Global Information Assurance Certification Paper

Donna M. Shuart

Hyperlink and Web Spoofing: Identifying and Defending Against Hacker Attacks

Administrivia Version 2.0

GSEC Practical Assignment
Version 1.3

As you know, hackers can attack systems in many ways. This paper discusses a specific attack hackers commit against network servers – spoofing. Using spoofing, the hacker fakes an IP address to simulate a trusted server within an existing network connection (Klander, 247). IP spoofing is a complex technical attack that is made up of several components. In actuality, IP spoofing is not the attack, but a step in the attack (Daemon9). The attack is actually trust-relationship exploitation. However, in this paper, IP spoofing will refer to the whole attack.

Hackers can use hyperlink spoofing to attack Secure Socket Layer (SSL) server installations. Web spoofing provides hackers with a way to intercept all transmissions a user or server forwards during an HTTP transaction series (Klander, 282). It is easy to detect spoofing attacks, as long as the user is aware of the signs that they are under attack.

Overview of Spoofing

Fundamentals of an IP Spoofing Attack

Transport Protocol (TCP) and Uniform Datagram Protocol (UDP) services assume that a host's Internet Protocol (IP) address is valid, and therefore trust the address. A hacker can use IP source routing to specify a direct route to a destination and a return path back to the origination. By using routers or hosts not normally used to forward packets to the destination, the hacker is able to intercept or modify transmissions without encountering packets destined for the true host (Klander, 262-263).

In order to impersonate a particular server's trusted client, a hacker would have to change the impersonating host's IP address to match the trusted client's address. The hacker would then construct a source route to the server that specifies the direct path the IP packets should take to the server and should take from the server back to the hacker's host. The source route is used to send a client request to the server. The server accepts the client request as if the request came directly from the trusted client, and then returns a reply to the trusted client. Using the source route, the trusted client forwards the packet on to the hacker's host (Klander, 262-263).

Many Unix hosts accept source-routed packets and will pass those packets on as the source route indicates. Many routers will accept source-routed packets as well. However, it is possible to configure some routers to block source-routed packets. The figure below shows the fundamentals of an IP spoofing attack (Klander, 263).

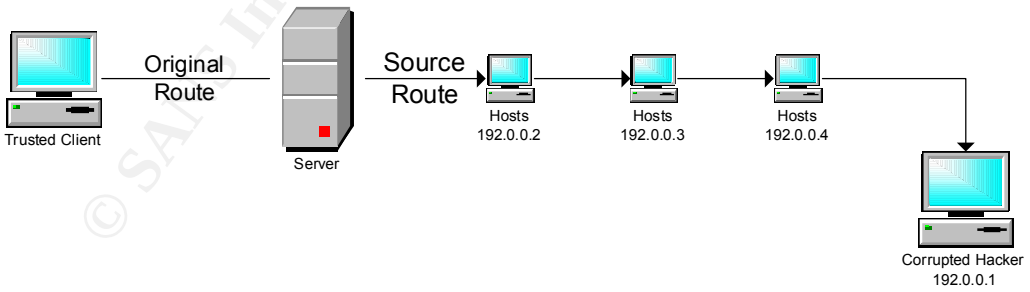

Figure 1 *The fundamentals of an IP spoofing attack.*

A simpler method for spoofing a client is to wait until the client system has shut down and then impersonate the client's system. A hacker could pose as the real client and configure a personal

computer with the same name and IP address as another computer's, and then initiate connections to the Unix host. Such a spoofing attack could be easy to accomplish, especially for an "insider" because only an insider is likely to know which computers within the protected network are shut down (Klander, 263-264).

A common misconception is that IP spoofing can be used to hide your IP address while surfing the Internet, chatting on-line, sending email, and so forth. This is generally not true. Forging the source IP address causes the responses to be misdirected, meaning you cannot create a normal network connection (Network ICE).

Spoofing Email

Email on the Internet is especially easy to spoof, and you should not trust email without enhancements such as digital signatures. The exchange that takes place when Internet hosts exchange mail uses a simple protocol that uses ASCII-character commands. An intruder could easily enter these commands manually using Telnet to connect directly to a system's Simple Mail Transfer Protocol (SMTP) port. The receiving host trusts the sending host's identity, so the intruder can easily spoof the mail's origin by entering a sender address that is different than the intruder's actual address. As a result, any user without privileges can spoof email (Klander, 264).

The only way to be sure that an email is from who it claims to be from is through signature verification. If, however, you receive a large number of spoofed emails, you will often be able to track the spoofer through viewing the email's header information, which will often include the actual originating server of the intruder. Knowledge of the originating server allows you to speak with the systems administrator at the originating server and see if there is any means of blocking the intruder from future spoofing attacks (Klander, 265).

Detecting Spoofing

IP spoofing attacks are difficult to detect. If your site has the ability to monitor network traffic, you should audit incoming traffic passing over the router. When you audit traffic, you keep a record of the traffic within a system log. Using the audit record, you should examine incoming traffic for packets with both a source and destination address contained within your local domain. You should never find packets containing both an internal source and a destination address entering your network from the Internet. If you find packets containing both addresses crossing your router, it likely indicates that an IP spoofing attack is in progress. Two freely available software tools, tcpdump and netlog, can assist in packet monitoring on Unix systems (Klander, 265). Among the SANS top-20 vulnerability list is the failure to filter packets for correct incoming and outgoing addresses (Lyman).

Preventing Spoofing

As discussed above, both addresses within a spoofed packet will most often be contained within your local domain, although the spoofed packet may contain the IP address or a trusted host outside the network. The best defense against IP spoofing attacks is to filter packets as the packets enter your router from the Internet, thereby blocking any packet that claims to have originated inside your local domain. Several router brands support this packet-filtering feature, known as an input filter. Some of the router brands that support packet-filtering include:

- Bay Networks/Wellfleet, version 5 and later
- Cabletron with LAN secure
- Cisco, RIS software version 9.21 and later

- Livingston

If the router hardware does not support packet-filtering on inbound traffic, a second router can be installed between the existing router and the Internet connection. The second router can be used with an output filter and be used to filter spoofed IP packets (Klander, 266).

Hyperlink Spoofing

Attacks on SSL Server Authentication

Hyperlink spoofing is one common attack hackers can use against computer communications using the hypertext transport protocol (HTTP). Hackers can perform attacks on the Secure Socket Layers (SSL) server authentication protocol used in creating secure Web browsers and servers. A "man-in-the-middle" hacker can persuade the browser to connect to a fake server while the browser presents the usual appearances of a secure session. A "man-in-the-middle" hacker is a hacker who inserts himself into the packet stream between a client and a server. The hacker can then persuade the user to reveal information such as credit card numbers, personal identification numbers (PINs), insurance and bank details, or other private information to the fake server. Another risk of hyperlink spoofing is that the user may download and run malicious Java applets from the fake server, believing the applets to be from the real server (Klander, 267).

Hyperlink spoofing attacks take advantage of flaws in the way that most browsers employ digital certificates to secure Web sessions. Hackers can impersonate any SSL-enabled server following the usual certificate conventions or by accessing the browser. In addition, server certificates are susceptible to the hyperlink spoofing attack when the browser uses either Internet Explorer or Netscape Navigator. The best long-term solution is probably modification to both the certificate content and the normal Web browser (Klander, 268).

Hyperlink Spoofing's Background

When a user makes an SSL connection, the browser and the server share a protocol in order to authenticate the server and sometimes the client. The hyperlink spoofing attack concerns itself only with server authentication. During the SSL's initial protocol exchange, the server presents the browser with the server's certificate. The server's certificate is a digitally-signed structure binding the server's public key to certain attributes. The SSL protocol uses a domain-name server (DNS) name in the certificate. Alternately, the certificate may include a "wildcard" rather than a complete DNS name. By transporting the protocol correctly, and by presenting a valid certificate that the client trusts, the server proves to the browser that it has the corresponding private key, which only the server knows. The browser accepts the proof and knows that the server has the right to use the claimed DNS name. It is important to recognize that, for the hyperlink spoofing attack, SSL is not the problem. Rather, the certificate contents and the browser user interface are at issue (Klander, 268).

The Hyperlink Spoofing Attack

The hyperlink spoofing attack is successful because most users do not request to connect to DNS names or to URLs – they follow hyperlinks. Just as DNS names are subject to DNS spoofing, URLs are subject to hyperlink spoofing, in which a page lies about a URL's DNS name. Both spoofing forms will direct you to the wrong Internet site. However, hyperlink spoofing is technically much easier than DNS spoofing (Klander 268). Today's browsers can detect if you have a secure connection even though a hacker has just spoofed you. The hacker will use certain tricks to tell the browser to indicate that you have a private connection with the intended server. Unfortunately, while you have a private connection, it is with the wrong server, and the hacker

would have made the target page look like the genuine Web page, which eventually may have prompted you for your credit card information or other personal information. However, if you dig deeper into the browser's menus and view the document source or the document information, you will notice that the server's authenticated identity is not what you expected (269).

According to Klander, as server certificate use becomes more widespread, defeating server authentication becomes easier, not harder. As more servers have certificates, hackers have more choices of sites to which they can redirect the trusting Web surfer. Also, many users will turn certificate dialogs off if their browser notifies them each time they enters a new Web page. Moreover, if every connection and document is secure, then knowing you have requested a secure document is not particularly helpful (Klander, 269).

Despite heavy authentication, no audit trail exists to tell the user what happens in the event of a hyperlink spoof. If the user is lucky, a local cache may store the doctored page. However, the hacker can easily force the page form the cache (Klander, 269).

A hacker does not want to send you their secure site, thereby giving you their certificate and a huge identity clue. Instead, the hacker can send you to someone else's SSL box that they have broken into. The hacker can send you somewhere else within the secure site to which you actually wanted to go. This type of misdirection can occur on virtually-hosted Web sites, or Web sites where the URLs represent common gateway interface (CGI) scripts or Java classes (Klander, 269).

Possible Fixes to Prevent Hyperlink Spoofing

One possible solution to prevent hyperlink spoofing is to make the users' browsers start up on a secure page, so that users can trust their initial links and a hacker can never send them anywhere suspicious (Klander, 270).

If you want a user's browser to open on an SSL page, you must send the URL for that page by some difficult- or impossible-to-intercept means such as a floppy disk or a paper memo through the mail. Otherwise, the page creates an opening to the attack you want to prevent. All links out of this page should send users to trustworthy sites, and preferably all links should be SSL links. You can determine what sites you categorize as trustworthy sites based on the following two criteria:

1. The site is securely-run (that is, the entire site is secured against attack and page interception).
2. The site only serves pages with hyperlinks to sites that are run securely.

Most sites fail on the first criterion, so secure-page access fix is probably only viable for intranet applications behind a firewall, or for specialized Internet applications where users will access only your corporate Internet sites and therefore do not require general Internet access. Another possible fix to consider is one that can be deployed quickly. Most commercial Web browsers provide security options. One of these security options lets you monitor site-based certificates (Klander, 270).

A browser plug-in that displays the site certificate for each site that browsers on your network access can be quickly programmed. After installing the plug-in, your users will always see who owns the site the user connected to, which may prevent some attacks. The plug-in could turn the information dialog box on by default, maybe with a password-dependent option for the network administrator to turn the dialog box off. In addition, the certificate information could be shown

in the browser window after the browser and server connection is made. The certificate display would give constant feedback about the owner of the side delivering the current page (Klander, 271). Unfortunately, this certificate display will probably not help prevent the spoofing attack if the spoof redirects the client HTTP request or the server's HTTP response from one page to another page within the same Web site (272).

A final possible fix is the use of trusted bookmarks. Internal security personnel would verify each trusted bookmark before distribution to the individual bookmark file. Additionally, security personnel would transfer each trusted bookmark using only manual means, such as a floppy disk. The trusted bookmark would be marked in some way within the browser's bookmark file so that it is clearly apparent which bookmarks are trusted and which are not trusted. The trusted bookmark fix requires that your organization create a browser plug-in which enforces the trusted bookmarks. A trusted bookmark would map from hyperlink text or images to domain names or Universal Resource Locators (URLs) kept in the browser. In the event the browser connects to a third-party domain, the browser would warn the user that the connected domain does not match the expected domain (Klander, 272).

The Long-Term Fix for Hyperlink Spoofing

Hyperlink spoofing presents a serious risk to a user's security as the user browses the Web. The basic problem that hyperlink spoofing takes advantage of is that the SSL-provided certificate contains the wrong information – the name of the domain name server (DNS). The DNS name is a more technical detail than the URL, which is itself a more technical detail than the hyperlink that the user clicks upon. As more and more users connect to the Web, the less technical the information the certificate displays, the better. Most people guess at URLs; however, just because a URL seems as if it should belong to a certain company, does not mean that it does (Klander, 273).

Despite the fact that URLs are generally guessable and typically reflective of the company who owns the Web site the URL refers to, it is mostly convention and not law that makes it that way. When a domain name is registered, Internet authorities ensure that the registered DNS does not violate copyright laws. The DNS does not have to be reflective of the company's name or even their business. Basically, most users access the Web in terms of URLs, not DNS names, and they expect the text or graphical link to those URLs to be reflective of the URLs destination (Klander, 273).

The problem that Internet certification bodies must solve is determining what server certificates should certify. Determining what the certificates should certify probably depends on the application, because the application is, in many ways, indicative of the user's sophistication level. For some applications, the DNS name is fine for certificate display, since the DNS name is what the user enters to access the site, and the user probably has a greater understanding of the DNS name's meaning. However, for browsing, the certificate should include the hyperlink image or text, or something meaningful provided to the browser by the certified page (Klander, 274).

If the certificate includes an image, the browser could put the image within a dialog box. Alternately, the browser could display the image within the browser image box instead of the Netscape or Explorer logo. Placing the certificate within the image box gives the user constant feedback as to who they are currently connected to (Klander, 274).

The browser could automatically verify image certificates with some simple modifications. If the user followed an image link to reach the current page, the browser could check the certificate to ensure the image appeared. If the image did not appear in the certificate, the browser would alert the user. Similarly, if the user followed a text link to reach the current page, the browser could look for that text. For example, the browser could parse the certificate for a sub-string of the text link, or a known hash function's result of the text link. Verifying the image or text link would positively authenticate the link the user followed as being accurate, in a comprehensive means that does not allow spoofing (Klander, 274).

The Computer Incident Advisory Capability (CIAC) suggests the following in regards to packet filtering:

> With the current IP protocol technology, it is impossible to eliminate IP spoofed packets. However, you can take steps to reduce the number of IP spoofed packets entering and exiting your network. Currently, the best method is to install a filtering router that restricts the input to your external interface (known as an input filter) by not allowing a packet through if it has a source address from your internal network. In addition, you should filter outgoing packets that have a source address different from your internal network to prevent a source IP spoofing attack from originating at your site. The combination of these two filters would prevent outside attackers from sending you packets pretending to be from your internal network. It would also prevent packets originating within your network from pretending to be from outside your network. These filter will not stop all attacks, since outside attackers can spoof packets form any outside network, and internal attackers can still send attacks spoofing internal addresses. We strongly urge Internet service providers to install these filters in your routers. In addition, we strongly recommend customers of Internet service providers to contact your service provider to verify that the necessary filters are in place to protect your network (U.S Department of Energy).

Web Spoofing

Introduction to Web Spoofing

Web spoofing is another type of hacker attack in which the hacker creates a convincing, but false, copy of the entire Web. The false Web looks like the real one, including all the same pages and links as the real Web. However, the hacker completely controls the false Web so that all network traffic between the victim's browser and the Web goes through the hacker. The figure below illustrates a conceptual model of Web spoofing (Klander, 275).

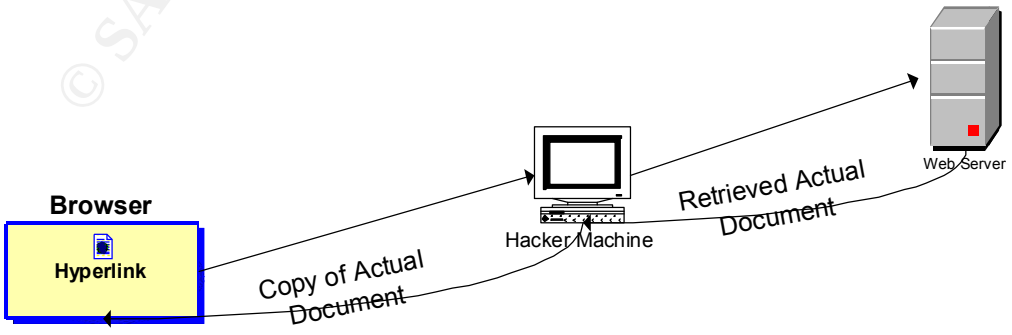

Figure 2 *The conceptual model of the Web-spoofing attack.*

Web Spoofing's Consequences

Web spoofing allows the hacker to observe or modify any data going from the victim to Web servers. Also, the hacker can control all return traffic from Web servers to the victim. According to Klander, spoofing is one of two of the most common methods that hackers use to break into networks. The other method is sniffing, a surveillance-type activity because the hacker passively watches network traffic. Spoofing is a tampering activity because the hacker convinces a host computer that the hacker is another, trusted host computer, and therefore should receive information (Klander, 275).

While Web spoofing, the hacker records the contents of the pages the victim visits. When the victim fills out a form on an HTML page, the victim's browser transmits the entered data to the Web server. Because the hacker is interposed between the client and the server, the hacker can record all client-entered data. Additionally, the hacker can record the contents of the response the server sends back to the client (Klander, 275).

It is even possible for the hacker to spoof the entire World Wide Web because the hacker does not need to store the Web's entire contents. By definition, the whole Web is available on-line, so the hacker can fetch a page from the real Web when it needs to provide a copy of the page on the false Web (Klander, 276).

How the Attack Works

The key to the Web spoofing attack is for the hacker's Web server to sit between the victim and the rest of the Web, known as the "man-in-the-middle attack." The hacker's first step is to rewrite all URLs on some Web page so that the URLs point to the hacker's server rather than to a real server. When the victim reaches the rewritten URL page, the URLs will look normal to them because the hacker has spoofed the URL. After the hacker's server has retrieved the real document needed to satisfy the request, the hacker rewrites the URLs in the document into the same special form the hacker used to initially spoof the victim. Because all URLs in the rewritten page point to the hacker's server, if the victim follows a link on the new page, the hacker's server will again retrieve the page (Klander, 276-277).

Forms and Secure Connections

For the same reasons that hackers can spoof any URL, hackers can also spoof any form on a Web page. Just as Web page requests go to the hacker's server, so do victim-submitted forms. The hacker's server can observe and modify the victim-submitted data. Therefore, the hacker can change the data as much as he desires before passing the data on to the real server. The hacker's server can also modify the data returned in response to the form submission (Klander, 277).

The Web spoofing attack works even when the victim requests a page with a secure connection. For example, if the victim tries to do a secure Web access using S-HTTP or the Secure Sockets Layer in a false Web, the browser display will appear as usual. The hacker's server will deliver the page, and the victim's browser will turn on the secure connection indicator. The browser will inform the victim that the browser has a secure connection with a server because the browser does have a secure connection. Unfortunately, the secure connection is to the hacker's server, and not to the desired Web page (Klander, 277).

Starting the Web Spoofing Attack

It is difficult to escape a Web spoofing attack once it begins. However, starting the Web spoofing attack requires action on the part of the victim. To start the attack, the hacker must somehow lure the victim into the hacker's false Web. A hacker can make the false hyperlink more accessible to victims in several easy ways, including the following:

- A hacker can put a link to the false Web onto a popular Web page.
- If the victim uses Web-enabled email, the hacker can email the victim a pointer to the false Web.
- Alternately, the hacker can email the victim the contents of a page in the false Web.
- The hacker can trick a Web search engine into indexing part of a false Web.
- If the victim uses Internet Explorer, the hacker might write an ActiveX control that Explorer executes each time the victim runs the browser. The hacker's ActiveX control might replace a normal, correct URL with a hacked URL.

The important issue is that the hacker must draw you into the false Web somehow (Klander, 278).

The Status Bar

The Web spoofing attack is not perfect because the attack must convince victims that they are still within the real Web. If the hacker is not careful, or if the victim has disabled certain options within the browser, spoofed Web pages will display certain page information within the status bar. The page information may provide the victim with enough information to realize their entry into the false Web (Klander, 278).

The Web spoofing attack leaves two kinds of evidence on the status bar. First, when you hold the mouse pointer over a hyperlink, the browser's status line will display the URL the link contains. Therefore, the victim might notice that the hacker has rewritten the hyperlink's URL. Second, when the browser is retrieving a page, the status line will briefly display the name of the server the browser contacted. Therefore, the victim might notice that they are connected to the incorrect server. Unfortunately, there are ways for the hacker to eliminate such evidence. Because the hacker can write content to the status line, the hacker can arrange things so that the status line participates in the illusion. In addition, the hacker can bind his program to relevant events, always showing the victim the expected status line for the real Web, even when connecting to a new page (Klander, 279).

The Location Line

Like the status bar, the location line can give away the Web spoofing attack. The browser's location line displays the URL of the page the victim is currently viewing. The victim can also type a URL into the location line, instructing the browser to request the resource at that URL. Without further modification, the Web spoofing attack will display the rewritten URL. If the victim notices the rewritten URL, the victim will probably realize that they are under attack (Klander, 279).

Again, the hacker can hide the rewritten URL using an embedded program within the spoofing server that hides the real location line and replaces it with a fake location line that looks correct. The fake location line can show the URL that the victim expects to see. The fake location line can also accept keyboard input, letting the victim type in URLs normally. The embedded program can rewrite typed-in URLs before the browser requests access (Klander, 280).

Viewing Document Information

The final clue that the victim can use to identify an attack is document information. If the victim selects the browser's View Document Information menu item, the browser will display information about the document. This document information includes the document's URL. Like the View Documents Source item menu, the hacker can replace the document information using a spoofed menu bar. If the hacker creates a spoofed menu bar, the hacker can display the document information dialog box using manipulated information. In effect, the hacker can use scripting languages to override all of the possible clues that the victim could access to determine a false Web connection. The only defense the victim might have, once spoofed, is to disable scripting languages within the browser (Klander, 280).

Tracing the Hacker

Because of the nature of the attack, the hacker's server must reveal its location in order to carry out the attack. If the victim detects the attack, the server's location will most likely be available. Unfortunately, hackers performing a Web spoofing attack will probably do so from a stolen computer. Stolen machines are the most likely base for Web spoofing attacks (Klander, 280).

Remedies to the Web Spoofing Attack

Although Web spoofing is nearly an undetectable security attack, there are some preventative measures to protect yourself and network users from this attack. The best defense, according to Klander, is to follow a three-part strategy:

1. Disable JavaScript, Java, and VBScript in your browser so the hacker cannot hide the evidence of the attack.
2. Make sure your browser's location line is always visible.
3. Pay attention to the URLs your browser's location line displays, making sure the URLs always point to the server to which you think you are connected.

This three-part strategy will significantly lower the risk of attack, though a hacker could still victimize users, particularly if those users do not remain conscientious about the status bar and the location line (Klander, 280-281).

Long-Term Solutions to Web Spoofing

Solving the majority of the problems presented by Web spoofing requires action on the part of the browser manufacturers. Changing browser code so that the browser always displays the location line would provide additional security, as would securing the browser from exterior modification; that is, making sure that Web programs could not create false menu bars, status bars, and so on. However, both solutions still presume that users are attentive and know how to recognize rewritten URLs. Without significant internal limitations on modification, the browser is not capable of securing itself from the Web spoofing attack (Klander, 281).

For pages the browser retrieves over a secure connection, an improved secure-connection indicator within a browser could help to ensure security. Rather than simply indicating a secure connection, browsers should clearly state the server name completing the secure connection. The browser should display the connection information in plain language that novices can understand (Klander, 281).

Fundamentally, however, every approach to the Web spoofing problem seems to rely on Web users' attentiveness.

There are several reasons why hackers do what they do. (Rent-A-Hacker, "Hacking, Information Technology Security Services"):

- Cyber pirates seeking profits and information
- Hackers seeking a thrill
- Your competitors seeking to destroy you
- Your ex-employees seeking revenge
- Your employees seeking knowledge and power
- Vendors and business associates seeking leverage

When asked what it is about the computer that makes it become such an obsession, the anonymous hacker stated, "Well, it's power at your fingertips. You can control all these computers from the government, from the military, from large corporations. And if you know what you're doing, you can travel through the Internet at your will, with no restrictions. That's power; it's a power trip (Frontline)."

To sum up, hyperlink spoofing lets hackers attack SSL server installations, while Web spoofing provides hackers with a way to intercept all transmissions a user or server forwards. Although almost every hacker leave trails that you can use to catch or stop them, detection of an attack rests mainly on your attentiveness. Unfortunately, hackers invent new attacks as security professionals defeat each old attack.

References

Daemon9. "IP Spoofing – Demystified." Phrack Magazine. June, 1996.
URL: http://www.fc.net/phrack/files/p-48/p48-14.html (December 14, 2001).

"Hacking, Information Technology Security Services." Rent-A-Hacker.
URL: http://www.rent-a-hacker.com (Sept. 30, 2001).

"Interview: Anonymous." Frontline.
URL: http://www.pbs.org/wgbh/pages/frontline/shows/hackers/intervfiews/anon.html
(Sept. 30, 2001).

Klander, Lars. Hacker Proof: The Ultimate Guide to Network Security. James Press. 1997. Pgs 247, 262 – 281.

Lyman, Jay. "Feds, Security Groups Release Top-20 Vulnerability List." NewsFactor. October 3, 2001. URL: http://www.newsfactor.com/perl/story/?id=13907 (Oct. 2, 2001).

"Spoofing." Network ICE.
URL: http://advice.networkice.com/Advice/Underground/Hackning/Methods/Technical/Spoofing/default.html (December 14, 2001).

U.S Department of Energy. "G-48: TCP SYN Flooding and IP Spoofing Attacks." Computer Incident Advisory Capability Information Bulletin. September 20, 1996.
URL: http://ciac.llnl.gov/ciac/bulletins/g-48.shtml (December 14, 2001).

"Web Spoofing." Secure Internet Programming.
URL: http://www.cs.princeton.edu/sip/WebSpoofing (December 14, 2001).