

Web 2.0 Creates Security Challenges

George Lawton

By its very nature, Web 2.0 is interactive, allowing users to add input to Web sites such as participatory book reviews, blogs, wikis, social networks, and video- and photo-sharing services. Web 2.0 has consequently become very popular with users and Web site operators. However, the approach's interactivity has also made it popular with hackers.

Web 2.0 is basically a new way to use existing Internet technologies—such as XML and JavaScript—to enable participation, interaction, and collaboration among users, content providers, and businesses, rather than just the traditional viewing of static Web pages, said Hewlett-Packard security evangelist Michael Sutton.

Today's Web 2.0 applications are openly accessible and dynamically generated, Sutton said. This makes them more interesting but also bigger security risks, he noted.

For example, hackers can add malicious content to or exploit vulnerabilities in Web 2.0 sites. Moreover, Sutton said, many site owners are asking developers to focus on functionality rather than security. Thus, developers might not always take steps such as ensuring that pages validate user input.

Even properly developed Web 2.0 pages can cause security problems for



both clients and Web servers, noted Jeremiah Grossman, chief technology officer of WhiteHat Security.

These pages are attractive to hackers because they are so popular and because security vendors have reduced the effectiveness of other, more traditional attack vectors such as e-mail attachments.

Meanwhile, vendors and Web site operators are concerned because security measures frequently violate the openness and interactivity that Web 2.0 was designed to enable.

WEB 2.0 AND SECURITY THREATS

Web 2.0 sites inherently carry more risk than traditional Web sites because they let users upload content and require scripting capabilities—which can run code or carry malware—to function properly, said Will Dormann, a vulnerability analyst at the CERT Coordination Center, part of the Carnegie Mellon University-based Software Engineering Institute.

As a result, hackers have exploited Web 2.0 to launch worms that execute harmful operations outside the browser, leaving users unaware of their activities.

They also upload legitimate-looking malicious content to social networks. This could occur, for example, if someone visited a wiki linking to code that is supposed to be virus-removal software but that instead loads a Trojan horse.

The harmful code could include keyloggers that capture victims' keystrokes—including those used for bank and credit-card numbers and passwords—and send them to the hacker. The code could also turn victims' machines into remote-controlled zombies that hackers could use to launch spam, denial-of-service, or other attacks.

Hackers take advantage of several Web 2.0 elements and applications.

Ajax

Ajax (asynchronous JavaScript and XML) is a programming technique for creating rich, interactive Internet applications that behave like desktop programs.

It uses asynchronous JavaScript, a cross-platform technology an HTML page can use to fetch XML documents by making calls asynchronously to the server from which it was loaded.

This capability lets an application make a server call, retrieve new data, and simultaneously update the Web page without having to reload the contents, all while the user continues interacting with the program. For example, Google Maps, written with Ajax, lets users move a map around a screen with the cursor without having to wait for the page to reload each time.

Using Ajax yields highly interactive, fast programs with responsive interfaces, overcoming Web applications' typical slow performance and limited interactivity.

Because much of the processing and most of the data requests occur outside the browser window in the

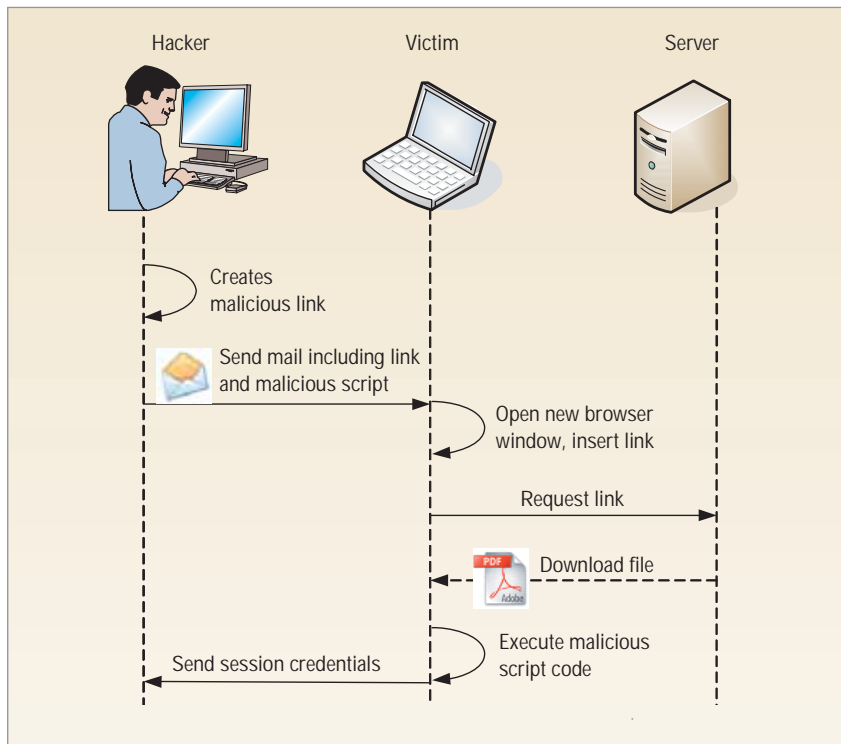


Figure 1. In a cross-site-scripting attack, hackers inject their own executable code into a legitimate Web page. In this case, hackers launch an XSS attack to use a victim's computer to get a Web page from a server that the victim is authorized to access. The hackers include a malicious script that, when the requested page downloads, executes and then sends the attackers the session credential for the victim's Web site visit. The hackers can use the credential to access and take actions on the site.

background rendering engine, users may not detect security problems with Ajax code. For example, they might not see the hacker's software either downloading code through a security hole in the browser or performing harmful operations.

Meanwhile, JavaScript—which lets developers embed behavior such as opening windows and changing images—into Web pages can run arbitrary server-provided code on a client with full privileges

Today, about 70 percent of malicious code in the wild is downloaded via Ajax, according to Yuval Ben-Itzhak, chief technology officer of security vendor Finjan.

Complicating the problem is many developers' unfamiliarity with Ajax, leading them to design programs with inadequate security.

XML syndication

Blogs, wikis, and other Web 2.0

applications support content that real-simple-syndication (RSS) or Atom newsfeed-distribution systems can send to subscribers.

Syndication systems distribute this code directly into browsers or encapsulate it into browser-based or freestanding online newsreader applications. Newsfeed authors can create content themselves or let third parties upload material into the feed.

Hackers could insert code either directly into a feed or via a compromised newsfeed server. The client then automatically accepts the malicious content.

Mashups

Mashups combine data or services from multiple Web sites into one user experience. For example, a mashup might combine Google Maps with data from an online, real-time, highway-traffic site.

Mashups typically work via a set of APIs published by the Web site provider. The APIs let the mashup take information from the various sites and mix features from multiple sources.

Security thus depends to some extent on the reliability of the provider's content. However, mashups can also connect dynamically to Web sites not necessarily under the provider's control, which presents further security challenges.

Therefore, the content providers should secure their servers and validate content, which they don't always do, said HP's Sutton.

Social networking

Web 2.0 lets users create and distribute content on social-networking sites such as Flickr, MySpace, Wikipedia, and YouTube. Users can upload video, audio, photo, text, and other files for subsequent downloading by others.

However, hackers could include malicious code in the uploaded files.

TYPICAL ATTACK TECHNIQUES AND TARGETS

There are multiple attack vectors for targeting a browser's or computer's Web 2.0 vulnerabilities. The majority of attacks use some form of cross-site scripting (XSS) or cross-site request forgery (CSRF), noted WhiteHat's Grossman.

XSS

In XSS attacks, hackers inject their own executable code into existing, legitimate, dynamically generated Web pages, as Figure 1 shows. When someone downloads the page, the embedded programming accompanies the requested page and can execute on the user's computer.

The code generally lets attackers gain elevated access privileges to the victim's system and, for example, steal data or change user settings.

XSS can thus have disastrous consequences, noted Ed Skoudis, senior security consultant at Intel-

guardians, an information-security consultancy.

One way that hackers sometimes inject the harmful code is via the many popular online guestbook and forum programs that let users submit posts that include embedded HTML and JavaScript.

In particular, XSS takes advantage of Ajax-style applications, which let code execute outside a user's browser.

CSRF

XSS exploits the trust a user has in a Web site. CSRF, on the other hand, exploits the trust a Web site has in a user.

In a CSRF attack, a hacker gains access to an unsuspecting user's computer and sends unauthorized requests to an e-commerce or other Web site to which the victim has been authenticated. The hacker can also send requests via the user's computer to a company intranet to which the victim has access, thereby bypassing firewall protection.

To authenticate and thus gain access to a Web site or corporate intranet, a hacker uses either the compromised computer's IP address or cookies that the site placed on the machine.

This enables the hacker to act as the computer's owner and initiate harmful actions such as taking money from the person's bank account, ordering products from an e-commerce site, stealing data from a company intranet, or changing settings on a local firewall or router.

Dynamic code obfuscation

Most antivirus programs and many other security applications, such as antispyware tools, use scanners and pattern-matching software to look for known malware's code signatures.

Hackers sometimes conceal the signatures via dynamic code obfuscation. DCO uses algorithms to add randomly generated code to a JavaScript-based Web page that includes malware. The code doesn't affect the way browsers render the

page and doesn't make the malicious code the page contains less harmful. But it keeps pattern-matching software from recognizing the malware.

In addition, DCO keeps mutating the code it adds to malicious files. With no single set of added code to key on, antivirus products can't develop subsequent pattern-matching software to recognize DCO-altered malware.

Most antivirus programs and security applications use scanners and pattern-matching software to look for known malware.

To lure victims to a compromised Web page, hackers use techniques such as spam or phishing, which can include e-mail messages with links to malicious sites. The links have URLs that are similar to those of popular legitimate sites or that appear to advertise sexual or other content that appeals to some people. They can also work via hidden redirect code that hijacks visitors from their apparent destination to the attacker's Web site.

The attack exploits browser vulnerabilities, such as buffer overflows, to place the payload—which can be a virus, Trojan horse, worm, or other type of malware—on a victim's computer.

Because DCO affects visitors to sites written in JavaScript, it is a particular threat to Web 2.0 sites.

Web 2.0 worms

Web 2.0 worms can propagate in the background of a user's browser without being displayed in an open window if someone visits an infected site. Such worms have afflicted major Web 2.0 sites such as MySpace and Yahoo! Mail.

The Samy worm hit MySpace late last year. The author created a piece of JavaScript code that loaded into a browser whenever someone visited an infected MySpace page. Within a

day, Samy spread to over 1 million pages, with the resulting traffic volume forcing MySpace to shut down temporarily.

The Yamanner worm was spammed to Yahoo! Mail users. When they opened the attachment, the worm sent a copy—outside the browser window—to everyone in their contact lists.

According to HP's Sutton, we have only seen the tip of the iceberg with Web 2.0 worms.

Exploiting Web 2.0 attacks

Once hackers compromise a client via a Web 2.0 attack, they can use the malicious code running within the browser to fool the victim's local system into believing that requests are coming from a local user. This could let a hacker connect to local resources. A hacker could thus reprogram a router or firewall to permit outside access to local services.

This gives hackers an opportunity to launch egregious attacks, said Pete Lindstrom, senior analyst with the Burton Group, a market-research firm.

For example, hackers could use victims' browsers to initiate requests to a company's internal servers. This would let hackers access sensitive company data, even if protected by a firewall.

Not just browsers

Many Web 2.0 sites use applications such as Flash, QuickTime, and WinZip to play video clips, view documents, or otherwise handle files.

These applications have become popular targets for hacker intrusions into systems because vendors have improved browser security, said Fred Cohen, research professor at the University of New Haven and founder of the Fred Cohen & Associates information-security consultancy.

FIGHTING BACK

With Web 2.0 attacks, the Burton Group's Lindstrom said, malicious

code does the same things that legitimate users do, such as logging into a banking Web site and withdrawing money. Security systems thus have trouble telling the difference, he noted.

Therefore, users should keep their browsers up-to-date with security patches, said Allen Paller, research director at the SANS Institute, an information-security training and research organization.

Operators can help protect their sites from Web 2.0 attacks via Web-application firewalls, as well as application and source-code vulnerability scanners, according to Paller.

Other helpful measures, he said, include better programmer training and education, improved employee training in safe computing practices, and basic computer hygiene.

Intelguardians' Skoudis said he expects to see more major Web 2.0 attacks, particularly those involving XSS, during the next couple of years. He also predicted that hackers will soon begin using automated tools to discover system flaws that their attacks can exploit.

Finjan's Ben-Itzhak said the potential profits from exploiting Web 2.0 problems have inspired an online industry that sells toolkits for finding and exploiting Web 2.0 vulnerabilities. The kits cost from \$100 to \$1,000, depending on their capabilities.

Web 2.0 is extending the scope of the cat-and-mouse game that security researchers have been playing with hackers since the dawn of the computer industry. Paller explained, "As you build up more defenses, people find more creative ways to get around them." ■

George Lawton is a freelance technology writer based in San Francisco, California. Contact him at glawton@glawton.com.

Editor: Lee Garber, *Computer*,
l.garber@computer.org