Shumaila Aslam
BSc Combined Honours Computer Science

Information Security

Spoofing

Spoofing: E-mail, server….. How it is done, how it is detected, how to defend against it.

What is Spoofing?

Definition

spoof (DECEIVE) [Show phonetics]
verb [I or T] US INFORMAL
to try to make someone believe in something that is not true, as a joke

(from Cambridge Advanced Learner's Dictionary)

Web spoofing is the act of secretly tricking your Web browser into talking to a different Web server than you intend. How? By attacking the DNS (domain name system) that maps the "*www.site.com*" in a URL to a network address, or by modifying a Web page to have a bad URL, or by tricking your browser as it interprets CGI data, JavaScript, etc.

After your browser has been fooled, the spoofed Web server can send you fake Web pages or prompt you to provide personal information such as your login ID, password, or even credit card or bank account numbers. If done carefully, you probably will not even notice that you have been duped.

How to Spot a Spoofed Page

Some Web spoofing may be noticeable, so it is helpful to keep these tips in mind:

- If you hold your mouse over a URL that is a link, the status line displays the corresponding URL. Be suspicious if the status line URL is different from what you think you should see.
- When the Web page is being requested, the status line will show the name of the server. Beware if the server name is different from what you expected.
- Your browser's location line is the place to watch for anything unusual about a site's URL.

Unfortunately, clues to a Web spoofing attack can be hidden if the attacker is using JavaScript (which can write to the status line and rewrite location line URLs) or a similar program that makes all requests for a particular URL go to the attacker's system. After obtaining the desired information, the spoofed Web site might even send you to the correct site.

Another way to think about Web spoofing is to be aware of where a link goes--whether to a place you expected or to someplace odd.

Private Information Requests

If Web pages with which you are familiar suddenly ask you to fill in private information, weigh the situation carefully before supplying it. If possible, call or send mail to the official source to verify that this change is legitimate. When in doubt, do not enter any information you feel uncomfortable providing.

Even a secure "https" connection (with Secure Sockets Layer) does not guarantee against surveillance or modification of information you send. If you are already connected to the attacker's system, you may simply be securely connected to the Web spoofer's server.

What to Do

If you think you are a victim of a Web spoof, report it to the official source of the page by phone or via an email address that you know to be correct. If you have been tricked into supplying your password, you should change it immediately.

To learn more about Web spoofing, start with this Web site: "Web Spoofing: An Internet Con Game" at *bau2.uibk.ac.at/matic/spoofing.htm*

Web Spoofing

Web spoofing allows an attacker to create a "shadow copy" of the entire World Wide Web. Accesses to the shadow Web are funneled through the attacker's machine, allowing the attacker to monitor the all of the victim's activities including any passwords or account numbers the victim enters. The attacker can also cause false or misleading data to be sent to Web servers in the victim's name, or to the victim in the name of any Web server. In short, the attacker observes and controls everything the victim does on the Web.

In a *spoofing attack*, the attacker creates misleading context in order to trick the victim into making an inappropriate security-relevant decision. A spoofing attack is like a con game: the attacker sets up a false but convincing world around the victim. The victim does something that would be appropriate if the false world were real. Unfortunately, activities that seem reasonable in the false world may have disastrous effects in the real world.

Spoofing attacks are possible in the physical world as well as the electronic one. For example, there have been several incidents in which criminals set up bogus automated-teller machines, typically in the public areas of shopping malls [1]. The machines would accept ATM cards and ask the person to enter their PIN code. Once the machine had the victim's PIN, it could either eat the card or "malfunction" and

return the card. In either case, the criminals had enough information to copy the victim's card and use the duplicate. In these attacks, people were fooled by the context they saw: the location of the machines, their size and weight, the way they were decorated, and the appearance of their electronic displays.

People using computer systems often make security-relevant decisions based on contextual cues they see. For example, you might decide to type in your bank account number because you believe you are visiting your bank's Web page. This belief might arise because the page has a familiar look, because the bank's URL appears in the browser's location line, or for some other reason.

To appreciate the range and severity of possible spoofing attacks, we must look more deeply into two parts of the definition of spoofing: security-relevant decisions and context.

Security-relevant Decisions

By "security-relevant decision," we mean any decision a person makes that might lead to undesirable results such as a breach of privacy or unauthorized tampering with data. Deciding to divulge sensitive information, for example by typing in a password or account number, is one example of a security-relevant decision. Choosing to accept a downloaded document is a security-relevant decision, since in many cases a downloaded document is capable of containing malicious elements that harm the person receiving the document [2].

Even the decision to accept the accuracy of information displayed by your computer can be security-relevant. For example, if you decide to buy a stock based on information you get from an online stock ticker, you are trusting that the information provided by the ticker is correct. If somebody could present you with incorrect stock prices, they might cause you to engage in a transaction that you would not have otherwise made, and this could cost you money.

Context

A browser presents many types of context that users might rely on to make decisions. The text and pictures on a Web page might give some impression about where the page came from; for example, the presence of a corporate logo implies that the page originated at a certain corporation.

The appearance of an object might convey a certain impression; for example, neon green text on a purple background probably came from *Wired* magazine. You might think you're dealing with a popup window when what you are seeing is really just a rectangle with a border and a color different from the surrounding parts of the screen. Particular graphical items like file-open dialog boxes are immediately recognized as having a certain purpose. Experienced Web users react to such cues in the same way that experienced drivers react to stop signs without reading them.

The names of objects can convey context. People often deduce what is in a file by its name. Is manual.doc the text of a user manual? (It might be another kind of document,

or it might not be a document at all.) URLs are another example. Is MICR0S0FT.COM the address of a large software company? (For a while that address pointed to someone else entirely. By the way, the round symbols in MICR0S0FT here are the number zero, not the letter O.) Was dole96.org Bob Dole's 1996 presidential campaign? (It was not; it pointed to a parody site.)

People often get context from the timing of events. If two things happen at the same time, you naturally think they are related. If you click over to your bank's page and a username/password dialog box appears, you naturally assume that you should type the name and password that you use for the bank. If you click on a link and a document immediately starts downloading, you assume that the document came from the site whose link you clicked on. Either assumption could be wrong.

If you only see one browser window when an event occurs, you might not realize that the event was caused by another window hiding behind the visible one.

Modern user-interface designers spend their time trying to devise contextual cues that will guide people to behave appropriately, even if they do not explicitly notice the cues. While this is usually beneficial, it can become dangerous when people are accustomed to relying on context that is not always correct.

TCP and DNS Spoofing

Another class of spoofing attack, which we will not discuss here, tricks the user's software into an inappropriate action by presenting misleading information to that software [3]. Examples of such attacks include TCP spoofing [4], in which Internet packets are sent with forged return addresses, and DNS spoofing [5], in which the attacker forges information about which machine names correspond to which network addresses. These other spoofing attacks are well known, so we will not discuss them further.

Web Spoofing

Web spoofing is a kind of electronic con game in which the attacker creates a convincing but false copy of the entire World Wide Web. The false Web looks just like the real one: it has all the same pages and links. However, the attacker controls the false Web, so that all network traffic between the victim's browser and the Web goes through the attacker.

Consequences

Since the attacker can observe or modify any data going from the victim to Web servers, as well as controlling all return traffic from Web servers to the victim, the attacker has many possibilities. These include surveillance and tampering.

Surveillance The attacker can passively watch the traffic, recording which pages the victim visits and the contents of those pages. When the victim fills out a form, the entered data is transmitted to a Web server, so the attacker can record that too, along

with the response sent back by the server. Since most on-line commerce is done via forms, this means the attacker can observe *any account numbers or passwords the victim enters*.

As we will see below, the attacker can carry out surveillance even if the victim has a "secure" connection (usually via Secure Sockets Layer) to the server, that is, even if the victim's browser shows the secure-connection icon (usually an image of a lock or a key).

Tampering The attacker is also free to modify any of the data traveling in either direction between the victim and the Web. The attacker can modify form data submitted by the victim. For example, if the victim is ordering a product on-line, the attacker can change the product number, the quantity, or the ship-to address.

The attacker can also modify the data returned by a Web server, for example by inserting misleading or offensive material in order to trick the victim or to cause antagonism between the victim and the server.

Spoofing the Whole Web

You may think it is difficult for the attacker to spoof the entire World Wide Web, but it is not. The attacker need not store the entire contents of the Web. The whole Web is available on-line; the attacker's server can just fetch a page from the real Web when it needs to provide a copy of the page on the false Web.

How the Attack Works

The key to this attack is for the attacker's Web server to sit between the victim and the rest of the Web. This kind of arrangement is called a "man in the middle attack" in the security literature.

URL Rewriting

The attacker's first trick is to rewrite all of the URLs on some Web page so that they point to the attacker's server rather than to some real server. Assuming the attacker's server is on the machine www.attacker.org, the attacker rewrites a URL by adding http://www.attacker.org to the front of the URL. For example, http://home.netscape.com becomes http://www.attacker.org/http://home.netscape.com. (The URL rewriting technique has been used for other reasons by two other Web sites, the Anonymizer and the Zippy filter. See page 9 for details.)

Once the attacker's server has fetched the real document needed to satisfy the request, the attacker rewrites all of the URLs in the document into the same special form by splicing http://www.attacker.org/ onto the front. Then the attacker's server provides the rewritten page to the victim's browser.

Since all of the URLs in the rewritten page now point to www.attacker.org, if the victim follows a link on the new page, the page will again be fetched through the attacker's

server. The victim remains trapped in the attacker's false Web, and can follow links forever without leaving it.

Forms

If the victim fills out a form on a page in a false Web, the result appears to be handled properly. Spoofing of forms works naturally because forms are integrated closely into the basic Web protocols: form submissions are encoded in URLs and the replies are ordinary HTML Since any URL can be spoofed, forms can also be spoofed.

When the victim submits a form, the submitted data goes to the attacker's server. The attacker's server can observe and even modify the submitted data, doing whatever malicious editing desired, before passing it on to the real server. The attacker's server can also modify the data returned in response to the form submission.

"Secure" connections don't help

One distressing property of this attack is that it works even when the victim requests a page via a "secure" connection. If the victim does a "secure" Web access (a Web access using the Secure Sockets Layer) in a false Web, everything will appear normal: the page will be delivered, and the secure connection indicator (usually an image of a lock or key) will be turned on.

The victim's browser says it has a secure connection because it *does* have one. Unfortunately the secure connection is to www.attacker.org and not to the place the victim thinks it is. The victim's browser thinks everything is fine: it was told to access a URL at www.attacker.org so it made a secure connection to www.attacker.org. The secure-connection indicator only gives the victim a false sense of security.

Starting the Attack

To start an attack, the attacker must somehow lure the victim into the attacker's false Web. There are several ways to do this. An attacker could put a link to a false Web onto a popular Web page. If the victim is using Web-enabled email, the attacker could email the victim a pointer to a false Web, or even the contents of a page in a false Web. Finally, the attacker could trick a Web search engine into indexing part of a false Web.

Completing the Illusion

The attack as described thus far is fairly effective, but it is not perfect. There is still some remaining context that can give the victim clues that the attack is going on. However, it is possible for the attacker to eliminate virtually all of the remaining clues of the attack's existence.

Such evidence is not too hard to eliminate because browsers are very customizable. The ability of a Web page to control browser behavior is often desirable, but when the page is hostile it can be dangerous.

The Status Line

The status line is a single line of text at the bottom of the browser window that displays various messages, typically about the status of pending Web transfers.

The attack as described so far leaves two kinds of evidence on the status line. First, when the mouse is held over a Web link, the status line displays the URL the link points to. Thus, the victim might notice that a URL has been rewritten. Second, when a page is being fetched, the status line briefly displays the name of the server being contacted. Thus, the victim might notice that www.attacker.org is displayed when some other name was expected.

The attacker can cover up both of these cues by adding a JavaScript program to every rewritten page. Since JavaScript programs can write to the status line, and since it is possible to bind JavaScript actions to the relevant events, the attacker can arrange things so that the status line participates in the con game, always showing the victim what would have been on the status line in the real Web. Thus the spoofed context becomes even more convincing.

The Location Line

The browser's location line displays the URL of the page currently being shown. The victim can also type a URL into the location line, sending the browser to that URL. The attack as described so far causes a rewritten URL to appear in the location line, giving the victim a possible indication that an attack is in progress.

This clue can be hidden using JavaScript. A JavaScript program can hide the real location line and replace it by a fake location line which looks right and is in the expected place. The fake location line can show the URL the victim expects to see. The fake location line can also accept keyboard input, allowing the victim to type in URLs normally. Typed-in URLs can be rewritten by the JavaScript program before being accessed.

Viewing the Document Source

There is one clue that the attacker cannot eliminate, but it is very unlikely to be noticed.

By using the browser's "view source" feature, the victim can look at the HTML source for the currently displayed page. By looking for rewritten URLs in the HTML source, the victim can spot the attack. Unfortunately, HTML source is hard for novice users to read, and very few Web surfers bother to look at the HTML source for documents they are visiting, so this provides very little protection.

A related clue is available if the victim chooses the browser's "view document information" menu item. This will display information including the document's real URL, possibly allowing the victim to notice the attack. As above, this option is almost never used so it is very unlikely that it will provide much protection.

Bookmarks

There are several ways the victim might accidentally leave the attacker's false Web during the attack. Accessing a bookmark or jumping to a URL by using the browser's "Open location" menu item might lead the victim back into the real Web. The victim might then reenter the false Web by clicking the "Back" button. We can imagine that the victim might wander in and out of one or more false Webs. Of course, bookmarks can also work against the victim, since it is possible to bookmark a page in a false Web. Jumping to such a bookmark would lead the victim into a false Web again.

Tracing the Attacker

Some people have suggested that this attack can be deterred by finding and punishing the attacker. It is true that the attacker's server must reveal its location in order to carry out the attack, and that evidence of that location will almost certainly be available after an attack is detected.

Unfortunately, this will not help much in practice because attackers will break into the machine of some innocent person and launch the attack there. Stolen machines will be used in these attacks for the same reason most bank robbers make their getaways in stolen cars.

Remedies

Web spoofing is a dangerous and nearly undetectable security attack that can be carried out on today's Internet. Fortunately there are some protective measures you can take.

Short-term Solution

In the short run, the best defense is to follow a three-part strategy:

1. disable JavaScript in your browser so the attacker will be unable to hide the evidence of the attack;
2. make sure your browser's location line is always visible;
3. pay attention to the URLs displayed on your browser's location line, making sure they always point to the server you think you're connected to.

This strategy will significantly lower the risk of attack, though you could still be victimized if you are not conscientious about watching the location line.

At present, JavaScript, ActiveX, and Java all tend to facilitate spoofing and other security attacks, so we recommend that you disable them. Doing so will cause you to lose some useful functionality, but you can recoup much of this loss by selectively turning on these features when you visit a trusted site that requires them.

Long-term Solution

We do not know of a fully satisfactory long-term solution to this problem.

Changing browsers so they always display the location line would help, although users would still have to be vigilant and know how to recognize rewritten URLs.

For pages that are not fetched via a secure connection, there is not much more that can be done.

For pages fetched via a secure connection, an improved secure-connection indicator could help. Rather than simply indicating a secure connection, browsers should clearly say who is at the other end of the connection. This information should be displayed in plain language, in a manner intelligible to novice users; it should say something like "Microsoft Inc." rather than "www.microsoft.com."

Every approach to this problem seems to rely on the vigilance of Web users. Whether we can realistically expect everyone to be vigilant all of the time is debatable.

Related Work

We did not invent the URL rewriting technique. Previously, URL rewriting has been used as a technique for providing useful services to people who have asked for them.

We know of two existing services that use URL rewriting. The Anonymizer, written by Justin Boyan at Carnegie Mellon University, is a service that allows users to surf the Web without revealing their identities to the sites they visit. The Zippy filter, written by Henry Minsky, presents an amusing vision of the Web with Zippy-the-Pinhead sayings inserted at random.

Though we did not invent URL rewriting, we believe we are the first to realize its full potential as one component of a security attack.

Acknowledgments

The URL-rewriting part of our demonstration program is based on Henry Minsky's code for the Zippy filter. We are grateful to David Hopwood for useful discussions about spoofing attacks, and to Gary McGraw and Laura Felten for comments on drafts of this paper. The figure was designed by Gary McGraw.

For More Information

More information is available from our Web page at http://www.cs.princeton.edu/sip, or from Prof. Edward Felten at felten@cs.princeton.edu or (609) 258-5906.

References

[1] Peter G. Neumann. Computer-Related Risks. ACM Press, New York, 1995.

[2] Gary McGraw and Edward W. Felten. Java Security: Hostile Applets, Holes and Antidotes. John Wiley and Sons, New York, 1996.

[3] Robert T. Morris. A Weakness in the 4.2BSD UNIX TCP/IP Software. Computing Science Technical Report 117, AT&T Bell Laboratories, February 1985.

[4] Steven M. Bellovin. Security Problems in the TCP/IP Protocol Suite. Computer Communications Review 19(2):32-48, April 1989.

[5] Steven M. Bellovin. Using the Domain Name System for System Break-ins. Proceedings of Fifth Usenix UNIX Security Symposium, June 1995.

[6] Web site at http://www.anonymizer.com

[7] Web site at http://www.metahtml.com/apps/zippy/welcome.html

---

E-Mail Impersonators

How to identify "spoofed" e-mail.

By Bill Barnes

Posted Tuesday, March 12, 2002, at 4:46 PM PT

Editor's note: *To read the complete explanation of how* ***Slate*** *was duped by an e-mail spoofer, see this "Press Box" column.*

After my wife cast her ballot on the morning of Election Day 1996, she arrived at work to find an e-mail from none other than the president of the United States (president@whitehouse.gov). He thanked her for her vote and promised to address her hot-button issues of education and women's rights. She was a little disturbed, but as it turned out the sanctity of her secret ballot hadn't been compromised. Someone (her husband) had merely sent her a spoofed e-mail.

E-mail is considered "spoofed" when the e-mail address in the "**From**" field is not that of the sender. As *Slate* learned so publicly last week, believing what you read in spoofed e-mail can cause huge embarrassment, so if you receive an e-mail from George W. Bush or a man purporting to be an executive

of a European carmaker, trust us: It might not be on the level. The bad news is that it's not very hard to spoof e-mail, but the good news is that it can usually be detected. To detect spoofed e-mail (and boy, do ***Slate***'s editors wish I'd written this piece last month!) you need to understand how e-mail is sent on the Internet.

1. First, your e-mail program (e.g., Outlook, Eudora, Hotmail) sends mail to an SMTP (Simple Mail Transport Protocol) server, a computer that understands how to relay your e-mail
2. from SMTP server to SMTP server across the Internet, until
3. it arrives at its penultimate destination, the recipient's mailbox. The mailbox stores this e-mail until
4. finally it's fetched by an e-mail program, so its recipient can read it.

Like a well-paid courier, SMTP just passes along what it was given. I tell Outlook my e-mail address, but neither it nor the SMTP server provided by my Internet service provider has any way to verify that it's true. Just this minute, I changed my Outlook settings to say that my name is Mork, e-mail address mork@ork.planet, and Outlook happily sent more mail to my wife, who is tiring of my little shenanigans. ISPs smarter than mine configure their mail servers to be more restrictive about the e-mail they'll accept, attempting to verify the veracity of the sender's address, but a determined spoofer usually knows how insert e-mail further along the transmission chain.

E-Mail Spoofing and How to Fight It

January, 1997
Chad Carson
CCARSON@LACA.ORG

Email spoofing may occur in different forms, but all have a similar result: a user receives email that appears to have originated from one source when it actually was sent from another source. Email spoofing is often an attempt to trick the user into making a damaging statement or releasing sensitive information (such as passwords).

- This description quoted from http://mwir.lanl.gov:8080/E-Mail_Spoofing.html

E-Mail spoofing, as the above description states, is nothing more than forging E-Mail. This can easily be done through Netscape Mail, by sending a message through LACA.ORG's SMTP port, or any other SMTP ready system, such as another company or university's site.

Spoofing could be used in many creative ways by students with access to Netscape, such as:

- It can be used to send rude comments, making them appear to come from a teacher to a principal or a student to a teacher. (I have not had a report of this happening yet, but it probably will).
- It can be (and has been) used to send off-color or inappropriate messages to users not on LACA's system, and make them appear to come from teachers.

Using spoofing to get sensitive information from someone (such as their password) is a little harder on LACA's system, because the "Reply-To" address is shown when viewing the message in Pine. If you get a message that looks like this....

```
Date: Wed, 22 Jan 1997 11:02:17  -0500
From: Chad Carson
Reply to: NE123456@LACA.ORG
To: JDOE@LACA.ORG
Subject: System Maintenance

John Doe,

        I need your password to be able to test a problem
        with your account.  Please respond to me through E -Mail.

Thanks,
Chad
```

...it would be very unwise to respond to it. The "From" and "Reply To" lines don't even match. It says it is from Chad, but the reply is going to go to a Newark HS student account.

Checking for Spoofed E-Mail

Here is how a message typically looks from within Pine mail (or just plain PMDF Mail). There is NO indication that this message did not come from Mickey Mouse (even though we're pretty sure it didn't).

```
Date: Wed, 22 Jan 1997 10:41:33  -0500
From: Mickey Mouse
To: CCARSON@LACA.ORG
Subject: Spoofed Message

This message is spoofed.
```

If you receive a questionable message such as this, type an "H" while reading the message in Pine, full header mode is turned on, and more information about the true origin of the message will be revealed. (In PMDF Mail, type READ /NOTRIM to read messages with full headers). Here is the same message with full headers:

```
Return-path:
Received: from 156.63.145.9 by LACA.ORG (PMDF V5.0 -6 #6770)
 id <01ieiebka6a8000cxu@laca.org> for CCARSON@LACA.ORG; Wed,
 22 Jan 1997 10:41:35 -0400 (EDT)
Date: Wed, 22 Jan 1997 10:41:33 -0500
From: Mickey Mouse
Subject: Spoofed Message
To: CCARSON@LACA.ORG
Reply-to: MMOUSE@DISNEY.COM
Message-id: <32e6352d.535c@disney.com>
Organization: Walt Disney World
MIME-version: 1.0
X-Mailer: Mozilla 3.01Gold (Win95; I)
Content-transfer-encoding: 7BIT

This message is spoofed.
```

It looks like a bunch of gibberish, but the line that reads:

```
Received: from 156.63.145.9 by LACA.ORG (PMDF V5.0 -6 #6770)
```

is very important. This line shows that the message originated from the IP address 156.63.145.9 (which happens to be the IP of my computer), then it was sent through LACA.ORG. This tells you that the message did not originate at DISNEY.COM, as the From: line incorrectly informs you. The IP address is the key that tells where the message originated from initially. You may need to contact LACA to determine which particular PC that IP address is assigned to.

The line that reads:

```
X-Mailer: Mozilla 3.01Gold (Win95; I)
```

tells you that the message was sent with Netscape.

Mozilla is Netscape's mascot, so any mention of Mozilla tells you it was sent with a Netscape product. This particular message was sent with version 3.01 Gold for Windows 95. Interpreting this information may also prove helpful in determining which PC actually sent the message.

Spoofing - arts of attack and defence

Introduction

Spoofing means pretending to be something you are not. In Internet terms it means pretending to be a different Internet address from the one you really have in order to gain something. That might be information like credit card numbers, passwords, personal information or the ability to carry out actions using someone else's identity.

Some potential spoof attacks are technically difficult. Others are commonplace. There are different ways in which you can defend against actual or attempted spoofing. Some attacks rely upon user error or misunderstanding. These are more difficult to prevent because users do not necessarily have the technical subtlety needed to understand what is happening, whilst the technology they use does not explain itself very clearly in non-technical terms.

In this white paper, several spoofing attacks, and possible defenses against them are considered.

DNS server spoofing attack

The most complex attack is to alter the address the master DNS servers will resolve for a given URL. The URL that an Internet user types in is not the numeric address of the site required, but an alphanumeric address structure. The DNS servers convert, say, www.articsoft.com, into a real Internet address, say 195.217.192.145 (not the correct address, but the point is made). This has to be done because people don't generally remember and associate 12 digit numbers with anything except telephone numbers, and then they generally file them on the telephone with a 'friendly name' that they have some relationship with.

An attack of this type has been successfully mounted that altered the server list, so that, for a period of time, users requesting some sites were directed to the wrong addresses.

This type of attack is a major threat and the Internet naming and addressing authorities have taken it very seriously indeed. DNS servers have incorporated numerous security measures to prevent repetitions of this attack from being successful. These include having the servers mirror and monitor each other as well as controlling very carefully how updates are introduced into the servers.

This kind of problem can be resolved by positive site identification, where the end user is able to automatically check the claimed web site URL against the content provided, as provided by the ArticSoft approach.

Web site names and addresses

There are many ways in which a web site may be spoofed. This section covers popular methods.

Content theft

A copy of a site can be created from the original by copying all the publicly accessible pages from a site to another server. Copying a publicly accessible site is automated through the use of programs called 'spiders'. You will find many programs freely available that are designed to copy whole websites so that the program user can read a web site offline rather than have to stay connected to the Internet.

Some spider activities are legitimate - maintaining mirror copies of the site to improve accessibility, or search engines looking for text and keywords to add to their catalogues. Search engines also maintain caches of pages for their users so that load times can be reduced. Other spider activities may not be. A web site owner may be aware that a 'spider' is reading his site, but he cannot know its real intent any more than you can know why a person reads a book.

The technical defenses against this attack are few. Sites have to make a large amount of their content publicly available - so naturally they want people to obtain that content. Software is badly placed to be able to decide what name a user really intended (what is a typing error?) and there would be an adverse reaction if systems were written to decide such things on behalf of the user.

Sometimes the copying may be to present your information as that of another site. This is sometimes referred to as deep linking, not spoofing. A number of schemes for 'watermarking' images have been invented in recent years to help detect this kind of attack, and an excellent reference is http://www.cl.cam.ac.uk/~fapp2/steganography/ where the background, history and actual effectiveness of the technique(s) that have been developed is discussed.

Name similarity

The simplest spoof is to catch the people who mistype the web URL they are looking for, or put the wrong locator at the end. For examples of similar names try www.whitehouse.com (it should have been .org) or www.nasa.com (similar problem). Other famous examples include www.nescape.com and

www.mcrosoft.com. Sometimes the content makes it obvious to the user that the site is not the one they were expecting, but it doesn't have to. That is down to the creator of the similar site. To give some examples where sites are being directed to what might be considered an unexpected destination, most likely because of a spelling mistake: (please note that there is no suggestion that these are hacker sites, but the redirection stated here was verified on 2 April 2002):

www.mcrosoft.com = 195.184.248.163/ML_HomePage.aspx
www.whithouse.org = www.amatuervideos.nl
www.harods.com = www.shopndrop.com
www.walmaart.com = www.search.linksponsor.com

On 24 April 2002 the author was offered the ability to register the web site WWW.VERISIGN.COM. (Shown in a different font this is www.verisIgn.com but how would you have known?) So registering web sites with similar names to real businesses is not so difficult, and a fraudster will have used a stolen credit card so they won't be found!

For an excellent tutorial on how to carry out this kind of attack and the uses it can be put to see the web site www.reamweaver.com which claims to provide downloadable software for this purpose.

The most effective defense against this kind of confusion is probably procedural. Users still have to take care that they do not put in the wrong name. But you cannot expect users to get their typing to be perfect on every occasion.

Altering the registration rules for Internet names to prevent registering names that are very close to those of registered companies or organizations could well help prevent this problem.

Usually national laws prevent people from registering company names that are similar to existing ones for exactly this reason (there is, in some jurisdictions, the offence of passing off), so the justification for allowing it on the Internet is muddy at best. It has taken us many hundreds of years to establish the rules for trade in the terrestrial world and it seems unreasonable to take a stance that says somehow the Internet can ignore the experience that made those laws necessary, any more than Newton's apple could have tried to ignore his law of gravity. (Selling domain names is a business whilst registering a company is something controlled by law. As a result, there may be differences in approach. Law evolved to make company registration a formal process for good reasons that the Internet does not appear to have fully recognized.)

Link alteration

Another attack, that offers far more gain to the hacker for rather less actual work, is to alter the return address in a web page sent to a user to make it go to the hacker's site rather than the legitimate site. This is done by adding the hacker's address before the actual address in any page that has a request going back to the original site. Literally, where they see a reference to <http://www.mysitemname.com> they add their own address to it to make <http://hackersite/http://mysitename.com>. You will notice that the fake site is recognized as a valid URL address.

The hacker only need to do this once to get a link into the communication between browser and server and they can reprocess all the communication from then on, including SSL connections. Since the user is familiar with seeing the site connections and names and even server certificate details constantly changing without any explanation or obvious reason, they are not likely to notice this change at all.

The commonest form of defense used by web sites at the moment is to apply 'digital signatures' to their web pages, which are checked as they are leaving the server to ensure that nothing has been changed. The idea is to prevent altered pages from being able to enter the Internet. The big drawback with the approach is that pages can easily be altered if cached on web servers. The end user cannot see either the original web site checks (and caches have no checks) and therefore has no idea if the pages are valid when they arrive. They may have been cached at one or more locations where they could have been attacked, which is rather easier than trying to alter them as they go by.

Recent developments by ArticSoft have produced a system that provides end users with continual verification of pages back to site URLs. This is a much more powerful technique for several reasons:

- Most important is that it gives the end user actual information to act on rather than leaving them guessing;
- Secondly, because, as well as detecting page alteration, their method prevents pages from being routed through a site that is not the originating site, and it therefore prevents this type of spoofing;

- Thirdly, it also resolves the problem in SSL that a hacker can, by altering addresses, get into the middle of the connection between the user and the real site and read all the supposedly protected information.

This latter feature is possible because most sessions do not, or cannot verify the user identity to the server, and the user does not know what identity the SSL connection should have.

SSL is a technology that has succeeded largely because few users understand it (or the padlock on the browser) at all. An SSL link for commercial sites is started by the browser, without validating where it is linking to. Nothing happens in the browser to confirm what the link is with, or if it is valid and there is no checking that the source of the information passing across the link has come immediately from the expected site. A great deal of faith is put in the user checking all the details for themselves, which stands in great contrast to the idea that systems are intuitive and easy to use.

A variety of academic papers have been published detailing attacks that defeat SSL and demolish many of the claims made for its capabilities have been published. These include:

www.cs.princeton.edu/sip/pub/spoofing.html
www.bau2.uibk.ac.at/matic/spoofing.htm
www.cs.dartmouth.edu/~pkilab/demos/spoofing/

User lack of understanding is further undermined by techniques that are common industry practices which confuse security. 'Wildcard' server certificates are used by many sites rather than proper individual names. ISP certificates are often used as the common certificate for all their hosted web sites. The use of third party secure services for payments systems with completely different site names also confuses the situation. Users can hardly be expected to understand what to them are arcane practices that have no apparent explanation.

On balance, it is just as well that the end user remains blissfully ignorant. However, this is the very ignorance that fosters hacking and spoofing. Changing this situation is not a matter of expecting all users to become expert technologists. Education is required, but so are appropriate methods of development behaviour that increase understanding and security as principles and best practice.

IP addresses changing attacks

Hackers are able to configure themselves (their messages over the Internet) to have any IP address that they want, so they can appear to be part of an internal network when in fact they are external, or appear to be the address that you want to connect to. This is a subtle attack because it may be used on the Internet, intranet or extranet equally well. Many networks are set up to dynamically allocate addresses, and software monitoring techniques to reveal information flowing around networks allow hackers to select valid addresses so that they can impersonate valid sessions. Alternatively, the hacker may try to capture a valid available address.

Defenses against this kind of attack are often firewall based. Firewalls can be set to perform network address translation so that internal addresses are not disclosed to the outside world. Also, firewalls can be set to discriminate between connections that are internal from those that are external but appear to be using internal addresses. However, if an attacker can gain access to an internal network they can bypass external firewall checks. To guard against this situation some organizations, particularly financial ones, use internal firewalls to control and limit the potential for this kind of attack.

E-mail address changing

In particular services, such as e-mail, the potential to spoof the apparent source address continues to be a problem. Most users are unaware that the apparent address is unreliable, and that replying to the apparent address may actually send a message to an unintended destination. (An example is replying to a message from an individual that was forwarded by a distribution group. The reply goes to the group, not the individual who appeared to be the source. This is not regarded as an error, although it is actually a security failure because the user is not aware of what is happening and in theory they are in charge.)

E-mail with secure attachments may be prone to spoofing as well. Users may assume that the header of the e-mail and the secure body are related to each other when that is not the case. User education is much more difficult in this case because it is not in the least clear to the user why secure messages

aren't. The only way to protect e-mail effectively is to ensure that the whole object is protected, not just parts of it. Users will also have to accept that the addressing information in e-mail cannot be private.

Review of the current situation

Spoofing attacks are based upon the ability to make a user believe that they are securely connected to a network address, or receiving e-mail from a specific source, when that is not the case. The problem stems from the fact that the whole of the addressing system on the Internet is not secure. This creates problems of spoofing in many areas outside web addresses, including e-mail.

Since there are currently no effective means of securing the addressing (unless everyone 'knows' everyone else, the attempts to secure links between address points are flawed, and unless there is a move to mandate absolute identification of all Internet users (politically unlikely given requirements for anonymity that exist in US law for certain types of transactions) they will remain so.

The best way forwards

A change is needed to move from relying on networking systems that don't solve the problem to content management - signing and protecting the actual information itself and not just the unproven link(s) it is traveling over. That prevents all the typical network IP attacks from having any effect, and provides genuine control over the information itself.

A change to securing content, rather than links, offers the e-business community significant benefits. For e-business, there is an imperative for the honest trader to identify themselves by clearly identifying their content. (How you link to them is then, actually irrelevant.) That way all their users can verify any content reaching them, and rely upon what that content is, regardless of how it got to them. The same would go for instructions to computer systems, services and networks.

By switching to that approach, the business community can achieve major trading benefits: certainty that the quality of their information can be proven; certainty of secure trade for them and their customers; certainty of privacy for them and their customers; certainty that payment details cannot be misused.

Conversely, traders not following such an approach identify themselves as leaving their customers open to fraud, misrepresentation, uncertainty and lack of confidence. Right now schemes to separate the good from the bad have little effect.

ArticSoft have provided some novel steps in the direction of proof by content rather than proof by network connection. For Internet technologies this is a more pragmatic way to proceed because content may reside anywhere on the Internet. It also allows for protecting information that is confidential by much simpler methods than are offered by network based solutions.

Such a change faces significant opposition, not least from the network providers, network analysts and managers, who risk being relegated to a lower status (and relative income) as a result. In practice, with the tools available, they have done the best job that could be done. Unfortunately, scripting attacks and cook book hacking methods are making those methods more vulnerable, and a change in approach is needed moving forwards.