Securing Web Applications via PKI

James A. Rome
Oak Ridge National Laboratory
P.O. Box 2008, Oak Ridge, TN 37831
E-mail: jar@ornl.gov

Submitted for
IAEA International Workshop on Information Security

October 25–27, 2000

This work was supported by the Mathematical, Information, and Computational Sciences Division of the U.S. Department of Energy

Introduction

Since 1996, the U.S. Department of Energy (DoE) has sponsored the Materials Microcharacterization Collaboratory (MMC) as a means of testing methods for doing "science at a distance." The MMC members are Oak Ridge National Laboratory (ORNL), Lawrence Berkeley National Laboratory (LBNL), Argonne National Laboratory (ANL), The University of Illinois at Champaign-Urbana (UIUC) and the National Institute for Standards and Technology (NIST). Distributed among these sites are dozens of electron microscopes and a handful of beamlines, all of which are used for materials science research. (See the MMC homepage at http://tpm.amc.anl.gov/mmc/.)

WHY RESEARCH AT A DISTANCE?

The ability to do research at a distance is motivated by convenience and economics. Travel is becoming expensive and unpleasant. Congress has required DoE to cut travel by 30% which adds to the urgency. But in fact, other than the requirement to transport the samples being examined to the instruments being used, there is no real reason for the scientist to accompany the sample. Indeed, because sophisticated scientific instruments have a tendency to break at times, the remote scientist is not inconvenienced by such events and can do other useful work while the instrument is fixed. An important motivator for a scientist to use new techniques and to overcome any learning curve is that his life will be made easier; remote operations can fulfill this requirement.

Remote operations are especially useful when instruments are left to monitor events at remote sites in possible hostile environments. And when a scientist visits a remote site he would like to remotely access the facilities at his home base. Security is essential in both circumstances.

THE CHALLENGES

In the MMC environment, we must support users in all major computing environments—PCs, Macs, and Unix. In addition, we require solutions that function through firewalls and VPNs so that site-specific security implementations can be observed. The MMC must support encryption of traffic as well as strong authorization and authentication.

The MMC security approach

PKI INFRASTRUCTURE

From the start of the MMC, we considered the above challenges and concluded that the best approach was to create our own public key infrastructure (PKI) and to integrate it into all of our applications. We were ready, but the PKI infrastructure was not; it is only in the last year that the tools to deploy and actually use PKI have become readily available.

We purchased Netscape Certificate Authority (CA) software and issued certificates to our members. After all, we trust ourselves (for our purposes) more than we do a commercial CA. However, when Windows NT Service Pack 4 came along, our CA software broke and we had to wait almost a year before Netscape issued CMS 4.01 which we are now using.

But issuing certificates was only part of the problem. Handling them was, and still is, awkward in the two most popular Web browsers — Internet Explorer (IE) and Netscape (NS). Only since IE 4.0 and Netscape 4.5 have the two browsers enabled reasonably complete and user-friendly handling of certificates.

BROWSER FLAWS

Browsers come preloaded with a list of "recognized" certificate authorities. Our MMC CA is not among them. If someone outside the MMC is presented with a certificate issued by the MMC CA, he gets an "Unable to find certificate authority" message.

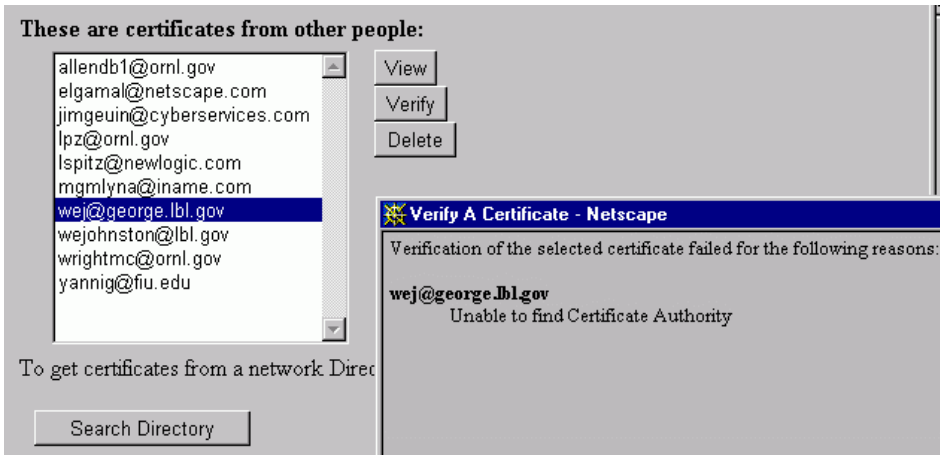

FIGURE 1. Unable to find certificate authority

The browser may or may not let you use a certificate (to encrypt mail, for example) from an unrecognized CA. The issue here is that there is no way to find out the location of the CA in question so you can go and get the CA certificate if you want to recognize that CA. Figure 1 also shows that other people's certificates are stored in a browser by their e-mail address. But what if I have certificates from different organizations that must be used for different purposes, and they all have the same e-mail address?

Storing your own certificates is not a lot better in IE. When you go to a Web site that requires a user certificate, the Web server tells the browser which certificates will be accepted, and the browser gives you a choice of them as shown in Fig. 2.

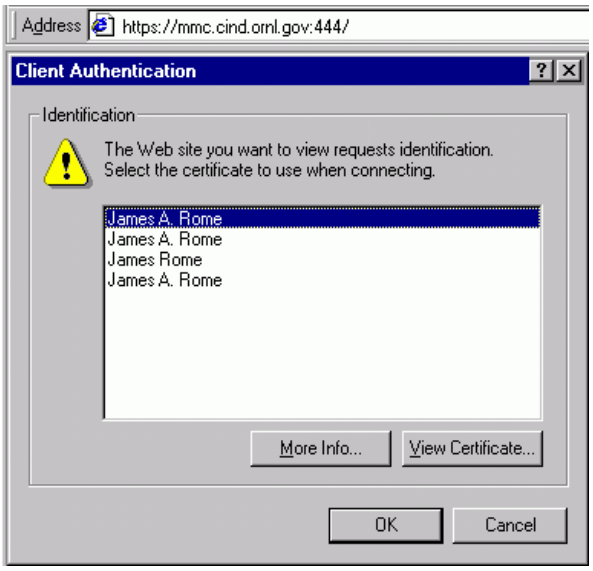

FIGURE 2. IE allows you to select your certificate from a useless list. NS allows users to give their certificates friendly names.

The user must manually view each certificate to see which one is correct.

Both IE and Netscape store the user certificates in a PKCS#12 certificate bag. One problem is that these databases can become corrupted over time, and the user becomes unable to delete some certificates (to replace them with new versions). Netscape has command line tools for managing the certificate databases; they allow you to add certificates, but not to remove them.

In a high-security environment, a user's certificate can be stored within a hardware token and protected by an access code. For example, the Fortezza card (containing the much-maligned Clipper chip) is supported by the Netscape browser.

SERVER PROBLEMS

Web servers are still not up to snuff when it comes to handling user certificates. Microsoft's Internet Information Server (IIS) is particularly bad it this, even in Version 5. The first problem is that Microsoft has seen fit to merge the security contexts of IIS and IE. The easiest, fastest way to set up a secure Web site is to issue a server certificate and to have the server require user certificates, and to have it only accept those that you issue. In IIS, the list of CAs that the IIS will accept is stored in IE! And it is quite unstraightforward to add a new CA. The user must manually choose the correct certificate store for this to work, and I have found no guidance on this from Microsoft.

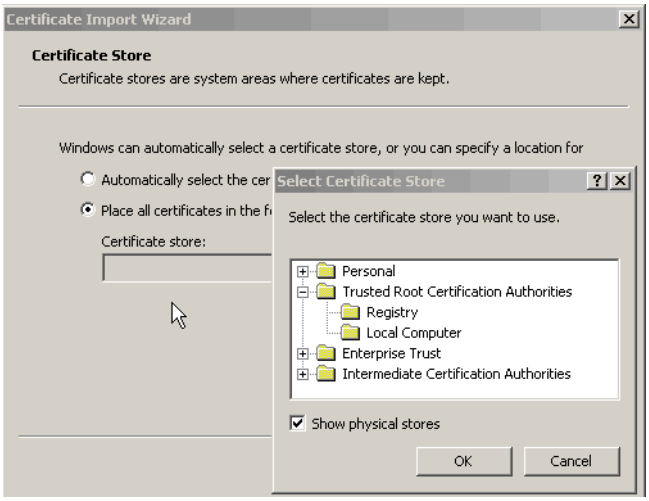

FIGURE 3. The user must manually select the correct certificate store in IE.

After you add your CA, you must manually untrust all of the other CAs so that a user with a certificate from another CA cannot gain access to your Web site.

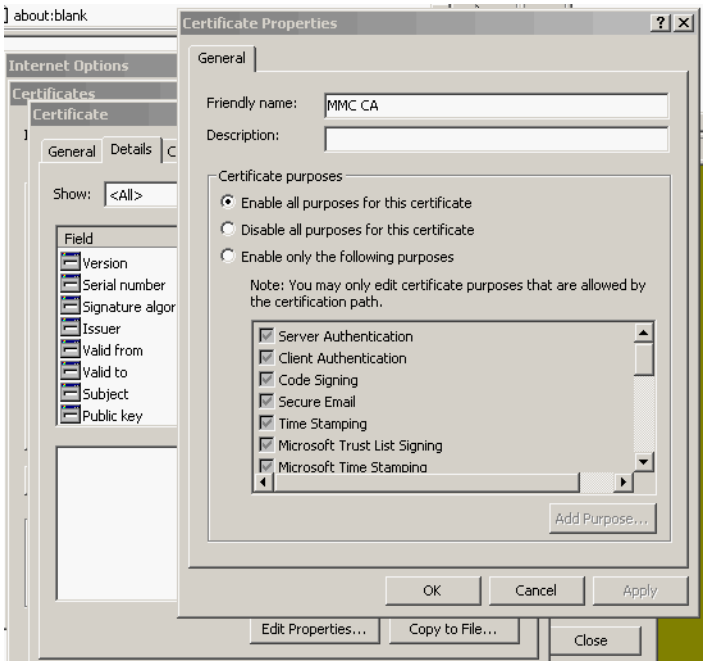

FIGURE 4. It is a tedious task to trust or untrust a CA in IE5.

But, if you succeed in untrusting all the other CAs, you will not be unable to securely install signed security patches from Microsoft. I believe it is a very bad decision to merge the security contexts of the browser and the server for this reason.

Servers also differ on how they handle user certificates, and in particular on whether they allow a server process to access the certificate that was presented by the user. We will discuss this in a later section.

SECURITY CONTEXT

Users store their certificates in their Web browsers. Is it possible (easy?) for them to use their certificates outside of the context of their browser? In order to extract the user's certificate (on the client), the user must provide the location of his certificate store together with the password for unlocking this store. From a security viewpoint, this is a rather risky thing for the user to do. If it is not necessary for the user to sign things, it is possible to obtain the user certificates from an LDAP server or on Web servers when they are presented by the user.

If a user initiates a session via an SSL-Web page, and presents a user certificate, it is useful and necessary to be able to track the user, i.e., to do some kind of session control. One reason we need session control is to be able to implement role-based access control; each user should be presented with only the options he is allowed to use. Thus, students might be allowed to focus and move the stage of an electron microscope, but not, for example to change the high voltage. A researcher needs to do this and more. The usual methods for session control do not work given our broad spectrum of users. Many MMC users will not allow cookies on their browsers, which prevents their use for this purpose. URL rewriting is also hard to apply because it would be necessary to rewrite too much of the software, and rewritten URLs can be spoofed.

SECURITY REQUIREMENTS

Our original plan of a global security architecture fell apart due to the inhomogeniety of our collaboratory. It seemed clear that many different solutions must be applied. Here is a short summary of the MMC requirements:

- Certificate-based user authentication and session control
- Role-based access control
- Ability to run on any client platform
- Ability to run on any server platform
- Encryption when desired
- Penetrate firewalls
- Auditing of resource use
- Ability to use legacy software and hardware

MMC SOLUTION

We have implemented a Java-based servlet solution that seems to solve most of these problems. By moving the security apparatus from the client to the server, we divorce ourselves from the user's environment. We only require that the user has a Web browser and an MMC user certificate. Such a solution was only made possible in the last year when the Servlet 2.2 API was released together with the Sun reference Java Security Services implementation that enabled the handling of certificates.

Users are required to use their MMC certificate to access secured MMC Web resources. This requires that the user enter a single password to unlock his local certificate bag, which is less work than if a UID and password were used. If the Netscape or Sun Web servers are used, a servlet can actually access the user's certificate in its entirety. Apache and IIS extract information from the certificate and place it in environment variables. Unfortunately, the user's public key is conspicuously missing. In addition, servlet engines typically do not list these extra header names in the *HttpServletRequest.getHeaderNames()* Enumeration.

Nonetheless, this suffices for most purposes. We intercept every access to *html*, *pl* (perl), and *jsp* (Java server page) URLs, check the user access, and log the user's name to a file. We extract the user's distinguished name from the presented, so we can implement any security policy that involves the fields contained in it. In particular, the MMC has chosen to use the ST (State) field to represent the user's status (role).

FIGURE 5. The author's MMC certificate as displayed by Netscape. His role is Administrator.

The roles used by the MMC are *Guest*, *Student*, *Researcher*, *Operator*, *Administrator*, and *Server*. (*Server* is used to identify MMC Web Servers.)

We use the Allaire JRun 3.0 servlet engine and the Netscape, Apache (with MOD-SSL extensions), or IIS Web servers. Because the user certificate is presented automatically on every access, we can track the session on the server and by extracting the user's role, we can present responses that are customized to prevent unauthorized actions.

AN EXAMPLE

As an example of this infrastructure, we present a scheme that allows a remote user to create a secure electronic lab notebook (written in perl by Al Geist of ORNL), to specify and maintain an access control policy, and to ensure that only authorized users can actually access the notebook.

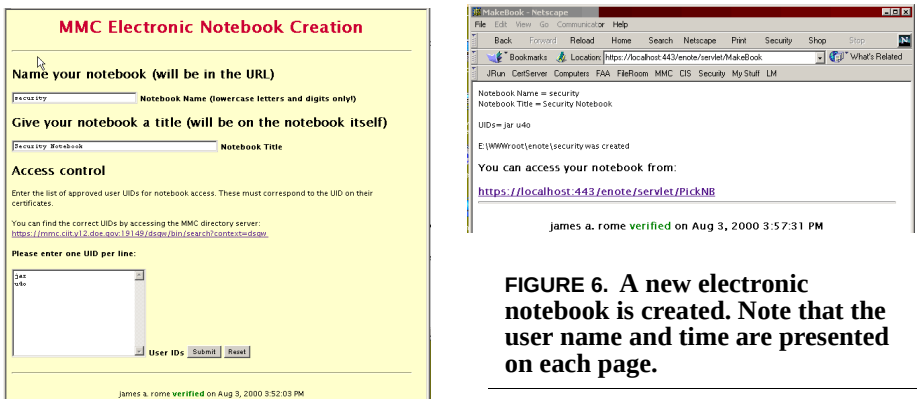

FIGURE 6. A new electronic notebook is created. Note that the user name and time are presented on each page.

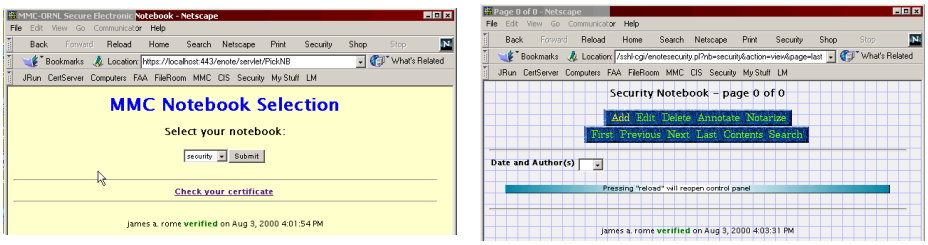

FIGURE 7. The new notebook appears on the selection list and can only be accessed only by users with UIDs *jar* or *u4o*.

New pages that are created in the notebook are all stamped with the user's name and time of upload, and even uploaded images have this information appended.

To implement more complicated security policies, we retrieve all the published certificates from the LDAP server, extract all the current values for each DN field, and present this information to the user on a dynamic Java server page. We use the Netscape LDAP software development kit to query the LDAP from the Web server.

The author runs a secure Lab Notebook on his Windows 2000 notebook computer. The files could be encrypted using the NTFS encryption features. The requirement for certificate presentation and an HTTPS connection to IIS5 prevent access to the notebook by unauthorized users when it is connected to the Internet. This computer is also protected by running ZoneLab's ZoneAlarmPro personal firewall.

In general, for small PKI infrastructures it is unnecessary to use an LDAP, and it is a nuisance to maintain the LDAP because certificate publishing is not as automatic as is advertised. Certificates will not publish unless the user's entry (with correct matching UID, Common Name, and possibly e-mail) already exists in the LDAP server when the certificate is issued by the CA. With the Netscape LDAP, these entries must be done manually. In the MMC we are not worried about certificate revocation lists. It is unlikely that a scientist suddenly becomes a criminal. However, we do issue short term

certificates to occasional users that expire a few days after their scheduled session. Expiration dates are checked as part of the certificate validation process.

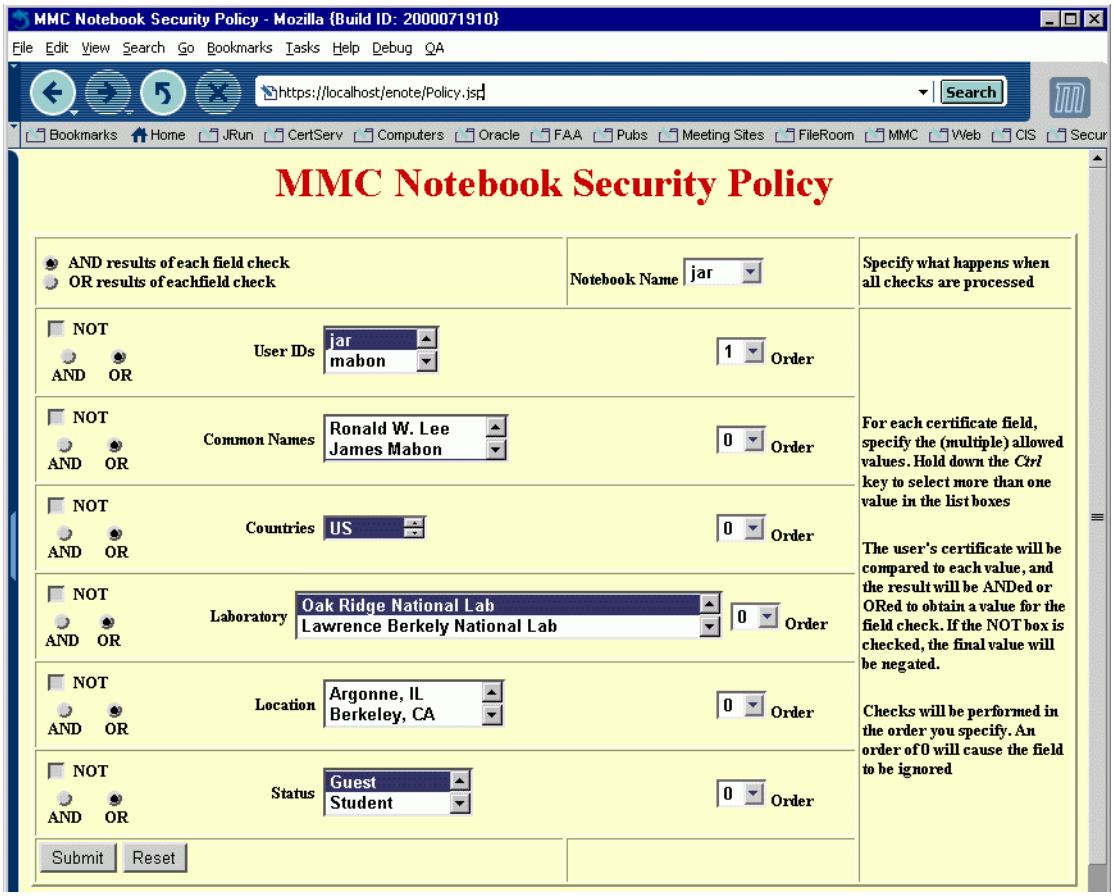

FIGURE 8. This policy definition Java Server Page gets the list box information by querying the LDAP server.

OTHER APPLICATIONS

The MMC microscopes can be rudimentarily controlled over a Web interface that can be secured in the same manner as the Lab notebook example. Other Web-based applications are also straightforward.

However, for more serious remote sessions, we employ a CORBA-based java application called DeepView [1] that acts as a middleman between the control functions of the different instruments and the user GUI. Originally, we planned to create a common user interface to lower the learning curve, but the microscopists complained that this would prevent access to the important and unique features of each instrument. Accordingly, we now create a unique user interface for each instrument. We are currently converting the

user interface to a (signed) applet that will allow us to extract the user's certificate from his Web browser so that we can use it in the CORBA security infrastructure.

Conclusions

It is very challenging to implement a PKI-based security infrastructure in a distributed heterogeneous environment. The solution implemented by the MMC collaboratory moves as much of this infrastructure as possible to the server. It is based upon Java for cross-platform compatibility, and makes heavy use of the servlet architecture.

Acknowledgements

The author wishes to thank Bahram Parvin and John R. Taylor (LBNL), and Ronald W. Lee, David Walker, and Michael C. Wright (ORNL) for their assistance.

References

[1]. Bahram Parvin and John R. Taylor, LBNL (http://vision.lbl.gov).