New Attack Tricks Antivirus Software

Karen Heyman

Traditionally, antivirus products stop malicious software by recognizing code signatures unique to different types of malware. When the applications encounter a file with a code string that matches one in their database for a known virus, they block its access to the intended victim's computer.

However, with the advent of Web 2.0 (see the sidebar "DCO and Web 2.0"), in which Web sites make it easy for users to add content, hackers have found a new way to spread malicious code and short-circuit the pattern-matching antivirus approach.

Dubbed *dynamic code obfuscation* by security vendor Finjan, the new approach is a way to exploit browser vulnerabilities by hiding JavaScript exploits.

DCO employs algorithms to change and disguise JavaScript-based malware code without making it less harmful, thereby keeping pattern-matching antivirus software from recognizing exploits. DCO keeps mutating malicious files, which makes a simple string match impossible, explained Finjan chief technology officer Yuval Ben-Itzhak.

"DCO makes it more difficult to detect the actual [malicious] JavaScript code that comes embedded in a Web page," noted senior analyst Pete Lindstrom with the Burton Group, a market research firm.

To detect DCO threats, said Carey Nachenberg, chief researcher at security vendor Symantec "You'd need a thousand signatures for thousands of different [malicious] executables."

"This makes it much more difficult for anyone to analyze malicious code," said Rob Murawski, vulnerability analyst with the CERT Coordination Center, an Internet security research and training organization.

DCO IN DETAIL

DCO works with JavaScript. Shortly after Netscape Communications released the language in 1995, developers began obfuscating, and thereby hiding, their JavaScript code for self-protection, said technology author and JavaScript expert Dori Smith. They spent hours writing JavaScript code, she explained, but people using a browser's "View Source" capabilities could easily see what was written and then copy and paste it into their own files.

DCO also uses polymorphism, which has been used previously with viruses. The first known polymorphic virus—which could change its code to fool security software—was written in 1990.

Today, malicious hackers—many of whom are sophisticated financial criminals—are cloaking their JavaScript exploits with obfuscation and polymorphism to create DCO attacks in an effort to generate effective malware that's difficult to detect and stop.

DCO has become particularly popular in part because writing obfuscating code is relatively easy. In fact, commercial automated JavaScript obfuscators are available from companies such as Stunnix.

Ensnaring victims

DCO attacks affect visitors to Web sites that hackers set up to lure victims. To attract people, hackers use techniques such as spam or phishing, which works via e-mail messages that have links to malicious sites, said Shimon Gruper, vice president of technologies for the eSafe Business Unit of software vendor Aladdin Knowledge Systems.

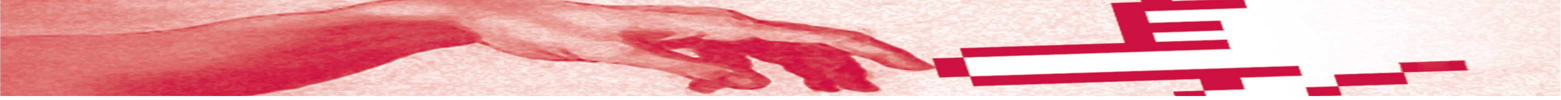
These links lure the unwary by having URLs that are similar to those of popular legitimate sites or that appear to advertise sexual or other content that appeals to some people. The links can also work via hidden redirect code that hijacks visitors from their apparent destination to the attacker's Web site.

The malware on the site takes advantage of browser vulnerabilities such as buffer overflows to place the malware, which is wrapped in DCO code, on a victim's computer.

Hackers use a simple trick to catch even those suspicious of clicking on links, warned Finjan's Ben-Itzhak. They include a hidden iFrame—an HTML element that lets developers embed an HTML document inside another document, viewable only with the source code—along with the text in an e-mail message.

The iFrame can issue a call to a server that then uploads malicious code from the hacker's Web site the moment the visitor opens the e-mail, explained Smith.

Published by the IEEE Computer Society



Inside a DCO attack

DCO combines obfuscation to disguise a payload—which can be a virus, Trojan horse, worm, or other type of malware—and polymorphism to mutate the obfuscated code so that antivirus products can't detect it reliably.

Code obfuscation takes several forms, including interspersing random letters, numbers, symbols, spaces, or even blocks of material; removing or renaming variables; reassigning function pointers; or breaking up code.

The DCO polymorphic engine, located on the hacker's Web site, generates unique obfuscations every time victims download the JavaScript code, noted CERT's Murawski. Thus, each time a hacker sends the same virus to multiple victims, the code looks different.

A subroutine within the obfuscation code returns the material to its original form so that the malicious program can execute as intended once on the victim's computer.

DCO attacks quickly load malicious code in the background, so they are difficult to notice or stop.

A nasty twist

A traditional polymorphic virus uses its own engine to make a functionally equivalent version of the payload with altered code. Because the polymorphic engine is within the malware, antivirus vendors can analyze it and make sure their products can recognize it, no matter how the malicious code itself changes, noted Symantec's Nachenberg.

With DCO attacks, on the other hand, the hacker's server that handles the malicious code also hosts the polymorphic engine.

"We can't see what that looks like," said Nachenberg. This is why DCO works so well against automated, pattern-matching antivirus programs, he explained.

DCO and Web 2.0

Dynamic code obfuscation (DCO) has begun to flourish in part because of Web 2.0's growing popularity.

Web 2.0 is basically a new way to use existing Web technology to enable participation, interaction, and collaboration among users, content providers, and businesses, rather than just the traditional viewing of static pages. For example, users could upload a book review to a Web 2.0-based bookseller's page.

Web 2.0 includes applications such as podcasts, blogs, wikis, and RSS feeds.

Security concerns

The ability of users to access and add code to Web 2.0 pages has raised concerns about security, which is largely in the hands of site developers. They might not always take steps such as validating user input or ensuring that generated pages are properly encoded.

The failure to develop pages securely can create vulnerabilities that hackers can exploit with DCO, data-stealing, phishing, and other attacks.

For example, XML syndication, which sends updated information from Web sites to subscribers, automates the retrieval of content that might include malware. Mashups—Web sites or applications that combine content or services from more than one source—connect dynamically to third-party sites that might not be well-protected.

Not only does Web 2.0 provide a way for hackers to launch DCO attacks, but the approach also makes users more vulnerable. For example, many users work with URL filtering, which provides security by automatically filtering out sites known to have malicious code or content objectionable to the user.

However, Web 2.0 can enable hackers to place malicious code on otherwise reliable sites. Thus, URL filtering users might go to a site they regard as safe and still suffer an attack.

JavaScript

DCO is used with exploits based on JavaScript, which is often used to enhance interactions on Web sites, making it a popular Web 2.0 developer's tool. JavaScript is best known for its use in Web sites and its ability to enable scripting-based access to objects embedded in other applications.

Developers can use JavaScript to write functions embedded in or included with HTML pages. These functions interact with retrieved pages and browser-based capabilities to perform tasks not possible with HTML alone. The JavaScript in pages can also make calls to Web and Web-service servers after a page has loaded.

Because JavaScript can run arbitrary server-provided code on a client with a browser that includes a full-blown interpreter, it could create security problems.

FIGHTING BACK

Despite the challenges, antivirus vendors are looking for ways to fight DCO. For example, Symantec is researching *in vivo blocking*, which would identify browser vulnerabilities even before vendors release a patch.

Such a system would monitor the browser and, if malware tries to exploit a vulnerability, intercept the code and block it, the company's Nachenberg explained.

Behavioral techniques

Because DCO makes detecting attacks difficult for pattern-matching antivirus software, behavioral antivirus approaches should be more effective, said the Burton Group's Lindstrom.

To determine whether a downloaded file is malicious, behavioral techniques examine the actions that it causes a system to begin taking. This approach thus avoids the need to look for code signatures while still stopping attacks before they cause problems.

Symantec, said Nachenberg, is looking into these types of approaches. "We want to see what the software is doing once it's on your computer, and if it's doing things such as system calls that are inappropriate, that's something we can detect," Nachenberg explained. "We can then remove the infection."

Other approaches

Finjan's Vital Security Web Appliance performs a static analysis of DCO code and tries to identify the pattern behind the obfuscation. For example, if the obfuscation shifts all characters by 5 bytes, the appliance looks for a built-in function that shifts them back to their original form to enable the host system to execute the code as originally written, the company's Ben-Itzhak explained.

"If we can detect this function and see the results of the descrambling, we can understand this code," he said. Once the appliance figures out the code's intended function, the product can block it in real time.

There might even be hope for a variation on the pattern-matching antivirus approach, according to CERT's Murawski. The key here is the subroutine that returns the obfuscated file to its original form.

Generally, these subroutines are in plain code, Murawski noted. If antivirus vendors can identify a big enough part of a subpiece that stays the same through numerous malware samples, they can use it as a code signature for the entire malicious file, he explained.

Finjan warns that the future might be full of DCO attacks. An audit the company conducted on UK service providers' traffic indicates that the vast majority of hacking attempts already use obfuscated code, explained Ben-Itzhak.

Murawski said the best advice for coping with this problem is the same advice that security experts have given for years. "We recommend utilizing a defense-in-depth approach," he explained. "This includes securing your Web browser, staying up to date on patches, running antivirus software, and utilizing a firewall."

"Because of all the stars that must be aligned for DCO attacks to succeed, this is typically going to be a mid-level priority amid issues like social engineering, more traditional malware, and application-layer attacks against databases," said the Burton Group's Lindstrom. "However, if you make your living on the Web, it should be a top priority." ■

__Karen Heyman__ is a freelance technology writer based in Santa Monica, California. Contact her at klhscience@yahoo.com.

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