Preparing for uniform resource identifier (URI) exploits

By Michael Cobb

Published: 2007-10-11 · Archived: 2026-04-06 01:49:05 UTC

By

- Michael Cobb

Published: 11 Oct 2007

Most people using the Internet know what a Web address is, or at least use the term as a non-technical synonym for a URL or uniform resource locator: a string of characters used to identify a resource and a means of locating it.

A URL is, in fact, a subset of uniform resource identifiers, or URIs. URIs use a defined syntax to provide a simple and extensible means for recognizing and accessing an Internet resource. The identifiers can do so without regard to the application or platform used. The URI syntax is essentially a URI scheme name, such as 'http' (Hypertext Transfer Protocol), followed by a colon and then a scheme-specific part. For example, the URL:

http://www.microsoft.com/en/us/default.aspx

…is a URI that identifies the resource of Microsoft's home page. The identifier also confirms that the page can be located using HTTP from a network host named www.microsoft.com.

Mozilla developers often use a URI that begins with 'rdf,' which enables access to a particular datasource. The URI 'rdf:history,' for example, returns the datasource that holds information related to a user's browsing history.

URIs are also used to launch an application from within a browser. During the installation process, browsers automatically store, or register, various URL protocol handlers, such as mailto and nntp, in the Windows registry. Each of these protocol handlers is associated with an application so that the browser launches appropriate software when requested. So, clicking on a Web link that begins "aim:goim," for example, will open an AIM instant message window.

Although this functionality helps make interaction between applications less complicated for the user, many software developers do not fully understand the complexity of URIs and the possible consequences of placing

them into the registry. Basically, a URI handler is going to increase the attack surface of an application. Let's look at why this is.

Security expert Thor Larholm recently highlighted an interesting fact about Firefox. When the browser is installed, it registers a URL protocol handler called "FirefoxURL," which potentially allows a URI in a Web page to launch Firefox. Because of the way in which the URL handler is registered, Windows cannot tell what type of input or request is valid. So when Internet Explorer encounters a reference to content inside the FirefoxURL URL scheme, it calls the ShellExecute command and passes the entire request URI without any input validation. That means there is no check on the data being passed to the ShellExecute command. By crafting a malicious URL, an attacker can then pass arguments, parameters and data to an external application that will run when the requested URI is loaded. The malicious link can be embedded in a Web site or sent via an HTML email.

For more information on URI security

Learn more about Mozilla Firefox's input validation error.

Common handlers can possibly be exploited with 'a single unexpected URI.' See how.

Should we be scaling back our Web browser security expectations?

Though Mozilla Corp. has released a patch, URI problems are not solely browser issues. Researchers Billy Rios and Nathan McFeters claim to have discovered a "functionality-based exploitation." Using the legitimate features of a popular software program launched via the protocol handler, the two claim to have found a way to steal data from a victim's computer and upload it to a remote server.

Such URI exploits are going to start a fresh round of problems for developers and users alike. Developers need to assess whether their applications really warrant the registering of a URI. Any application that registers an identifier needs to validate and sanitize any input. If attackers can execute applications using the exploit technique, they will be doing so with the privileges of the targeted user.

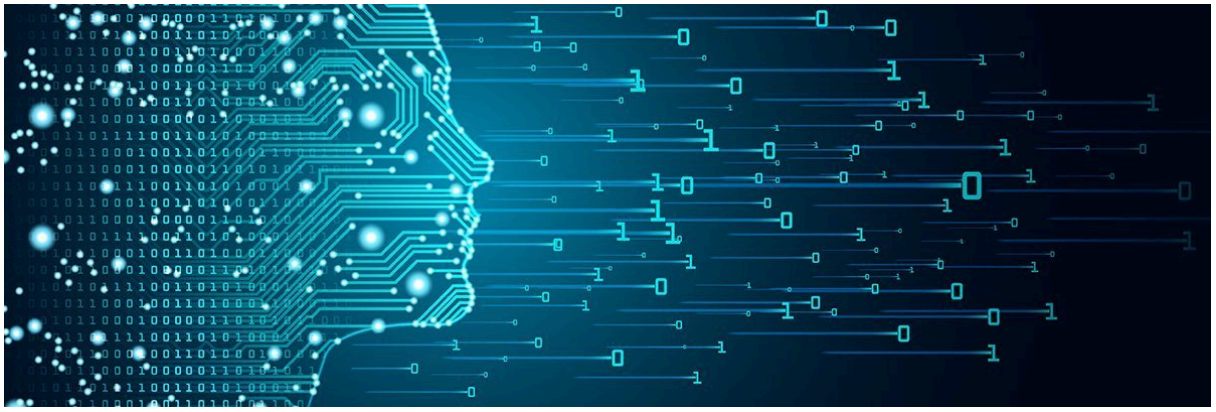
URI schemes are a valuable resource and are intended only to address information spaces that are globally useful. Developers who create new URI schemes to address spaces which are not useful to the Web in general, which aren't registered, or which break some axioms of Web architecture, are also creating another exploit for hackers to use.

The best way to protect against a possible URI attack is to install a browser vendor's latest fixes. Network administrators should remind their users to never follow links from untrusted sources or open unsolicited HTML email. The attack relies on user interaction, so for such an attack to be successful, the victim needs to follow a link to a malicious site or open a malicious email. Finally, security professionals must ensure that users' accounts only have the minimum access rights that are necessary for them to do their work.

About the author:

Michael Cobb, CISSP-ISSAP is the founder and managing director of Cobweb Applications Ltd., a consultancy that offers IT training and support in data security and analysis. He co-authored the book IIS Security and has written numerous technical articles for leading IT publications. Mike is the guest instructor for several SearchSecurity.com Security Schools and, as a SearchSecurity.com site expert, answers user questions on application security and platform security.

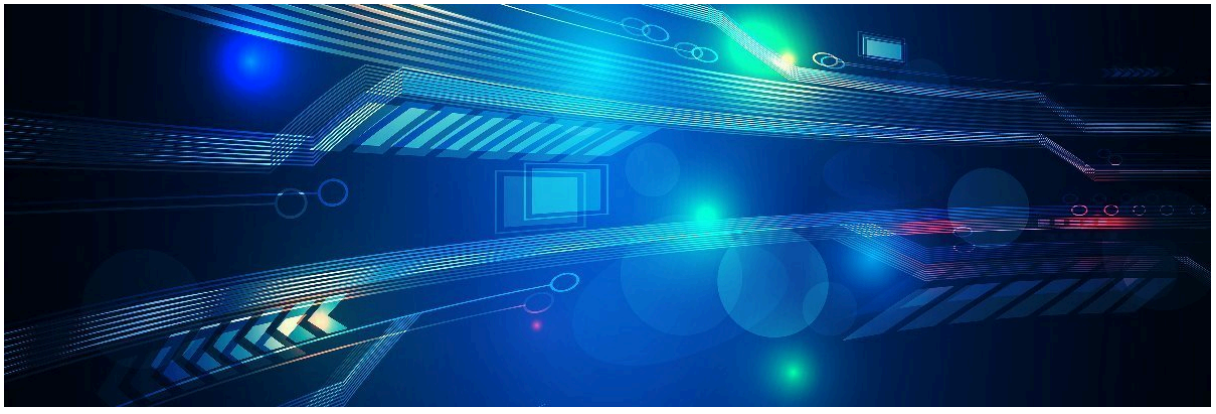
Dig Deeper on Application and platform security

- 

  What is a uniform resource identifier (URI)?

  By: Rahul Awati

- 

  What is a unique identifier (UID)?

By: Gavin Wright

- 

What is a URL (Uniform Resource Locator)?

By: Jessica Scarpati

- 

Top REST API URL naming convention standards

By: Raghu Karan Adapala
