

Restricting Access with Certificate Attributes in Multiple Root Environments – A Recipe for Certificate Masquerading

Capt James M. Hayes, USAF
Systems and Network Attack Center
National Security Agency
Suite 6704 – 9800 Savage Road
Fort George G. Meade, Maryland 20755-6704
jimh@thematrix.ncsc.mil

Abstract

The issue of certificate masquerading against the SSL protocol is pointed out in [4]. In [4], various forms of server certificate masquerading are identified. It should also be noted that the attack described is a man-in-the-middle (MITM) attack that requires direct manipulation of the SSL protocol. This paper is a mirror of [4] and involves client certificate masquerading. The motivation for this paper comes from the fact that this anomaly has shown up in commercial products. It is potentially more damaging than [4] since a MITM attack is not involved and the only requirement is that the application trust a given root certificate authority (CA). The problem arises when applications use multiple roots that do not cross-certify. The problem is further exasperated since the applications themselves do not have the ability to apply external name constraints and policies. Unfortunately, the problem is a fairly well known problem within the public key infrastructure (PKI) community, but continues to persist in practice despite this knowledge.

1. Introduction

PKI has been established as one of the major buzzwords for Internet, extranet, and intranet security. Although PKI shows much promise, it is not without its own subtle misgivings. Ford and Baum state "...the certificate user may hold multiple root public keys and may make decisions that one root key is trusted for some purpose and another root key is trusted for another purpose. (In saying the root key is trusted, we actually mean that all certification paths starting from that root key are trusted.)" [2]

In the past couple of years, some product manufactures have attempted to improve their products by including features that would allow a user to define the purpose of a

CA in terms of how certificates issued by a CA can be used, e.g., a specific type of application or protocol. For example, some products will allow an administrator to determine if a CA certificate can be used to validate certificates that are used for server authentication, client authentication, code signing, or secure e-mail; however, excluding these products, one would be hard pressed to find a product that would allow a user or an organization to enforce a policy such that an application could only accept certificates from a given CA when the certificates conform to a particular usage and/or name constraint. Generally, CAs are either trusted or not trusted for all purposes, regardless of the intended purpose.

The remaining sections of this paper will explain the details of attribute-based client masquerading. Section 2 will give a definition of PKI and Section 3 will define certificate masquerading. Section 4 will discuss trust implications of multiple root environments. Sections 5, 6, and 7 will illustrate the problem. Section 8 will present an analysis of the problem. Lastly, Section 9 will present possible solutions. When reading this paper, keep in mind that based on [3]'s 2001 annual "Computer Crime and Security Survey", 34 of the 538 respondents stated that they suffered a \$151,230,100 loss in proprietary information and 21 respondents stated they suffered a \$92,935,500 loss to financial fraud.

2. PKI Defined

[1] defines PKI as the following:

The set of hardware, software, people, policies and procedures needed to create, manage, store, distribute, and revoke public key certificates based on public-key cryptography.

The overall goal of a PKI is to ensure that a certificate that is used in any given transaction is valid at the instance that a transaction is executed. In order to assure that this is the case, the PKI will incorporate various functions such as registration, initialization, certification, key pair recovery, key generation, key update, key expiry, key compromise, cross-certification, revocation, and certificate and revocation notice distribution and publication. [1] If a root makes a decision to imitate a peer root, severe consequences can result, [4] i.e., it can then undermine the trustworthiness of its peers.

3. Certificate Masquerading Defined

Certificate masquerading, as based on [4], allows a masquerader to substitute an unsuspecting certificate holder's *valid* certificate with the masquerader's *valid* certificate. A *valid* certificate is defined as a certificate that has been signed by a CA that a relying party (user, computer, service, etc.) is willing to accept without challenge, i.e., the certificate has a CA trust point, the certificate's signature can be verified, the certificate is not revoked and the certification path can be constructed and is valid. [6] In addition, it's important to note that masquerading can take place where invalid certificates are used as well, i.e., no CA trust point.

4. Trust Implications in Multi-Root Environments

Whenever an organization trusts an external organization's CA without cross certifying, it creates by default an unbounded cross-certification with that organization. Often, this is done for convenience or expedience. What is the result? The issue of trust management is not resolved. [6] Trust decisions in terms of types of certificates that will be accepted, e.g., client, web server, timestamp, etc., are not addressed. Also, limits on certificate names, path lengths and other PKI policy issues are not addressed. If two organizations attempted to enter a formal trust relationship, these issues would need to be addressed in a formal cross-certification between the organizations. Unfortunately, these issues are often not addressed in informal relationships. An example of this is when one trusts the default CAs that come with a web browser or web server. However, some products are beginning to address this issue with respect to policy and allowing administrators to configure policies to limit the types of certificates accepted from a CA, i.e., only client certificates from a given CA will be accepted, but code signing certificates will not. This capability is yet to be extended to include path length constraints or name constraints both of which are needed even in the case of an informal relationship.

5. A Case Study of the BIMM Corporation, PPC and SRPC

So let's start off looking at the relationship between the Bureaucratic Institution for Mismanagement (BIMM) Corporation, the Popular Products Corporation (PPC), and the Second Rate Products Corporation (SRPC). In this illustration, the author has enlisted the service of Bob, Mallory, Trent, and Victor to demonstrate the problem at hand. Mallory will play the role of a network penetration tester and CA administrator who works for SRPC. Trent will play the role of an administrator of the CA for PPC. The BIMM Corporation has a web application that was created by Victor and used by BIMM, PPC, and SRPC. The web application uses SSL 3.0 and requires client authentication. Client certificates are issued by BIMM, PPC and SRPC CAs.

For many years, the BIMM Corporation has provided marketing data to PPC and SRPC. SRPC is having difficulties competing with PPC. SRPC management approaches Mallory and offers him a substantial amount of money if he can get any restricted information about PPC from the BIMM web application.

6. Mallory's Reconnaissance

Mallory and Victor often had conversations regarding security. One day Mallory asked Victor about how the web application was actually protected. Victor told Mallory the following:

1. It uses SSL 3.0 with client authentication.
2. The web application uses four trusted root certificates: BIMM, PPC, SRPC and Ultra Trust. Ultra Trust is a commercial CA.
3. The web application uses certificate mapping rules, as shown in Table 1, to determine which user accounts should be mapped to a given certificate. The mapping rule that is used is determined by the certificate's issuer distinguished name.
4. Since BIMM certificates are stored in a directory, BIMM client certificates are compared to certificates in the directory; however, PPC, SRPC, and Ultra Trust certificates are not stored in the directory because BIMM does not wish to allow other CAs to publish directly to their directory. In addition, collecting certificates from the other CAs would not allow the web application to be robust, i.e., BIMM would have to collect all user certificates and publish them to the directory before access could be granted to a legitimate user. This could cause unacceptable delay to PPC and

SRPC clients as well as additional overhead for BIMM.

5. It uses a certificate's subject distinguished name (SDN) to determine branch points in the directory and then applies the search criteria for a specific entry in the directory. It only allows certificates signed by PPC to map to PPC accounts and groups and certificates signed by SRPC to map SRPC accounts and groups, or so Victor thought.

After performing this basic reconnaissance, Mallory decided to test the mapping capability of the web application.

Table 1. Certificate Mapping Rules

Mapping Rules
CAName: default
SDN: *
Search: e
CAName: CN=BIMM CA, O=BIMM
SDN:OU, O
Search: e, uid
CAName: CN= SRPC CA, O=SRPC
SDN: OU, O
Search: uid
CAName: CN=PPC CA, O=PPC
SDN: OU, O
Search: e

7. Test of the BIMM Certificate Mapping

Mallory knows that the web application has several options that are restricted to PPC users. So Mallory targets the PPC's vice president of research and development (R&D). He creates a certificate that looks like the vice president of R&D's certificate and retrieves the requested information for SRPC management. How was Mallory able to complete this feat?

Victor's CA product is a product that did not support cross-certification. Without this ability to cross-certify with the PPC or SRPC CAs, Mallory realized that he could create a subordinate CA that looked like PPC's CA.

Now that Mallory had a look-alike PPC CA, he decided to create a certificate of PPC's vice president of R&D. Mallory had a copy of the vice president's authentic certificate and decided to copy the attributes into the fraudulent certificate. The only difference between

Mallory's certificate and the original were the public key and authority key identifier; however, it's important to note that because of the default mapping, Mallory could have created a SRPC certificate with SDN attributes that reflected Bob's certificate and still gained access. The certificate appears below in Figure 1.

Masquerading SRPC User Certificate
Bob
bob@rd.ppc.com
Vice President Research and Development
Research and Development
PPC
SRPC CA
SRPC

Figure 1. Bob's SRPC certificate

Mallory decides to visit his favorite café, The Black Hatters Internet Café. He accesses the web application in question and when prompted, selected the fraudulent certificate as his authentication certificate. The certificate chain was sent to the web application and Mallory was granted access as Bob. The complete chain is shown in Figure 2.

Root SRPC Certificate
SRPC CA
SRPC
SRPC CA
SRPC

Masquerading PPC CA Certificate
PPC CA
PPC
SRPC CA
SRPC

Masquerading User Certificate
Bob
bob@rd.ppc.com
Vice President Research and Development
Research and Development
PPC
PPC CA
PPC

Figure 2. Mallory's fake certificate chain

8. Trent's Analysis of Mallory's Attack

Several months later, PPC noticed that SRPC had offered a new product line that was similar to PPC prototypes. BIMM initially thought Bob must have divulged sensitive information since the web application log files showed that he had made several accesses from unscrupulous locations. Unfortunately, Victor's log files were very similar to the Common Logfile Format and therefore the only identity data collected was IPs and usernames—no certificate information. Trent was sent to the BIMM Corporation to investigate. Trent visited Victor and explained the situation. Victor went on to explain the design of the web application.

After listening to Victor, Trent told Victor how he believes Mallory may have compromised Victor's security measures. The basic problem uncovered by Trent was the following:

1. The path validation module checked to see if the client certificate chained to a trusted root. None of the root CAs had imposed constraints on what names could appear in their respective certificate paths.
2. The certificate mapping used by the web application to search the directory uses unrestricted values contained in the attributes of the SDN of client certificates.
3. Although the web application attempts to bind user accounts to certificates, its design is similar to that of binding attribute certificates to identity certificates whereby authentication is highly dependent upon loosely coupled attributes. [8]

Trent informed Victor that since there was no cross-certification by Victor's CA with PPC or SRPC, then both were able to imitate one another because of the lack of name constraints. Trent demonstrated this capability by creating a fake subordinate SRPC CA.

9. Possible Solutions

In order to solve the problem, Trent and Victor knew that they had to bind certificates using a restricted naming convention for any given CA. There are three possibilities: cross-certification with name constraints, attribute certificates, or name constraints and other policies applied by the web application or policy engine.

9.1 Cross-certification and Name Constraints

Cross-certification is the process by which a CA issues a cross-certificate to another CA. Essentially, the cross-

certificate contains the public key of a CA, which is associated with the private key for that CA. The purpose of cross-certification is to allow users in one domain to communicate securely in another. For example, BIMM could issue a cross-certificate to PPC, thereby allowing Bob's certificate validation path to end with the BIMM CA certificate instead of the PPC CA certificate. [1]

"The X.509 name constraint model allows any certification authority to specify, when it certifies another certification authority, exactly what names are allowed in subsequent certificates in the certification path." [2] Name constraints can be applied to various attributes such as uniform resource indicators (URI), Internet mail addresses, domain name system (DNS) names, directory names, and Internet Protocol (IP) addresses. [5]

Victor believes that cross-certification and name constraints are a potential technical solution, but not a viable solution in his case because of the reasons listed in [9].

...the representatives of each CA organization sign legal documents that specify security policies in both domains, and define specific liabilities or limitations. Complexity and financial cost increase with greater numbers of trust relationships between CAs. The management and risk assessment become expensive, not to mention the associated legalese between the multiple organizations in establishing the cross-certification trust...Path processing mechanisms derive success or failure of a validation through domain-specific policy sets and critical key extensions; these are neither standardized nor widely interoperable between domains.

9.2 Binding Identity Certificates to Attribute Certificates

Victor had toyed around with the idea of using attribute certificates, but once again, cost was a factor and the BIMM Corporation did not want to manage another authority. In addition, a tight binding between the attribute certificate and the identity certificate would need to be established, e.g., hash of the public key or certificate. In order to get the value, Victor would have to collect certificates, and therefore this was not an acceptable solution. He decided to use the Internet mail address, organizational unit, and organization attributes as a binder between a certificate and a user account. He believed that he could constrain these attributes using the name constraint concepts in [5]. He decided not to use userids because it would make it difficult to enforce uniqueness across CAs.

9.3 Application Enforced Name Constraints and Policies – A Moderate Coupling

Victor modified his web application so that it enforces its own form of name constraints and path length constraints for each CA. He wanted to keep his solution simple, yet limit the scope of any type of client masquerading within the scope of a CA's own validation paths. If a given CA had concerns about intra-organizational masquerading, then it could apply its own name constraint in accordance with [5]. He decided to apply permittedSubtrees to organizational CAs and commercial CAs in the web application as well.

The permittedSubtrees he chose to apply were the Internet mail address and directoryName. [5] Using this simple definition, Victor constructed a constraint table as in Table 2.

Table 2. Permitted subtrees and constraints

Root CA	Permitted Internet Mail Subtrees	Permitted Directory Name Subtrees
PPC CA PPC	.ppc.com ppc.com	O=PPC
SRPC CA SRPC	.srpc.com srpc.com	O=SRPC
Ultra Trust Commercial CA Inc.	.bigcars.com bigcars.com walker@free-email.com	O=Big Cars O=Walker Inc.

The application name constraint policy basically states that each organizational CA can only have its organization name in certificates. Ultra Trust CA can have a limited set of organizational names (Walker Inc., Big Cars). Victor could have used excludedSubtrees for Ultra Trust so that PPC and SRPC do not appear in certificates, but any other name could. In addition, Internet mail addresses are limited to the respective domains (.ppc.com), host (ppc.com), or specific mailbox (walker@free-email.com).

Victor also included the ability to apply path length constraints. This ability was added so that he could limit chain lengths for CAs where he knew he did not need to accept subordinate CAs; however, for those root level CAs where any given user could have access to the application, restrictions on names was the overriding factor.

10. Conclusion

Many organizations are looking at the possibility of using certificates as authorization objects. When environments exist where there are multiple CAs involved, careful consideration should be given to answering the question "How should this CA be trusted?" If the answer is for a limited purpose(s), then the application should take into consideration that purpose(s). If traditional path validation cannot resolve the issue of how the CA will be limited to that purpose(s), then the application should address the restrictions required. This must be weighed against the risk. Adding any external policy will increase cost and overhead. It may be determined that the risk is cheaper than the solution.

References

- [1] Arsenault, A., Turner, S., *Internet X.509 Public Key Infrastructure*, draft-ietf-pkix-roadmap-06.txt, Internet Society, November 2000.
- [2] Ford, W., Baum, M.S., *Secure Electronic Commerce*, Prentice Hall PTR, Upper Saddle River, N.J., 1997.
- [3] *Financial losses due to Internet intrusions, trade secret theft and other cyber crimes soar*, http://www.gocsi.com/preleas_000321.htm, Computer Security Institute, March 12, 2001.
- [4] Hayes, J.M., The Problem with Multiple Roots in Web Browsers – Certificate Masquerading. In IEEE Computer Society *Proceedings of WETICE 1998*, 17-19 June 1998 at Palo Alto, California.
- [5] Housley, R., Ford, W., Polk, T., Solo, D., *Internet X.509 Public Key Infrastructure Certificate and CRL Profile*, RFC2459, Internet Society, January 1999.
- [6] Housley, R., Polk, T., *Planning for PKI*, John Wiley & Sons, Inc., New York, 2001.
- [7] Larson, E., Stephens, B., *Web Servers, Security, & Maintenance*, Prentice Hall PTR, Upper Saddle River, N.J., 2000.
- [8] Park, J.S., Sandhu, R., Binding Identities and Attributes Using Digitally Signed Certificates. In IEEE Computer Society *Proceedings of Computer Security Applications Conference*, 11-15 December 2000 at New Orleans Louisiana.
- [9] Prasad, V., Potakamuri, S., Ahern, M., Lerner, M., Balabine, I., Dutta, P., Scalable Policy Driven and General Purpose Public Key Infrastructure (PKI). In IEEE Computer Society *Proceedings of Computer Security Applications Conference*, 11-15 December 2000 at New Orleans Louisiana.