHelloResponse“/>
     </wsdl:operation>
</wsdl:portType>
```

The operation object defines the SOAP actions and the encoding of each message, for example, "literal." An operation can be thought of as a method in a traditional programming language.

```
<wsdl:operation name="sayHello">
    <soap:operation soapAction="sayHello" style="document"/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
```

Instead of portType, WSDL v. 2.0 uses interface elements which define a set of operations representing an interaction between the client and the service. Each operation specifies the types of messages that the service can send or receive.
Unlike the old portType, interface elements do not point to messages anymore (it does not exist in v. 2.0). Instead, they point to the schema elements contained within the types element.

The *Message* object (available only in v. 1.1) contains the messages required and returned by any operation. For example, the operation named sayHello requires the message sayHelloRequest and returns the message sayHelloResponse.

Attacking

When dealing with web service security, accessing the WSDL file is the first step; this gives us the full list of operations and types allowed by the server as well as the correct syntax to use, inputs, outputs and all the useful information we may need to run successful attacks.

Getting WSDL file :
1- Below are simple Google dorks you can use to find a WSDL:
filetype:wsdl --> To filter by WSDL file type
site: <yourTarget> --> To filter by site
inurl:wsdl --> To filter all URLs with the string WSDL

Example :
To search all of the indexed WSDL files on the target www.vuln.att, you would use the following search string:
site:www.vuln.att filetype:wsdl

2- Once the SOAP service has been identified, another way to discover WSDL files is by appending ?wsdl,.wsdl or ?disco to the end of the service URL:
http://www.soap.com/index.php?wsdl

3- DISCO is a Microsoft .NET Web Services discovery tool used to discover the URLs of XML Web services located on a Web Server.
The .disco documents are XML files that, similarly to WSDL, describe the Web service.

4- UDDI (Universal Description, Discovery, and Integration) is a directory service used for storing information about web services.
Thanks to UDDI, anyone can search for information about Web Services that are made available by or on behalf of a business.

5- You can use online resources such as soapclient.com to search for public web services.

The attack consists of the following few steps:

1. The attacker starts the client application and gets the WSDL file

2. The attacker analyzes the WSDL file to look for hidden methods and to get general information about the structure of each operation

3. The attacker invokes some (hidden) methods

Mitigation
- By disabling the SOAPAction header.
- By configuring the firewall to inspect the SOAPAction header when filtering the coming requests.

U can do SQL injections with these