

Writeup by Amit Klein (Trusteer): Address Bar Spoofing for IE6

Source: <http://www.derkeiler.com/Mailing-Lists/securityfocus/bugtraq/2008-10/msg00206.html>

- *From:* Amit Klein <amit.klein@xxxxxxxxxxxxx>
 - *Date:* Mon, 27 Oct 2008 18:14:20 +0200
-

Address Bar Spoofing Attacks against Microsoft Internet Explorer 6

Amit Klein, Trusteer

Summary

=====

IE6 is the second most popular web browser (after IE7), with market share of around 25% (according to recent surveys e.g. <http://marketshare.hitslink.com/report.aspx?qprid=2>).

This write-up presents two new phishing attack techniques, abusing an address bar issue (security vulnerability) with IE6 in combination with non-standard DNS domain names. The net result is that a phishing site may present itself via a link that when clicked in IE6 displays an almost indistinguishable URL from the one in used by the genuine site. The technique is new, i.e. it's different than the ASCII similar characters and IDN homographs attacks.

There are two techniques: the first technique presents an address bar which is very similar (visually) to the address bar expected for the genuine domain, by abusing the NBSP character. The second technique presents an address bar visually identical to the one expected for the genuine domain, using the fact that a non-DNSish characters are not displayed in the address bar in some cases. This technique requires registration of a non-standard domain, hence it is probably theoretic only (although "site down" imitation is still possible).

The attacks were verified with Windows XP SP2 and Windows XP SP3.

Introduction

=====

URLs typically include host name, which tells the browser (after DNS resolution) where to fetch the resource from. While regular host names contain alphanumeric characters (a-z, A-Z and 0-9), dots, hyphens and (in Intranets only) underscores, it is possible to construct (at least syntactically) URLs whose host part contain any octet (as explained in RFC 1035 section 3.1). The interpretation of such characters when presented as links (when IDN is not supported by the browser, see below) by the browser and by the DNS infrastructure, as well as the way those characters are presented by the browser (in the address bar) are the subject of this write-up.

Non-DNS characters can be provided to the browser in several ways (assuming e.g. an anchor HTML tag context):

- * In raw form, i.e. as a byte (octet), e.g. \$
- * In HTML-encoded form, e.g. $
- * In URL-encoded form, e.g. %24

In raw form, the data is provided as-is. In HTML-encoded form, the data is considered Unicode, and may undergo encoding. In URL-encoded format, the data is (again) directly decodable into raw form. The difference is subtle, but important. The octet values 00-7F (corresponding to the ASCII characters) have a single interpretation across all systems. However, octet values 80-FF may have different interpretation depending on the code page and encoding system in use.

Address bar spoofing in IE6

=====

Non-DNS characters

Within the ASCII range (00-7F), only the DNS subset of ASCII characters is allowed.

As for higher values (e.g. A9 or %A9): IE6 uses DnsQuery_A to resolve the name. DnsQuery_A assumes that the characters are in the "current" Windows ANSI codepage (e.g. Windows-1252 or Windows-1255, see <http://www.microsoft.com/globaldev/reference/WinCP.msp> for a list of Single Byte code pages). It translates the characters into UTF-8 representation and sends them this way. So %A9 is URL-decoded into the byte (\xA9) by IE6, then this raw byte is forwarded to DnsQuery_A, which interprets it according to the current codepage (e.g. Windows-1252 or Windows-1255) as COPYRIGHT_SIGN, moves to Unicode (U+00A9), and UTF-8 encodes this symbol (into the 2 byte sequence (\xC2) (\xA9)). The net result is that <http://www.foo%A9bar.com> goes out as a DNS query on `www.foo(\xC2)(\xA9)bar.com`.

As it happens, almost all single-byte character sets (Windows-1250...Windows-1258) interpret (\xA9) as COPYRIGHT_SIGN, and the one exception being Windows-874 (Thai) which does not.

NOTE: the code page for a particular Windows box is determined through the Control Panel (Regional and Language Options -> advanced [tab], in the Languages for non-Unicode programs). The Windows ANSI code page is derived from the language specified via the table as provided in <http://msdn.microsoft.com/en-us/library/ms776260.aspx>. For example, if the language is English (all variants) then the Windows ANSI code page is 1252, whereas if the language is Hebrew, then the Windows ANSI code page is 1255. As can be seen, the only languages whose code page is not Windows-1250...Windows-1258 are the far east languages Chinese, Japanese, Korean and Thai. So with the exception of these languages, IE6 will request a DNS resolution for [www.foo\(\xC2\)\(\xA9\)bar.com](http://www.foo(\xC2)(\xA9)bar.com) when it navigates to <http://www.foo%A9bar.com/>.

Attack #1: Raw/HTML-encoded characters

IE6 allows "raw" high-bit characters to be typed in the address bar, e.g.

[http://www.foo\(c\)bar.com/](http://www.foo(c)bar.com/)

In such case, the character is displayed in the address bar (unlike %A9 which is not).

It is possible to present this URL in a link, e.g.:

```
<a href="http://www.foo&#x00A9;bar.com">FooBar</a>
```

NOTE: An HTML-encoded character is displayed as the corresponding Unicode symbol. However, if this symbol is not mapped to the current code page, IE will not resolve the host name (it shows an "invalid syntax" error page).

A more interesting, and phishing related example is using the Non-Blocking Space character (NBSP, Unicode U+00A0). This character is rendered in the address bar as a space (NBSP is mapped as 0xA0 in all single-byte character set codepages, i.e. Windows-1250...Windows-1258 and Windows-874). Thus it opens up an address bar spoofing trick similar in effect to a one already disclosed (first reported in BugTraq December 2003: <http://www.securityfocus.com/archive/1/346948>, then picked up by CERT <http://www.kb.cert.org/vuls/id/652278> and fixed by Microsoft as MS04-004).

For example, consider the following phishing link (mimicking

Now, here's where it gets interesting: high-bit characters will not be displayed in the address bar. So instead of showing visually as "http://www.foo%A9bar.com/ (or "http://www.foo(c)bar.com/") as one may expect, the address bar will show "http://www.foobar.com/.

Theoretically, this can be used for phishing. A phisher can register, say foo(\xC2)(\xA9)bar.com and use that in a phishing URL (http://www.foo%A9bar.com/). When clicked, the IE6 address bar will display the expected URL, http://www.foobar.com/. However, this vulnerability seems to be theoretic only, since (in the author's limited experience), it's not possible administratively to register such domain names.

As for domain security, as far as IE6 is concerned, these are two different domains. Cookies are not shared, access across domains is denied, SSL certificate will not match, etc. Also, the Host header includes the value with the original raw character – i.e. the Host header is:

Host: www.foo(\xA9)bar.com

Even if no real domain can be registered, this can still be somewhat of an annoyance. For example, spam can offer a URL as evidence that a company's site is not available, or was hacked. So if an attacker wants to defame www.foobar.com, he may do so by sending spam with text such as "foobar inc. went chapter 11 – site is down. Check out http://www.foo%A9bar.com/. This will end up in DNS resolution failure.

Auto-completion applies to the address bar string (not the real URL), hence auto-completing, say, www.fo will result in www.foobar.com (the real domain name), and the browser will navigate to the genuine site.

Vendor status

=====

Microsoft (MSRC) was informed of the two issues on January 13th, 2008. MSRC acknowledged the two problems and assigned the first one the ticket MSRC7899, and the second one MSRC7900. However, Microsoft declined to fix the issues.

Additional notes and observations

=====

Label and name lengths

Labels are limited (per RFC 1034 section 2.3.4) to 63 octets. This means that no more than 31 consecutive NBSPs can be used. The trick is to split them into labels (by inserting a dot). Names are limited to 255 octets (per RFC 1034 section 2.3.4). This includes the accumulated length of all labels, plus a length octet preceding all labels (including the 0-length root label). Apparently, both restrictions are enforced by DnsQuery_A (in fact, names are limited to 254-256 bytes, including dots).

Additionally, it seems that a non-DNS character counts as 3 octets towards the total name limit (but not towards the label length limit). A possible explanation is that when a non-ASCII character is encountered, the worst case UTF-8 representation length is used (3 octets) rather than the actual UTF-8 representation length (2 bytes for characters whose Unicode index is smaller than 0x800 e.g. NBSP, 3 bytes for all other Unicode characters). Thus, it is impossible to use more than 85 such characters.

HTTP Caching

There seems to be an additional bug in IE6 regarding how resources whose URL contain high-bit set bytes are cached. The key URL is constructed in an erroneous manner. It seems that the key is constructed as following: take the string consisting of the URL-encoded URL, e.g. http://www.foo%A9bar.com/, and overwrite it with the decoded (shorter) URL, http://www.foo(\xA9)bar.com/, in this case resulting in http://www.foo(\xA9)bar.comom/. Obviously no regular URL will match this key, so the caching is meaningless.

The key used for caching retrieval is probably the URL-encoded version of the URL. And since it's never there, the effect is of no-caching.

DNS caching

It was also verified that BIND 9 (the most popular DNS server software) is capable of serving such domains, both as an authoritative server and as a caching DNS server (verified with BIND 9.2.4 as an authoritative and caching server, and 9.4.1-P1 as a caching server). In order to configure BIND to serve the domain as an authoritative name server, the high-bit bytes (\xC2)(xA9) should be inserted to the zone file. Care should be

exercised here with the choice of editor, since some text editors don't handle high-bit bytes well. It is advised to review the file contents with a hex-dump tool (e.g. od) to ensure that the correct bytes were entered.

Windows DNS server (verified with Windows 2003 for Small Business Server SP2) as a cache server also supports such domain names (no testing was done regarding Windows DNS server as an authoritative server).

Root and TLD support

Apparently, the root servers and the .COM/.NET gTLD servers are indifferent to non-ASCII characters in sub-domains, and they will happily respond to such queries pointing at the authoritative DNS server (which would be the attacker's server).

IDN and homographs

The first attack, while abusing the same underlying phenomenon (different logical symbols which are rendered into graphically identical or almost indistinguishable forms – SP and NBSP in our case) is nonetheless completely different than the homograph attack (<http://www.shmoo.com/idn/homograph.txt>), which makes use of the IDN extension to DNS.

The second (theoretic) attack is completely different because the non-DNS characters simply do not appear in the address bar.