Foundstone®
Professional Services
A DIVISION OF McAFEE

Foundstone WSDigger™ 1.0
Web Services Security Testing Framework

By Kartik Trivedi
Foundstone Professional Services, a division of McAfee

June 20, 2005

Foundstone®

Table of Contents

Foundstone®

Growth of Web Services

The term web services refers to architecture, standards, technology and business models that provide an implementation-independent way for applications to communicate with each other. They are self-contained, self-describing, modular applications that can be published, located, and invoked across the Web.

Web services are poised for significant growth as demonstrated by the following predication;

- Gartner predicts "by 2006, 45% of US companies will be using some form of IT utility enabled by Web Services. Through 2007, web services will enable the emerging RTI capability underlying 80% of hybrid IT utilities deployed by US companies".[1]
- International Data Corp. estimates "Web services would be a $15.2 billion market by 2008.[2]
- "Web Services Enabled eBay To Become What It Is Today", Matt Ackley Senior director of eBay developers program.[3]

Web Services Security

Security has been identified as one of the most important challenges to interoperable Web services and one of the most common reasons for not implementing them. Significant interest in web services specification development from organizations like Microsoft, IBM, BEA Systems, Oracle, Sun Microsystems and Hitachi has led to myriads of security standards which often have significant overlap. At the same time as these standards are being created, web services developers are falling prey to the same or similar attacks that have plagued web applications for the last five years and which these new standards are not designed to address. Attacks like SQL injection have now become XPATH injection and in doing so, become even more powerful.

This paper discusses Foundstone WSDigger™ - a new free open source tool designed by Foundstone to automate black-box web services security testing (also known as penetration testing). WSDigger is more than a tool, it is a web services testing framework. Version one of this framework contains sample attack plug-ins for SQL injection, cross site scripting and XPATH injection attacks. A sample web service vulnerable to XPATH injection accompanies the tool. By releasing the framework as an open-source tool, users are encouraged to develop and share their own plug-ins. The tool, along with the source code, can be downloaded from Foundstone's website at http://www.foundstone.com. A workspace for developing and sharing plug-ins will be available at SourceForge. Go to http://www.sourceforge.net , and search for Foundstone.

[1] "Predict 2005: The Impact of web services still grows" – Gartner Nov 2004
[2] http://searchwebservices.techtarget.com/originalContent/0,289142,sid26_gci916071,00.html
[3] http://webservices.sys-con.com/read/48251.htm

Foundstone®

WSDigger Features

The major features of WSDigger 1.0 are listed below:

- Built-in sample attack plug-ins: SQL injection, Cross Site Scripting and XPATH injection
- Sample Web Service: Includes example web service vulnerable to XPATH injection
- Open source: The source code of the main tool and the sample plug-ins are included with the installer and the license allows users to develop their own customizations and attack plug-ins
- Automated Web Services Discovery: The tool analyzes public and / or private UDDI's to discover web services related to search specific strings
- Automated Attack Vector Discovery: The tool analyzes the web service to determine potential points of attack
- Automated and Manual Exploit Testing: The tool can be used to manually inject malicious data and generates test cases which can be automatically executed through the attack plug-ins
- HTML Reporting: The tool generates HTML reports of the results and test case history for further analysis

Installing WSDigger

Pre-requisites

WSDigger is a .NET managed assembly built using C#. It requires the use of the Microsoft .NET framework version 1.1. This maybe obtained using Windows update or by visiting the following URL: http://msdn.microsoft.com/netframework/howtoget/default.aspx. WSDigger has been tested on Windows XP workstations running the .NET v1.1. While it has not been tested on other versions of Windows, we do believe that it should execute successfully on all Windows operating systems that can support the 1.1 framework.

WSDigger_WS (Sample web service) requires the use of Internet Information Service (IIS). Installation of WSDigger_WS is optional.

Foundstone®

Step One

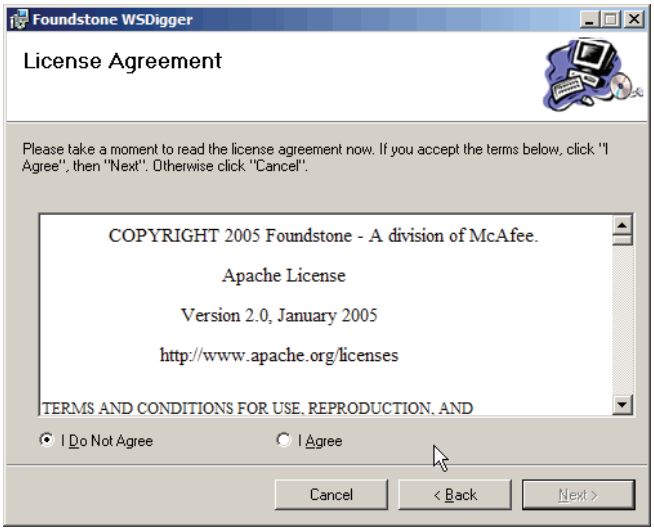
Download WSDigger v1.0.zip from Foundstone's website at http://www.foundstone.com/resourcses/freetools.htm. Unzip the file into a separate directory. The directory should now have the following four files:

WSDigger.msi: WSDigger Installer
WSDigger_WS.msi; Sample Web Service Installer
Code.zip: Source code of both WSDigger and WSDigger_WS
Readme.txt: Help file with installation instruction

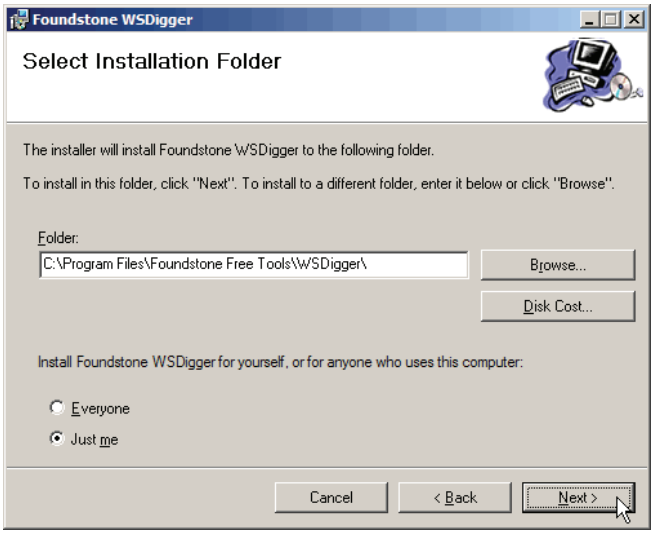
Step Two: Installing WSDigger

Double click the WSDigger.msi file to launch the WSDigger installer.

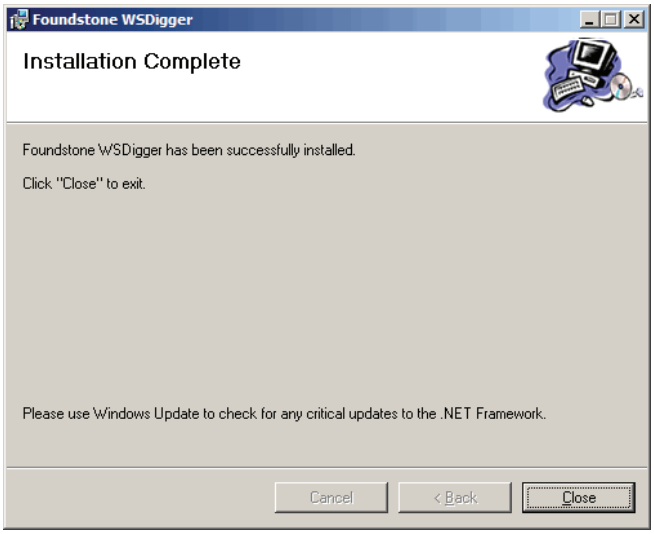
Accept the Foundstone's Term of use to install the software



Follow the installation steps. Select the installation folder and user permissions.

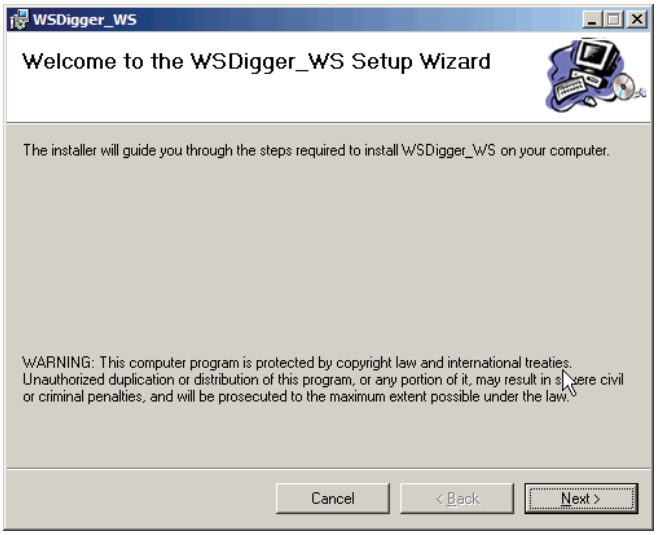


Click Next to Finish the installation

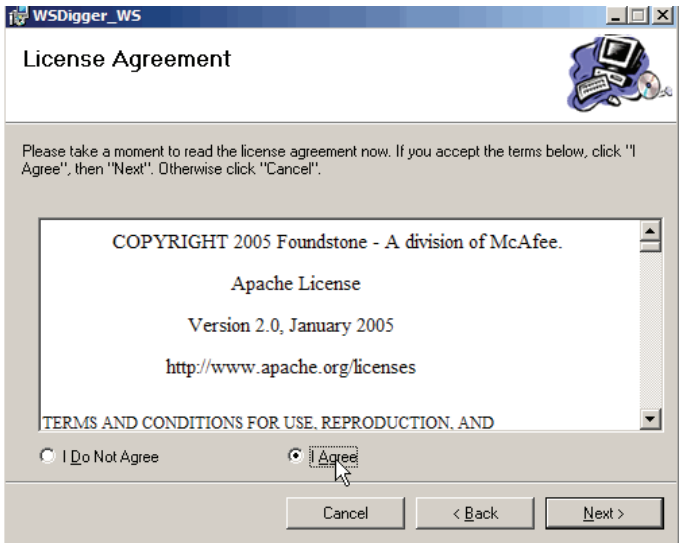


Step Three: Installing Sample Web Service (Optional)

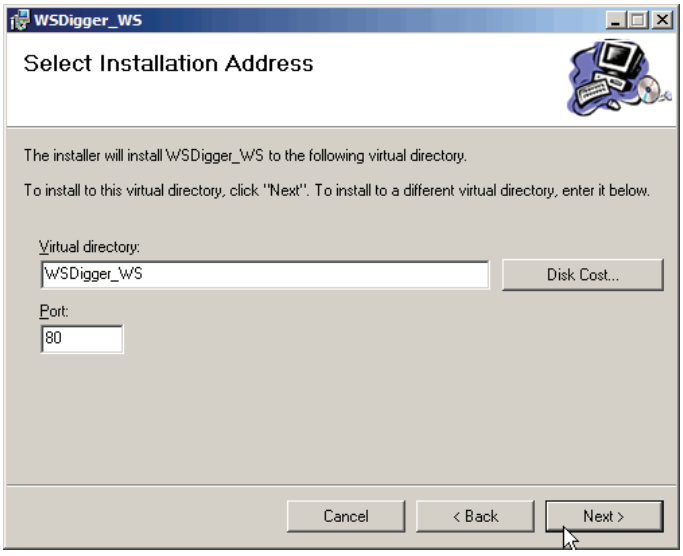
Double click the WSDigger_WS.msi file to install the sample web service.



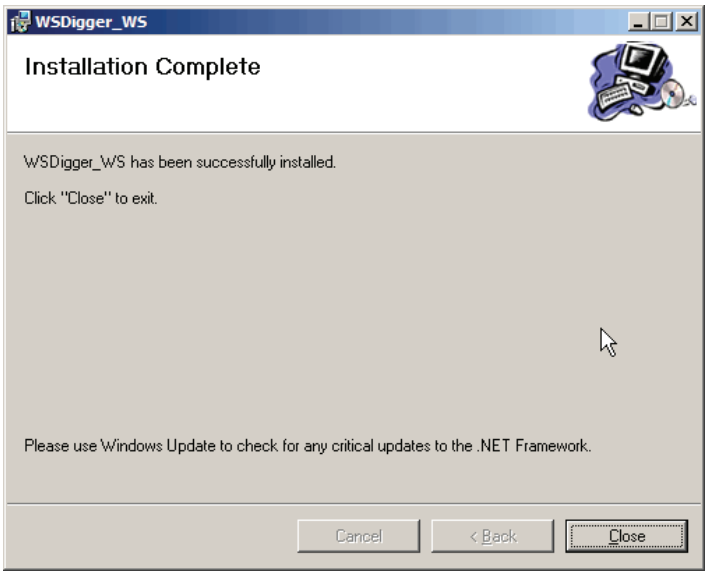
Accept the Foundstone's Term of use to install the software



Follow the installation steps. Select the installation folder and user permissions.



Click Next to Finish the installation



What to do if Installer Fails?

The installer checks to ensure that the .NET framework is installed and the computer meets the minimum requirements for the installation. If the installation fails for any reason, we suggest running Windows Update (http://windowsupdate.microsoft.com) and reinstalling the .NET framework. Further questions should be directed to tools@foundstone.com.

Foundstone®

Using WSDigger

WSDigger takes a black box penetration testing approach; that is to say it imitates a malicious user and has no internal knowledge of the code that drives the web service. It operates by acting as a web services client, discovering how to interact with the service and asking the user to make decisions.

Using the tool can be broken into several key steps;

- Service Discovery
- Attack Vector Discovery
- Exploit Testing
- Analysis

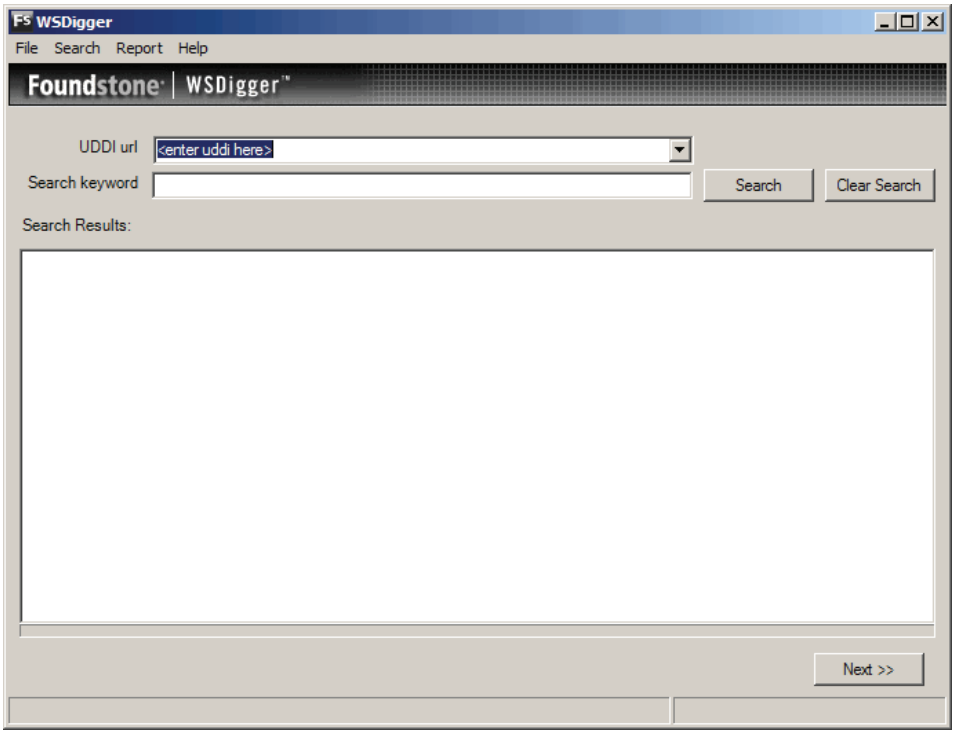
The following example shows how the tool can be used to conduct a security assessment of a web service.

Service Discovery

We will first use WSDigger to discover public web services from a public directory of services hosted at Microsoft. When testing, you should use the UDDI that hosts the WSDL file for the service you are interested in.

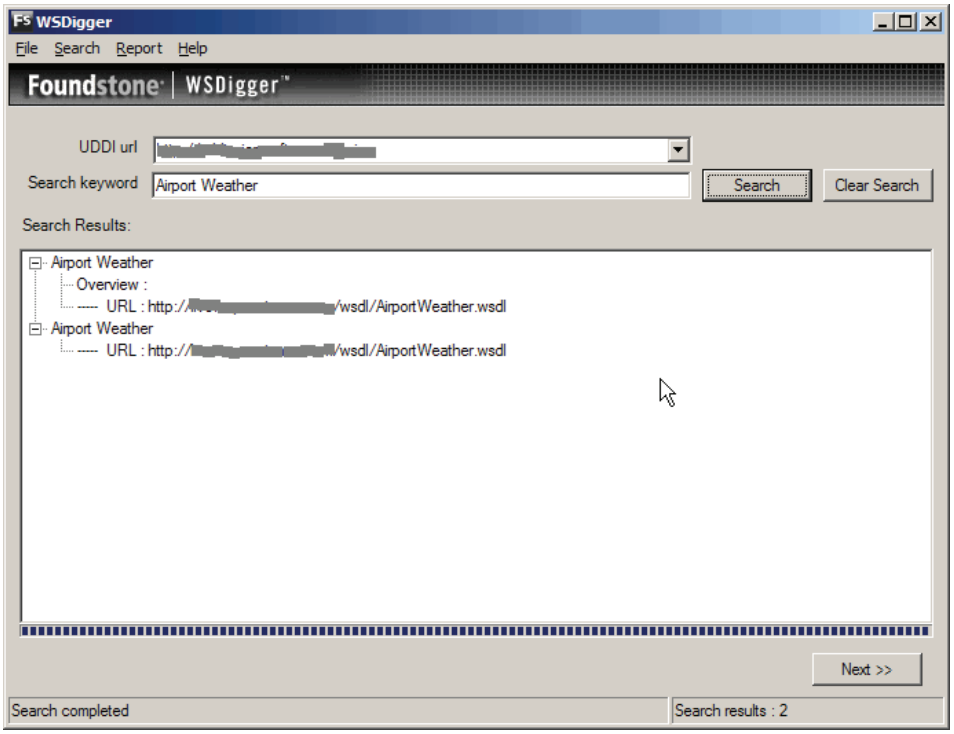
Step One

Launch WSDigger by browsing to Start -> Program -> Foundstone Free Tools -> WSDigger



Step Two

We select the public UDDI from the drop down box. To use the tool against a private UDDI, you must type the complete URL of the private UDDI, enter the search keyword and press search. For this example, we select Microsoft public UDDI (http://uddi.microsoft.com/inquire) from the drop down box and search for keyword 'Airport Weather'. The results display a list of WSDL's matching the keyword.



As we can see from the results there are two web services that will give us weather information about airports.

Attack Vector Discovery

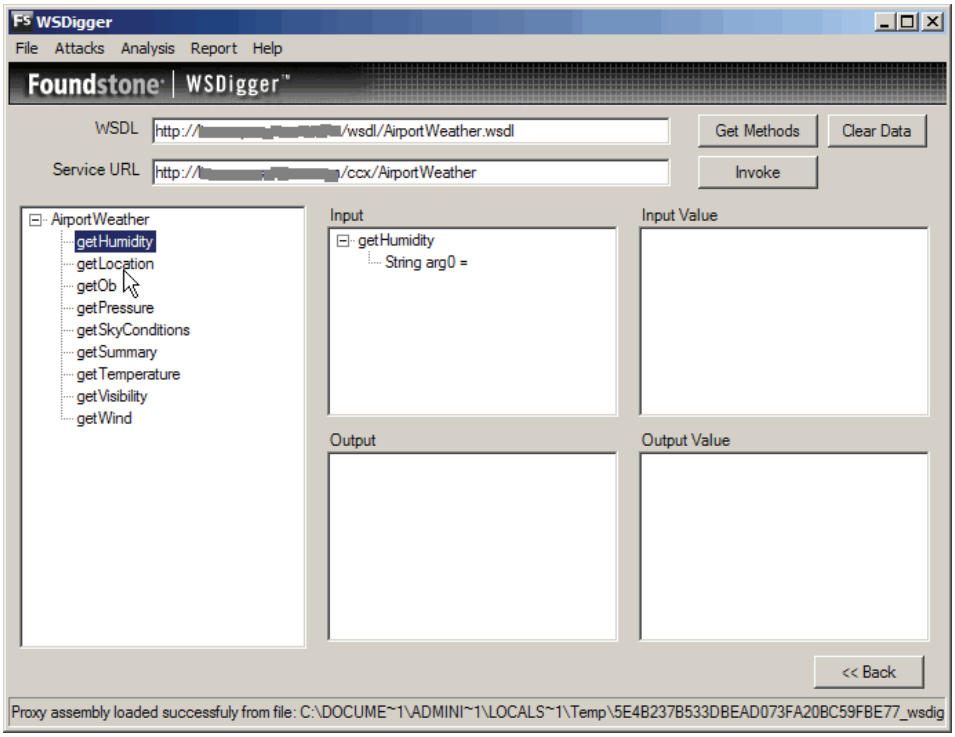
We then use the tool to parse the WSDL and catalog public interfaces and data types. These are the ways in which we can talk to the web service and exposed as a way to enable a web services client to know how to communicate. As an attacker the ways in which we can attack it.

Step one

We select the WSDL to assess and click Next.

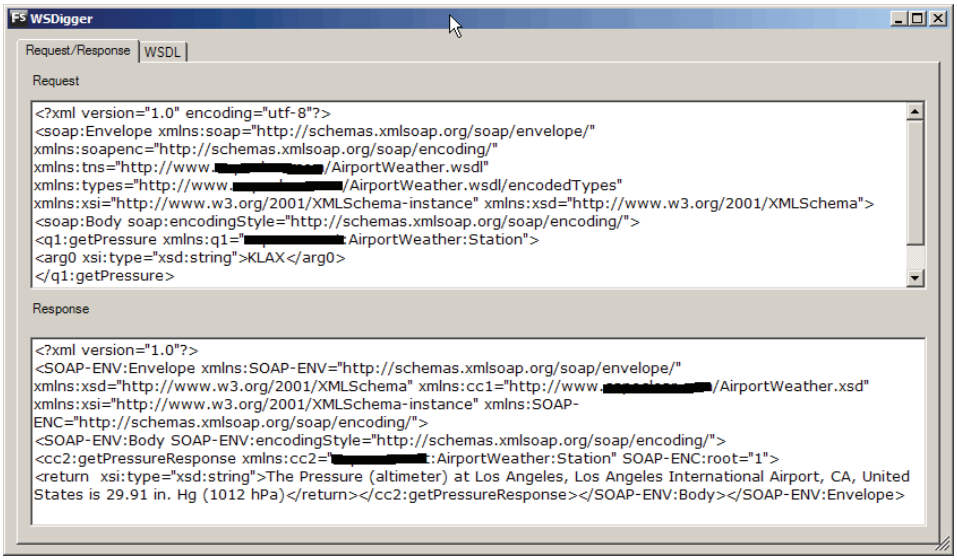
Step Two

We click 'Get Methods' to parse the WSDL and get the methods. Note the results. The left pane contains a list of interfaces provided by the Airport Weather web service. When we highlight an interface by clicking on it, the right pane displays the data type information.



Step Three

At any time, to view the raw request, response and WSDL information, click on Analysis on the Menu Bar.

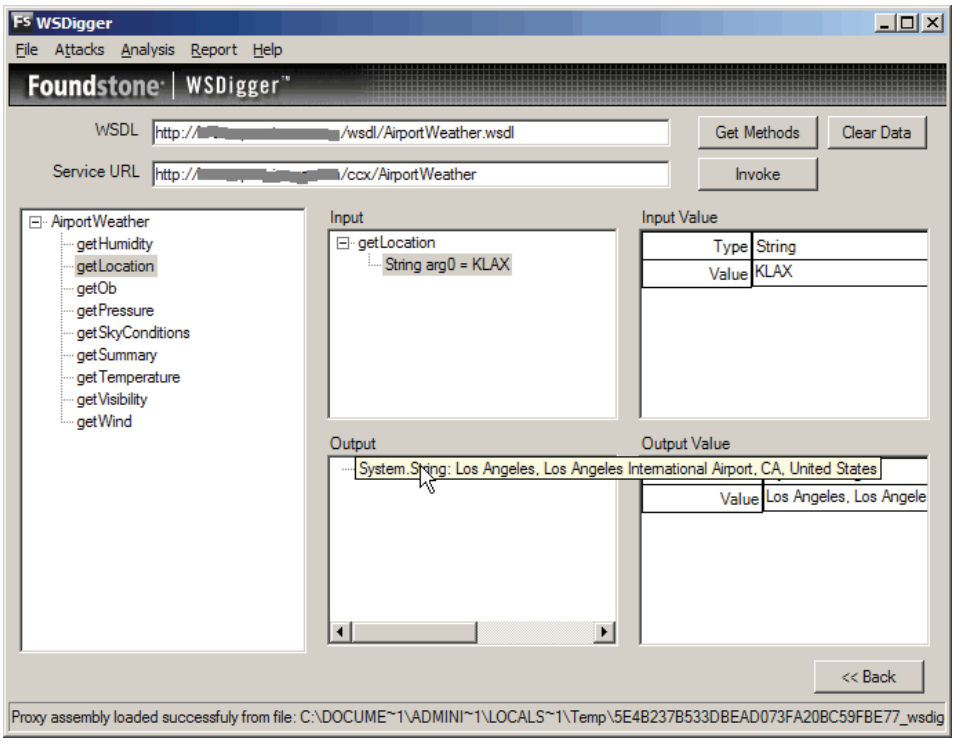


Exploit

This phase uses WSDigger to send malicious input to the web service.

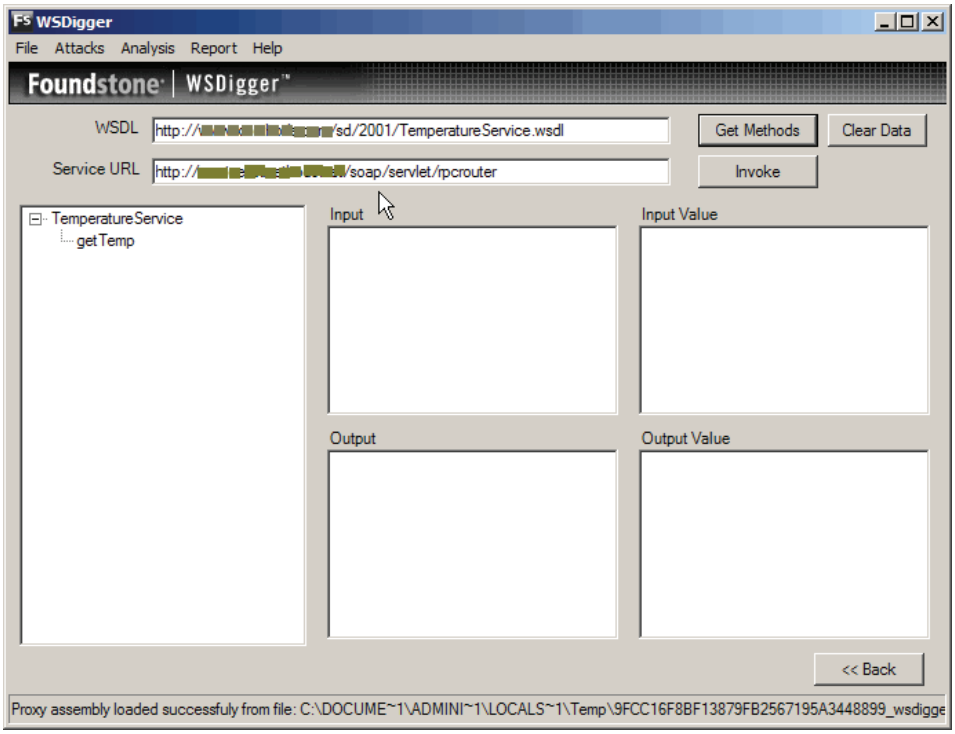
Example 1: Good input

Continuing from where we left, select the 'getLocation' interface and provide 'KLAX' as the input. Click the invoke button. The results displayed in the right bottom tab show the output generated by the web service. We have successfully invoked the 'getLocation' method of the 'AirportWeather' web service.

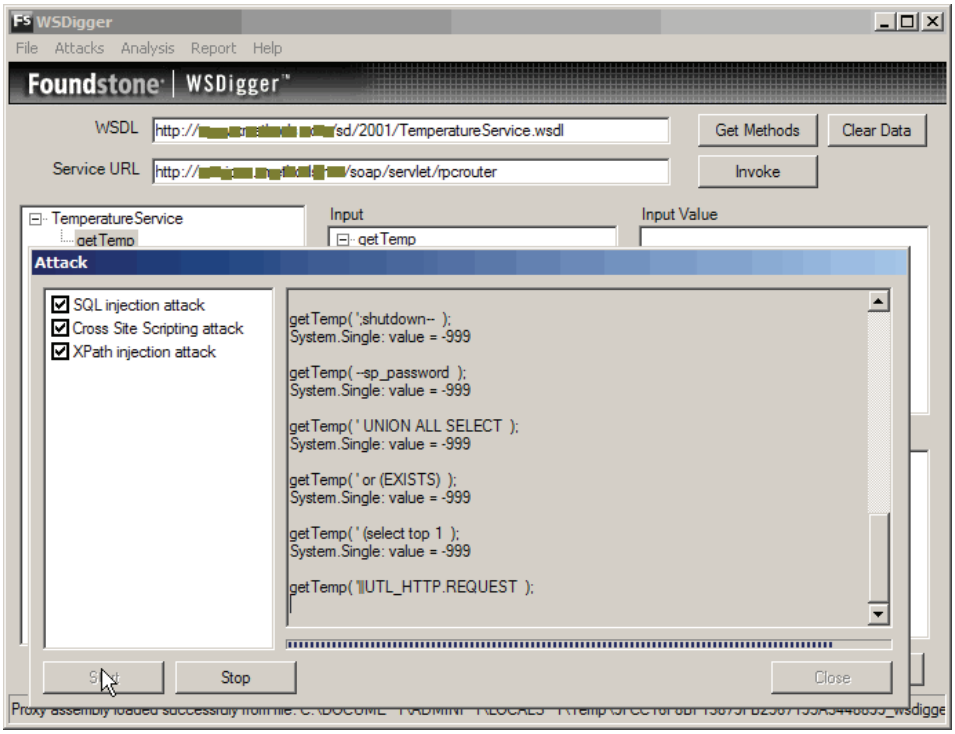


Example 2: Malicious Input

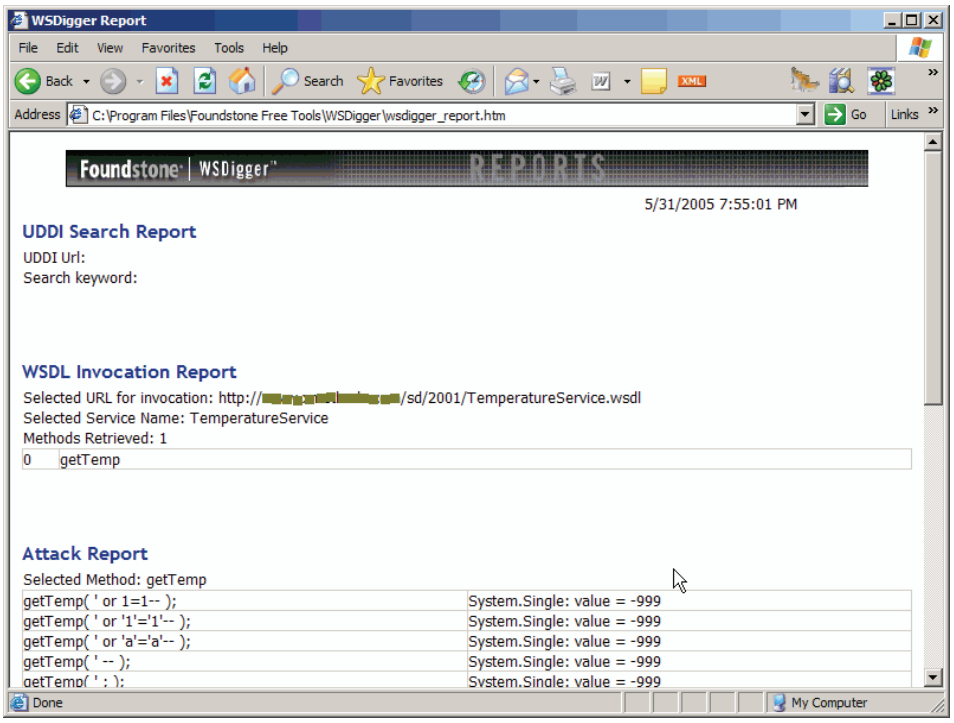
Launch a new instance of WSDigger. Click 'next' at the bottom right corner and enter the WSDL of web service to assess. Note: This time we are skipping discovery phase and directly jumping to enumeration. Click 'Get Methods' to enumerate methods.



Select a method and browse to Attack -> Select Attack Type. Check the attack and hit Start.

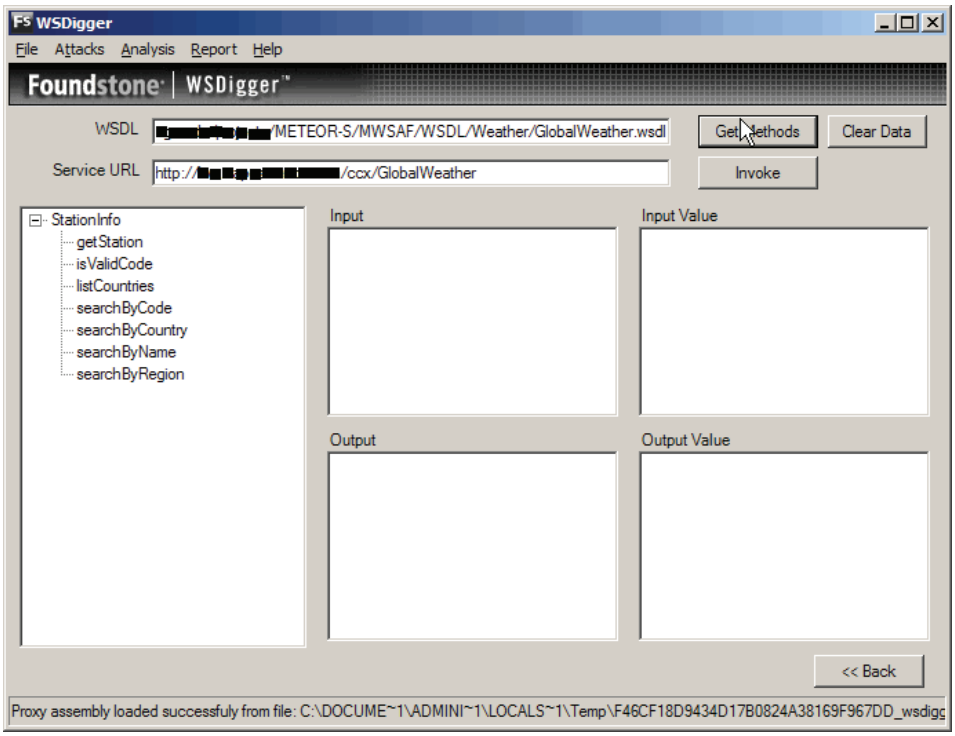


Browse to Report -> Generate Report after the scan is done. The report displays how the web services responds to automated attacks. Note that the web service does not seem vulnerable to any attack in this case.

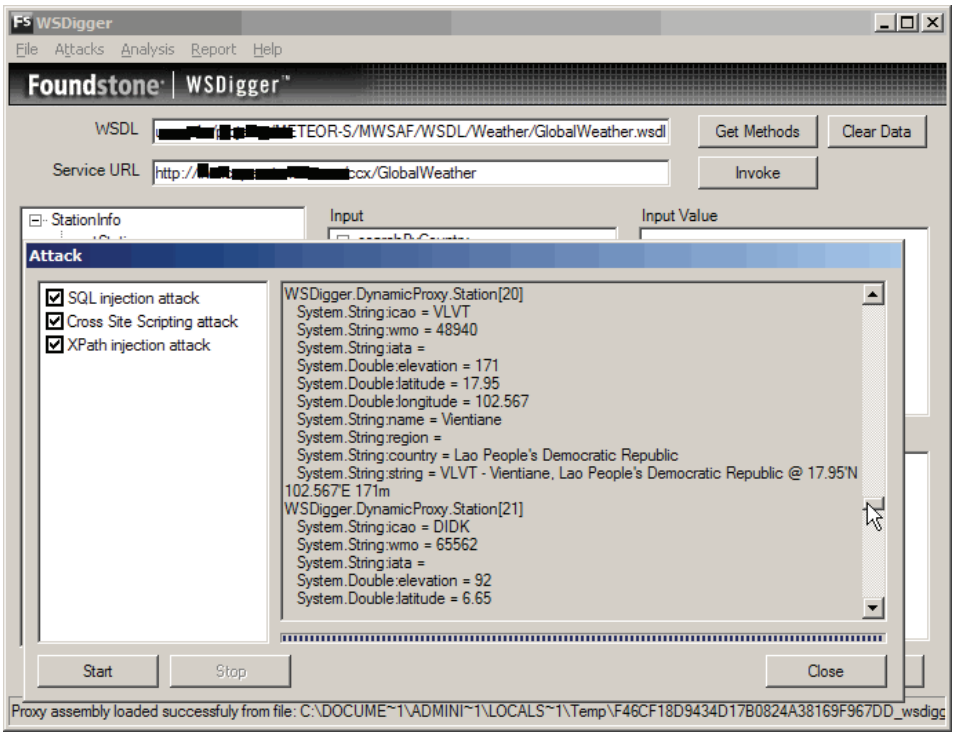


Example 3: Malicious Input

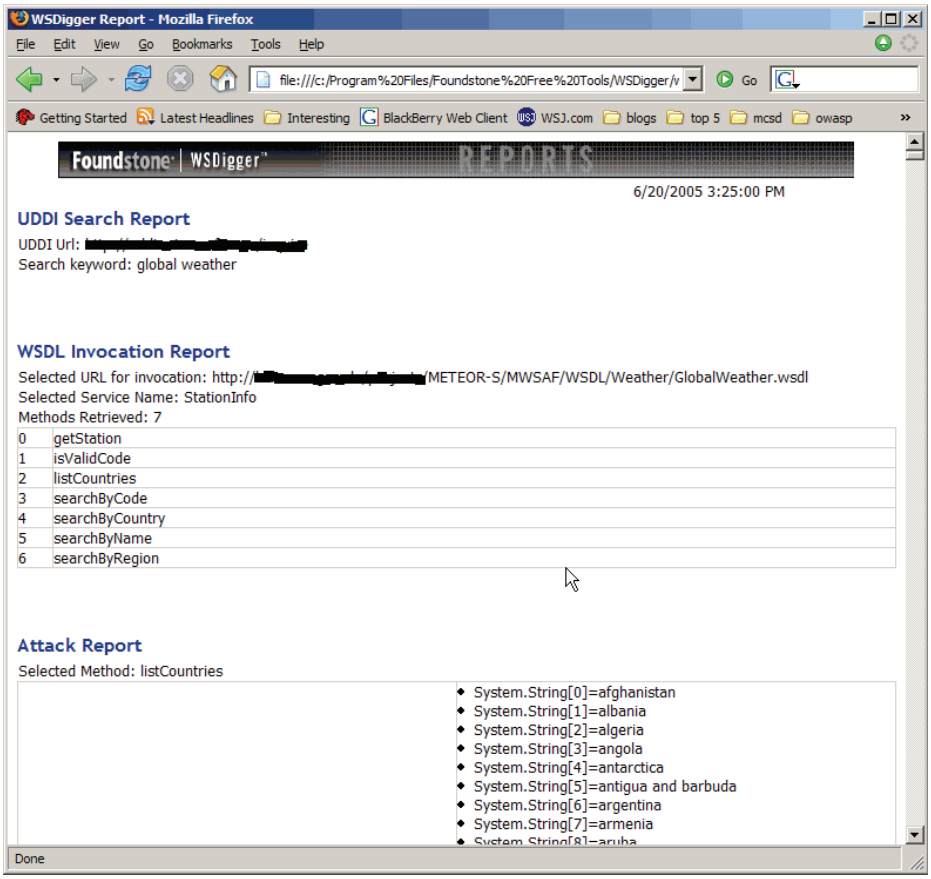
Let us use WSDigger to hack a vulnerable web service. Launch a new instance of WSDigger. Click 'next' at the bottom right corner and enter the WSDL of web service to assess. Note: This time we are skipping discovery phase and directly jumping to enumeration. Click 'Get Methods' to enumerate methods.



Select a method and browse to Attack -> Select Attack Type. Check the attack and hit Start.

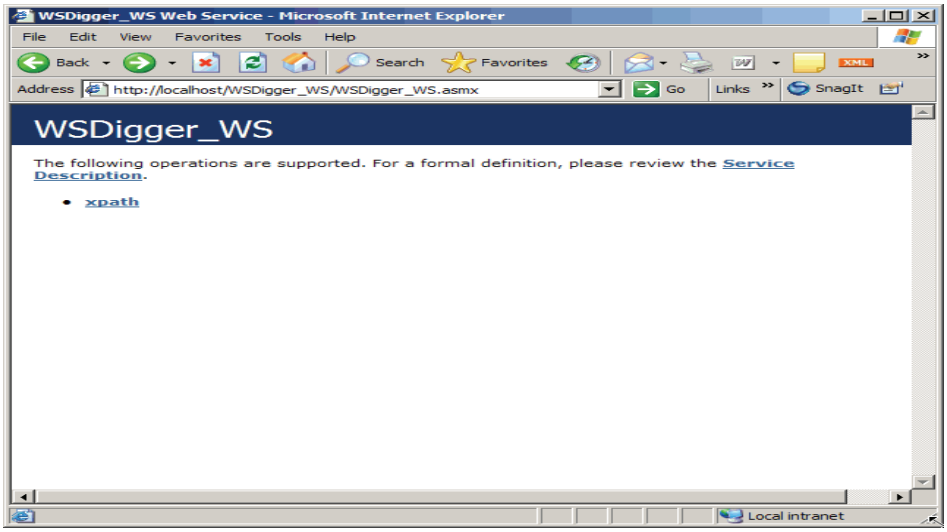


Browse to Report -> Generate Report after the scan is done. The report displays how the web services responds to automated attacks. Note that the web service is vulnerable to SQL injection attack.

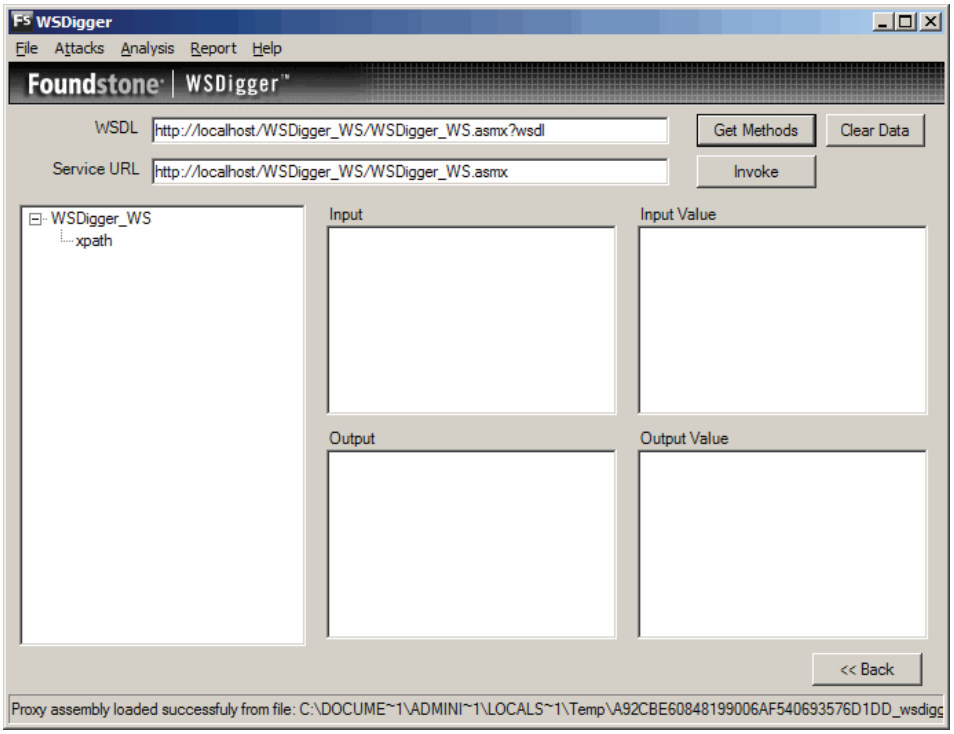


Example 3: Using the locally installed Sample Web Service

If the user ran WSDIgger_WS.msi installer, the sample web service is installed at http://localhost/WSDigger_WS/WSDigger_WS.asmx*. and the corresponding WSDL at* http://localhost/WSDigger_WS/WSDigger_WS.asmx?wsdl. You can try manually browsing to the URL and invoking the methods.

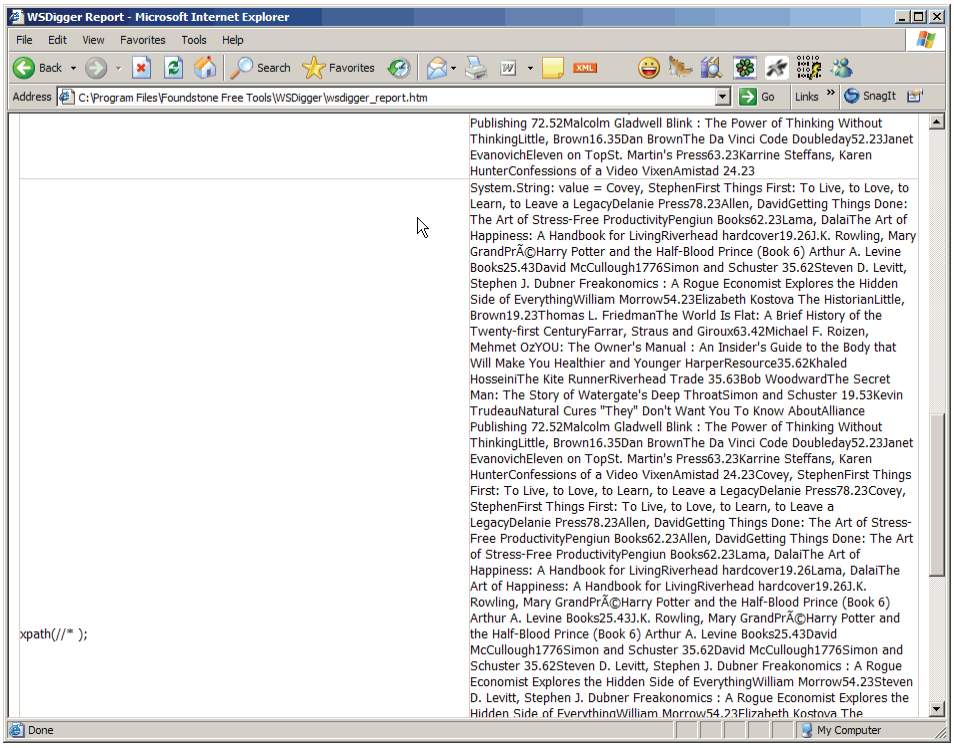


Launch WSDigger and hit next to get to the WSDL screen. Enter http://localhost/WSDigger_WS/WSDigger_WS.asmx?wsdl in the WSDL textbox and hit 'Get Methods'. The result displays methods enumerated.



Browse to Attack ->Select Attack Types. Select all 3 attack plug-in and hit enter. Once the tool is finished, browse to Report -> Generate Report. The report is generated and displayed in the default browser. The web service is vulnerable to XPATH Injection and dumps the complete database.

Foundstone®



Foundstone®

About Foundstone Professional Services

Foundstone Professional Services, a division of McAfee, offers a unique combination of services and education to help organizations continuously and measurably protect the most important assets from the most critical threats. Through a strategic approach to security, Foundstone identifies, recommends, and implements the right balance of technology, people, and process to manage digital risk and leverage security investments more effectively.

Foundstone's Secure Software Security Initiative (S3i™) services help organizations design and engineer secure software. By building in security throughout the Software Development Lifecycle, organizations can significantly reduce their risk of malicious attacks and minimize costly remediation efforts. Services include:

- Source Code Audits
- Software Design and Architecture Reviews
- Threat Modeling
- Web Application Penetration Testing
- Web Services Security Assessment
- Software Security Metrics and Measurement

For more information about Foundstone S3i services, go to www.foundstone.com/s3i.

Foundstone S3i training is designed to teach programmers and application developers how to build secure software and to write secure code. Classes include:

- Building Secure Software
- Writing Secure Code – Java (J2EE)
- Writing Secure Code – ASP.NET (C#)
- Ultimate Web Hacking

For the latest course schedule, go to www.foundstone.com/education.

