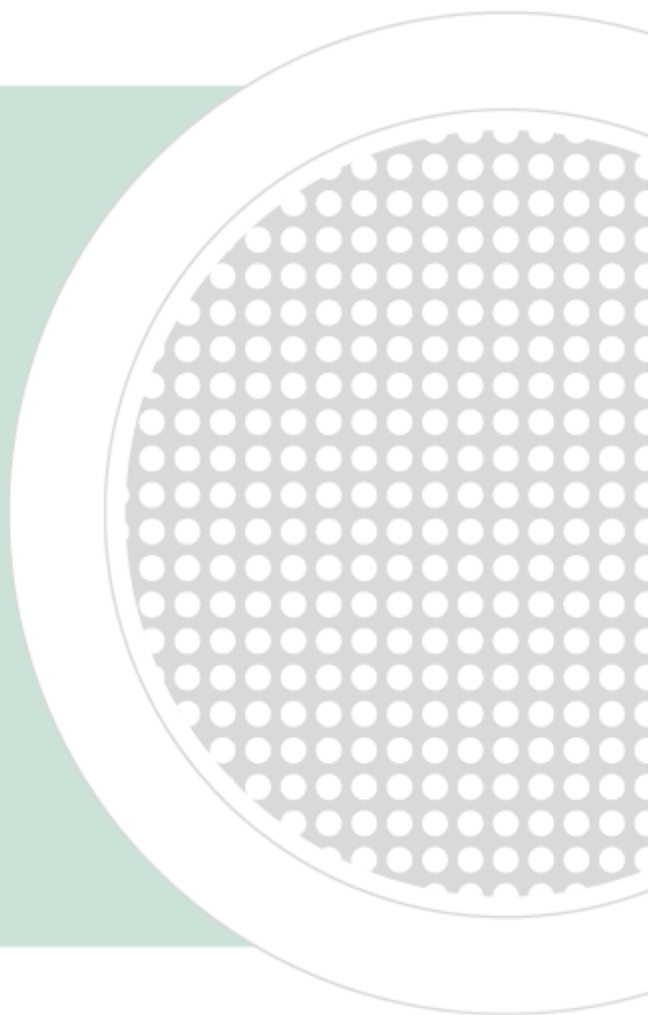


White Paper

Drive-by Downloads *The Web Under Siege*

By Ryan Naraine
Security Evangelist, Kaspersky Lab



Preface

The way that computer viruses and malware travel has evolved in much the same way that information itself has changed in the way it travels. In the early days, information was typically physically transported from one computer to another using a variety of storage media. By the early 1980's, information traveled over expensive private data networks. As the U.S. government pressured corporate suppliers to provide some consistency in transport and format of the information it received, the Internet sprang into real fruition. And with it came the ability for businesses of all sizes to transmit information over this “free” network, most often using e-mail and e-mail attachments. By the late 1990's, the highly publicized viruses that affected businesses and individuals worldwide followed suit – they relied on e-mail for replication and distribution.

Meanwhile, the World Wide Web was quickly maturing into a valuable platform for information exchange, global commerce, and workplace productivity. Slowly but surely, we saw the value of not e-mailing (pushing) information to all who might need it, but only sending a notification that included a link allowing users to browse the single copy of information accessible via the Web. 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. However, the amount of sophisticated, behind-the-scenes communication that occurs when Web browsers quietly interact with data stored on the PC, with desktop applications, and with Web servers would amaze not only most home users but also most (non-IT related) corporate professionals if they truly understood it.

Unfortunately, this maturity and sophistication has attracted the attention of well-organized malware purveyors who are now intent on using the Web to deliver their viruses, spyware, Trojans, bots, rootkits, and fake security software. The anti-virus industry refers to this covert downloading of malware, which occurs at Web sites without the user's awareness, as a “drive-by download.” In this white paper, we will explore what actually happens during a drive-by attack, the lures used to perpetrate attacks, the technology behind the attacks, and the use of drive-by download attacks in personal data theft and computer takeovers.

Table of Contents

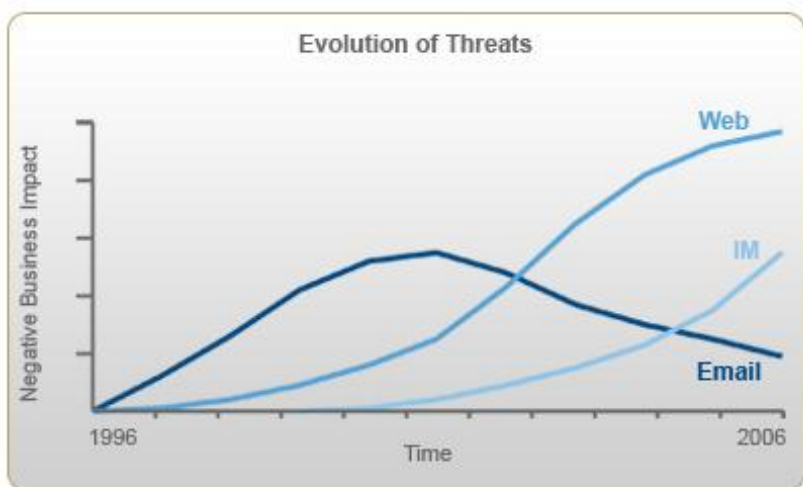


Preface.....	i
Understanding the Explosion	1
Browser Attacks.....	2
Anatomy of a Drive-by Attack.....	4
Exploit Kits.....	5
An Unpatched Monoculture.....	7
Conclusion: Avoiding Attacks	8

Understanding the Explosion

Before we explore drive-by downloads in more detail, it is useful to understand how this type of attack has exploded in recent years. It is also helpful to understand that the same malware (viruses, spyware, Trojans, bots, rootkits, and fake security software) can, and often is, delivered in different ways – sometimes by e-mail, sometimes by visiting a Web page, sometimes by other methods.

Drive by malware delivery is of increased appeal to cybercriminals simply because it is, in general, a more stealthy form of infection that results in more successful attacks. Figure 1 shows data from ScanSafe, a company that tracks Web-based malware threats, and illustrates how the impact on businesses has shifted from e-mail to Web and IM during the decade beginning in 1996.



Source: ScanSafe Threat Center

Figure 1 - Evolving Malware Delivery Methods

According to more recent data from ScanSafe, **74 percent of all malware spotted in the third quarter of 2008 came from visits to compromised Web sites.**

Now that you understand the growing magnitude of this problem, we will explain 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.

Browser Attacks

To fully understand the dramatic shift to using the Web browser as the attack tool, it is useful to revisit the history of major Internet-based computer attacks. During the “Internet worm era,” when attacks like Code Red, Blaster, Slammer and Sasser wreaked havoc on corporate networks, hackers used remote exploits against Windows operating system vulnerabilities. (A remote exploit is one in which the malware resides on a network-connected server, exploits legitimate code on the user’s computer, but doesn’t require prior access to the user’s computer to exploit the vulnerability in the code.) Malicious executables, such as Melissa, were also attached to e-mail or they arrived via instant messaging or peer-to-peer applications.

Microsoft reacted to the worm attacks in a positive way. They added a firewall, which is turned on by default in Windows XP SP2, and implemented several anti-worm mitigation mechanisms in the operating system. With automatic updates enabled on Windows, end users got some assistance with regularly applying operating system patches. Businesses and consumers also got smarter about blocking attachments or not clicking on strange executables. Both factors forced attackers to shift tactics, moving up the stack to target third-party applications and to perfect the art of social engineering.

This evolution also drove the emergence of a stealthy new technique – the drive-by download – that uses the browser as the mechanism to connect computer users to servers rigged with malicious exploits. 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. Figure 2, from the Google Anti-Malware Team, shows the basic structure of a drive-by download attack. These exploits can target vulnerabilities in the Web browser, an unpatched browser plugin, a vulnerable ActiveX control, or any other third party software flaws.

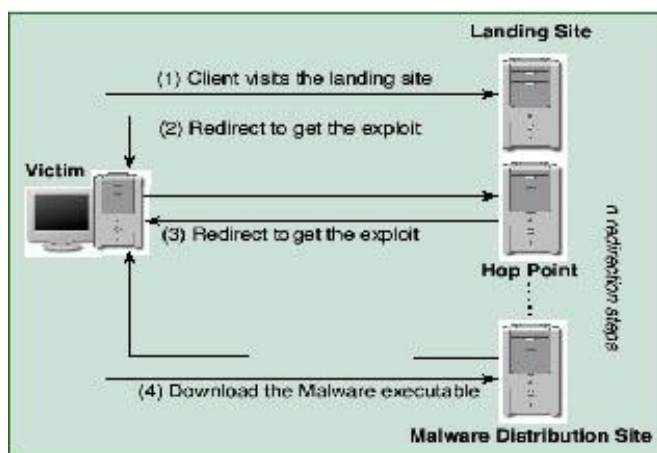


Figure 2 - Structure of a Drive-by Download Attack

As the figure indicates, there may be any number of redirections to different sites before the exploit is actually downloaded.

According to data from Kaspersky Lab and others 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. Figure 3, taken from that study, reveals an alarming upward trend occurring in the percentage of searches with an infected site during the study period.



Figure 3 – Search Results Containing a Harmful URL

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 either secretly exploit script or planted redirect code that silently launches attacks via the browser.

Anatomy of a Drive-by Attack

One high-profile Web site compromise in 2007 provides a glimpse at how drive-by downloads are launched against computer users. In the weeks leading up to the NFL Superbowl game, Miami's Dolphin Stadium site was hacked and rigged with a snippet of JavaScript code. (See Figure 4.)



```
Source of: http://www.dolphinstadium.com/ - Firefox
File Edit View Help
<!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://www.3.com/3.js"></script>
    <script src="/ssi/AL_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">
  </HEAD>
  <BODY>
  </BODY>
</HTML>
```

Figure 4 - JavaScript Code Used on Miami's Dolphin Stadium Site

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 attacker could later take advantage of the compromised computer in order to steal confidential information or to launch DoS 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 techniques¹ to avoid detection and to keep malicious servers online during the attack. Figure 5 shows a screenshot of the compromised Bank of India site with the malicious script used to launch the drive-by download attack.

¹ JavaScript obfuscation -- To avoid detection by security tools malware authors use this technique to make the malicious code difficult to decipher.

* iFrame redirects -- An iFrame is an HTML element that allows a Webmaster to embed an HTML document within an already existing one. Malware authors use this technique to embed code to redirect victims to malicious servers.

* Fast-flux -- This is a DNS technique used by botnets to hide phishing and malware delivery sites behind an ever-changing network of compromised hosts.

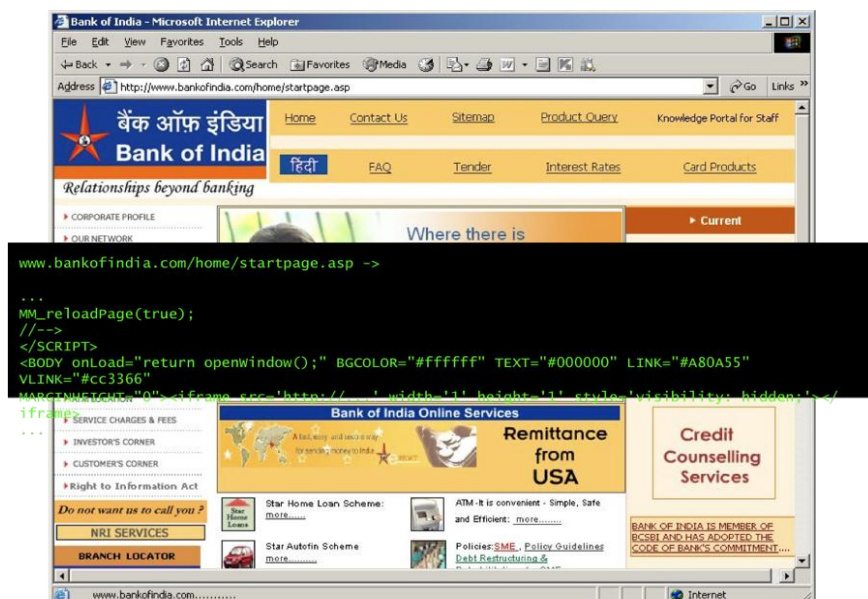


Figure 5 - Bank of India Site and Malicious Script

These are just two examples to highlight the extent of the problem on legitimate Web sites. In its tracking of Web-based malware threats, ScanSafe reported that by the middle of 2008, the majority of malware was being found on legitimate sites. Interesting highlights from ScanSafe's 3Q08 report are –

- The volume of Web-based malware increased 338 percent in 3Q08 compared to 1Q08 and 553 percent compared to 4Q07.
- Approximately 31 percent of all malware threats in September 2008 were zero-day malware threats. (A zero-day threat is one for which no patch exists.)
- The risk of backdoors and password stealing Trojans increased 267 percent in September 2008 compared to January 2008.

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.

Exploit Kits

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.

Figure 6 shows the variety of exploits contained in a single exploit kit intercepted during a JavaScript redirect attack. This example illustrates the popularity of exploits in Microsoft software, but also helps to illustrate how other software is simultaneously exploited to potentially increase the value of the exploit kit to cybercriminals.

Exploit	Microsoft Bulletin (if applicable)
MDAC remote code execution	MS06-014
ShockwaveFlash.ShockwaveFlash.9 exploit	
WebViewFolderIcon setSlice() exploit	MS06-057
Msdds.dll exploit	MS05-052
Microsoft Works exploit	MS08-052
Creative Software AutoUpdate Engine exploit	
Online Media Technologies NCTsoft NCTAudioFile2 ActiveX buffer overflow	
Ourgame GLWorld GLIEDown2.dll exploit	
DirectAnimation.PathControl buffer overflow	MS06-067

Figure 6- Content Present in a Single Exploit Kit

Identity thieves and other malware authors purchase exploit kits and deploy them on a malicious server.

An Unpatched Monoculture

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.

Looking at existing exploit kits, we see several old vulnerabilities, such as MS06-014 and MS05-052, remaining in circulation for years after the patch became available. (The third and fourth characters indicate the year the bulletin was issued.) Targeted exploit packs featuring only flaws in Adobe PDF Reader have been highly successful in spite of improvements to Adobe's security response process. Adobe Flash Player, which enjoys almost 100 percent penetration on Internet-enabled computers, is another big target, as is RealNetworks' RealPlayer.

With very few exceptions, the exploits in circulation target software vulnerabilities that are known – and for which patches are available.

Conclusion: Avoiding Attacks

In conclusion, it is important to note that most modern Web browsers – including Internet Explorer, Firefox, and Opera – have added anti-malware blockers that provide early-warning systems when users attempt to surf to a rigged Web site. These blockers provide good value but, because they are blacklist-based, they do not provide 100 percent protection to Web surfers.

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.

These steps toward managing the vulnerabilities continue to offer the greatest, most valuable protection against drive-by download attacks.



Kaspersky Lab, Inc. • 500 Unicorn Park • Woburn, MA 01801
phone: (781) 503-1800 • fax: (781) 503-1818
www.kaspersky.com

About Us

Kaspersky Lab delivers the world's most immediate protection against IT security threats, including viruses, spyware, crimeware, hackers, phishing, and spam. Kaspersky Lab products provide superior detection rates and the industry's fastest outbreak response time for home users, SMBs, large enterprises and the mobile computing environment. Kaspersky® technology is used worldwide inside the products and services of the industry's leading IT security solution providers. Learn more at www.kaspersky.com. For the latest on antivirus, anti-spyware, anti-spam and other IT security issues and trends, visit www.viruslist.com

Learn more at www.kaspersky.com