



Centro de Investigación
y Desarrollo Ecuador

SEMINARIO INTERNACIONAL DE SEGURIDAD EN EL DESARROLLO DE SOFTWARE

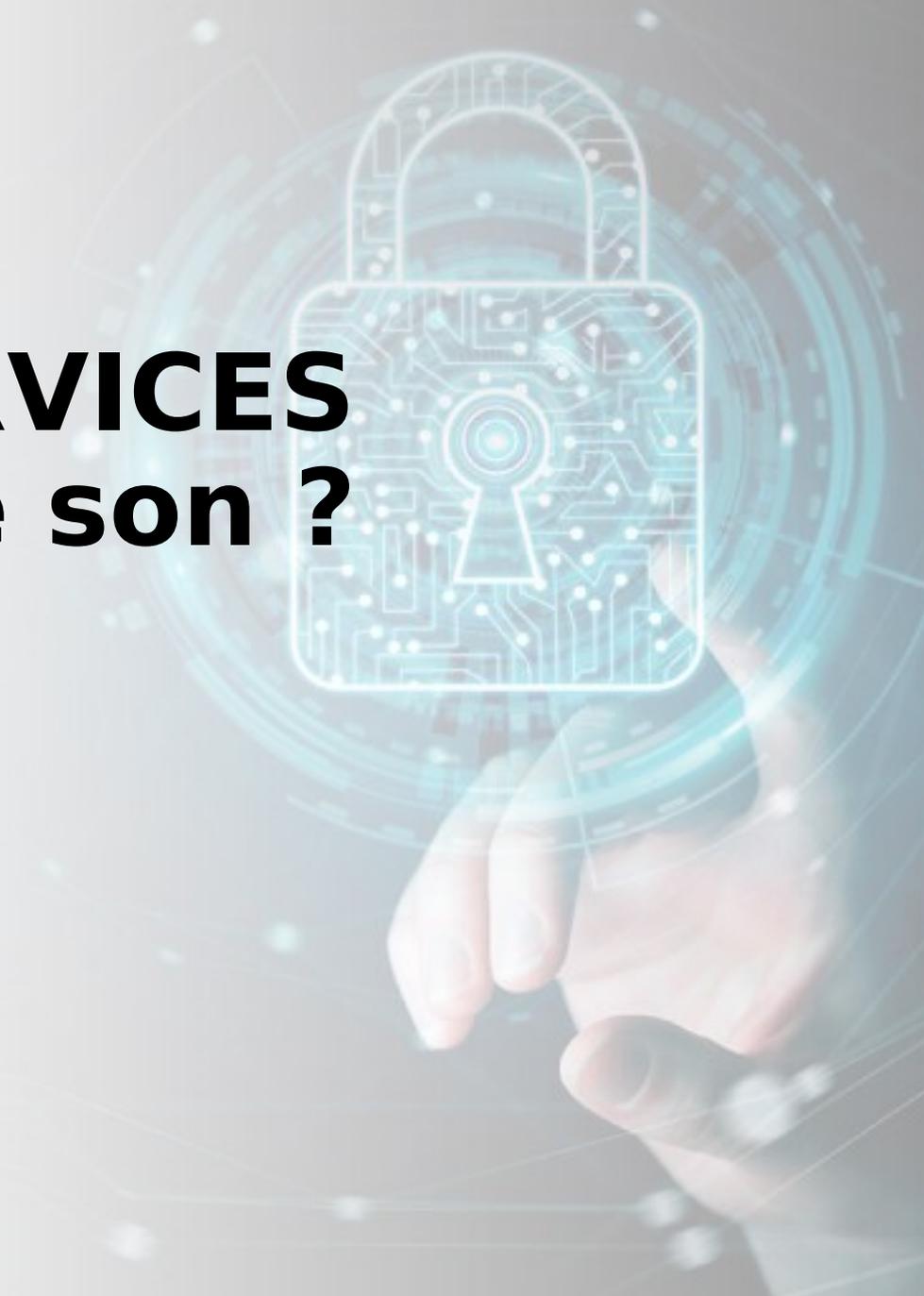


Hacking a Servicios Web

JUAN OLIVA / @jroliva

- Pentester - Security Researcher
- Proyectos de Ethical Hacking en Silcom VoIP & Security Assessment
- Consultor de soluciones de VoIP, enfocadas en seguridad
- Instructor de cursos de Ethical Hacking, Linux y VoIP
- Php y Python lover
- Linux user forever and ever
- Technical writer : Papper “Metasploitable 3 Laboratorios de Práctica”
- Twitter: @jroliva
- Blog: <http://jroliva.wordpress.com/>

WEB SERVICES **Qué son ?**



Web Services - Tipos

- **SOAP**
- **REST**
- **JSON-RPC**
- **XML-RPC**
- **BEPL**
- **WCF**



Web Services - Tipos

- SOAP
- REST

REST vs. SOAP

REST

- Exposes **RESOURCES** which represent **DATA**
- Uses HTTP Verbs (GET/POST/DELETE)
- Emphasis on simple point-to-point communication over HTTP
- Supports multiple data formats
- Emphasizes stateless communication

SOAP

- Exposes **OPERATIONS** which represent **LOGIC**
- Uses HTTP POST
- Emphasis on loosely coupled distributed messaging
- Supports only XML (and attachments)
- Supports stateless and stateful/conversational operations
- Supports asynchronous messaging
- Strong Typing

WEB SERVICES

Cómo funciona ?



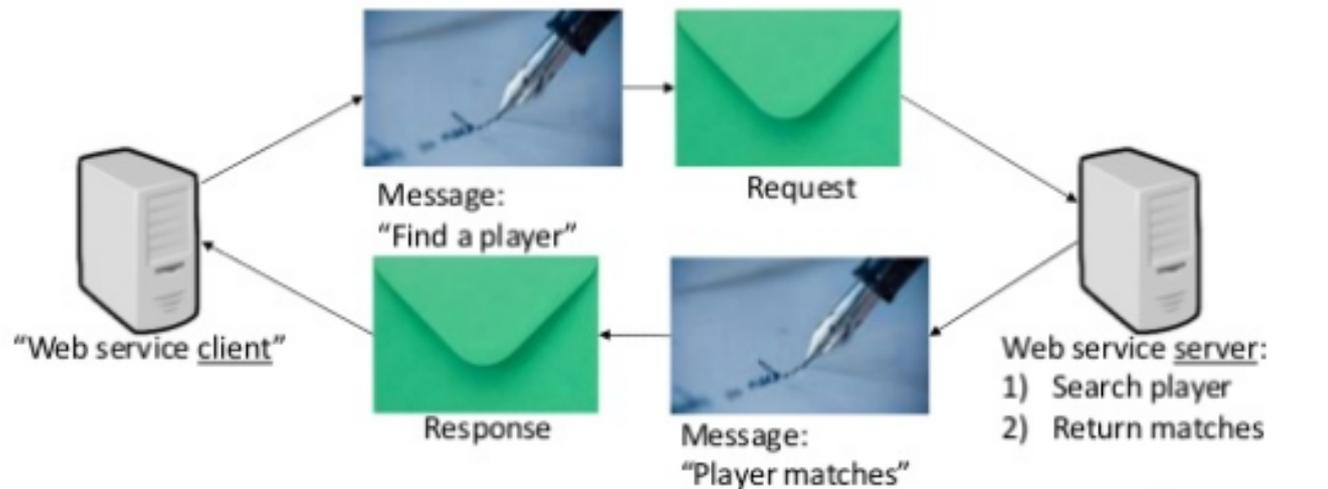
Web Services Funcionamiento SOAP (Simple Object Access Protocol).

- Basado en XML**
- Envío mediante HTTP POST**
- Mensajes SOAP-Request Y SOAP-Response**
- Soporta cualquier protocolo en “texto plano”**
- Trabaja con Protocolo WSDL (Web Service Description Language).**
- WSDL resuelve el descubrimiento de metodos**

Web Services

Ejemplo de funcionamiento

- **Buscar en el Web Service “Buscar id”**
- **El servicio web retorna el registro**



Web Services

Ejemplo de funcionamiento

- **Buscar en el Web Service “Buscar id”**
- **El servicio web retorna el registro**

Enter your Account Number:

SELECT * FROM user_data WHERE userid = 103

USERID	FIRST_NAME	LAST_NAME	CC_NUMBER	CC_TYPE	COOKIE	LOGIN_COUNT
103	Jane	Plane	123456789	MC		0
103	Jane	Plane	333498703333	AMEX		0

Web Services

Ejemplo de funcionamiento

- Internamente : WSDL (XML)

```
127.0.0.1:8080/WebGoat-5.4/services/WsSqlInjection?WSDL
Most Visited Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB Aircrack-ng Nessus / Loading

--<wSDL:definitions targetNamespace="http://127.0.0.1:8080/WebGoat-5.4/services/WsSqlInjection">
  <!--
    WSDL created by Apache Axis version: 1.2
    Built on May 03, 2005 (02:20:24 EDT)
  -->
  <!--
  -->
  <wSDL:types>
    <schema targetNamespace="http://127.0.0.1:8080/WebGoat-5.4/services/WsSqlInjection">
      <import namespace="http://schemas.xmlsoap.org/soap/encoding/">
        <complexType name="ArrayOf_xsd_string">
          <complexContent>
            <restriction base="soapenc:Array">
              <attribute ref="soapenc:arrayType" wsdl:arrayType="xsd:string[]"/>
            </restriction>
          </complexContent>
        </complexType>
      </schema>
    </wSDL:types>
    <wSDL:message name="getCreditCardRequest">
      <wSDL:part name="id" type="xsd:string"/>
    </wSDL:message>
    <wSDL:message name="getCreditCardResponse">
      <wSDL:part name="getCreditCardReturn" type="impl:ArrayOf_xsd_string"/>
    </wSDL:message>
    <wSDL:portType name="WsSqlInjection">
      <wSDL:operation name="getCreditCard" parameterOrder="id">
        <wSDL:input message="impl:getCreditCardRequest" name="getCreditCardRequest"/>
        <wSDL:output message="impl:getCreditCardResponse" name="getCreditCardResponse"/>
      </wSDL:operation>
    </wSDL:portType>
    <wSDL:binding name="WsSqlInjectionSoapBinding" type="impl:WsSqlInjection">
      <wSDL:soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
      <wSDL:operation name="getCreditCard">
        <wSDL:soap:operation soapAction="">
          <wSDL:input name="getCreditCardRequest">
            <wSDL:soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="http://lessons.webgoat.owasp.org" use="encoded"/>
          </wSDL:input>
          <wSDL:output name="getCreditCardResponse">
            <wSDL:soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="http://127.0.0.1:8080/WebGoat-5.4/services/WsSqlInjection" use="encoded"/>
          </wSDL:output>
        </wSDL:operation>
      </wSDL:binding>
    <wSDL:service name="WsSqlInjectionService">
      <wSDL:port binding="impl:WsSqlInjectionSoapBinding" name="WsSqlInjection">
```

Web Services

Ejemplo de funcionamiento

- Web Service “Update”
- El servicio web retorna el estado del update



WEB SERVICES

Superficie y seguridad ?

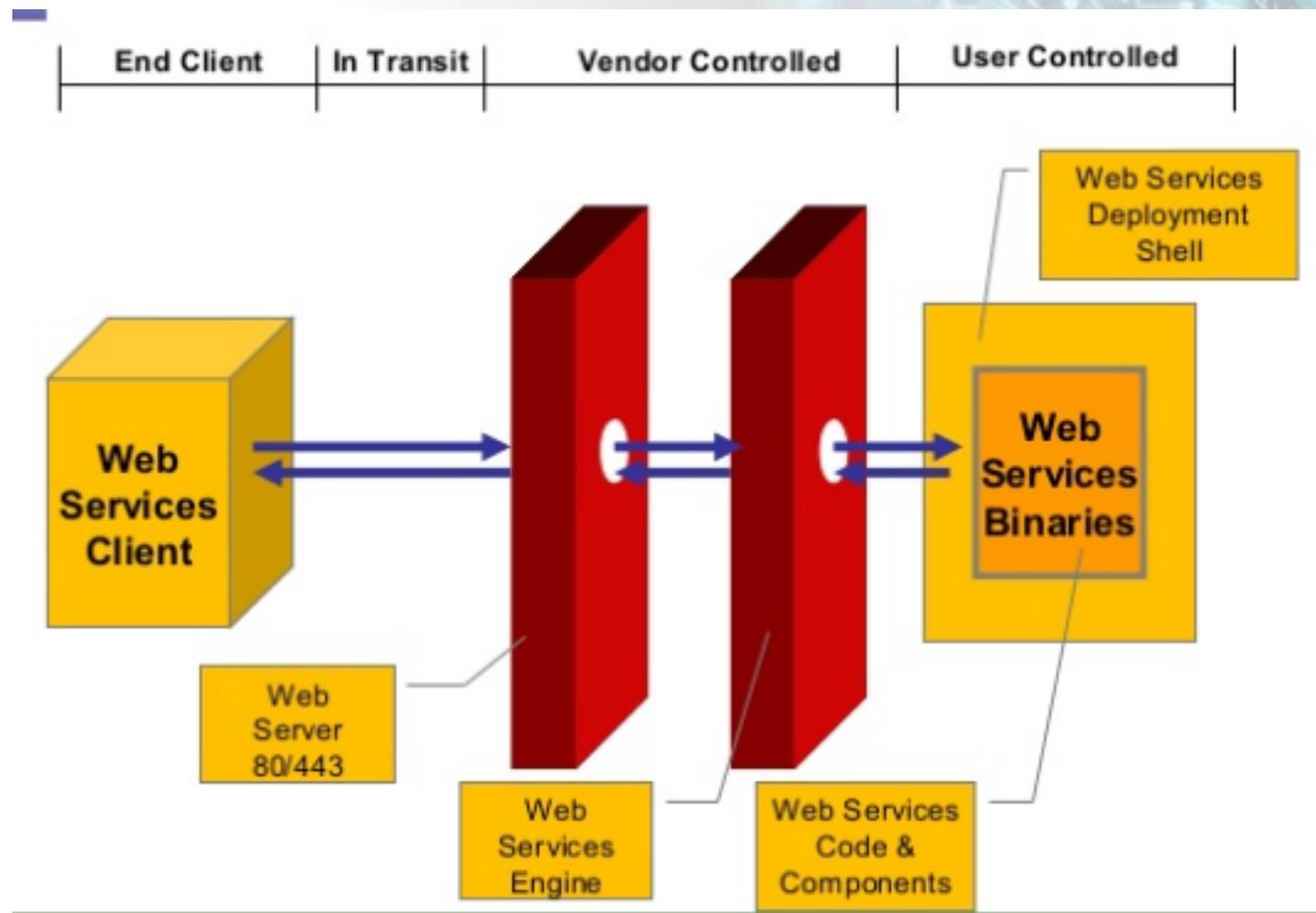


Web Services Superficie y Seguridad

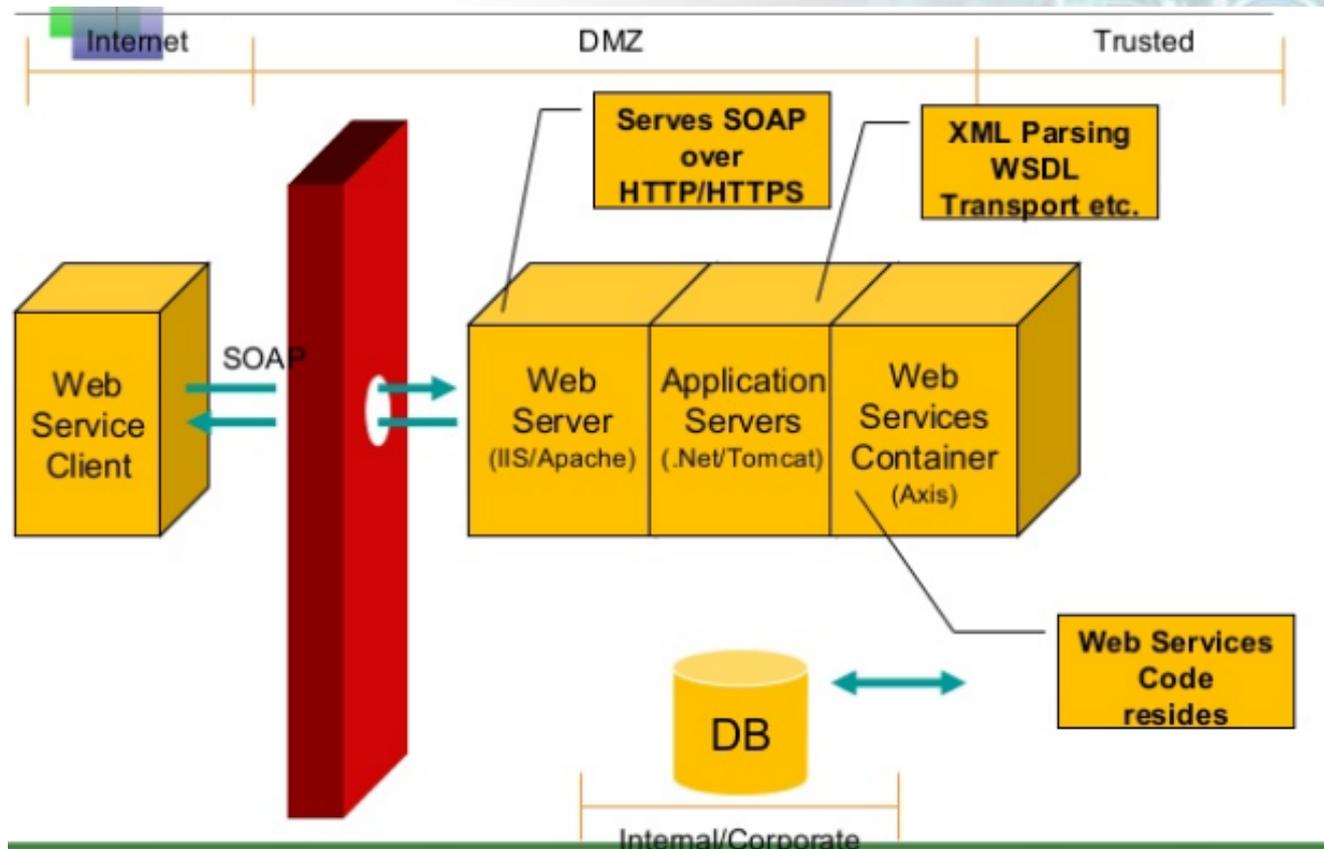
- **Se ha convertido en un nuevo punto de ataque**
- **Vienen apareciendo nuevas herramientas y formas de explotación**
- **Muchos protocolos causan confusión**
- **Malas implementaciones por falta de tiempo**
- **Los ataques estén creciendo ante el uso masivo de estos.**



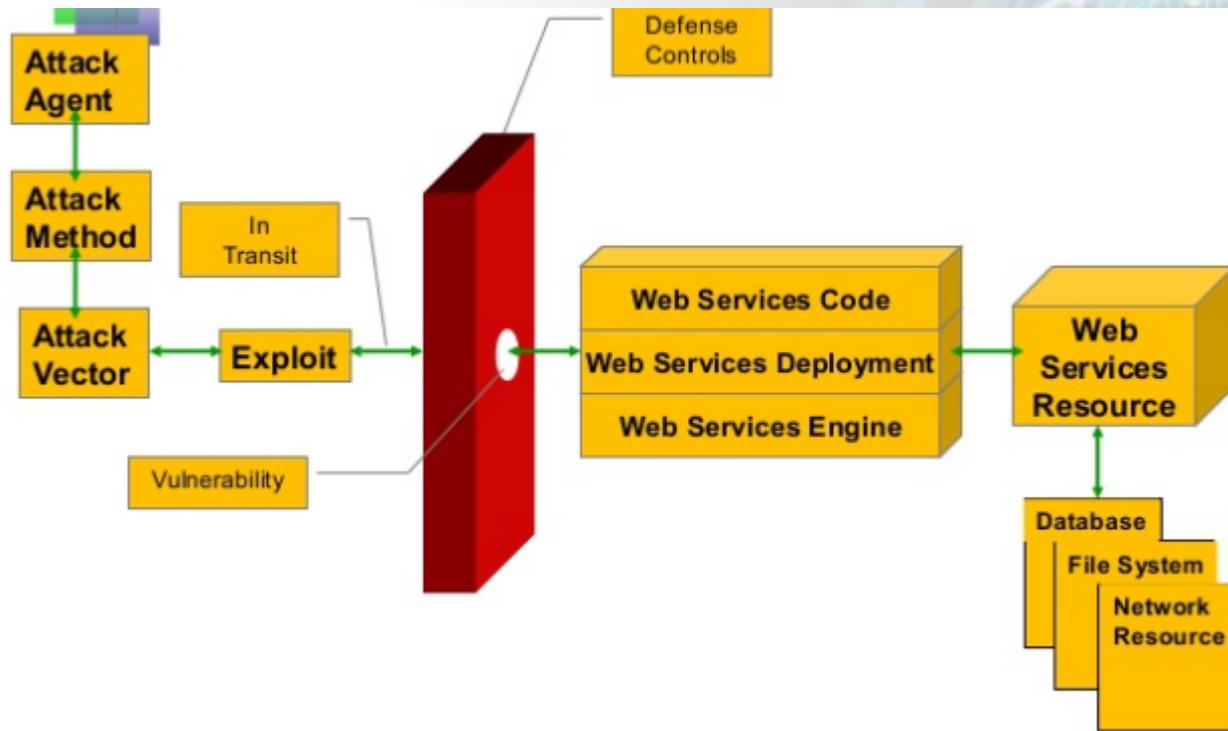
Web Services Superficie y Seguridad



Web Services Superficie de Seguridad



Web Services Superficie y Seguridad

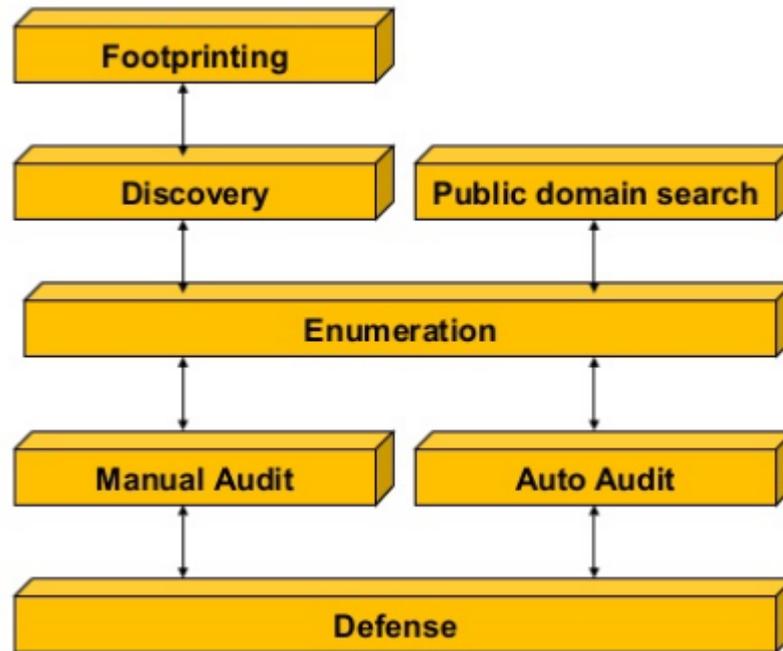


WEB SERVICES

**Cómo hacemos
Ethical Hacking ??**



Web Services Metodología de Evaluación



Web Services Footprinting - Quien dijo Google Dorks!!

- **inurl:"asmx?wsdl"**
- **inurl:"jws?wsdl"**
- **ext:wsdl**
- **ext:asmx**
- **ext:jws**

The screenshot shows a Google search interface with the query "inurl:asmx?wsdl" entered in the search bar. The search results are displayed below the search bar, showing a list of search results with titles and URLs. The results include:

- country.wsdl**
www.webservicex.com/country.asmx?wsdl
- Service Description**
www.dneonline.com/calculator.asmx?WSDL
- the w3schools.com WSDL**
https://www.w3schools.com/xml/tempconvert.asmx?wsdl
- stock WSDL**
ws.cdyne.com/delayedstockquote/delayedstockquote.asmx?wsdl
- WeatherForecast Web service - WebserviceX.NET**
www.webservicex.net/WeatherForecast.asmx?WSDL
- Service Description - Aonaware Web Services**
services.aonaware.com/DictService/DictService.asmx?WSDL
- TaarifCustoms.asmx?WSDL - IIS Windows Server**
62.219.95.10/TaarifWebService/TaarifCustoms.asmx?WSDL
- asp.net - asmx wsdl loading forever - Stack Overflow**
stackoverflow.com/questions/351078/asmx-wsdl-loading-forever
- c# - Azure API Management and ASMX/WSDL SOAP endpoint ...**
stackoverflow.com/.../azure-api-management-and-asmx-wsdl-soa...
- Service Description - AFIP**
https://servicios1.afip.gov.ar/wsfev1/service.asmx?WSDL

At the bottom of the page, there is a "Go" button and a page number "1" with a "Siguiente" link.

Web Services Scanning - descubrimiento y enumeración de metodos

- WebScarab
- Burp
- WSSAT
- ext:asmx
- ext:jws

Burp Intruder Repeater Window Help
 Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Options Alerts WSDLer
 1 x 2 x ...
 Target Positions Payloads Options

? **Payload Positions**

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

Attack type:

```

POST /InstantOrder.asmx HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: text/xml; charset=UTF-8
SOAPAction: http://schemas.microsoft.com/samples/sqlserver/storefront/2005/3/OrderItem
Host: 10.2.9.105
Content-Length: 778

<soapenv:Envelope xmlns:soapenv="$http://schemas.xmlsoap.org/soap/envelope/$"
xmlns:ns="$http://schemas.microsoft.com/samples/sqlserver/storefront/2005/3/$">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:OrderItem>
      <ns:userName>$gero et$</ns:userName>
      <ns:password>$sonoras imperio$</ns:password>
      <ns:productID>$3$</ns:productID>
      <ns:quantity>$3$</ns:quantity>
      <ns:addressLine1>$quae divum incedo$</ns:addressLine1>
      <ns:addressLine2>$verrantque per auras$</ns:addressLine2>
      <ns:addressCity>$per auras$</ns:addressCity>
      <ns:addressStateProvinceID>$3$</ns:addressStateProvinceID>
      <ns:addressPostalCode>$circum claustra$</ns:addressPostalCode>
    </ns:OrderItem>
  </soapenv:Body>
</soapenv:Envelope>
  
```

0 matches

11 payload positions Length: 1032

Web Services Vulnerabilidades y explotación

- **Spoofing** , hacerse pasar por un servidor o cliente
- **Tampering**, Ponerse en el medio de la comunicación
- **Injection Attacks**, inyecciones de código
- **Information Disclosure**, divulgación de paths, networks
- **DOS**, eso mismo :D
- **XML Reference Attack**
- **XML POISONING**

No es lo mismo que vulnerabilidades web tradicionales ??

Hacking Web Services Ejercicio

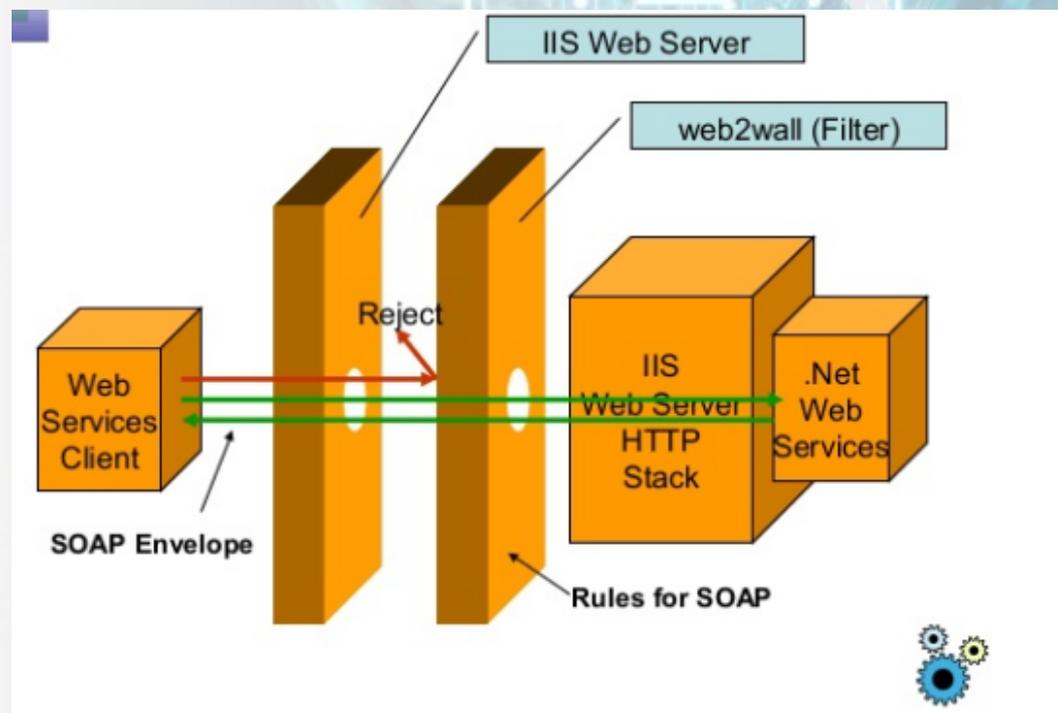


WEB SERVICES COMO HACEMOS PARA ASEGURAR ??

A hand is shown interacting with a futuristic digital interface. The interface features a glowing blue padlock icon with circuit patterns inside it, set against a background of concentric circles and light trails. The overall aesthetic is high-tech and digital.

Web Services Aseguramiento - Hardening

- SOAP sobre HTTPS
- Niveles de filtros y monitoreo activo
- Filtro de contenido a nivel de SOAP
- WSDL sobre HTTPS
- Sobre proveer los métodos necesarios
- WSDL ACL
- Uso de credenciales sobre web services
- Accesos a SOAP y XML filtrados x origen
- Control de excepciones, validación de entrada anti-sqli



Gracias !!!

Juan Oliva

Consultor en Ethical Hacking

joliva@silcom.com.pe

Hangouts : jroliva@gmail.com

Twitter : [@jroliva](https://twitter.com/jroliva)

Blog : <http://jroliva.net/>



Centro de Investigación
y Desarrollo Ecuador



SEMINARIO INTERNACIONAL DE SEGURIDAD EN EL DESARROLLO DE SOFTWARE

WWW.CIDECUADOR.COM

Una vez finalizado el evento esta presentación
será publicada en su respectiva página web