Real World Web Service Testing For Web Hackers

black hat®
BRIEFINGS & TRAINING

USA + 2011
EMBEDDING SECURITY

TOM ESTON

- Senior Security Consultant – SecureState
- Web Application / Network Penetration Tester
- Founder of SocialMediaSecurity.com
- Previous Security Research includes:
  - Hacking Social Networks, Privacy Issues and Social Media
- Co-Host of Security Justice and Social Media Security Podcasts
- Twitter: @agent0x0

JOSHUA “JABRA” ABRAHAM

- Senior Security Consultant / Researcher – Rapid7
- Web Application / Network Penetration Tester
- Founder / Contributor to Open Source projects
  - Fierce v2, Nikto, BeEF, Metasploit
- Previous Security Research includes:
  - Hacking SAP BusinessObjects, Browser Privacy
- Codes in Perl!
- Twitter: @Jabra

KEVIN JOHNSON

- Senior Security Consultant – Secure Ideas
- Web Application/Network Penetration Tester
- Founder of various Open Source projects
  - SamuraiWTF, Laudanum, WeaponizedFlash, Yokoso!, BASE, SecTools
- Author of SANS SEC542, SEC642 and SEC571
  - Web Penetration Testing/Advanced Web PenTest/Mobile Security
- Senior SANS Instructor and Internet Storm Center Handler
- Founder PenTesterScripting.com
- Twitter: @secureideas

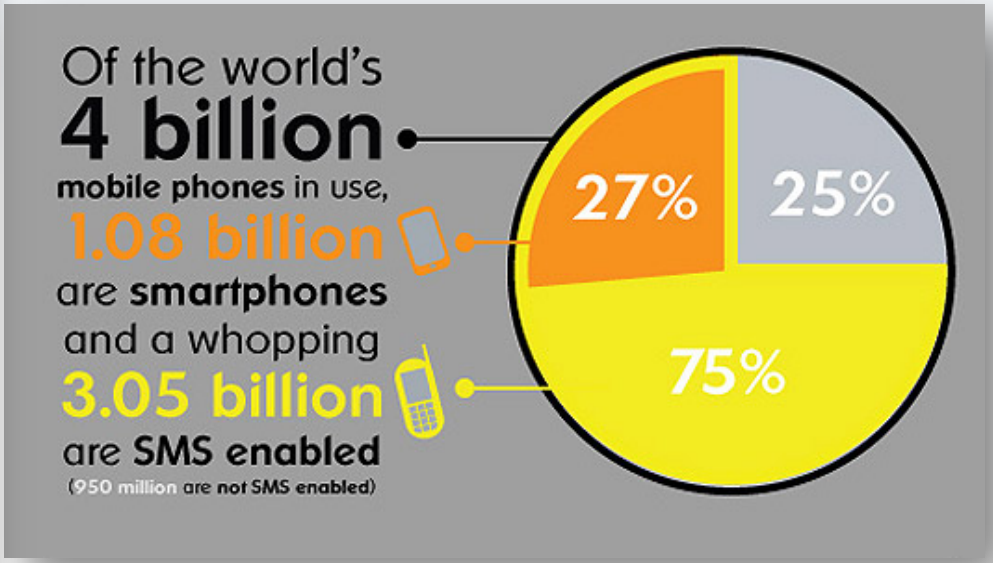
AGENDA

- State of the Union for Web Services Testing
- New Web Services threats and risks we need to address
- Process Improvements Needed
  - Methodology, testing techniques
  - Tools and Lab Environments for Testing

WHY ATTACK WEB SERVICES?

- Secondary attack vector
- Ability to bypass controls in the application
- Many developers don't implement proper security controls
- Installed outside the protections within the web application
- Assumed that the only client for a web service is another application
  - **You know what happens when we assume right?**

RECENT STATISTICS



» Statistics are from Microsoft Tag

WEB SERVICES STATE OF THE UNION

- There are issues with:
  - Scoping
  - Tools
  - Testing Process
  - Methodology
  - Testing Techniques
  - Education
  - Testing Environments
- Basically, it's all broken…

PENETRATION TESTERS DON'T KNOW WHAT TO DO WITH WEB SERVICES

- How do you scope?
  - Do you even ask the right scoping questions?
- Where do you begin?
- How do I test this thing?
  - Automated vs. Manual Testing
  - Black vs. Grey vs. White Box Testing

WHY IS THE TESTING METHODOLOGY BROKEN?

» OWASP Web Service Testing Guide v3

- It's good for web application testing "in general"
- It's the "gold standard"
- It's outdated in regards to web service testing
- Missing full coverage based on a complete threat model
  - Examples: MiTM, Client-side storage, host based authentication
- Testing focused on old technology
  - Example: No mention of WCF services, how to test multiple protocols
- Most testing uses standard Grey Box techniques, fails to address unique web service requirements

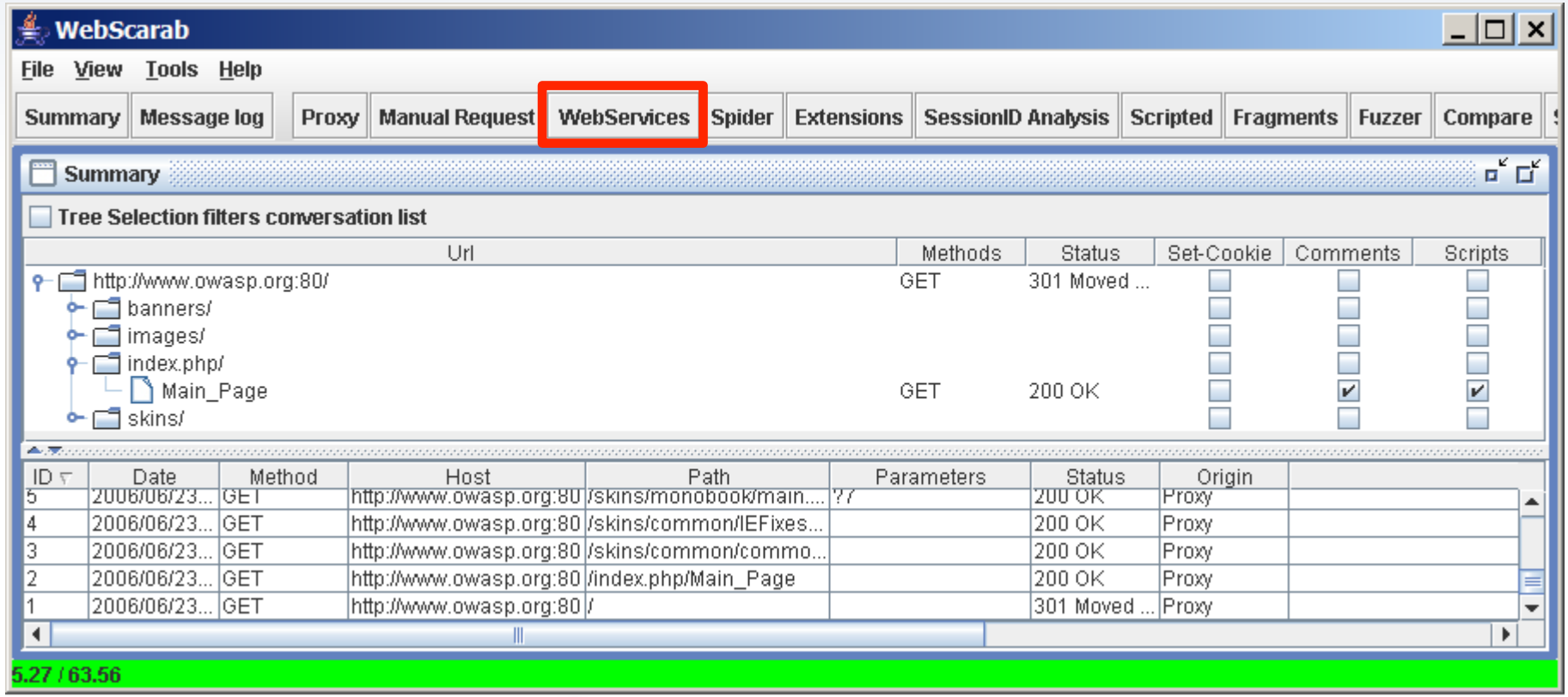
CURRENT TOOLS

- They SUCK ☺
- Mostly commercial tools (for developers, very little security focus)
  - soapUI, WCF Storm, SOA Cleaner
- Very little automation
  - Tester's time is spent configuring tools and getting them running, less hacking!
  - Minimal amount of re-usability
- Multiple tools built from the ground up
  - Missing features
  - Missing functionality (payloads)
  - Community support?

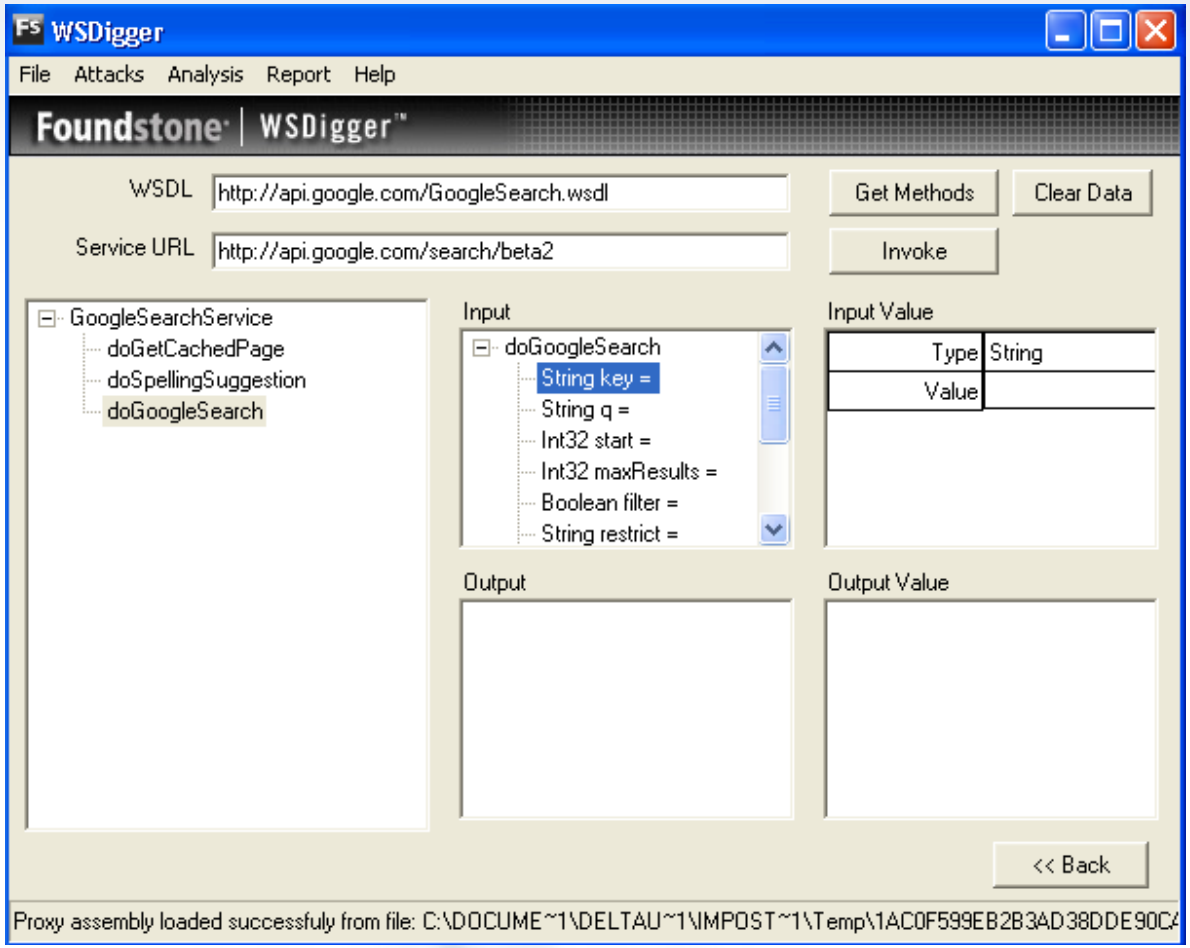
CURRENT TOOLS

- What happened to WebScarab?
- WS-Digger? No SSL?
- There are other tools but many are hard to configure or just don't work properly
- SOAP Messages written by-hand (THIS REALLY SUCKS!)
  - ~14 modules in Metasploit for web services

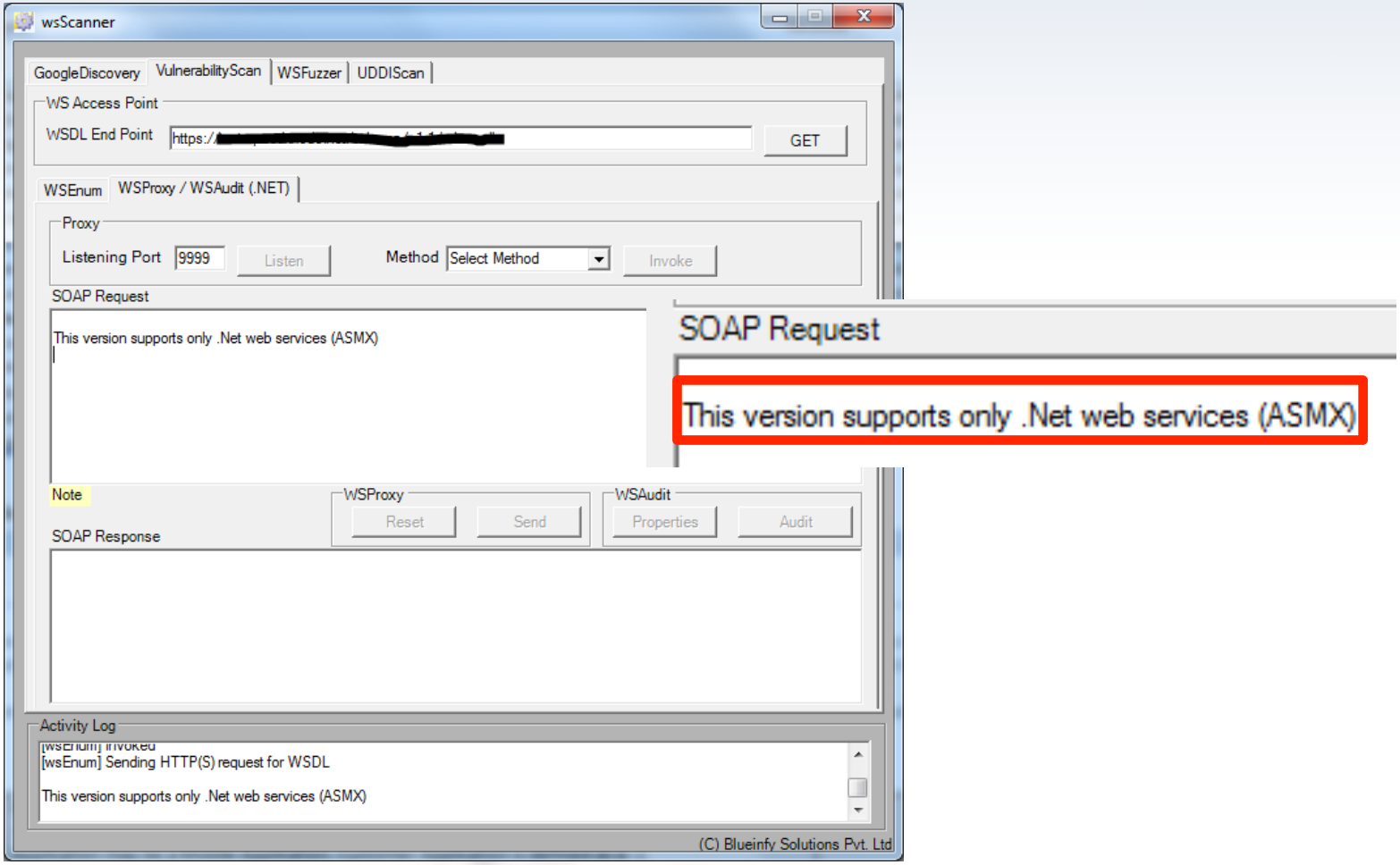
WEBSCARAB – WEB SERVICE MODULE



WSDIGGER


WSDigger
File Attacks Analysis Report Help
Foundstone | WSDigger™
WSDL
http://api.google.com/GoogleSearch.wsdl
Get Methods
Clear Data
Service URL
http://api.google.com/search/beta2
Invoke
GoogleSearchService
doGetCachedPage
doSpellingSuggestion
doGoogleSearch
Input
doGoogleSearch
String key =
String q =
Int32 start =
Int32 maxResults =
Boolean filter =
String restrict =
Input Value
Type
String
Value
Output
Output Value
<< Back
Proxy assembly loaded successfuly from file: C:\DOCUME~1\DELTAU~1\IMPOST~1\Temp\1AC0F599EB2B3AD38DDE90C

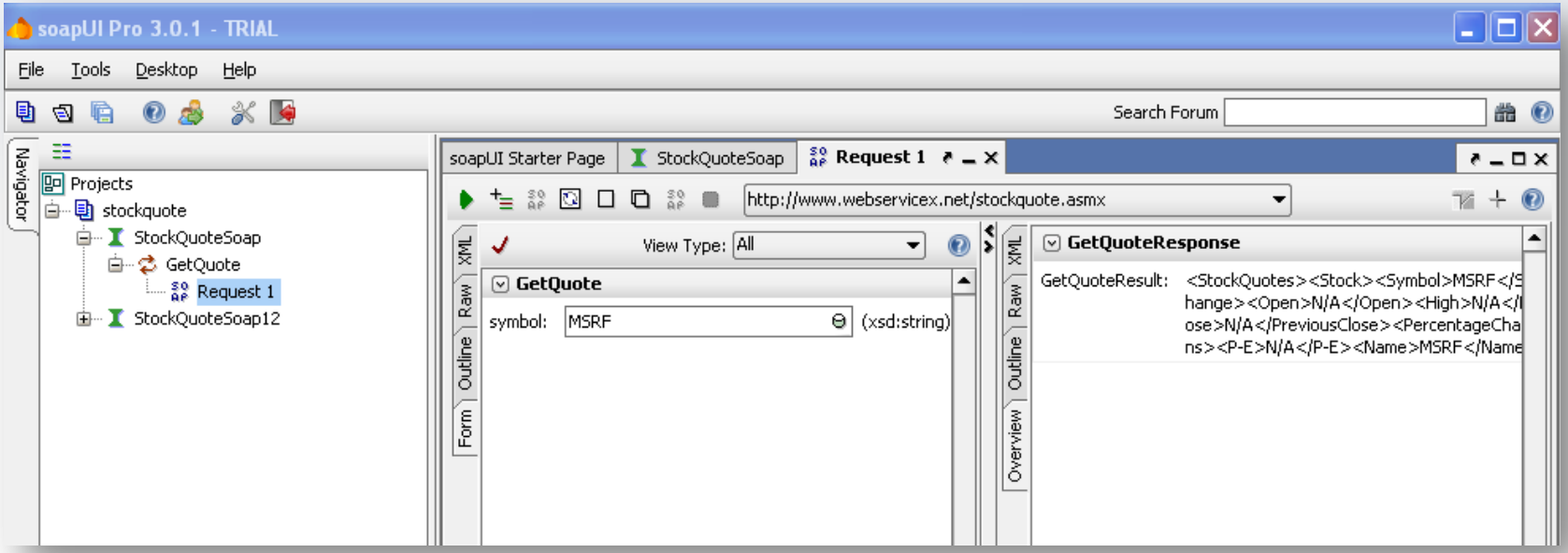
WSSCANNER


wsScanner
GoogleDiscovery
VulnerabilityScan
WSFuzzer
UDDIScan
WS Access Point
WSDL End Point
https:/
GET
WSEnum
WSProxy / WSAudit (.NET)
Proxy
Listening Port
9999
Listen
Method
Select Method
Invoke
SOAP Request
This version supports only .Net web services (ASMX)
Note
WSProxy
Reset
Send
WSAudit
Properties
Audit
SOAP Response
Activity Log
[wsEnum] Sending HTTP(S) request for WSDL
This version supports only .Net web services (ASMX)
(C) Blueinfy Solutions Pvt. Ltd
SOAP Request
This version supports only .Net web services (ASMX)

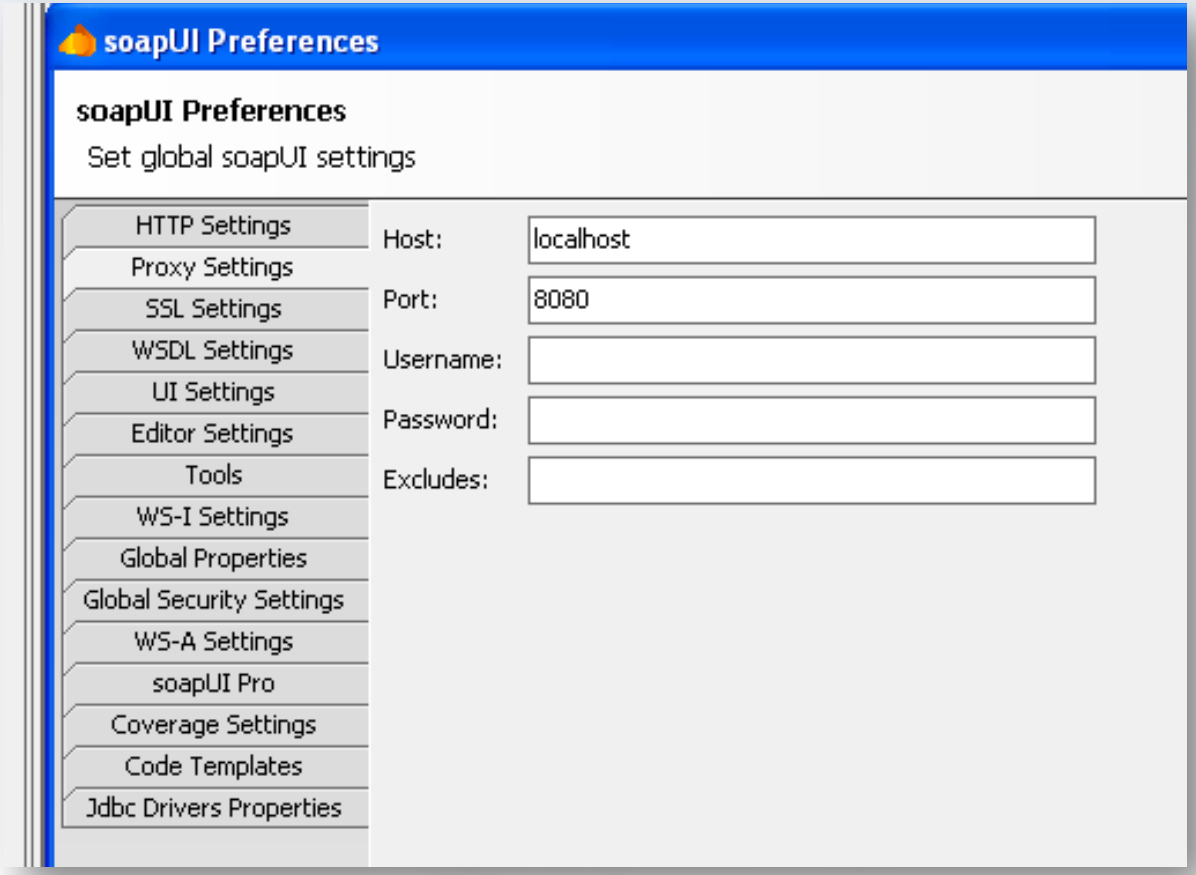
WHAT ARE WE USING?

- soapUI combined with BurpSuite are the bomb
  - Still could be better
- There are very good BurpSuite Plugins by Ken Johnson as well: http://resources.infosecinstitute.com/soap-attack-1/
- Custom built scripts for specific engagements
  - Takes time and billable hours

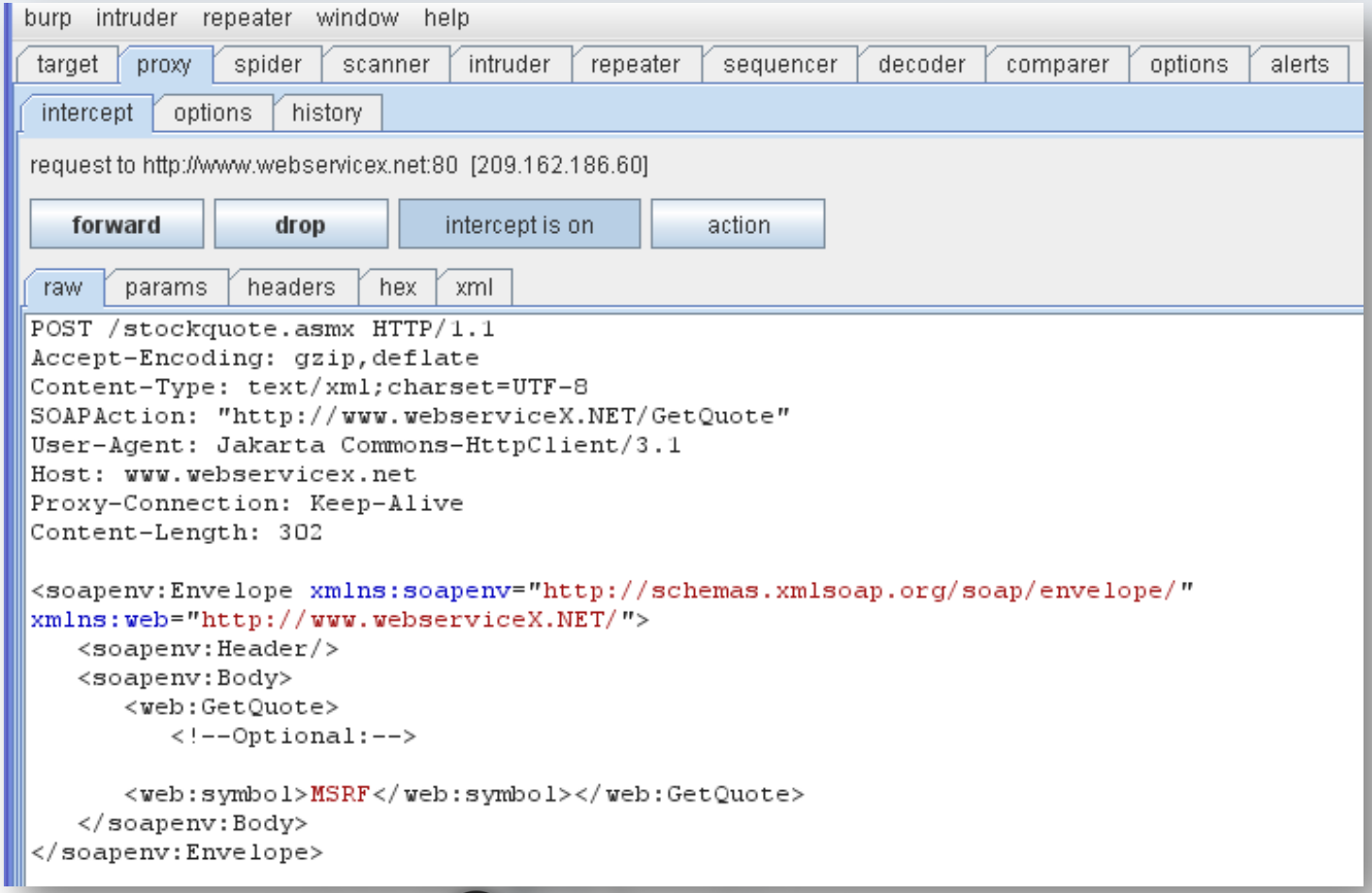
SCREEN SHOTS OF SOAPUI->BURP



SCREEN SHOTS OF SOAPUI->BURP (2)



SCREEN SHOTS OF SOAPUI->BURP (3)

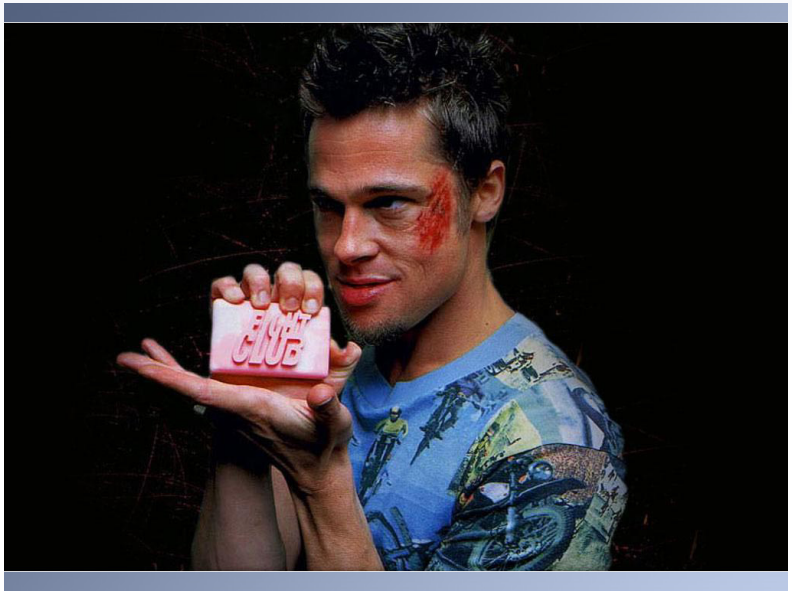


LACK OF TESTING ENVIRONMENTS

- Great! I have a new tool/script..where can I test this?
- Production systems will work….wait, what?
- I'll just build my own testing environment…wait, what?

WEB SERVICES AND THE OSI LAYERS

- Implemented by adding XML into layer 7 applications (HTTP)
- SOAP
  - **S**imple **O**bject **A**ccess **P**rotocol
- Think of SOAP like you would with SMTP
  - It's a message/envelope and you need to get a response



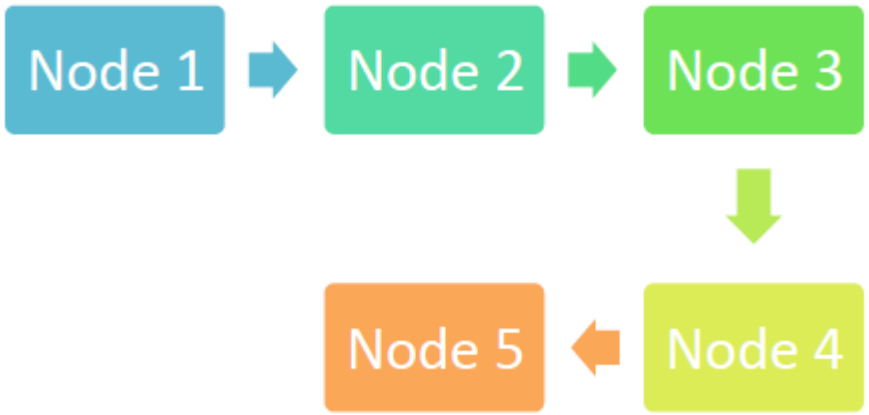
THE WEB SERVICE THREAT MODEL

- Web Services in Transit
  - Is data being protected in transit?
  - SSL
  - What type of authentication is used?
    - BASIC Authentication != Secure
- Web Services Engine
- Web Services Deployment
- Web Services User Code

*** From "Hacking Web Services" by Shreeraj Shah**

THE SOAP ENVELOPE AND TRANSPORT MECHANISM

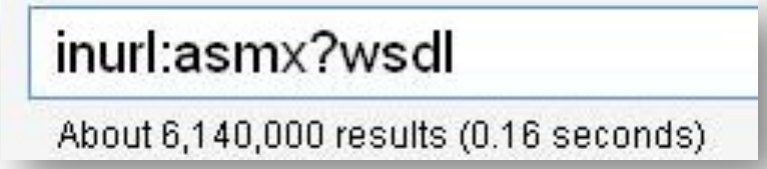
- Multiple endpoints become a problem
- SSL only protects the data between nodes
- What about the security of the message itself?



WEB SERVICES FINGERPRINTING

- Google Hacking for exposed WSDLs
  - filetype:asmx
  - filetype:jws
  - filetype:wsdl
  - Don’t forget about DISCO/UDDI directories
- Searches for Microsoft Silverlight XAP files
- Shodan search for exposed web service management interfaces

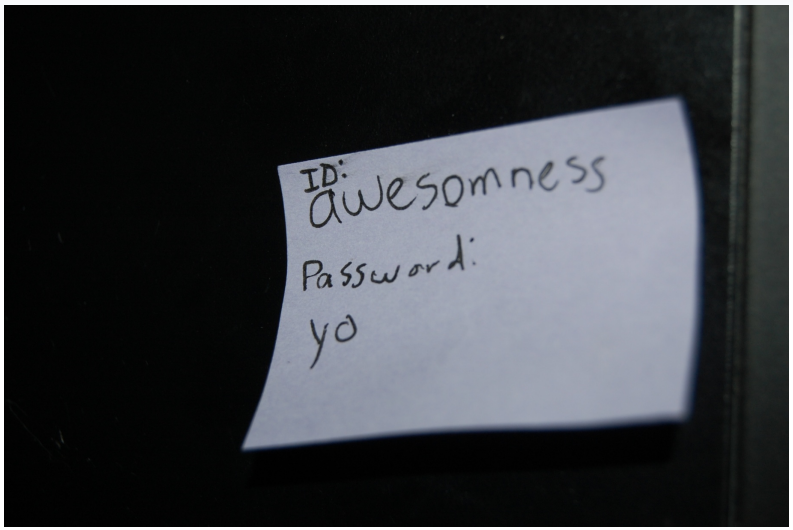
GOOGLE SEARCH



DIFFERENCES IN WEB SERVICE STANDARDS

- Some developer departure from XML based SOAP to RESTful services like JSON
- REST (Representational State Transfer) use HTTP methods (GET, POST, PUT, DELETE)
- RESTful services are lightweight non-complex
- However:
  - SOAP based services are complex for a reason!
  - Many custom applications use them in enterprise applications
- Large services still use SOAP:
  - Amazon EC2, PayPal, Microsoft Azure are a few examples

THE IMPORTANCE OF WEB SERVICE MANAGEMENT INTERFACES



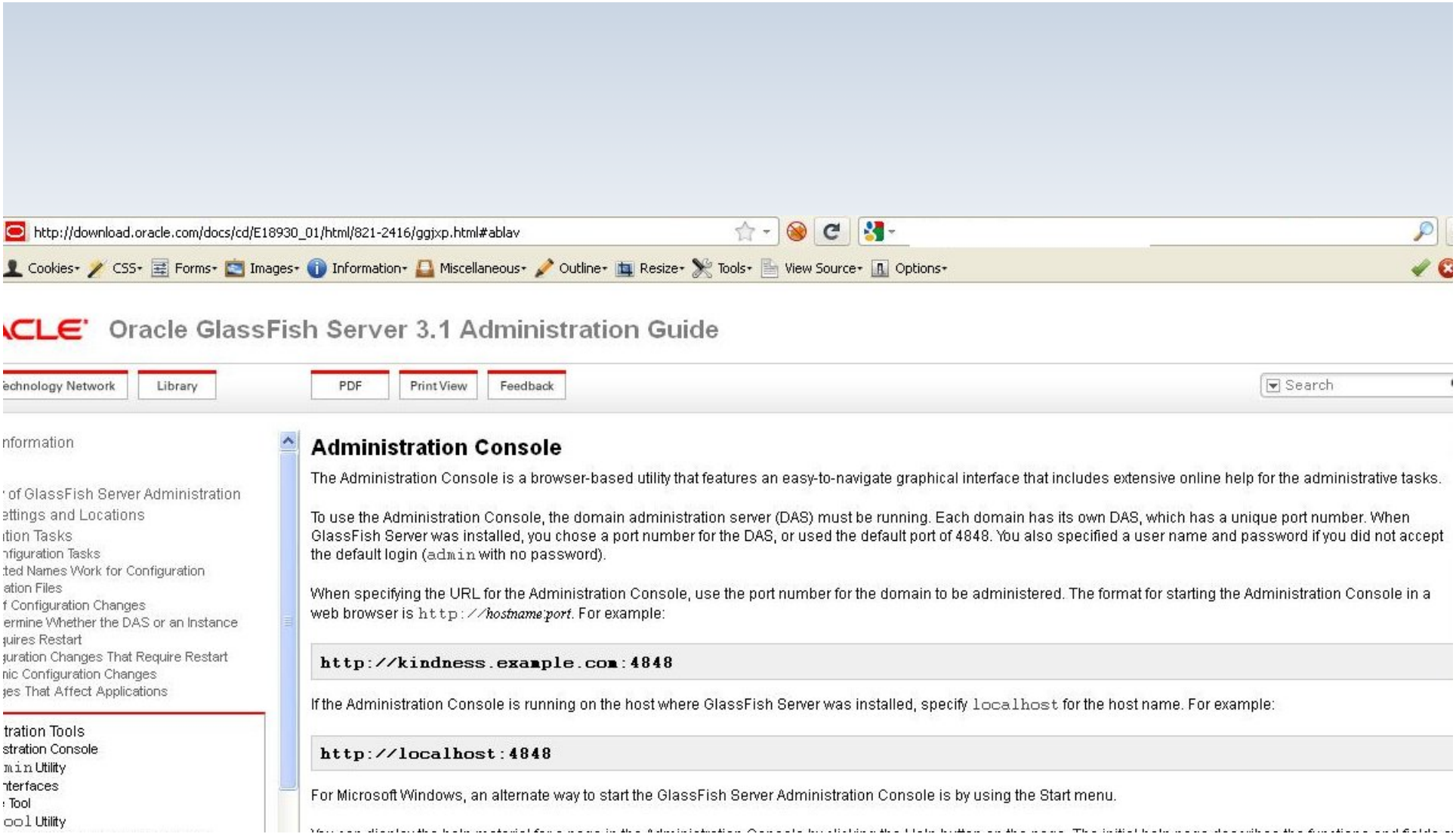
- If these interfaces are exposed an attacker could:
  - Control the system that has the web services deployed
  - Why bother even testing the web services at this point??
- How about weak or default passwords?
  - Most organization this is their biggest risk
  - Pass-the-Hash
- Administration interfaces
  - Axis2 SAP BusinessObjects
  - 2010 Metasploit module created for this
  - http://spl0it.org/files/talks/basc10/demo.txt

GLASSFISH CURRENT ATTACKS

- Web Application interface for managing web application and web services
- Unique port: 4848
  - Enumeration easy
- Sun Glassfish 2.x and Sun Application Server 9.x
  - Default credentials: admin / adminadmin
- Known authentication bypass: CVE-2011-0807 (released in April)
  - Affects: Sun Glassfish 2.x, Sun Application Server 9.x and Glassfish 3.0

GLASSFISH 3.1 ATTACKS

- Still unique port: 4848
  - Enumeration easy
- Oracle GlassFish 3.0 and 3.1 use a default credential: (admin / *blank password*)


http://download.oracle.com/docs/cd/E18930_01/html/821-2416/ggjxp.html#ablav
Cookies CSS Forms Images Information Miscellaneous Outline Resize Tools View Source Options
ACLE Oracle GlassFish Server 3.1 Administration Guide
echnology Network
Library
PDF
Print View
Feedback
Search
nformation
of GlassFish Server Administration
ettings and Locations
ition Tasks
nfiguration Tasks
ted Names Work for Configuration
ation Files
f Configuration Changes
ermine Whether the DAS or an Instance
uires Restart
guration Changes That Require Restart
nic Configuration Changes
es That Affect Applications
tration Tools
stration Console
min Utility
nterfaces
Tool
ool Utility
Administration Console
The Administration Console is a browser-based utility that features an easy-to-navigate graphical interface that includes extensive online help for the administrative tasks.
To use the Administration Console, the domain administration server (DAS) must be running. Each domain has its own DAS, which has a unique port number. When GlassFish Server was installed, you chose a port number for the DAS, or used the default port of 4848. You also specified a user name and password if you did not accept the default login (admin with no password).
When specifying the URL for the Administration Console, use the port number for the domain to be administered. The format for starting the Administration Console in a web browser is http://hostname:port. For example:
http://kindness.example.com:4848
If the Administration Console is running on the host where GlassFish Server was installed, specify localhost for the host name. For example:
http://localhost:4848
For Microsoft Windows, an alternate way to start the GlassFish Server Administration Console is by using the Start menu.

GLASSFISH ENUMERATION

GLASSFISH 3.1 ATTACK

Administration Console

The Administration Console is a browser-based utility that features an easy-to-navigate graphical interface that includes extensive online help for the administrative tasks.

To use the Administration Console, the domain administration server (DAS) must be running. Each domain has its own DAS, which has a unique port number. When GlassFish Server was installed, you chose a port number for the DAS, or used the default port of 4848. You also specified a user name and password if you did not accept the default login (`admin` with no password).

When specifying the URL for the Administration Console, use the port number for the domain to be administered. The format for starting the Administration Console in a web browser is `http://`*hostname*`:`*port*. For example:

```
http://kindness.example.com:4848
```

If the Administration Console is running on the host where GlassFish Server was installed, specify `localhost` for the host name. For example:

```
http://localhost:4848
```

For Microsoft Windows, an alternate way to start the GlassFish Server Administration Console is by using the Start menu.

Reference: http://download.oracle.com/docs/cd/E18930_01/html/821-2416/ggjxp.html#ablav

GLASSFISH 3.1 METASPLOIT DEMO

- Auxiliary Scanner
- Exploit Module
  - Thanks to Juan and Sinn3r for helping with the module
- Works on :
  - Glassfish 3.1 (commercial and open source)
    - default credentials (admin / *blank password*
  - Glassfish 3.0 (commercial and open source)
    - default credentials (admin / *blank password*) and auth bypass
  - Sun Glassfish 2.1 and Sun Application Server 9.1
    - Default credentials (admin / adminadmin) and auth bypass

THE IMPORTANCE OF WEB SERVICE MANAGEMENT INTERFACES

- If these interfaces are exposed an attacker could:
  - Control the system that has the web services deployed
  - Why bother even testing the web services at this point??
- How about weak or default passwords?
  - Example: Axis2 SAP BuisinessObjects
  - 2010 Metasploit module created for this (Josh you want to show an example?)

NEW WEB SERVICE THREATS

- Microsoft Silverlight
  - Client side application that can use web services
  - SOAP or REST
  - Can use WCF (Windows Communication Foundation) services
  - Attacker can directly interface with the web services…really no need for the client
  - Security depends on the configuration of the services!

NEW WEB SERVICE ATTACKS

- WS-Attacks.org by Andreas Flakenberg
  - Catalogs most (if not all) attacks for modern SOAP and BPEL web services
- SOAP requests to web services that provide content to the web app
- AJAX, Flash and Microsoft Silverlight add to the complexity

NEW ADVANCEMENTS

- Client side applications like Microsoft Silverlight
- Increased complexity with AJAX and Flash implementations
- Multiple web services being used within applications
- Organizations exposing web services for mobile applications

BPEL

» WS-BPEL

- Web Service Business Process Execution Language (BPEL)
- Separates the business process from the implementation logic
- Usually a White Box approach is required to understand the business logic fully

SCOPING A WEB SERVICE PENTEST

- Pre-Engagement Scoping is CRITICAL!
- Not only for pricing but for proper testing
- Questions such as:
  - What type of framework being used? (WCF, Apache Axis, Zend)
  - Type of services (SOAP, REST, WCF)
  - What type of data do the web services provide
  - Provide all WSDL paths and endpoints
  - What type of authentication does the web service use?
  - SOAP attachment support?
  - Can you provide multiple SOAP requests that show full functionality?
- There are MANY more questions. Our White Paper has the full list ☺

THE NEW WEB SERVICE TESTING METHODOLOGY

- OWASP Testing Guide v3 was a great start
- It's old, outdated and doesn't address new concerns
- Our research will be included in OWASP Testing Guide v4
- We are aligning the methodology with:
  - PTES: **P**enetration **T**esting **E**xecution **S**tandard
  - PTES provides a standard penetration testing methodology framework
  - Created with the help from information security practitioners from all areas of the industry (Example: Financial Institutions, Service Providers, Security Vendors)
  - Can be used by all penetration testers and outlines essential phases of ANY penetration test

PTES AND WEB SERVICE TESTING

- Pre-Engagement Interactions
  - Scoping Questions and Goals
  - Assessment type (Black, Grey, White Box)
  - Rules of Engagement
- Intelligence Gathering
  - Identify WSDLs and Enumerate
  - WS-Security Controls
  - Authentication Credentials
  - Sample SOAP requests
  - Identify Web Service Interfaces (GlassFish, Axis2)
- Threat Modeling
  - What is most valuable from a business perspective?
  - Outline scenarios for realistic attack vectors

PTES AND WEB SERVICE TESTING

- Vulnerability Analysis
  - Authentication Testing (Brute Force)
  - Transport Layer Testing
  - Web Service Interface Management Testing
  - Analyze Client Applications (Sliverlight)
- Exploitation
  - XML Structural, Content-Level Testing
  - Use new MSFWEBFUZZ module
  - Reply/MiTM Testing
  - BPEL Testing
- Post Exploitation
  - Got shell?
  - Prepare and document
- Reporting

*** Full Methodology is included in our White Paper!**

NEW WEB SERVICE TESTING MODULES FOR METASPLOIT

- Two tools released today:
  - HTTP request repeater (msfwebrepeat)
  - HTTP fuzzer (msfwebfuzz)
- Backend web services libs (alpha version)
  - Authentication support: BASIC/DIGEST and WS-Security
  - Ability to leverage existing payloads (php/java) thru native MSF libs

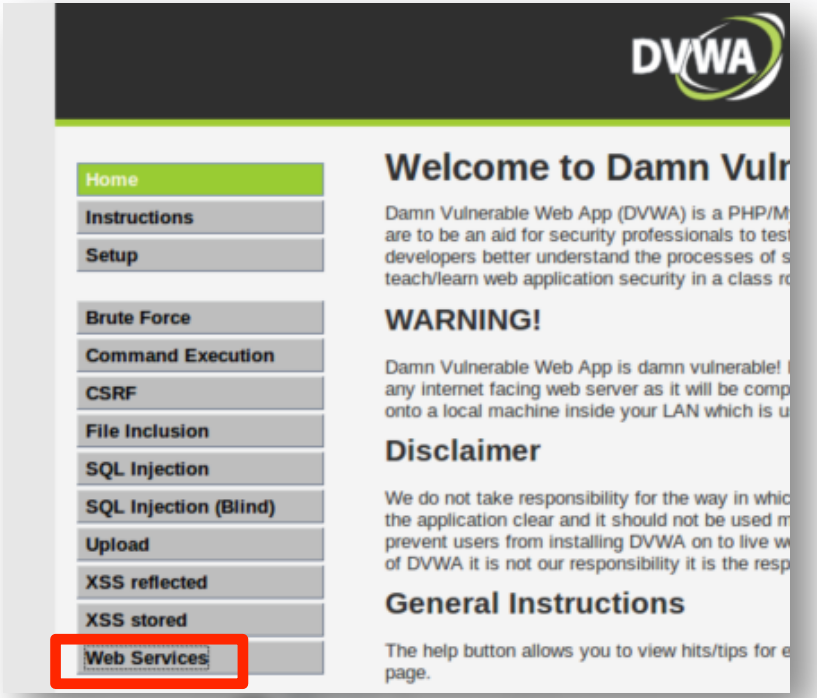
MSFWEBREPEAT - DEMO

MSFWEBFUZZ - DEMO

MSF WEB SERVICES MODULE - DEMO

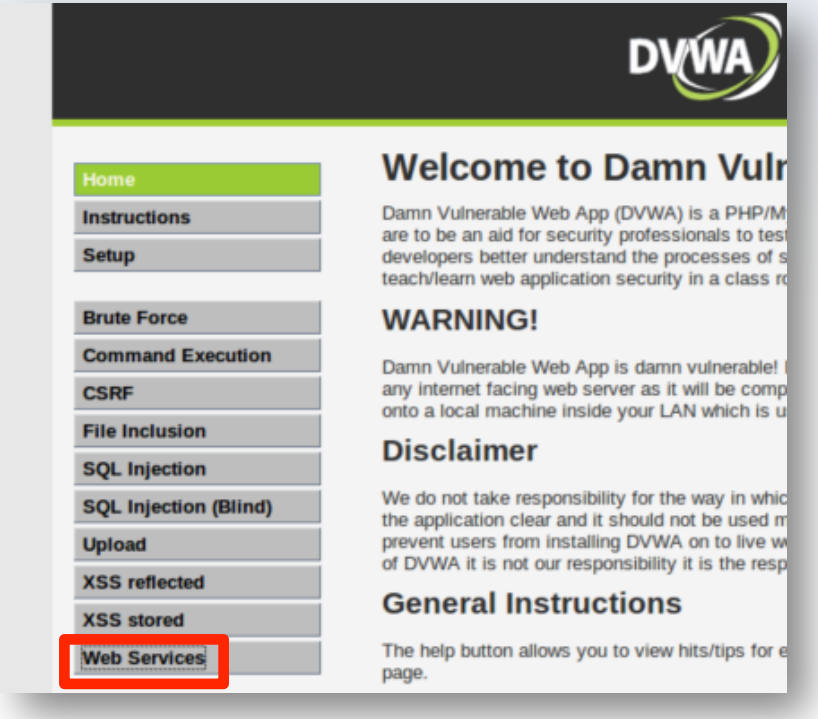
DAMN VULNERABLE WEB SERVICES

- Damn Vulnerable Web Services (DVWS) is a series of vulnerable web services
- Built within DVWA
- Provides a series of services to test



DVWS FEATURES

- Uses DVWA authentication
- High, medium and low difficulties
- WSDL available for each services
- Reflective and persistent vulnerabilities
- Extendable



WS-SQLI

- Allows for the testing of SQL injection
- Uses the DVWA database to be consistent
- Difficulty levels are used for more challenge

WS-COMMANDINJ

- Command injection allows for system commands delivered via SOAP
- Filtering based on select DVWA difficulty
- High level includes blind command injection

WS-XSS_P

- Persistent XSS flaw
- Service publishes content to the main web application
- Difficult for automated testing due to the remote display

CONCLUSIONS

- Pay attention to new attack vectors and web service technology
- Developers are ahead of the security community and we need to catch up
- Our work is only the beginning. Get involved with OWASP, contribute to open source projects (get developers to do the same)


black hat®
BRIEFINGS & TRAINING
USA + 2011
EMBEDDING SECURITY