

# URL Lock

Steven Lawrance  
Yue Zhao  
Dongwhan Kim

## Introduction

The increase in the number of households with Internet connections has brought with it an increased need for control of which web sites to allow and which web sites to deny. Protecting children from inappropriate content and blocking non-work-related sites in a company have emerged as requirements in modern networks. URL Lock addresses these requirements using string matching on the web site address. Steven Lawrance, a member of the team that developed URL Lock, had previously implemented an Internet Explorer (IE) version of the URL Lock, but its configuration interface had numerous usability drawbacks, hindering its proper use with users who do not know regular expressions. Furthermore, the IE URL Lock did not work with other browsers.

The team set out to create a simpler configuration user interface and to extend the URL Lock to Mozilla Firefox. While regular expression editing is possible, it is hidden from the main view and is intended only for advanced users. The main user interface makes rule editing simpler by letting administrators create rules either by typing in web site addresses or selecting a web site from their bookmarks or history lists. When pages are blocked, a friendlier page than the one that previously existed is now presented. Pages that link to disabled pages will now have those links disabled, and search results in Google for blocked results will have an opaque disabled appearance.

## Architecture

The user interface is structured into three main modules: the configuration user interface, URL address blocking, and search engine link disabling. Behind the scenes, a library acts as a data model to support the front-end and supports event propagation when changes are committed. The library is intended to also communicate with the operating system, the Perl-compatible regular expression library, and the Windows registry.

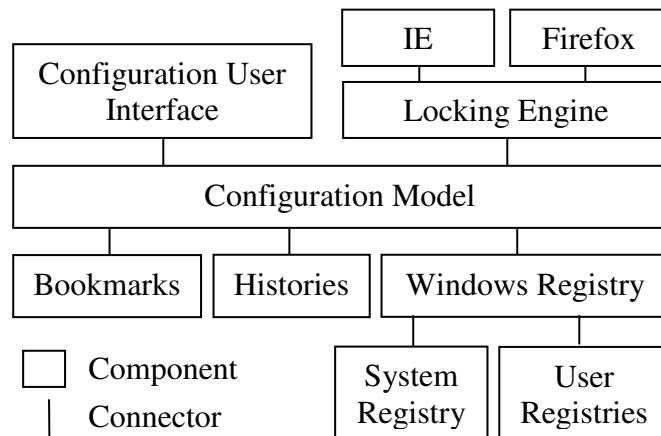


Figure 1: Runtime view of URL Lock

The runtime view in Figure 1 shows a higher level of abstraction of how the components and connectors relate to each other. The next three sections focus on the “Configuration User Interface” and “Locking Engine” components.

## Configuration User Interface

The group spent a substantial amount of time redesigning the user interface after devising a modified interface. A “think-aloud” session with three users quickly identified that the initial modified user interface had a substantial number of usability problems. Some examples include a search label was labeled “scroll to row” and a search box that was on the bottom of the list, causing users to have a hard time locating the search box. Furthermore, the blocked URLs list was too narrow as the URL addresses were typically longer. Our test users also gave the team feedback on how the initial modified interface was too cluttered overall. Other feedback items included the description on the top not making much sense, misaligned buttons, and an unattractive user interface.

For the new user interface design, the team added more images and icons and then addressed some of the usability problems. As examples, the “scroll to row” label was changed to “search,” and the entire search box was moved above the URL list so that users can see it easily. The width of URL list is larger so that the user can view the entire path of the URL. Instead of having a description on the top of the window, we decided to add a commonly used question mark beside the search textbox to save more display space as well as a tooltip to explain to user what the question mark was for. The user list has been moved below the blocked URL list. To make UI more attractive, we came up with URL Lock icons and a color scheme. An icon has been put onto each configuration window to keep consistency. The team also added a help menu on the new interface, which includes “what’s URL lock?” and “about”. The “what’s URL Lock?” menu item will redirect the user to IE URL Lock homepage. The “about” menu item will display the credits for URL Lock.

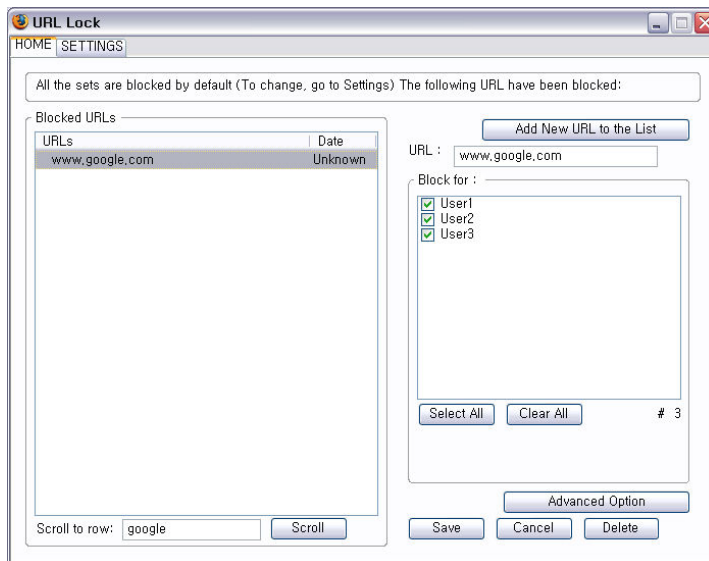


Figure 2: Initial modified user interface

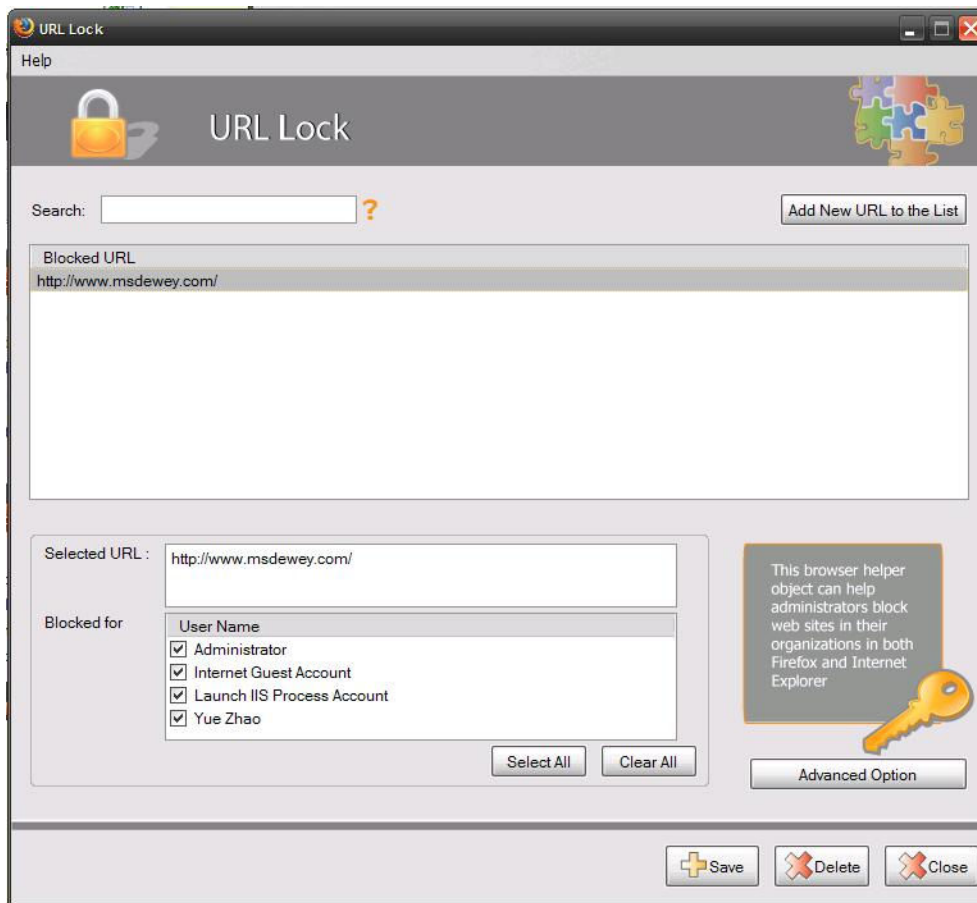
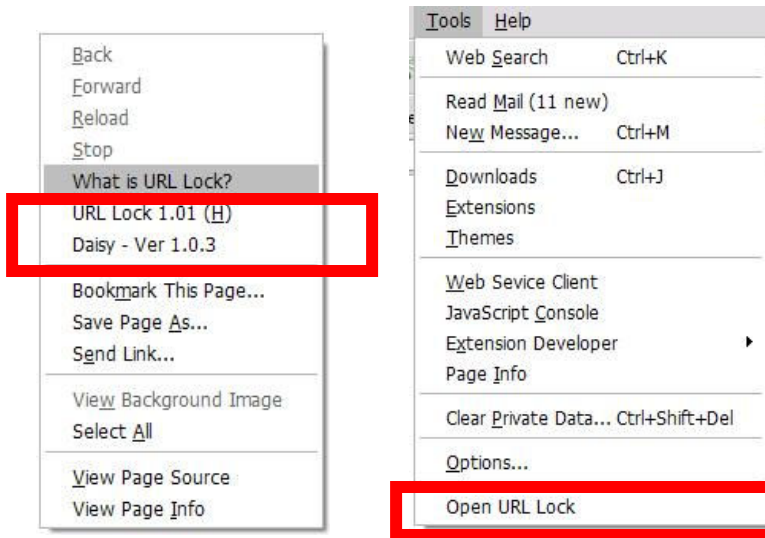


Figure 3: New user interface

Part of the main user interface is the browser overlay. The system adds a “Open URL Lock” menu item under the Tools menu that permits the user to open the URL Lock configuration window. The mouse right click menu has a “what is URL Lock” that redirects the user to the IE URL Lock main website. The “URLLock 1.01” menu item opens the URL Lock configuration window. This part is implemented in URLLockOverlay.xul and the Javascript functions are included in urlLockOverlay.js. The image below displays where these popup menu items exist.



Another part of the URL Lock is the configuration window. In the main window, shown in Figure 3, a list box on the top displays all the defined rules. The rules are intended to be retrieved from the Windows registry, though the present version does not persist rules. The search text field allows the user to search over the list by typing in search criteria, automatically updating the list of rules shortly after the user types in criteria. The question mark icon pops up a small window explain what the blocked URL is. Left clicking on a rule will show its detail. The detail includes the list of users that the rule applies to. The user list is read from the Windows user account database via the native class in the library. It includes the user name, full name, and administrative status.

To add new rule, click on the “Add New URL to the List” button on the top right corner. In the resulting dialog box, shown in Figure 4, an administrator can either manually type in a URL address or pick an from the bookmark or browser history. To open bookmark and history by simply click on the “Bookmark” or “History” button beside the URL text box, left click on the preferred link, and drag it into the URL text box. Double-clicking a link opens it in the main browser window, so dragging and dropping the link into the URL text box is necessary to block that bookmark or history item. The bookmark and history popup window code are both from Firefox browser content (chrome://browser/content/bookmarks/bookmarksManager.xul for Bookmark manager window, chrome://browser/content/history/history-panel.xul for history window). The interface was modified to fit the URL Lock extension’s code. The list box displays all the Windows user accounts. The rule becomes active for the selected users after the OK button is pressed. In a future version, saving a rule will change the URL Lock configuration in the Windows registry.

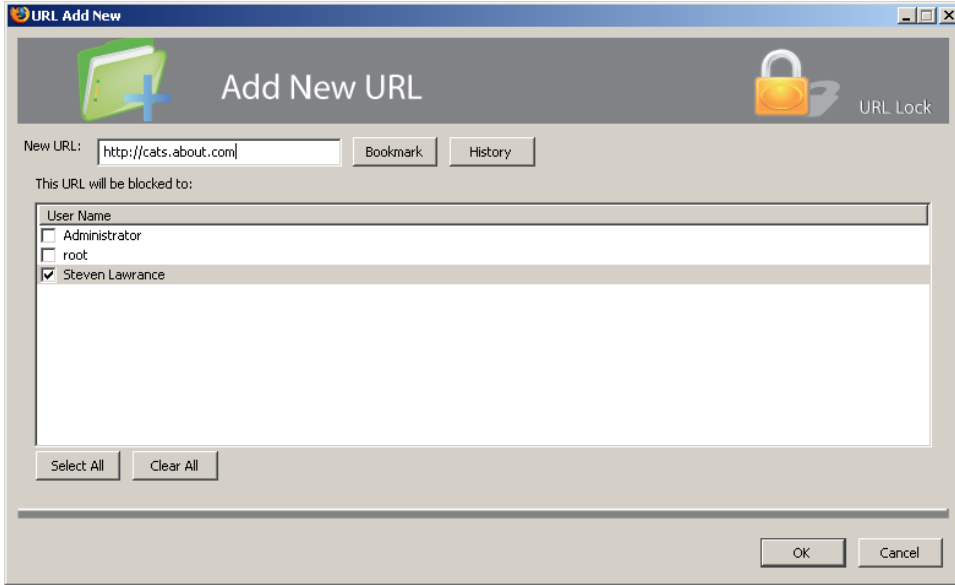


Figure 4: Add New URL dialog box

Advanced users can configure paths, ports, domains, and protocols in the “Advanced Options” window. This window is shown below in Figure 5. Super-advanced users can also edit the generated regular expression. While the advanced user interface is finished, the glue code and some of the support code in the library do not exist in this release. As a result, the advanced user interface is not yet functional.

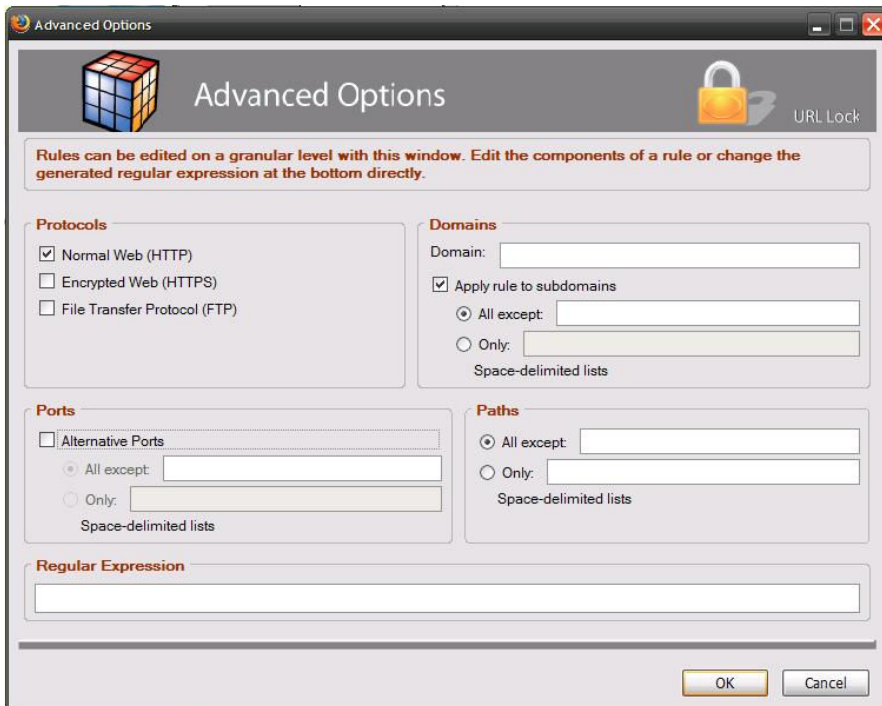


Figure 5: Advanced options

## Search Engine link disable

The team had decided to handle pages that contain disabled links, including search engines. In order to save the hassle of clicking on a blocked link and get a warning message, the URL Lock will simply disable the URL that was blocked and make it grayed out, giving it a disabled appearance. A URL Lock icon is added beside the blocked URL so that the user can recognize the icon and understand that the link was blocked by an administrator. The implementation concept is similar to URL blocking. A page load event listener was added to the current browser so that when Google search results are loaded, googleDisableOne() is called to check each result so that groups of HTML elements associated with an individual search result can be disabled together. The blocking and disabling implementations exist in linkDisable.js. The Google disabling is shown in Figure 6 below.

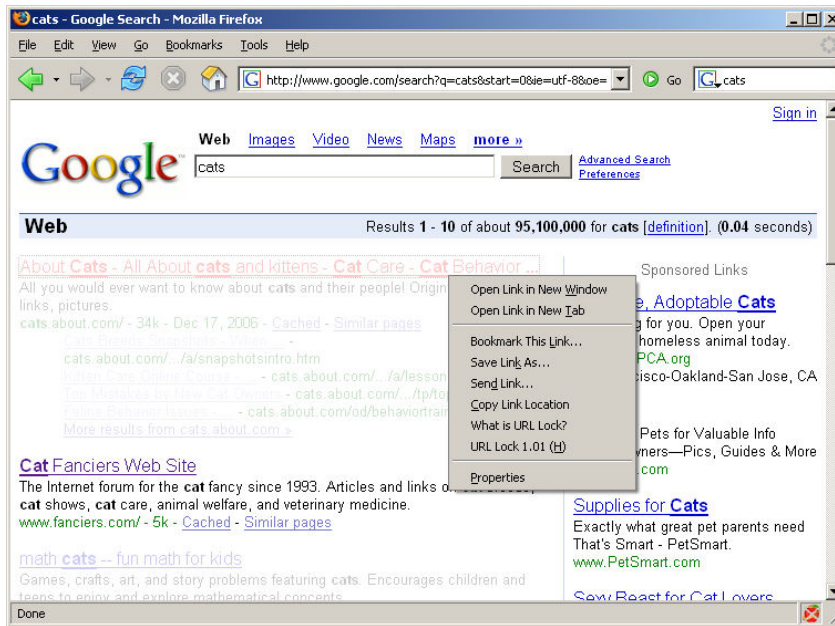


Figure 6: Search result disabling in Google

## URL Blocking

This applies to when a user is trying to access a blocked webpage. The browser will redirect the user to a warning page, as showed in Figure 7.

Within our extension's chrome/content folder, linkDisable.js contains a window load listener. It is called when a new browser window is about to be displayed. Once a new Firefox window is created, our extension calls the loadWindow method to add a progress listener to the current browser. The onLocationChange method was implemented so that it can compare the current URL link with the defined rules and, if blocked, redirect the browser to the warning HTML page.

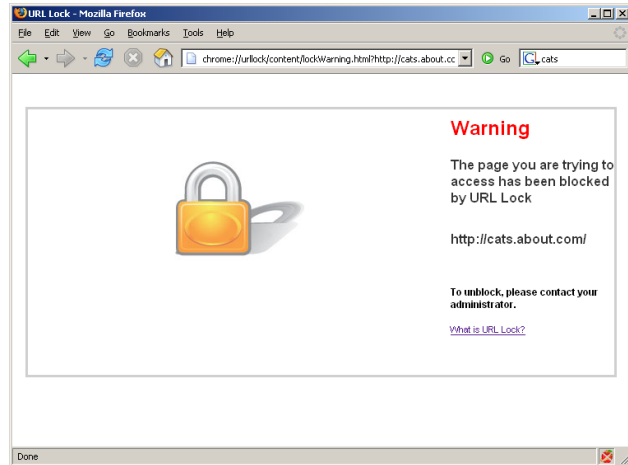


Figure 7: Blocked web site

### Library

The library was implemented in XPCOM using JavaScript for all interfaces except for LockNative, which was written in C++ using Microsoft Visual C++ 2005. An interface relationship diagram of these interfaces is shown in Figure 8 below. The user interface classes included a small JavaScript file that fetched the singleton lockEngine instance using getService().

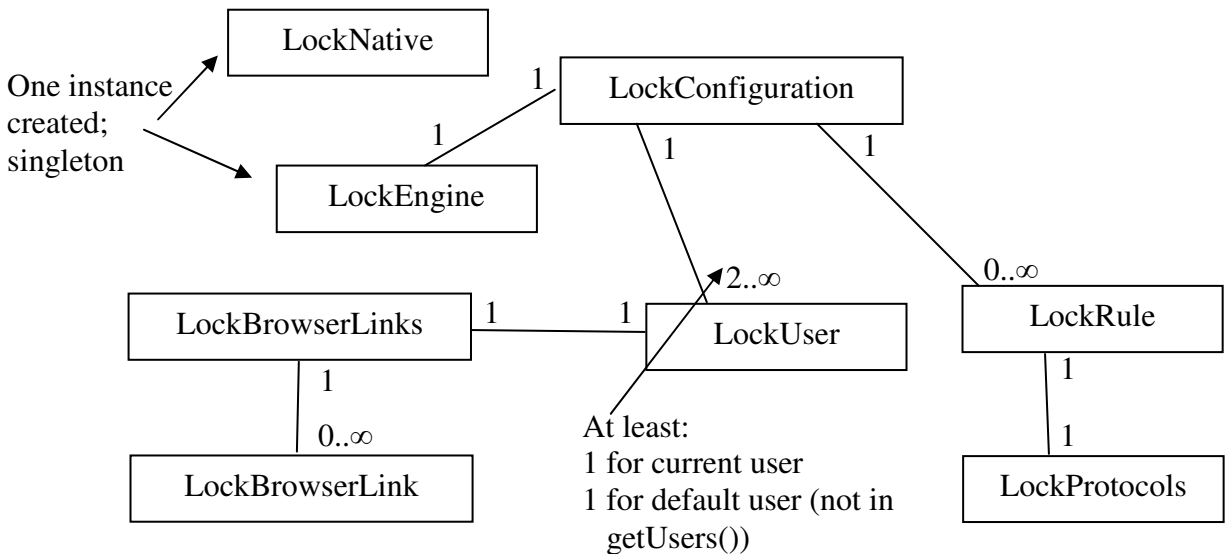


Figure 8: Library interface relationships

The library keeps track of changes to individual user and rule objects, providing methods to query whether or not any changes exist, saving the changes, or reverting the changes. Event notification takes place when properties are changed, permitting the GUI to respond to changes to the data without having to know who is changing what data. As an



example, this simplified how adding a rule from the rule addition screen automatically updated the rule list in the main configuration editor window. Those two screens do not know anything about each other, aside from the main configuration editor window knowing which chrome URL to pass to `window.open()` to launch the rule addition screen. The library propagates a `fieldChanged` event whenever properties change, and the main configuration editor window registers as a listener for these events. This promotes future extensibility while minimizing code complexity, which can be good quality attributes to have in a software architecture for user interfaces.

The team's software development process enabled parallel development between the library and the user interface by initially implementing the library as stubs that the user interface could call. This "skeleton" was then implemented over time to include functionality such as a loading the list of users from Windows. Not all functionality was implemented due to time constraints and strategic decisions, though they can be implemented in the future.

Due to a thought-out design, very few changes were required in the library during the project. Some details related to XPCOM did lead to some minor changes with how arrays were passed, but overall, the interface remained the same. XPCOM's learning curve contributed highly to the delays in getting the library completed. As a result, some functions are missing such as registry persistence and regular expression generation.

## **Limitations**

The present version of URL Lock has some limitations. The team originally wanted to implement the full set of features, but the original set was a bit too ambitious. As with many software projects, the scope had to be revised as deadlines approached. This section summarizes the limitations in this version of URL Lock

1. Double-clicking on a location in the history or bookmarks window from the rule addition screen browses to that web site instead of selecting that location for the new rule
2. Rules are not persisted to the registry nor loaded from the registry. The functions that do this, however, do exist as stubs
3. The Internet Explorer integration for the new configuration user interface does not exist
4. The advanced options window exists but does not do anything
5. Substring matching is used in blocking decisions instead of regular expressions. This is partly due to the absence of the regular expression generator, which was designed, but not implemented

6. Making changes and then moving to another rule or closing the window does not ask the user if they are sure that they want to lose their changes. The library supports this through `isChanged()` functions on the objects, but the current GUI does not make use of them
7. The system-wide “default user” is not implemented in the library, which is okay as it is not modeled directly in the new user interface
8. Regular expression generation from rule properties as well as regular expression parsing into rule properties are both not implemented
9. The Internet Explorer bookmarks and histories do not appear in the bookmarks and history screens
10. The bookmarks and histories for other users in the system do not appear in the bookmarks and history screens
11. Pressing the Back button after navigating to a blocked page brings the browser back to the block page. As a workaround, go back two pages. This did not happen in the pre-existing IE URL Lock, but happens in the current URL Lock on Firefox

## Future

As this project extended an existing open-source project, the code from this project will be improved, completed, and integrated into the main IE URL Lock extension. The new user interface is a vast improvement over the original one created in Microsoft’s Group Policy Editor, and the second iteration on the team’s user interface is an improvement from the initial configuration user interface design iteration. The architecture kept the code segments logically separated and logically organized, permitting its eventual inclusion into a production open-source product.

## Usage

To install the XPI file, drag it into Firefox and confirm the installation. After restarting Firefox, the URL Lock should work as described in this document. Try to add a rule such as <http://cats.about.com> and, after doing so, search Google for “cats”. Also try to navigate directly to <http://cats.about.com> to see the access denial page.