HTTP Authentication

1 Introduction

Web servers, like Apache, usually have three distinct ways of dealing with the question of whether a particular request for a resource will result in that resource actually be returned. These criteria are called *Authorization*, *Authentication*, and *Access control*.

Authentication is any process by which you verify that someone is who they claim they are. This usually involves a username and a password, but can include any other method of demonstrating identity, such as a smart card, retina scan, voice recognition, or fingerprints. Authentication is equivalent to showing your drivers license at the ticket counter at the airport.

Authorization is finding out if the person, once identified, is permitted to have the resource. This is usually determined by finding out if that person is a part of a particular group, if that person has paid admission, or has a particular level of security clearance. Authorization is equivalent to checking the guest list at an exclusive party, or checking for your ticket when you go to the opera.

Finally, *access control* is a much more general way of talking about controlling access to a web resource. Access can be granted or denied based on a wide variety of criteria, such as the network address of the client, the time of day, the phase of the moon, or the browser which the visitor is using. Access control is analogous to locking the gate at closing time, or only letting people onto the ride who are more than 48 inches tall - it's controlling entrance by some arbitrary condition which may or may not have anything to do with the attributes of the particular visitor.

Because these three techniques are so closely related in most real applications, it is difficult to talk about them separate from one another. In particular, authentication and authorization are, in most actual implementations, inextricable.

If you have information on your web site that is sensitive, or intended for only a small group of people, the techniques in this tutorial will help you make sure that the people that see those pages are the people that you wanted to see them.

2 Basic authentication

As the name implies, basic authentication is the simplest method of authentication, and for a long time was the most common authentication method used. However, other methods of authentication have recently passed basic in common usage, due to usability issues that will be discussed in a minute.

2.1 How basic authentication works

When a particular resource has been protected using basic authentication, the web server sends a *401 Authentication Required* header with the response to the request, in order to notify the client that user credentials must be supplied in order for the resource to be returned as requested.

Upon receiving a 401 response header, the client's browser, if it supports basic authentication, will ask the user to supply a username and password to be sent to the server.

```
WWW-Authenticate: Basic realm="By Invitation Only"
```

If you are using a graphical browser, such as Netscape or Internet Explorer, what you will see is a box which pops up and gives you a place to type in your username and password. These information is then base64-encoded and put in an Authentication header, which is used in the following-up requests to tell the server.

```
Authorization: Basic QWRtaW46Zm9vYmFy
```

The server base64-decodes the credentials and compares them against his username-password database. If the username is in the approved list, and if the password supplied is correct, the resource will be returned to the client.

Because the HTTP protocol is stateless, each request will be treated in the same way, even though they are from the same client. That is, every resource which is requested from the server will have to supply authentication credentials over again in order to receive the resource.

Fortunately, the browser takes care of the details here, so that you only have to type in your username and password one time per browser session - that is, you might have to type it in again the next time you open up your browser and visit the same web site.

Along with the 401 response, certain other information will be passed back to the client. In particular, it sends a name which is associated with the protected area of the web site. This is called the *realm*, or just the authentication name. The client browser caches the username and password that you supplied, and stores it along with the authentication realm, so that if

other resources are requested from the same realm, the same username and password can be returned to authenticate that request without requiring the user to type them in again. This caching is usually just for the current browser session, but some browsers allow you to store them permanently, so that you never have to type in your password again.

The authentication name, or realm, will appear in the pop-up box, in order to identify what the username and password are being requested for.

2.2 Apache configuration

For example, Apache servers may use the following directives to allow only users named *rbowen* and *sungo* to access the resources in the realm of "*By Invitation Only*"

```
AuthType Basic
AuthName "By Invitation Only"
AuthUserFile /usr/local/apache/passwd/passwords
Require user rbowen sungo
```

2.3 Security concerns

Basic authentication should not be considered secure for any particularly rigorous definition of secure.

Although the password is stored on the server in encrypted format, it is passed from the client to the server in plain text across the network. Anyone listening with any variety of packet sniffer will be able to read the username and password in the clear as it goes across.

Not only that, but remember that the username and password are passed with every request, not just when the user first types them in. So the packet sniffer need not be listening at a particularly strategic time, but just for long enough to see any single request come across the wire.

And, in addition to that, the content itself is also going across the network in the clear, and so if the web site contains sensitive information, the same packet sniffer would have access to that information as it went past, even if the username and password were not used to gain direct access to the web site.

Don't use basic authentication for anything that requires real security. It is a detriment for most users, since very few people will take the trouble, or have the necessary software and/or equipment, to find out passwords. However, if someone had a desire to get in, it would take very little for them to do so.

3 Digest authentication

Digest authentication provides an alternative method for protecting your web content, which is relatively more secure.

Using digest authentication, your password is never sent across the network in the clear, but is always transmitted as an MD5 digest of the user's password. In this way, the password cannot be determined by sniffing network traffic.

Due to time limitation, we won't discuss digest authentication in class. If you are interested in it, you may read RFC2617 for details.

4 Access control

Authentication by username and password is only part of the story. Frequently you want to let people in based on something other than who they are. Something such as where they are coming from. Restricting access based on something other than the identity of the user is generally referred to as *Access Control*.

Different web servers may have different ways to specify access control. Apache web servers use Order, Allow and Deny directives to let you allow and deny access based on the host name, or host address, of the machine requesting a document. For example:

```
Order Deny,Allow
Deny from all
Allow from hostname.example.com
```