Contents

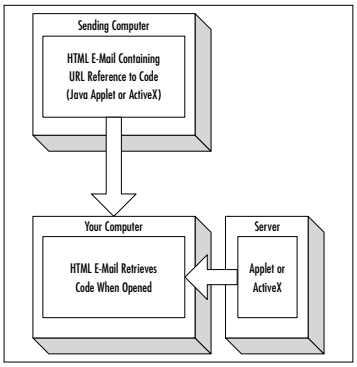
Understand how rogue applets can transmit bad code:

Mobile code applications, in the form of Java applets, JavaScript, and ActiveX controls, are powerful tools for distributing information. They are also powerful tools for transmitting malicious code. Rogue applets do not replicate themselves or simply corrupt data as viruses do, but instead they are most often specific attacks designed to steal data or disable systems.

Thinking Creatively When Coding

- Be aware of outside influences on your code, expect the unexpected!
- Look for ways to minimize your code; keep the functionality in as small a core as possible.
- Review, review, review! Don't try to isolate your efforts or conceal mistakes.

Understand how mobile code works for Java applets and ActiveX controls:



Mobile Code Residing on a Web Server

Tools & Traps…Beware of User Input

One of the most common methods of exploiting CGI scripts and programs is used when scripts allow user input, but the data that users are submitting is not checked. Controlling what information users are able to submit will reduce your chances of being hacked through a CGI script dramatically.

Answers All Your Questions About Hacking Techniques

Q: What should I do if I stumble across a back door in my code base?

A: First and most importantly, determine that it is a genuine back door. Segments of code often appear to have no authentication aspect and can do some rather powerful things, but nonetheless had proper authentication performed prior to their being called. If your best research still indicates that it is a back door, contact an associate in your security department who understands the language in which you're coding and request a review of the code. If that person determines it is a back door, it should be investigated to determine whether the code was introduced simply due to poor planning or actual malice.

How to Efficiently Trace through a Program

- ☑ Tracing a program's execution from start to finish is too time-intensive.
- ☑ You can save time by instead going directly to problem areas.
- ☑ This approach allows you to skip benign application processing/calculation logic.

Complete coverage of the Java Security Model:

- Class loaders
- Byte-code verification
- Security managers
- Digital signatures
- Authentication using certificates
- JAR signing
- Encryption

Damage & Defense: Debugging XSL

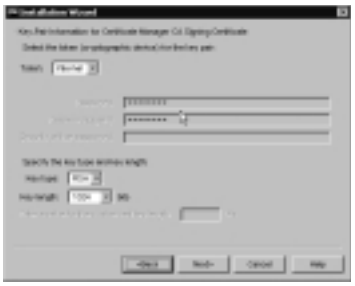
The interaction of a style sheet with an XML document can be a complicated process, and unfortunately, style sheet errors can often be cryptic. Microsoft has an HTML-based XSL debugger you can use to walk through the execution of your XSL. You can also view the source code to make your own improvements. You can find the XSL Debugger at http://msdn.microsoft.com/downloads/samples/internet/xml/sxl_debugger/default.asp.

Use ActiveX and understand the Authenticode Security Warning

Write Secure ColdFusion Code:

When writing a ColdFusion application, you must look out for a number of tags that involve the movement of data in ways that can be attacked. In most cases, validating the data sent to a page will prevent them from being misused. In others, not allowing attributes to be set dynamically is the answer. For each tag we examine, another solution may be to just turn the tag off (an option controlled by the administration panel). Other tags can not be turned off and must be coded properly.

Select Cryptography Token, Key Type, and Key Length



Set up a checklist of defects not easily detected through standard testing methods for working in a Java environment:

- Excessive copying of strings—unnecessary copies of immutable objects
- Failure to clone returned objects
- Unnecessary cloning
- Copying arrays by hand
- Copying the wrong thing or making only a partial copy
- Testing new for null
- Using == instead of .equals
- The confusion of nonatomic and atomic operations
- The addition of unnecessary catchblocks
- Failure to implement equals, clone or hashcode