# SOPHOS

## Not all malware detection is created equal

The internet is now the number-one conduit for infecting users with malware. Sophos detects a new infected web page every few seconds. This white paper outlines the terms you need to know and the steps you should take to stay safe.

By Fraser Howard, Principal Researcher, Sophos

### Not all malware detection is created equal

The internet is now the number-one conduit for infecting users with malware. SophosLabs reports that in 2009, 23,500 new infected web pages are discovered every day. That's one every four seconds or so, four times worse than what it was in the same period in 2008 [1]. Malware authors are very successful with a popular method: compromising popular, high-traffic, legitimate sites in order to kick-start the infection process [2].

Users visiting a hijacked site have no way of knowing the site has been compromised because the malicious code is invisible but executed as soon as the page loads in the user's browser. The code typically will utilize cross-site scripting to fetch an even more malicious payload from a third-party site that will then attempt to leverage one of dozens of known exploits in the browser or operating system to infect it, steal data or subvert it into a botnet.

The scope of these attacks cannot be underestimated, since all types of sites—from government websites to educational establishments to popular news portals, blogs and social networking sites—have been targeted. As security vendors add detection for this kind of malicious web code, the attackers constantly evolve it in order to evade being caught. As this game of cat and mouse intensifies, the attackers have turned to using JavaScript for delivering their attacks. Why?:

- » JavaScript is very powerful and universal with rich capabilities supported in all browsers and operating systems.
- » JavaScript provides great flexibility for them to hide (or obfuscate) the malicious code.

A simple illustration of this is shown in Figure 1. In A, the page has been injected with a simple HTML iframe, which would cause the browser to load malicious content from the specific remote site when that page is viewed. In B, the page is injected with a short JavaScript. The payload is not immediately visible to the eye—it is using very simple obfuscation. In fact, the payload is exactly the same as for A. The JavaScript simply writes the exact same iframe object to the page when it is viewed in the browser.



Figure 1: Simple example of a web page compromised in different ways, but where the payload is exactly the same. The injected iframe (A) and script (B) cause the browser to load content from the malicious remote site when the page is browsed.

The flexibility of JavaScript allows almost an unlimited number of ways for this payload to be obfuscated in B. This presents a problem for content scanners because the payload is effectively hidden when the page content is scanned.

Consequently, proactive generic detection becomes harder to achieve. We have seen numerous massdefacement attacks during 2009 in which tens of thousands of legitimate sites have been compromised (their pages injected with malicious JavaScript code). These attacks invariably use heavily obfuscated JavaScript as a means of evading detection for as long as possible.

A good example of this attack is known as *Gumblar* [3], in which many sites were injected with a malicious script that used simple character substitution to hide its payload (Figure 2A). As you can see, the payload is not visible in the injected script. But after manual deobfuscation, the payload is obvious (Figure 2B)—loading of a malicious script from a remote site.



Figure 2: Malicious script (A) injected into legitimate pages as part of the Gumblar mass-defacement attacks during 2009. The manually deobfuscated script is shown in (B).

The reach of mass-defacement attacks like such as Gumblar can be huge. Very quickly after adding detection (as Troj/JSRedir-R), that threat quickly rose to the top position in SophosLabs' web threat stats, dwarfing other threats at that time (Figure 3).



Figure 3: Top web threats detected between May 6 and 13, 2009. Troj/JSRedir-R (aka Gumblar) dwarfs all other detections at >40% of total reports.

#### **Commercial script packers**

There is nothing malicious in obfuscating JavaScript code. In fact, there are commercial tools available that individuals can use in order to obfuscate their code. Why?:

- » Protection of intellectual property: Users may choose to use such tools to obfuscate their code in an attempt to prevent others from copying it.
- » Efficiency: Some tools can produce smaller scripts that are quicker to download, resulting in more responsive sites.

This creates yet another problem for analysts and content scanners. When attackers use commercial tools to obfuscate their malicious scripts, antimalware analysts have to be careful not to generate a false positive on legitimate scripts that are obfuscated with the same tool [4].

#### Server-side polymorphism (SSP)

Many of today's threats also use advanced scripting techniques on the server in order to create polymorphic malicious code. For example, during 2009, SophosLabs identified several attacks aiming to infect users with Zbot [5,6] that were aggressively using SSP. In these attacks, the malicious scripts being used to exploit vulnerabilities on user machines were dynamically created on the server, resulting in a slightly different script on each request.

SSP is essentially a special case of obfuscation, which again poses a challenge to anti-malware analysts and content scanners.

#### **Malicious PDF documents**

SophosLabs has reported on the widespread use of PDF documents in attacks in 2009 [7]. The reason is that attackers have aggressively targeted vulnerabilities in common PDF reader applications as a means of infecting users. What is less known is that JavaScript is typically used in such attacks. Adobe Acrobat includes support for embedded JavaScript within PDF documents [] in order for users to create complex and dynamic documents. However, this provides a mechanism for attackers to construct malicious PDF files that use embedded JavaScript in order to exploit application vulnerabilities.

The script obfuscation and SSP techniques discussed above are equally applicable to JavaScripts embedded in PDF files.

### How Sophos does it better: Improved JavaScript handling

New functionality specifically designed to provide improved handling of JavaScript will be included in the core anti-malware engine. The improvements include:

- » Ability to tokenize and parse JavaScript: This enables SophosLabs analysts to write more efficient generic detections for JavaScript content.
- » JavaScript emulation: The core anti-virus engine includes a JavaScript emulator that will provide a generic mechanism to deobfuscate script contents. This enables the engine to see the payload of the script, therefore providing a significant boost to proactive generic detection.

The existing Sophos anti-malware engine (and some of the competitor engines) already has some ability to "unpack" obfuscated scripts. So how is this new technology different? Existing unpacking technologies rely on recognizing specific obfuscations and writing code to handle each one (where possible). This approach does not scale with the volume of malware today. There are virtually limitless numbers of ways in which scripts can be obfuscated, and we know that attackers modify their techniques frequently. A generic solution for deobfuscation is required. This is what JavaScript emulation provides.

The key benefits of this new technology to Sophos customers are:

» Increased detection rates of malicious script content and PDFs » Increased proactive (zero-day) protection against new attacks

In November 2009, Sophos Web Security and Control will include this functionality in the core anti-malware engine. This update will be rolled out to all customer appliances as part of the monthly update package (no additional work will be necessary).

From early 2010, the technology will be included in the Browser Helper Object (BHO) component of *Sophos Endpoint Security and Data Protection*, which runs and protects Internet Explorer from processing any malware in requests.

To learn more about Sophos please visit: http://www.sophos.com

- 1 http://www.sophos.com/blogs/gc/g/2009/07/24/threat-report-july-2009/
- 2 http://www.sophos.com/security/technical-papers/modern\_web\_attacks.html
- 3 http://www.sophos.com/blogs/sophoslabs/v/post/4405
- 4 http://www.theregister.co.uk/2009/09/04/mcafee\_false\_positive/
- 5 http://www.sophos.com/blogs/sophoslabs/v/post/882
- 6 http://www.sophos.com/blogs/sophoslabs/v/post/2090
- 7 http://www.sophos.com/blogs/sophoslabs/v/post/5968
- 8 http://partners.adobe.com/public/developer/pdf/topic\_js.html

Boston, USA | Oxford, UK © Copyright 2009. Sophos Plc

All registered trademarks and copyrights are understood and recognized by Sophos. No part of this publication may be reproduced, stored in a retrieval system, or transmitted by any form or by any means without the prior written permission of the publishers.

