If your domain name parser can't handle internationalized domain names, then maybe that's your parser's problem

devblogs.microsoft.com/oldnewthing/20211109-00

November 9, 2021

Raymond Chen

A security vulnerability report arrived that went roughly like this:

> Internet Explorer has a security vulnerability that allows an attacker to bypass domain filtering. Suppose my web site filters domain names, say with the following test:
>
> ```
> if ($frame_domain == "microsoft.com") {
>     block();
> }
> ```
>
> An attacker can construct a frame which targets an intentional misspelling:
>
> ```
> <iframe src="https://ⓜicrosoft.com" ...>
> ```
>
> Even though the "ⓜ" is the Unicode character CIRCLED LATIN SMALL LETTER M (U+24DC), the web page that is shown in the frame is indeed `microsoft.com`. The web browser rewrote the domain, allowing an attacker to bypass filtering.

Yes, this is all true. The web browser rewrote the domain, allowing an attacker to bypass filtering. But the bug is not in the web browser. The web browser is doing exactly what the standard says it's supposed to do: Unicode Technical Standard #46 describes how so-called *international domain names* are converted to ASCII for domain lookup purposes. One of the steps is to map the code points according to the IDNA Mapping Table, and the IDNA Mapping Table says that character CIRCLED LATIN SMALL LETTER M (U+24DC) is mapped to LATIN SMALL LETTER M (U+006D).

The bug is in the code which tries to block access to `microsoft.com`. It's performing a literal string comparison against `microsoft.com` without going through the IDN conversion process. Indeed, you didn't even need to use IDN to attack the filter.

```
<iframe src="https://microsoft.com." ...>
```

As we learned some time ago, `microsoft.com` is technically shorthand for the full name `microsoft.com.` with a trailing period. But the above filter doesn't handle that case, so appending a dot easily avoids the filter.

Hang on, it's even easier:

```
<iframe src="https://Microsoft.com" ...>
```

The filter used a case-sensitive comparison, but domain names are case-insensitive, so `Microsoft.com` sneaks past the filter.

And, of course, you could gain access to `microsoft.com` by using its IP address explicitly.

None of this is the fault of the web browser. The problem is in the attempt at filtering the domains that can be placed inside frames. If you set up an insecure system, don't be surprised that it has a security issue.

Raymond Chen

Follow