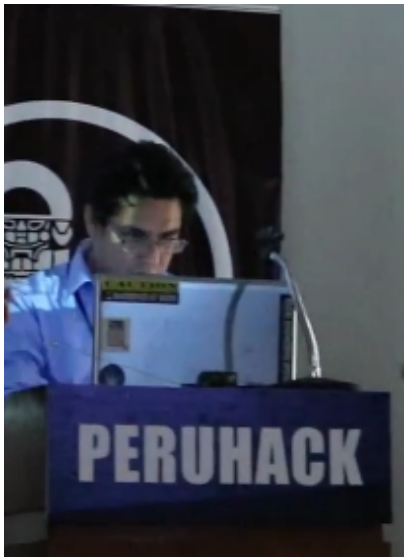
TECNOLÓGICO SUPERIOR CORDILLERA

25 AÑOS

CIDE
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 – Definición

- Es software
- Sirve para intercambiar información
- Diferentes arquitecturas y lenguajes
- Usa Protocolos y estándares

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

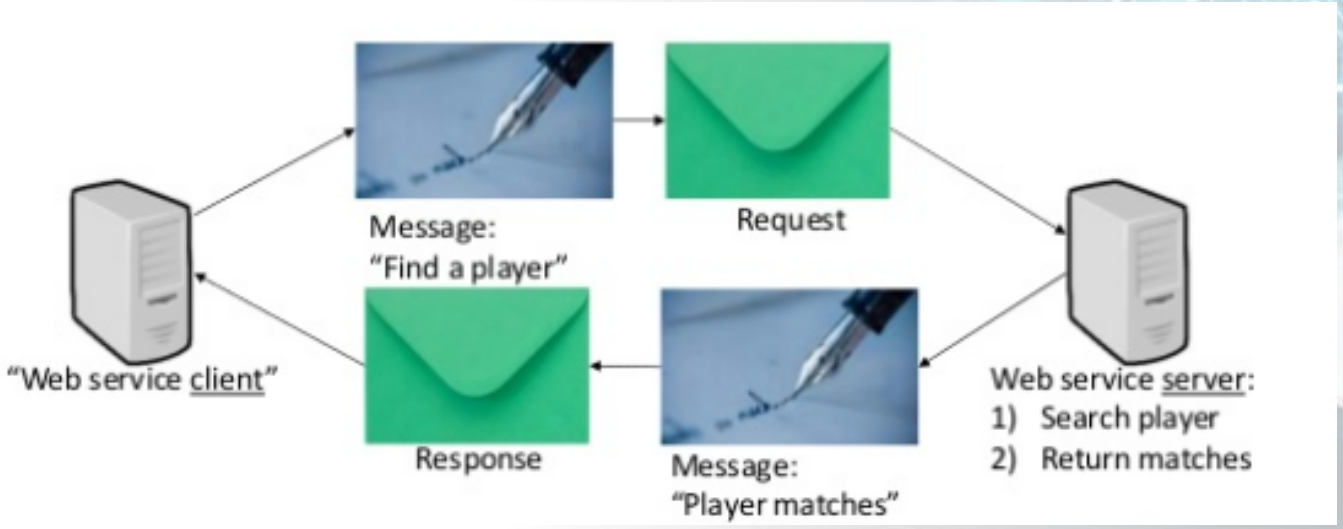
WEB SERVICES
Cómo funciona ?

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

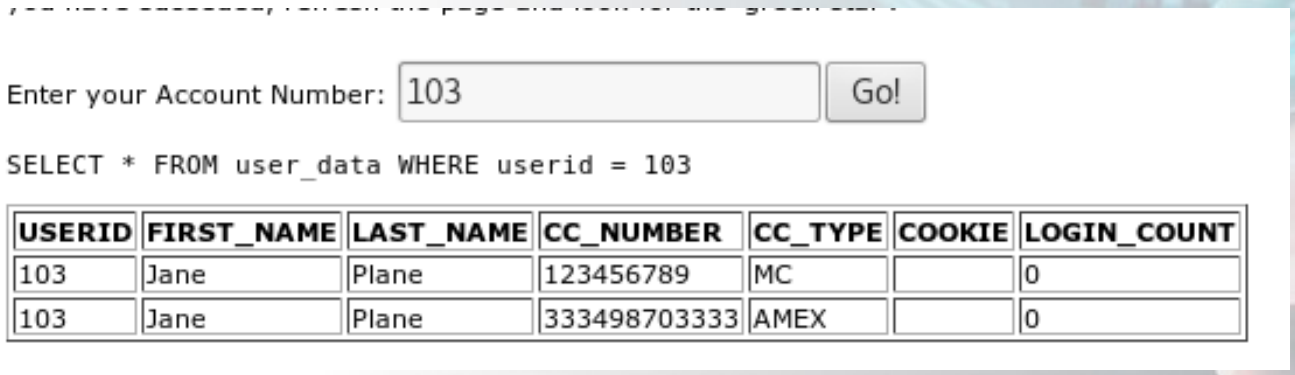
Web Services
Ejemplo de funcionamiento

- **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**



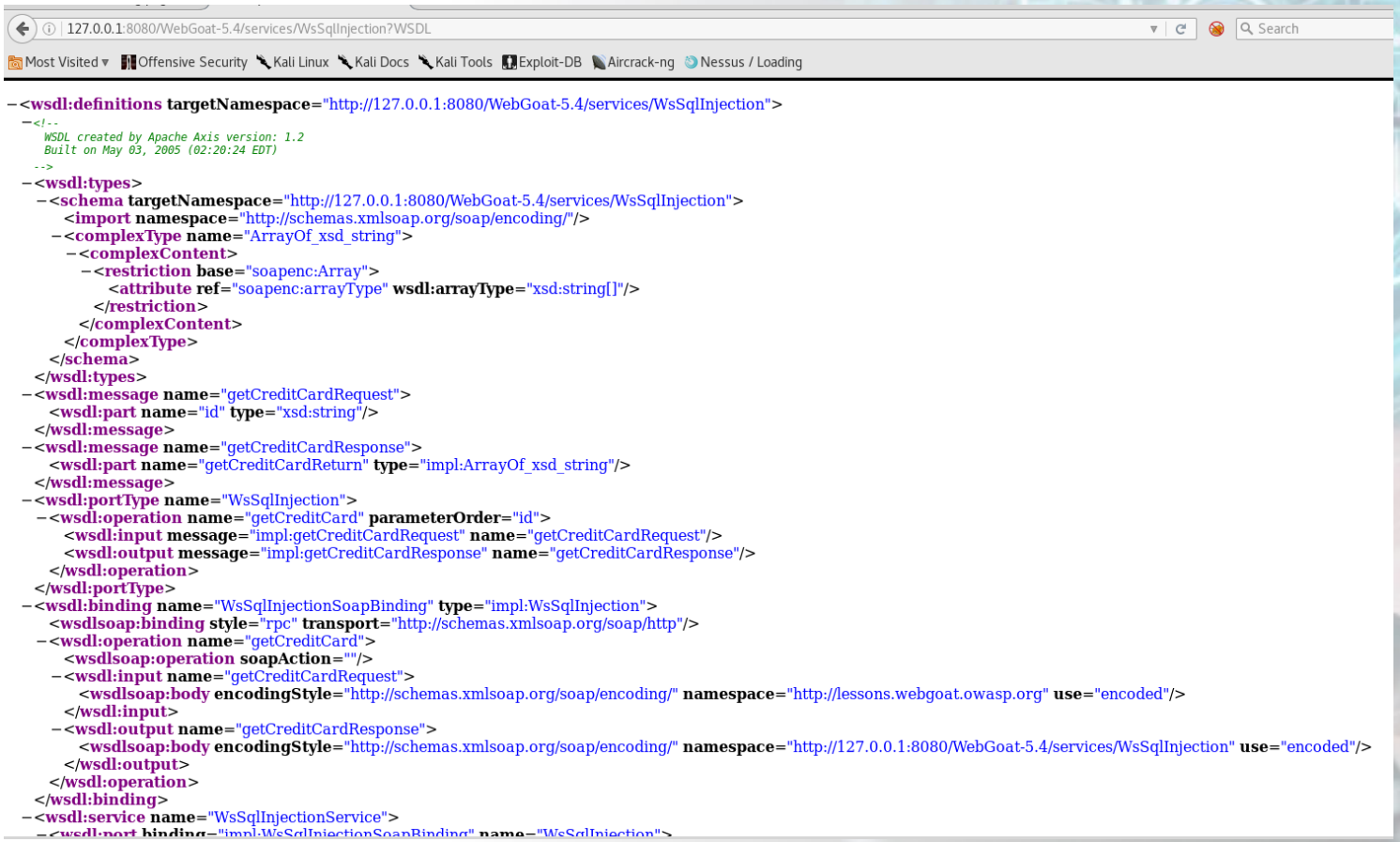
Enter your Account Number: 103 Go!

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)




```
−<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">
     <wsdlsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
    −<wsdl:operation name="getCreditCard">
       <wsdlsoap:operation soapAction=""/>
      −<wsdl:input name="getCreditCardRequest">
         <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="http://lessons.webgoat.owasp.org" use="encoded"/>
       </wsdl:input>
      −<wsdl:output name="getCreditCardResponse">
         <wsdlsoap: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">
```

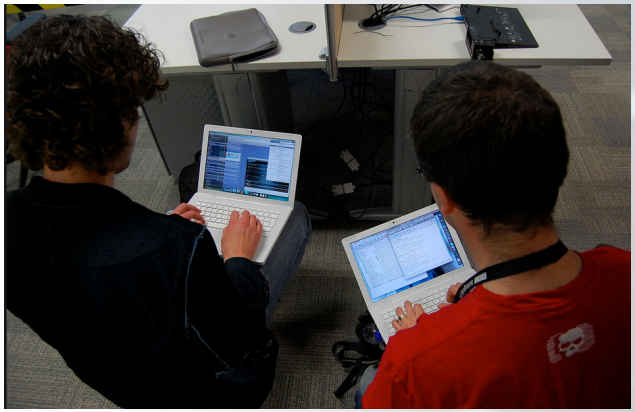
Web Services
Ejemplo de funcionamiento

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

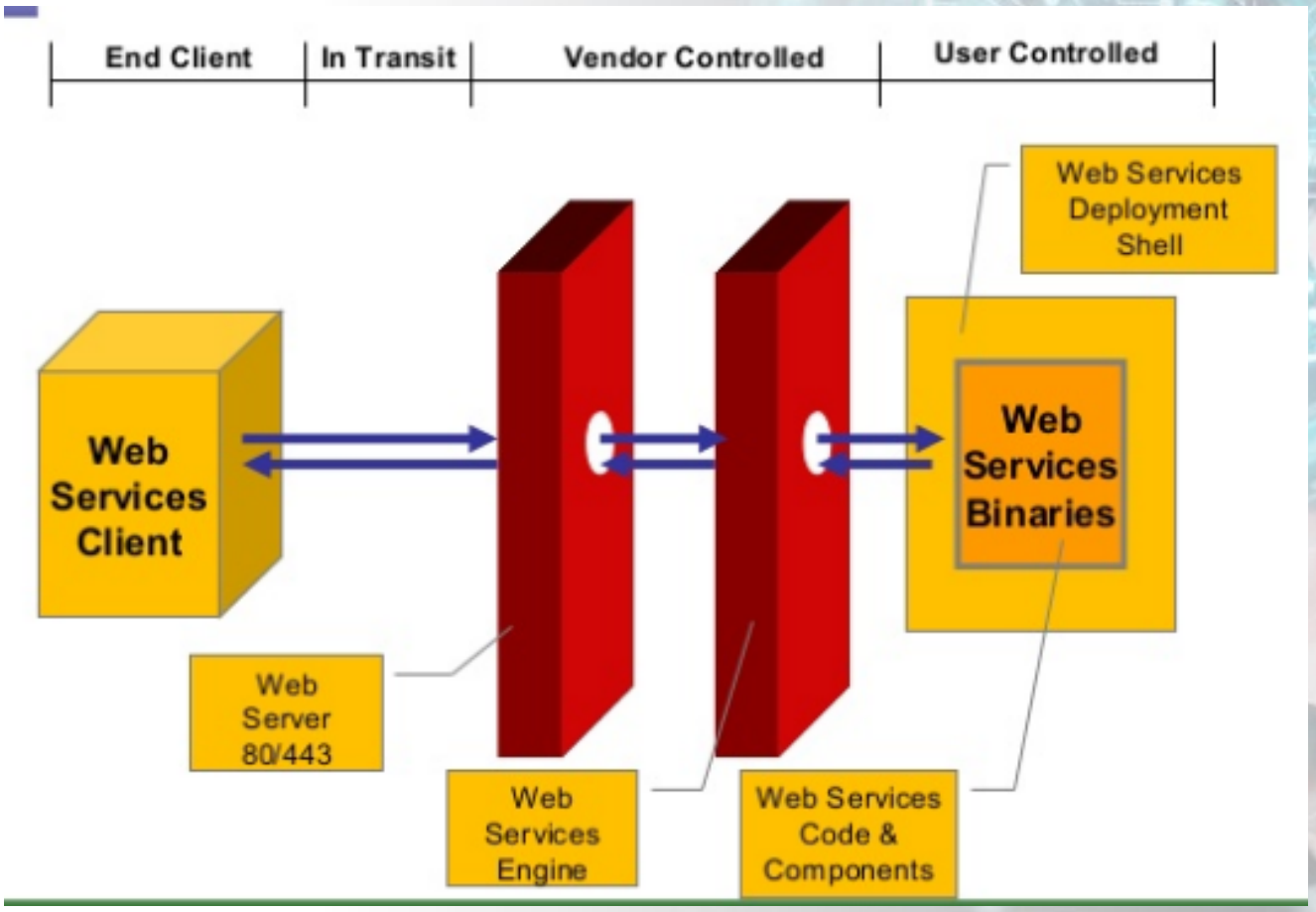
WEB SERVICES
Superficie y seguridad ?

Web Services
Superficie y Seguridad

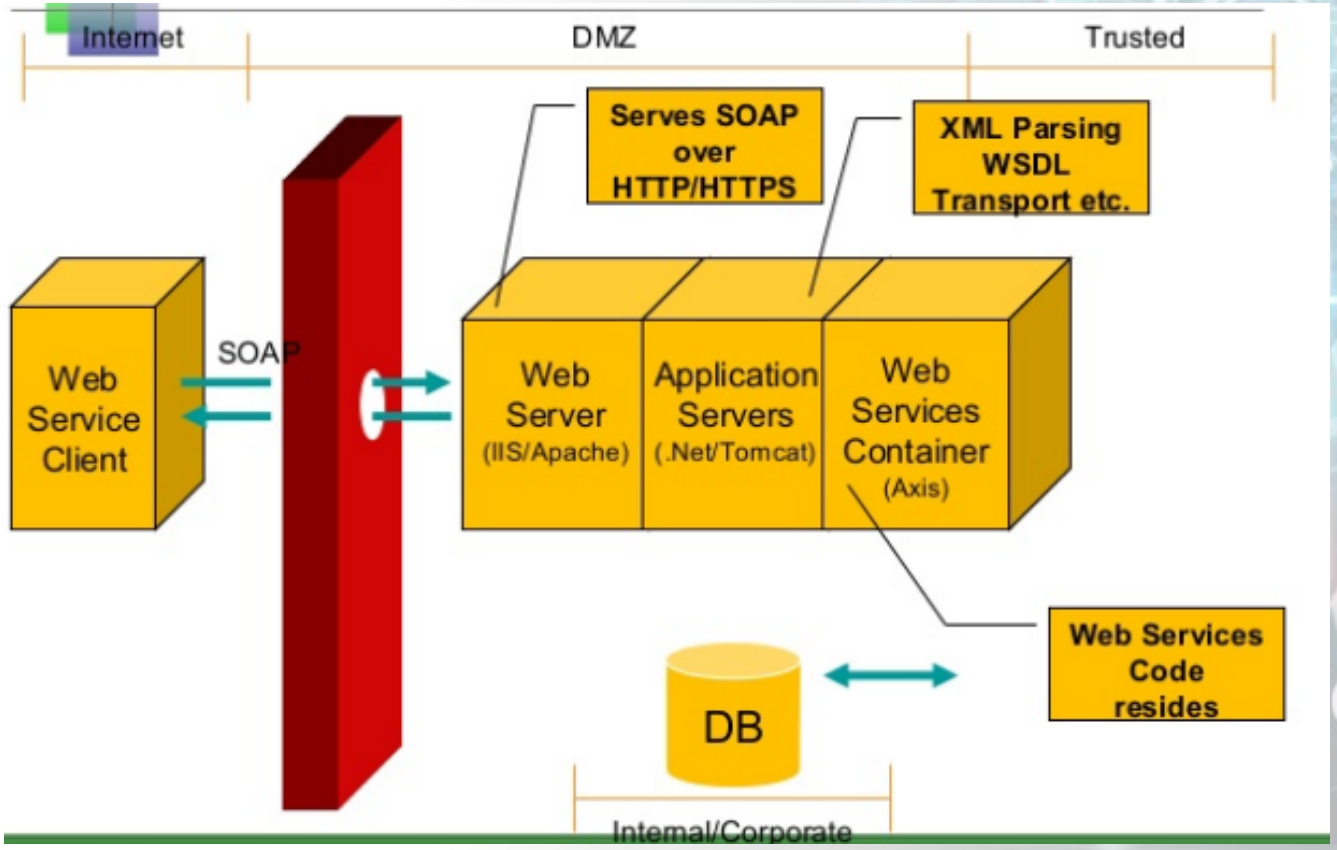
- Se ha convertido en un nuevo punto de ataque
- Vienen apareciendo nuevas herramientas y formas de explotación
- Muchos protocolos causan confusión
- Malas implementaciones por falta de tiempo
- Los ataques estén creciendo ante el uso masivo de estos.



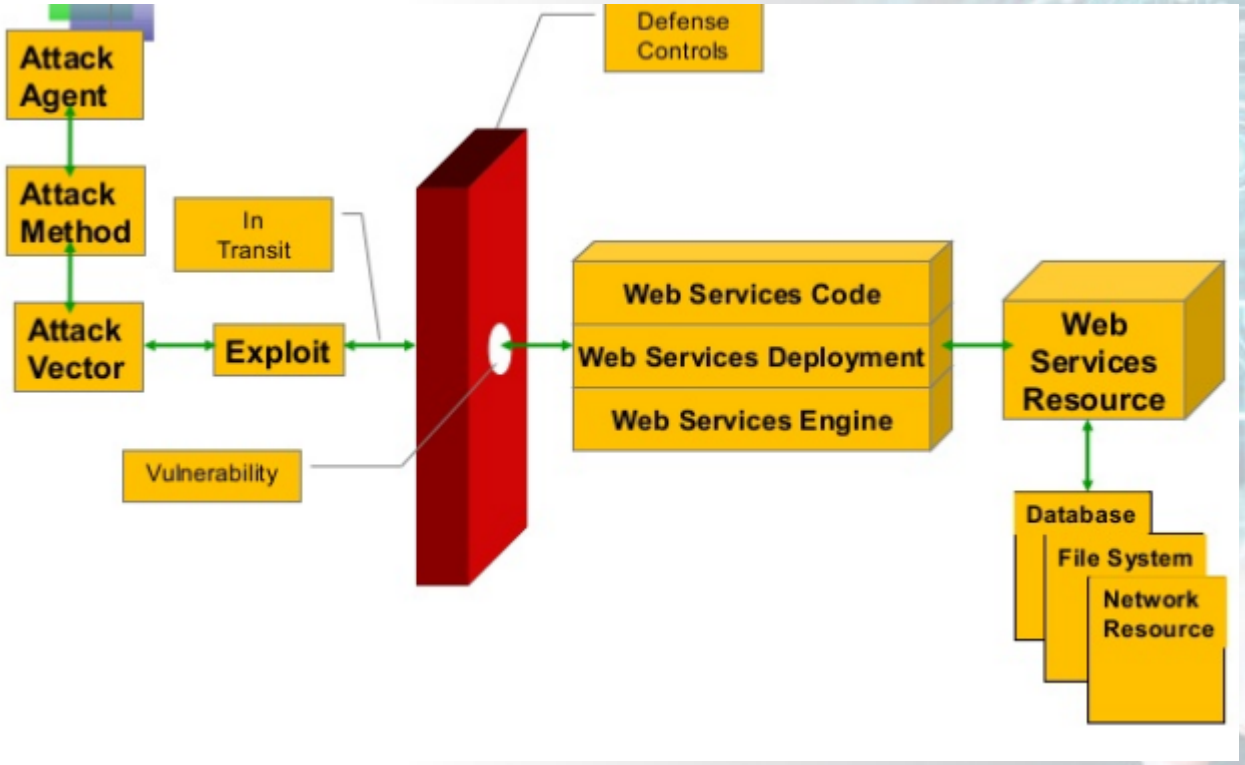
Web Services
Superficie y Seguridad



Web Services
Superficie de Seguridad



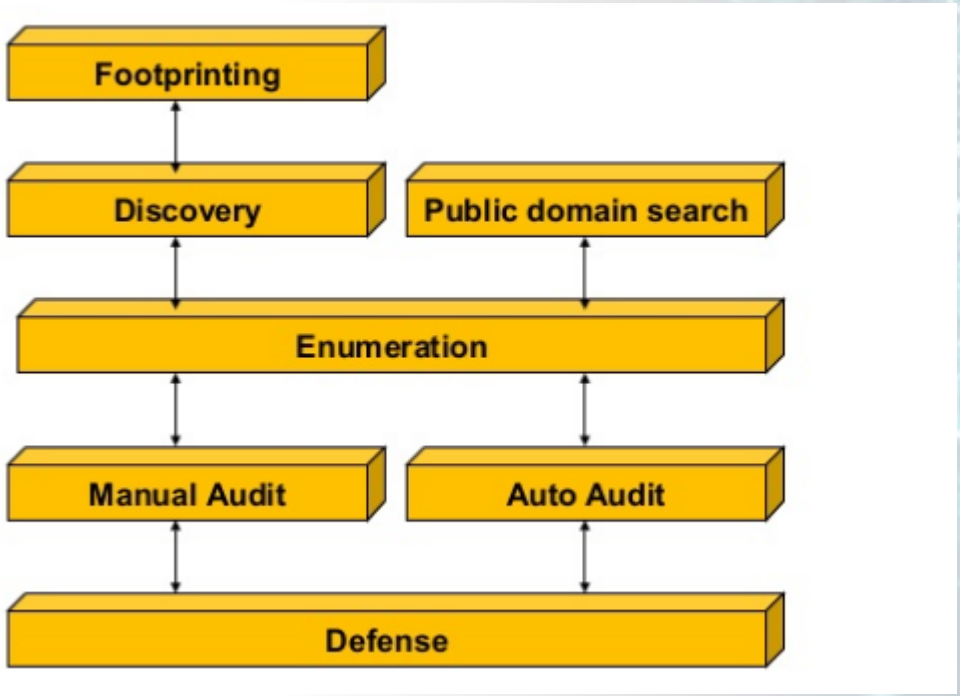
Web Services
Superficie y Seguridad



WEB SERVICES

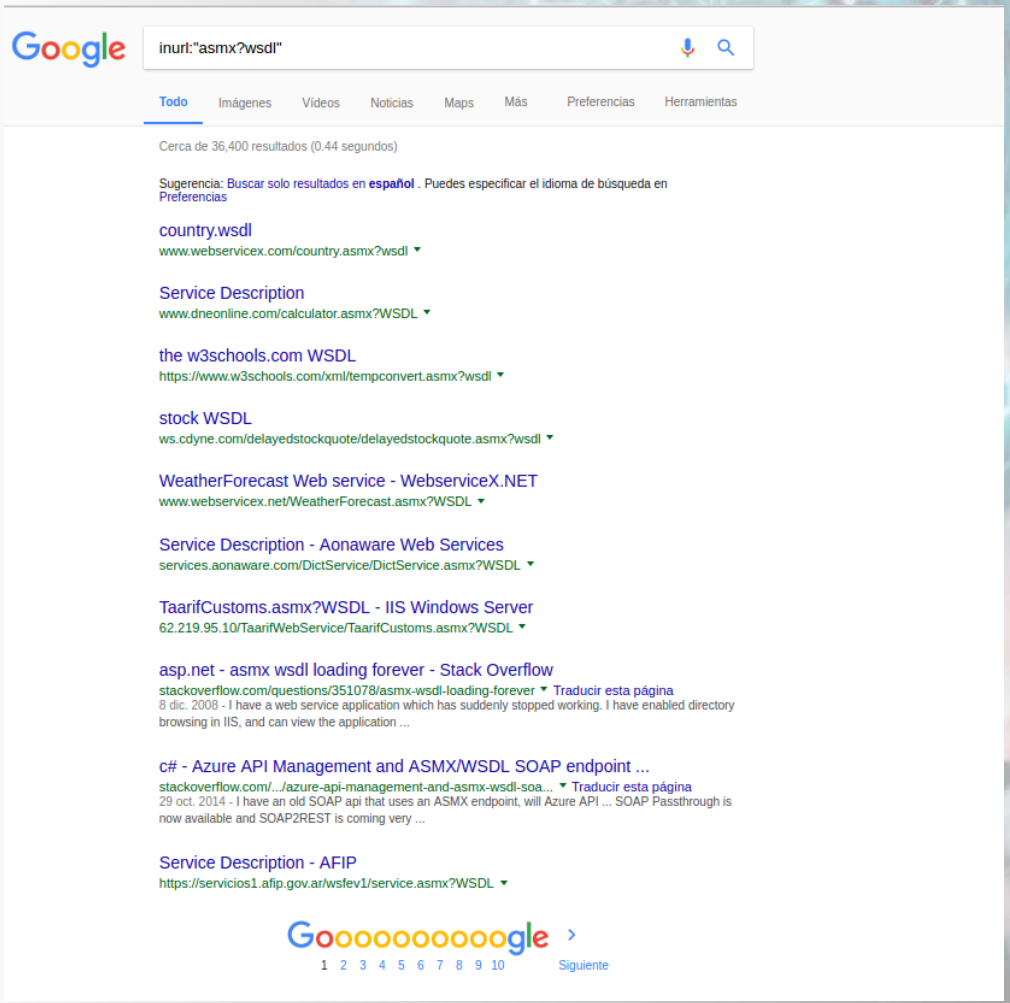
Cómo hacemos Ethical Hacking ??

Web Services
Metodología de Evaluación



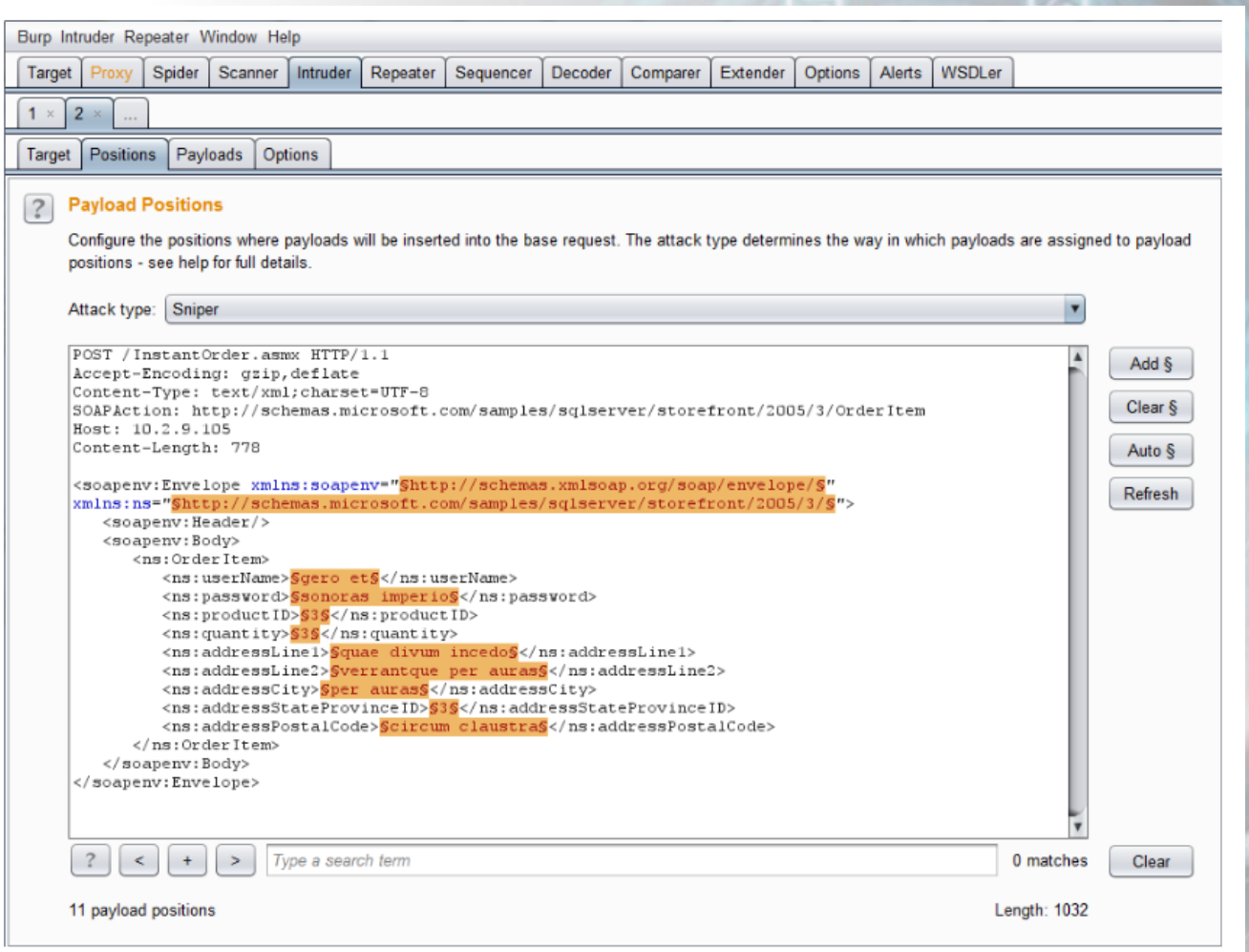
Web Services
Footprinting – Quien dijo Google Dorks!!

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



Web Services
Scaning – descubrimiento y enumeración de metodos

- **Webscarab**
- **Burp**
- **WSSAT**
- **ext:asmx**
- **ext:jws**

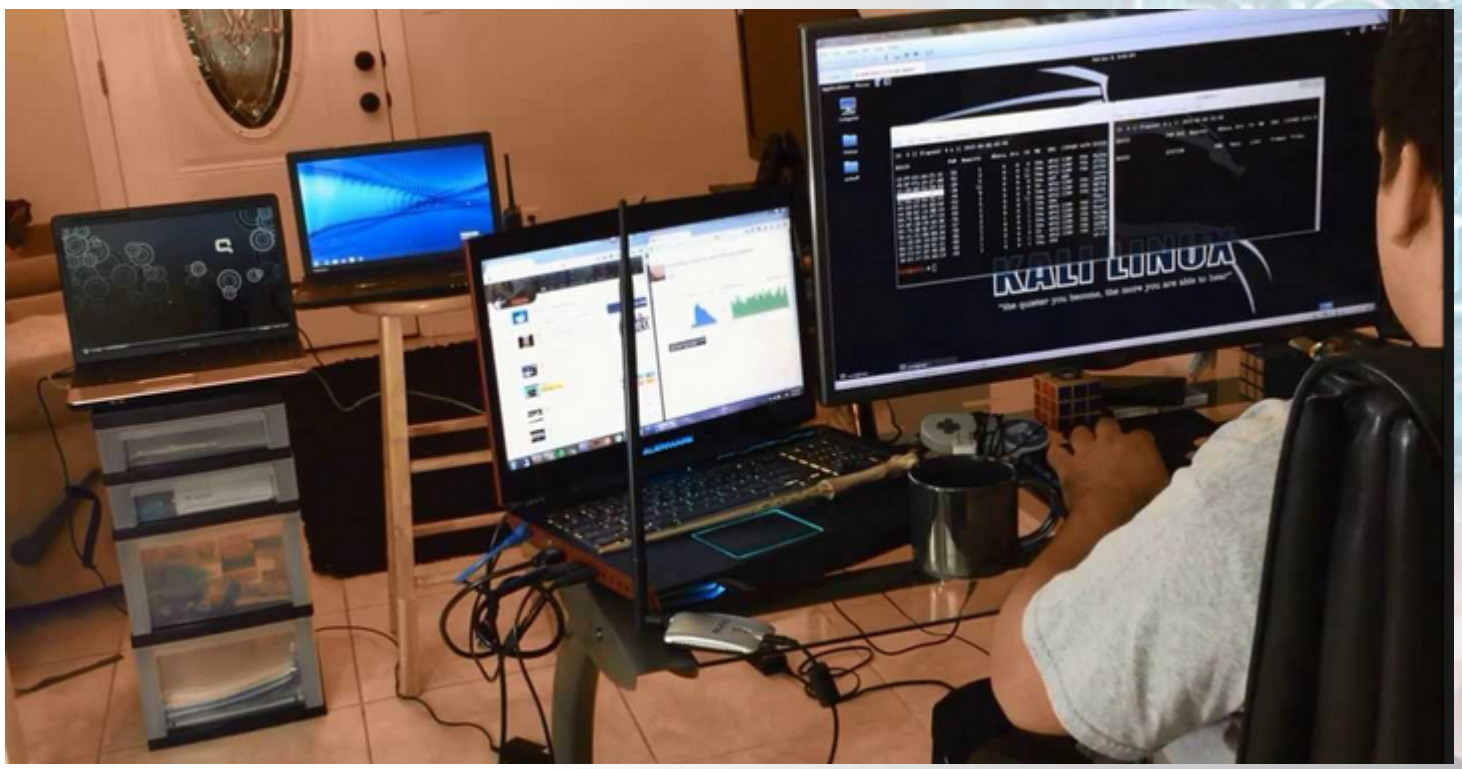


Web Services
Vulnerabilidades y explotación

- Spoofing , hacerse pasar por un servidor o cliente
- Tampering, Ponerse en el medio de la comunicación
- Inyection Attacks, inyeciones de codigo
- 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

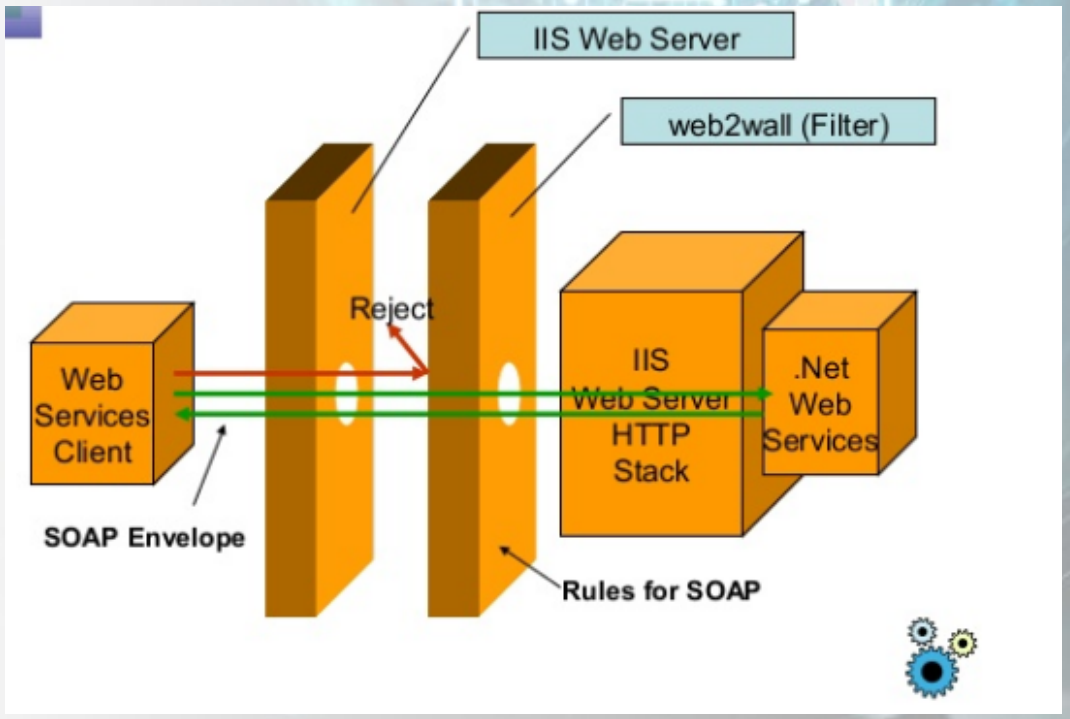


WEB SERVICES

COMO HACEMOS PARA ASEGURAR ??

Web Services
Aseguramiento - Hardering

- 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
Twiter : @jroliva
Blog : http://jroliva.net/

TECNOLÓGICO SUPERIOR CORDILLERA

25 años

CIDE
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 sera publicada en su respectiva página web