

How Web Spoofing Works



SystemExperts Corporation
Brad C. Johnson

Abstract

In networking lingo, spoofing means pretending to be something you are not. Let's look at some specific examples:

- Mail Spoofing is pretending to be somebody else in email (e.g., sending mail as though you are bill.clinton@whitehouse.gov when actually you are poor.slob@nowhere.com).
- IP Spoofing is pretending to be somebody else's machine (e.g., pretending you are the trusted Intranet host with IP address 127.35.214.16 when actually you are the untrusted Internet host with IP address 212.58.128.4).
- Web Spoofing is pretending to be somebody else's web site.

Given that a Universal Resource Locator (URL) is very specific about the protocol, host, and resource that is being accessed, how is pretending to be somebody else's web site possible? The short answer is, your request to read a page is intercepted by an intermediary, it makes a copy of the valid page, and make changes to the data before giving it back to you. This intermediary does not actually change the data at the web site you asked for, they change a copy of the data you wanted to see. Not only is all of this possible, but most importantly, it does not require any unusual skills or technology to execute. *Web Spoofing* allows somebody both to see (for example, credit card numbers or account passwords) and change data that you requested from another site **even if you are using SSL!**

How Web Spoofing Works

- What they meant, but didn't say in "Web Spoofing: An Internet Con Game," Princeton University"
- HTML Web Spoofing Commands

Inside

- Normal HTML Operations
- Abnormal HTML Operations
- Web Spoofing
- How to Protect Yourself

Contact Information

SystemExperts Corporation

Toll Free (USA only): +1 888 749 9800
From outside USA: +1 978 440 9388

<http://www.SystemExperts.com>
info@SystemExperts.com

Normal Operation

To lay the groundwork before preceding more deeply into *Web Spoofing*, let's first review normal browser-server interactions. Normally, a person accesses Web content on the network through their Web browser. Most people use Netscape, Mosaic, or Microsoft browsers. Some people use browsers supplied by their On-line Service company -- such as Prodigy, America On-line, or CompuServe. If you get confused and find yourself thinking that *Web Spoofing* can't happen to me, you are wrong. *Web Spoofing* can and does happen and it does not matter what browser you use if it is an HTML based browser (HTML is the language that Web pages are written in).^{*} Guess what? All browsers are HTML browsers. Yep, even Java based browsers are really using HTML.

OK, so you are using a browser and you want to see something. Everything on the net (both inTER or inTRA net) is referenced by something called a Universal Resource Locator (URL). Every time you go to get something you are really saying to your browser "go get me that URL" -- we will call that "GET stuff" from now on. Most people have seen a URL, they can look something like:

<http://search.yahoo.com/bin/search?p=science+magazines>

So the browser sends a message over the network to something called a server: a Web server to be exact. The Web server looks at the message and it sees "GET stuff." As long as the requested resource exists, the Web server says "OK" and sends back the Web stuff you were looking for. That is how things normally work.

Summary: Your Web browser says "GET stuff" to a Web server, and it sends back the page that was requested.

^{*} Web Spoofing is accomplished by changing the URL that a page is pointing to. This technique is referred to as *URL rewriting*. As pointed out in the Princeton paper (see next footnote), Fred Cohen described the use of URL rewriting as a useful attack technique in "50 Ways to Attack Your World Wide Web System," Computer Security Institute Annual Conference, Washington, DC, October 1995. There are at least two web sites that offer examples of the URL rewriting, they are:

<http://www.anonymizer.com/> and

<http://www.metahtml.com/apps/zippy/welcome.mhtml/>

Abnormal Operations

Under the right conditions, when you ask your browser to "GET stuff" instead of going directly from your browser to the Web server that has the content you want to see, your request goes through a "middle-man." Let's call this an intermediary. How can that happen? Well, there are many ways but the three most likely are this (the first is good and the other two are bad):

- Access to the Web site is directed through a proxy server: this is called an (HTTP) application proxy. This allows for more finely grained management of access to the server. This is a good technique that is used at many sites, however, it does not prevent Web Spoofing.
- Somebody puts a false link in a popular Web page (that is, they hacked it). This is really bad and probably illegal.
- You use a search engine (like Yahoo, Alta Vista, Excite, or one from your On-line Service company) to find out where links are to interesting topics. Unknown to you, some of these links have been put in by bad people that pretend to be somebody they are not. Such as, a search for banks gives you <http://www.chasebank.com> -- unfortunately, you might not know that Chase Manhattan Bank's actual URL is <http://www.chase.com>. This is bad and often malicious but probably not illegal.

Summary: It is easy to get pointed at the wrong Web server and you just can't tell. Whenever you access a web site, if you're not sure about the link you are following, you may be at risk of following the wrong path.

Web Spoofing

OK, so now you know how things are supposed to work and that somebody could actually be in the middle of the browser-server exchange (*that is not where you wanted to go today!*). Now what? Well, remember our friend the URL. Everything is built around that. The only way the intermediary can keep on being in the middle is by intercepting all of your requests, connecting to web sites on your behalf, and then returning all the content (possibly with changes) back to you. They do that by changing all of the URLs in a copy of the first page you requested to make sure they stay in the middle! That way, when you go to

make your subsequent requests to "GET stuff" it always goes through the intermediary first.*

Summary: Somebody can put a Web server in between your browser and the Web server you really want to "GET stuff" from and you just can't tell.

HTML Spoofing Commands

Remember that all browsers are using HTML? Well, there are many HTML commands. However, there are only a few that have URLs in them. So the intermediary has to make sure that it looks for all of these special commands and change them in the pages they copied for you. So, if the actual URL was

http://www.yahoo.com

the changed URL would be

http://intermediary/http://www.yahoo.com

When your browser goes to "GET stuff" it thinks "**http://intermediary**" is the place where the Web server is and "**http://www.yahoo.com**" is the content you are trying to get. The intermediary Web server sees the request and knows that "**http://www.yahoo.com**" is where you really wanted to go, and calls that Web server for you. After it makes a copy of the page you requested at Yahoo!, it looks for all of the special HTML commands that may reference a URL and changes them before giving it back to you.

Here are the special commands it looks for (this is not exact HTML syntax).

To link to something.

<APPLET CODEBASE="URL">
To define a Java Applet location.

<AREA HREF="URL">
To define the area of a section.

<BODY BACKGROUND="URL">
To define the background image.

<EMBED SRC="URL">
To insert an object into a page.

* See Figure 1 at the end of the document.

<FORM ACTION="URL">
To define a form.

<FRAME SRC="URL">
To define the source for a frame.

To display an image.

<INPUT "URL">
To define the source for input.

<META URL=URL">
To perform a client side pull.

Summary: The commands a Web Spoofer must look for are obvious.

Conclusion

At this point, all the intermediary has done is change special HTML commands that reference URLs to ensure that future requests will go through them first. That alone is bad enough. People don't normally react well to finding out they are being tricked. What is worse is that most people who would do this are bad and change other things as well. If they don't change things, they are looking at things they shouldn't be: such as, your credit card number, personal or private information, account passwords or IDs, or whatever.

So what can you do about it? If you are responsible for managing a web site, normal host and server hygiene will help. Such as:

- Using URL checking software to ensure that your links point to expected locations. Most of these programs will show you a listing of all of the URL links that are referenced in your web pages that can be visually scanned for inaccuracies.
- Using host security policies and procedures to ensure that critical files can not be modified without notice. For example, one could use some type of access control method to either deny access or log a message (or both) if somebody attempts to modify the files that contain your web pages.

If you are a user of web sites that require private or critical information (such as credit card numbers, bank account data, personal information), you can take precautionary measures like:

- Enabling your browser to show the URL you are accessing --this allows you to see the actual URL that is

be referenced when you follow a link. In Netscape, for example, you pull down the Options header and enable Show Location.

- Be appropriately suspicious of odd behavior when accessing critical Web sites ñ for example, if you are using a Web based banking transaction, be leery of unannounced changes to the application.
- Before giving out critical personal information (e.g., a credit card, your social security number, etc.), attempt to validate the Web site you are accessing. For example, you could review the URL to see if it is pointing to a place that “makes sense.” you could use the Internic WhoIs service to see if the domain is registered to the company you believe you are dealing with,* or you could call the contact phone numbers (hopefully a toll free call) listed on the Web page and ensure the program is supposed to be asking for personal information (*and you might as well ask how they intend on protecting that information!*).

In any event, it is important to remain vigilant and aware -- but not paranoid. Practical security is based on the concept of managed risk: providing appropriate policies and procedures for a given resource. If the resource being accessed is extremely important, you should be willing to act accordingly if that site is “misbehaving” -- that is, notify your system administrator, register a complaint with your manager or local web administrator, notify the web administrator at the real site, notify CERT (if the *Web Spoofing* is confirmed), or trigger a local security event or alarm.

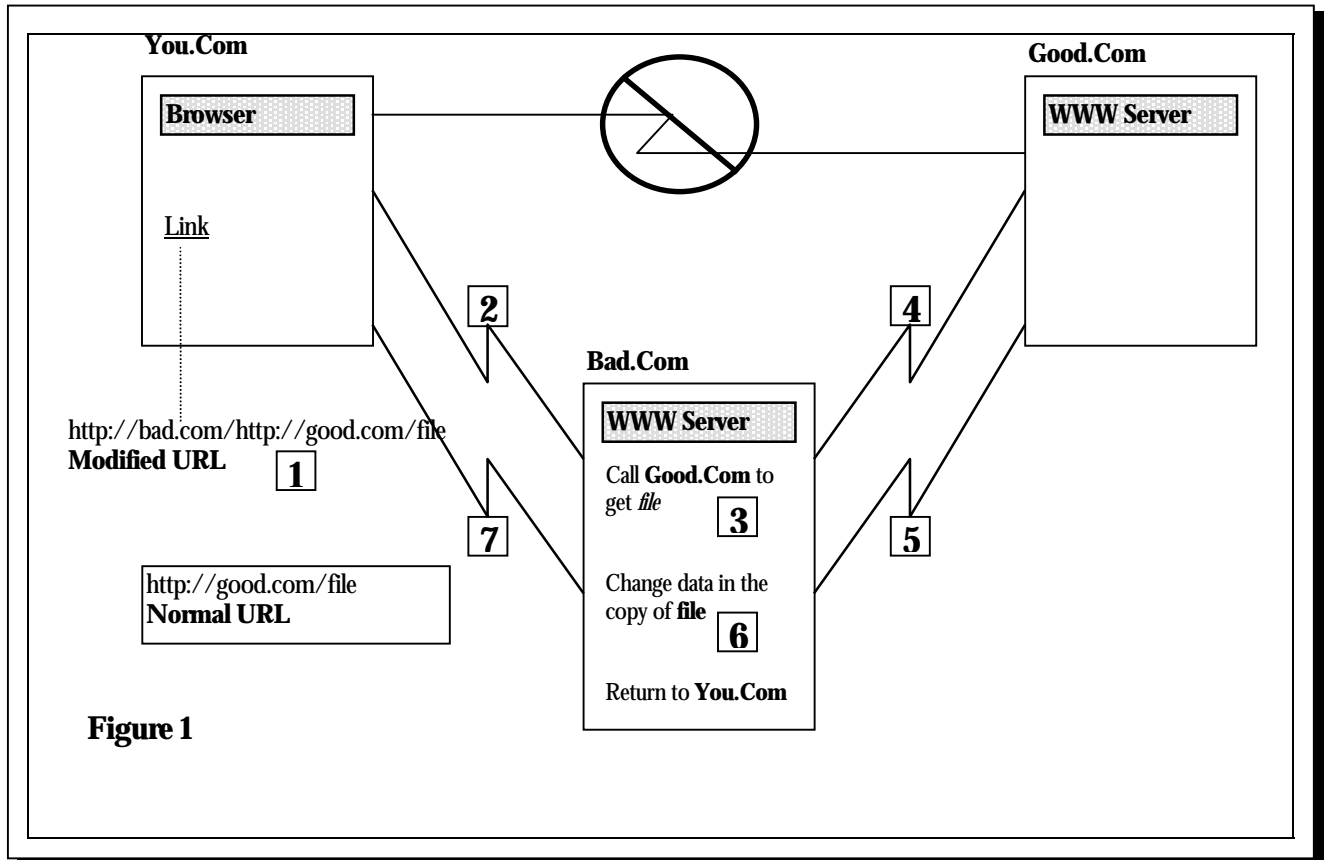
***Remember the old saying:
just because you're paranoid,
doesn't mean that somebody
isn't trying to get you!***

Summary: Web Spoofing can be done and some people already know how to do it. *

* Refer to <http://rs.internic.net/cgi-bin/whois>. For example, use the preceding URL and type in “internic.net”.

* The original paper on Web Spoofing was written by EdwardW. Felton, Dirk Balfanz, Drew Dean, and Dan S. Wallach. The paper is titled, “Web Spoofing: An Internet Con Game”. That document can be found at: <http://www.princeton.edu/sip/pub/spoofing.html>

Here is how a spoofed web session flows.



1. Somehow, the URL that your client browser is referencing on You.Com has been prefaced with the address of the intermediary web site – Bad.Com.
2. Your browser determines that Bad.Com should handle the URL request .
3. The web server on Bad.Com reads the URL and determines that the you are actually trying to reach a file on Good.Com.
4. The Bad.Com server calls Good.Com and asks for the specified file.
5. The Good.Com server returns the page as requested.
6. At this point, the Bad.Com server can change its copy of the page you asked for.
7. After these changes, the Bad.Com server returns the page to you.

About SystemExperts Corporation

SystemExperts Corporation is a recognized leader in the fields of host, network, and electronic commerce security. Our distinguished staff is drawn from forerunners in the open systems industry, including AT&T Bell Laboratories, MIT's Project Athena, BBN Corporation, the Open Group (Open Software Foundation & X/Open), CertCo, Open Market, Lawrence Livermore National Laboratories, as well as from distributed computing vendors like Sun Microsystems, Digital Equipment Corporation and Motorola, and users of client/server computing such as SAIC, the U.S. Navy, and Mobil Oil.

In addition to our hands-on and high level strategy and architectural work with our clients, the credentials of our staff are recognized industry-wide. Our courses on Secure Electronic Commerce, Kerberos, Penetration Testing, Firewalls, Intrusion Detection, and Sendmail/DNS, have consistently been among the highest rated and best attended tutorials at major conferences such as Usenix, LISA, SANs, NetWorld-Interop, Internet World, CSI, and the MIS Training Institute.

SystemExperts provides consulting services on a fixed-price basis. This approach enables our customers to budget accurately for our services and know, in advance, how long a project will take. We apprise our clients of just what to expect before a project begins by clearly defining and committing to specific deliverables as well as costs that we "cast in stone" before a project begins. As a result, our clients avoid "eternal projects" that are typical of traditional time and materials based consulting. Our consultants provide a wide range of services for a diverse customer base in many industries. The list below is a sampling of areas in which we have directed clients.

- Network & Host Security (including penetration testing, incident response, audit preparation, security policy development, and web application exposure profiling)
- Electronic Commerce and WWW Design & Implementation
- Technology Assessment, Strategies & Architectures
- Intrusion Detection and Event Management
- "Guru" level expertise in Firewalls, DNS, Sendmail, System Administration, and Databases
- VPN Design and Implementation
- Middleware, Distributed Computing, and OSF's DCE

Boston

<http://www.SystemExperts.com>

New York

Telephone: 888 749 9800

San Francisco

Washington

info@SystemExperts.com



Providing Leadership in System Security & Management