Global Information Assurance Certification Paper

Artem Kazantsev
GSEC Version 1.4b (option 1) submitted 05-17-2004

Web SSL Authentication Using Client X.509 Digital Certificates
(practical approach)

Abstract
I will discuss the benefits of using Client X.509 Digital Certificates for authentication to websites, compared to the traditional methods such as username and password combination, as well as the limitations of this method. We will demonstrate the practical aspects of implementing a basic, but inexpensive and effective way to use Client Digital Certificates with a website in a situation, when a limited number of users need to authenticate themselves, but physically are in remote, inaccessible to the administrator locations. As an illustration we will use Microsoft IIS 6.0 as a webserver and Mozilla 1.6 browser with Digital IDs from Thawte and Verisign. Other alternative setups, using Apache 2.0 and Microsoft IIS 5.0 as webservers, and other Certificate Authorities also will be discussed.

1. SSL and Internet standards
SSL (Secure Socket Layers) is a protocol that was developed by Netscape Communications in 1994-1996 and was widely adopted by the Internet community. In a nutshell, SSL uses public key cryptography for encryption and authentication of Internet communications. As the name implies, SSL intertwines with other layers (protocols), such as HTTP, IMAP4, POP3, SMTP and many others. The latest incarnation of SSL is called TLS (Transport Layer Security) and in version 1, as it is described in Request for Comments (RFC) 2246 [1], it became the de-facto Internet standard.

Let's briefly examine how SSL works. To understand the public key cryptography principles of SSL, we recommend Netscape Developer's original document [2], and in order to limit the size of this document, we will go straight to the practical implementation:

1) We have two users Alice and Bob; Bob obtains a certificate (supposedly, from a trusted Certificate Authority (CA)). Each certificate contains the following information:

- The certificate issuer's name
  # the Certificate Authority's name
- The entity for whom the certificate is being issued (aka the subject)
  # in our case, this will be Bob's distinguished, unique identifier
- The public key of the subject
  # generated by Bob during the certificate request
- Some time stamps
  # important to provide the expiration and synchronization functions

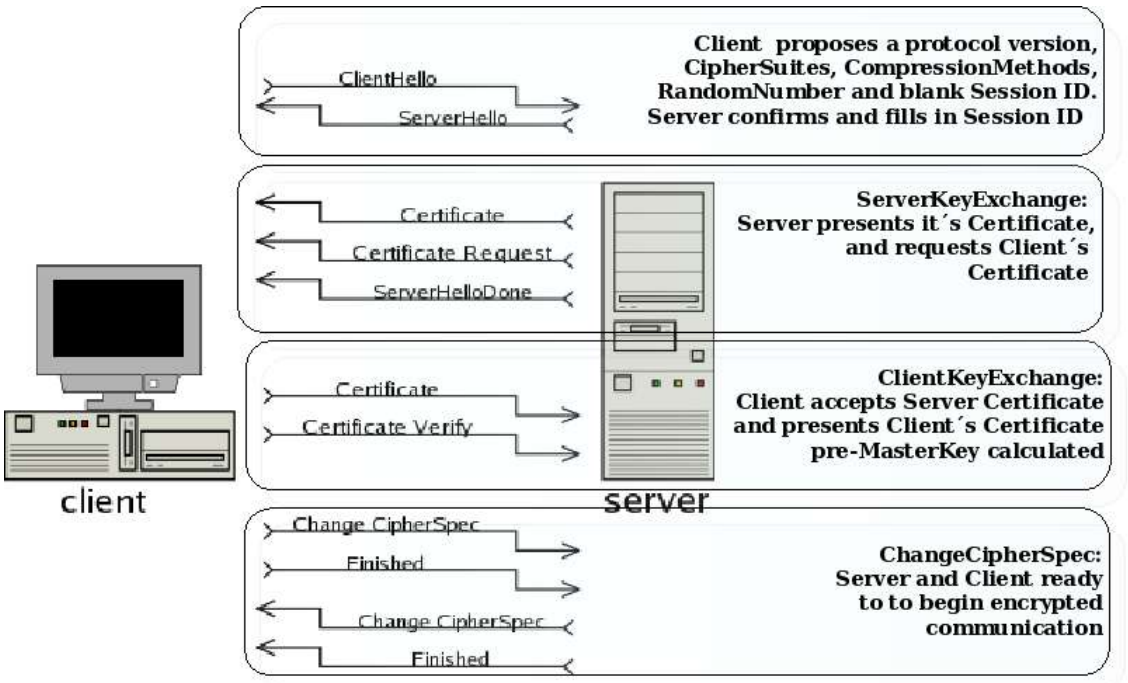
Also, the private key is generated together with the public key. Public key is represented in the certificate. Private key is never transported over the network.

2) Alice sends Bob '*hello Bob*' to initiate a handshake
3) Bob responds with '*hello Alice*' and sends his certificate
4) Alice examines Bob's certificate. She doesn't trust Bob yet, so she sends a '*challenge message*' asking Bob to verify his identity.
5) Bob creates '*another message*', calculates the *digest* of this message (to ensure the integrity of the message) and signs the *digest* with his private key. Finally, he sends the whole package to Alice.
6) Alice already has Bob's certificate, so she can compare '*another message*' against Bob's public key, and verify the integrity by computing the *digest* of '*another message*' herself. If Bob's message verification is successful, Alice proceeds with sending him a *secret* for the session. She encrypts the *secret* using Bob's public key.
7) Bob can decrypt the new arrival from Alice with his private key and derive the *secret.*
8) Now both parties, Alice and Bob, have a *secret* and can start a communication session, using encryption with *secret* being a symmetric key (because it's faster!). In order to prevent Man-In-The-Middle attack, they will also use Message Authentication Code (MAC), a function which uses a combination of *message digest* and *secret.* Since *secret* is not available to a potential eavesdropping party, it's very improbable for them to arrive at the right value of MAC. To secure it further, *secret* (essentially, a session key) could be recalculated frequently to prevent brute-force and replay Man-In-The-Middle attacks.

The algorithm 1)-8) reflects the authentication of only one party (Bob). If Bob represents a Server and Alice represents a Client, we arrive at a very common practice on Internet, when the authentication of Alice (Client) is not required. The example of this kind of communication would be e-commerce, where the identity of a client is not as important as the authenticity of the server, as long as the Client's credit card is valid.

Starting with the version 3, SSL protocol included the provision for a client authentication. That is where Client's Digital Certificates (also called Digital IDs) come into play.
Now with SSL v3, Bob can also request Alice to authenticate, if he chooses to. Alice would have to obtain from CA and present to Bob her certificate, and Bob could decide whether to accept the certificate from Alice or not.

The revised scenario, as it is described in the mod_ssl maual [3], is the following, Bob being the Server, Alice – the Client:



'Cipher Suite' consists of:

Key Exchange Method

(Asymmetric key)

RSA key exchange when certificates are used, and Diffie-Hellman key exchange for exchanging keys without certificates and without prior communication between client and server.

Key exchange provides *Authentication* and *Non-repudiation*

Cipher for Data Transfer

(Symmetric session key)

No encryption (*no Confidentiality -AK*)

Stream Ciphers

- RC4 with 40-bit keys
- RC4 with 128-bit keys

CBC Block Ciphers

- RC2 with 40 bit key

- DES with 40 bit key
- DES with 56 bit key
- Triple-DES with 168 bit key
- Idea (128 bit key)
- Fortezza (96 bit key)
- AES (256 bit key) [new standard, see RFC3268][4]

Message Digest for creating the Message Authentication Code

(MAC)

- No digest (Null choice) (*Null integrity -AK*)
- MD5, a 128-bit hash
- Secure Hash Algorithm (SHA-1), a 160-bit hash

Presenting Client Certificate according to the RFC2246 (7.3, 7.4.4 and 7.4.6) is optional.

2. To be...?

We demonstrated that the usage of Digital Certificates on the client side is optional according to the SSL standard. So why should we utilize them?

There are many advantages, let's discuss the most important ones.

1. *Issued by CA.* Unlike passwords, certificates issued by Certificate Authority. In turn, CA should take extra steps to ensure the identity of the Certificate requesting party. The whole SSL encryption / authentication model hinges on the trust relationship between specific parties: the CA and the Certificate presenter; and CA and Certificate accepter. Even if parties had no prior communications, but they both trust the CA, that issued the certificate, and as soon as the handshake with verification took place, accepter can start trusting the presenter and engage in communications.

2. *Stronger Authentication*. Compared to usual methods of authentication, such as username and password, Digital Certificates provide much higher level of authenticity confidence. Username / password combinations are notorious for being easy to sniff off the network, to steal or to brake by brute-force attack. Digital IDs are also compatible with plethora of secure devices, such as Smart Cards, Secure Tokens, i-Buttons and many others where private key generated and stored directly on the device. If a username / password combination constitute one-factor authentication (<something I know>), Digital IDs together with a secure device, such as Smart Card or a token, could easily provide

two-factor authentication(<something I have>+<something I know>). If private key resides on the Client's computer, it is still much harder to steal and use it, because most operating systems have means to protect it. Good Practices require a user to guard her private key regardless of OS protection.

3. *Easy to integrate*. All commonly used browsers, such as MS Internet Explorer, Netscape Navigator, Mozilla, Mac OS X Safari, Opera, Konqueror, and many others support x.509 Digital IDs.

4. *Access to many applications*. There are many ways to employ certificates: Email clients provide authentication and encryption for email; VPN software offers means to authenticate users and create secure tunnels. In addition certificates are used in document authentication, timestamps, file encryption and software code signing. Ideally, users should be able to accomplish as many tasks as possible with a single certificate. However, this is still far from reality.

3. ...Or not to be?

The drawbacks of using Client Certificates are few, but they might truly inhibit the implementation of PKI in one's company. Some limitations of Digital IDs are listed below:

1) Certificates are bound to their corresponding private keys and to hardware in general.

One cannot simply take a certificate with her. Unlike passwords, certificates are computer dependent. Even with Security Devices, such as Smart Cards, one has to have a special reader to be able to use that Smart Card with her Certificate.

2) Logistics.

The process of requesting a Digital ID from a Certificate Authority is elaborate, sometimes – tedious and lengthy, and often not intuitive. Requesting Digital IDs require a user to be familiar with many technical terms, be computer-savvy. The fact that this process often relies on obscure OS crypto-functions, that an average user doesn't deal with in her day to day operations, doesn't help either. Using Digital IDs require stricter discipline, and a general understanding of the whole process. The private key protection is a novel concept for users.

The bigger the number of users, the harder it becomes to implement client certificates in a swift and effective way.

With the wider acceptance of Digital Certificates as a standard mean of authentication, hopefully, the situation will change.

3) PKI management

The management of certificates also could be a problem, especially

without a centralized directory. As we will demonstrate further, web servers in their standard, most common configurations are not equipped with extensive PKI management options thus shifting the burden of managing Digital Certificates off to the site administrators. Other applications, such as email, VPN or remote access and many others also do not provide easy management solutions. That in turn translates into heavy workload for sysadmins.

That said, for small companies with a few number of employees, or sites with a centralized directory, Digital IDs are still very attractive means of authentication.

The other aspect of successful implementation of PKI in organization is the development of PKI policy. Even for a small organization with the closed infrastructure a clear and concise PKI policy "can decrease user errors and increase user awareness"[5].

4. Where to get a Client Certificate?

Suppose, we convinced the management that Digital Certificates are the way to go for our organization. Now, where to get them?

There are two choices: either issue them in house or get them from a CA. We would like to advocate the latter solution for the following reason: self-issued certificates will be trusted only within your organization, and unless it is CA itself, such as Verisign or Entrust, (in which case you should not be concerned with this document's topic in the first place), the benefits of portability and interoperability with other sites, organizations, even countries will not be there for you. The other argument against self-issued certificates is simple – if your time and resources are limited, why undertake a very complex and elaborate function of CA? You can leave it up to the companies, that are in this business, know this business, well-recognized in the industry and well-positioned to do the job! Better yet, some CAs provide Client Digital IDs for free.

Let's mention a few Certificate Authorities, that issue Digital IDs:

Certificate Authority	URL for Digital ID	Cost	notes
Thawte	http://www.thawte.com/email/index.html	free	+ very well established CA, reliable, CRL, SPKI, WOT (to get your name on a certificate) - long and tedious enrollment process

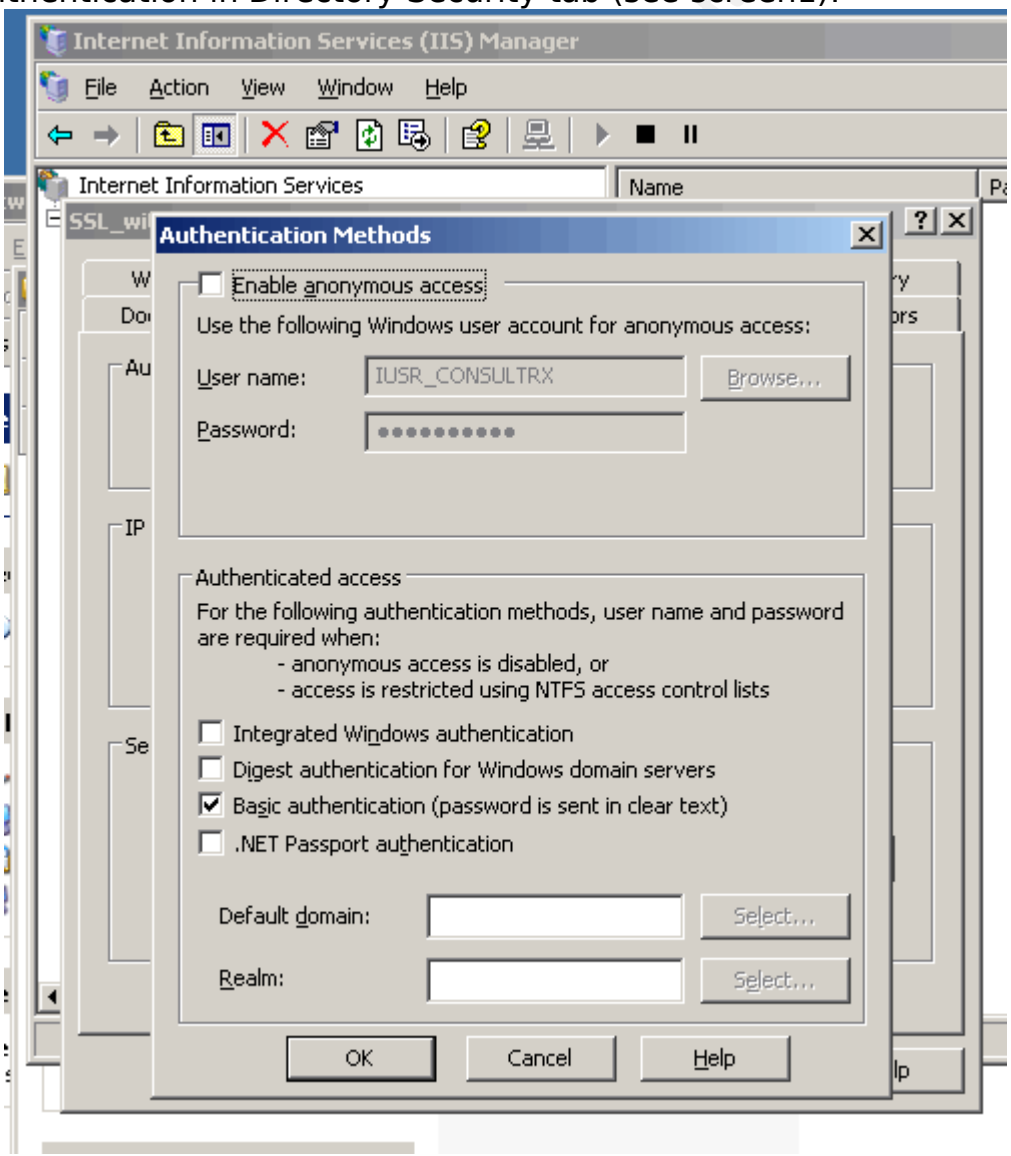
Certificate Authority	URL for Digital ID	Cost	notes
Verisign	http://www.verisign.com/products/class1/index.html	$14.95 /year	+ the most known and trusted CA, reliable, CRL, has free 60 days trial, easy to enroll - 1 year certificate is not free
CAcert	https://www.cacert.org/index.php?id=3	free	+ all certificates free, trust model similar to Thawte's, CRL - root cerificate is not distributed with browsers, new in business.
Swisssign	https://swisssign.net/cgi-bin/class1/request	free	+ free certificates, CRL - the root cerificate is not distributed with browsers.
Globalsign	http://www.globalsign.net/digital_certificate/personalsign/index.cfm	16 euro/ year	+ well established CA, CRL, website is easy to navigate, root certificate is distributed with many browsers - relatively expensive

Here are some recommendations regarding choosing vendors. If a vendor provides a trial version of a certificate (like Verisign does), it is very strongly recommended to install a trial version first. You will avoid a lot of problems, save time and money by testing waters first.. Some vendors call their certificates 'E-mail certificates' because primarily they are considered to be S/MIME Digital IDs and to be used for e-mail encryption and authentication. This should not confuse us, because most certificates can have a dual function, e-mail and web authentication, which is stated in RFC2459[6] (Internet X.509 Public Key Infrastructure), 4.2.1.3 (key usage) and 4.2.1.13 (Extended key usage field). The only known vendor who sets the restrictions to the 'extended key usage field' is Comodo Group http://www.instantssl.com/ssl-certificate-products/free-email-certificate.html . An Email certificate from www.comodogroup.net (the other name of their website) will not provide the Client web authentication.
Another important aspect is insurance. If our organization is dealing with financial or other sensitive data, it is wise to select a vendor who backs up their certificates with a hefty insurance. For our peace of mind we would pay hefty premium, of course.

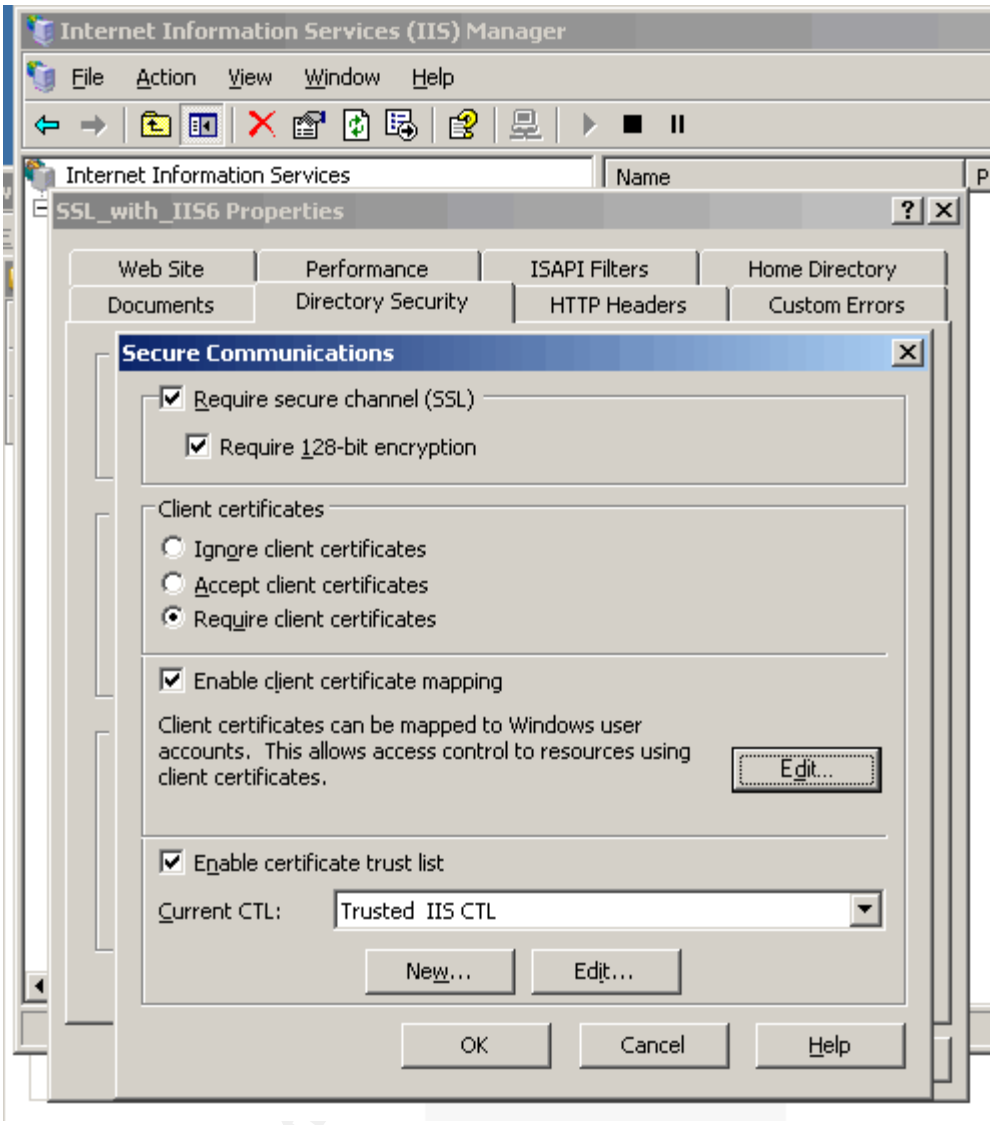
5. **Example: Setting up IIS 6.0 to require Client Certificate**

To demonstrate a simple, yet effective way to use Client Certificates for web authentication, we will use the following setup: Microsoft Internet Information Server (IIS) 6.0 that comes integrated with Windows 2003 Server platform; Mozilla browser 1.6 with two Digital IDs, one from Thawte and one from Verisign. In the Attachment 2 we provided the content of both certificates. For the the purpose of this demonstration the most important Subject field in Client Certificates

is e-mail address. Note that in certificate, issued by Thawte, email of the client is *akazantsev@ncaccesscare.ork*, in the Certificate, issued by Verisign email is *tema@bigfood.com.*
Unfortunately, Microsoft did not provide an easy way to export the configuration into text, so we had to use screen shots to prove the point.
The first configuration step is to install an SSL server certificate. We assume that this task is already performed. *[help with the installation of SSL server certificates could be found on many Internet sites (http://www.thawte.com/support/keygen/index.html http://searchsupport.verisign.com/content/kb/vs1895.html to name a few) and beyond the scope of this document.]*
The second step is to disable Anonymous access and enable Basic authentication in Directory Security tab (see screen1):



screen 1



screen 2

Next, we need to enable SSL access on Directory Security tab, see screen 2. By default, we will use the standard port 443.

we selected the following options:

- Require 128-bit encryption (it is optional -*AK*)
- Require client certificates (that's what we want)
- Enable client certificate mapping (below we will explain why this option is selected)
- Enable certificate trust list. Without this option selected, IIS 6.0 will not trust any Client Certificates and will generate the error message:

  HTTP Error 403.16 - Forbidden: Client certificate is ill-formed or is not trusted by the Web server.

  Creating a trust list with CAs, used by organization is a good way to reinforce the PKI policy, but should not be a

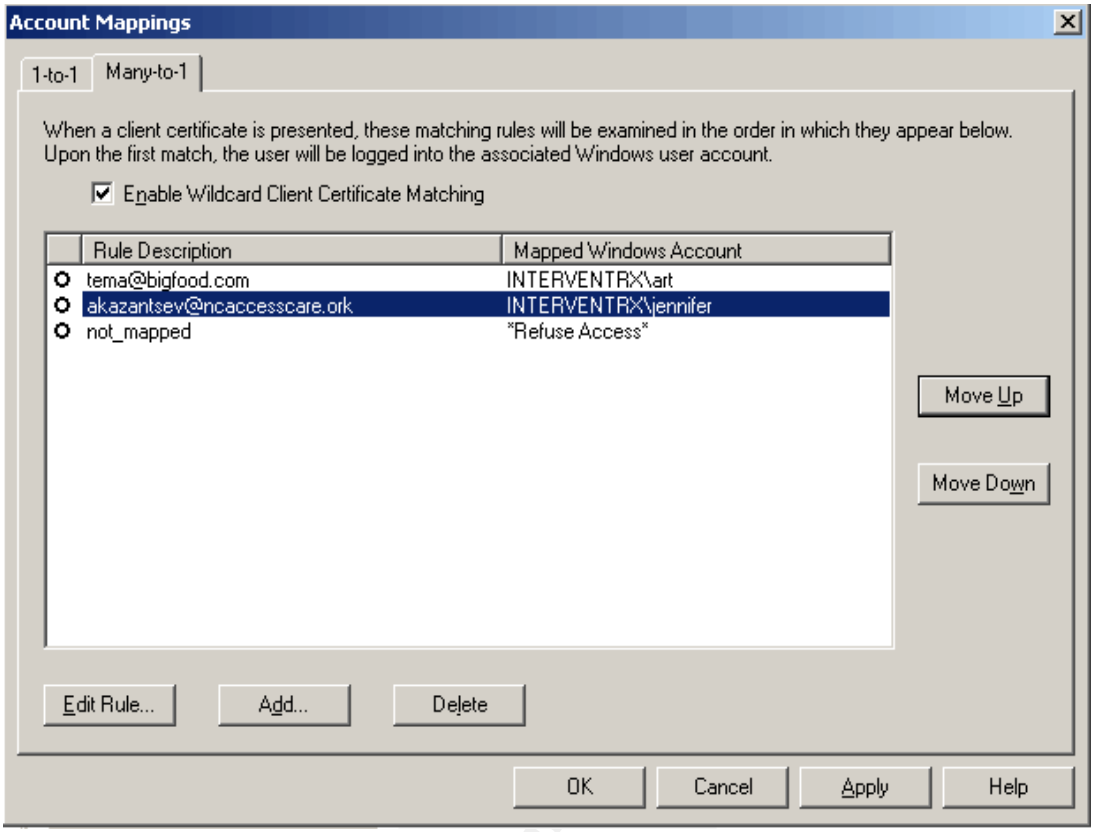
substitution of PKI Policy itself.
The caveat with adding CAs to a CTL (called Trused IIS CTL on the screen): only self-signed Root certificates are accepted. If we do not have one, we are out of luck. Fortunately, most of the self-signed Root Certificates from common CAs are already in the Local Computer Certificate store. But to make things more complicated, Microsoft included very few Intermediate Signing certificates in the certificate store. To make, say, Personal Freemail Client Certificate from Thawte trusted, one needs to import 'Personal Freemail RSA 2000.8.30 – Thawte Consulting' Intermediate certificate into the Local Computer Certificate store, using *mmc* and Certificates Snap-In.

In the next 3 screens we will show how mapping e-mail address field from a Digital ID to a local computer account could enable us to provide authentication and plus an enforcement of Access Control List (ACL) on the server. It should be stated, that although the mapping feature proves to be very useful in environments with a few number of users, it could become a management nightmare if the number of users grows beyond tens. Microsoft also did not provide any built-in mechanism to backup or export the existing maps, making re-creation or replication of such setups extremely difficult.

First, select 'Edit' in 'Enable client certificate mapping' from the previous screen 2.

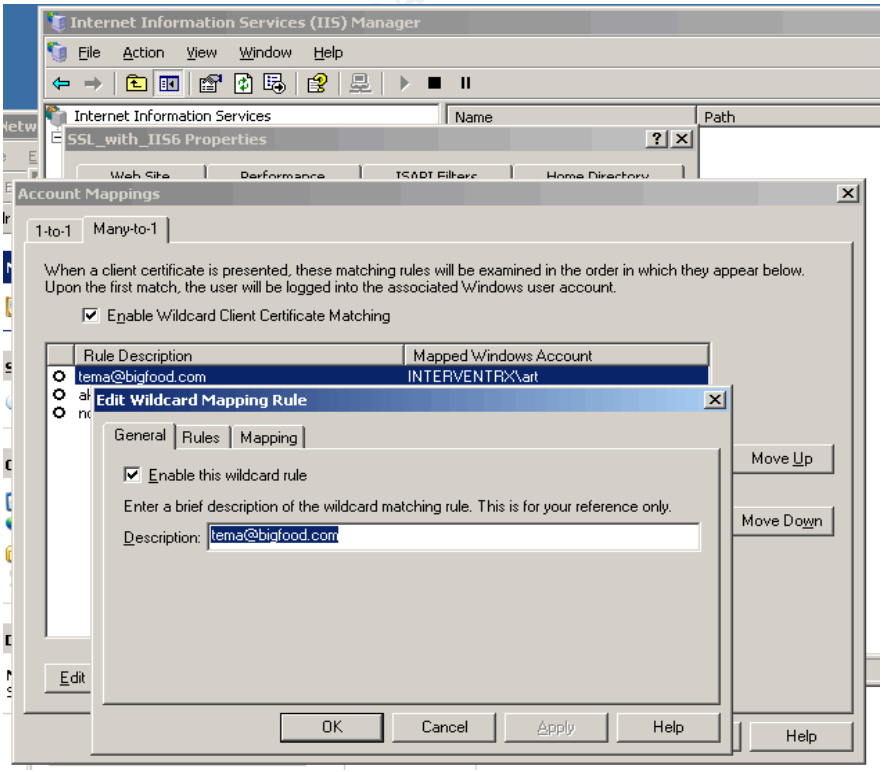
IIS offers two options in mapping Certificates to Local Computer accounts:

- 1-to-1 mapping
  To use this option, the webserver administrator needs to have all users' certificates in the public format beforehand. While this 'stricter' option could be useful, it is not practical in situations, when site users are inaccessible. 1-to-1 mapping would be a great choice for Intranet or sites with a live access to PKI directory.
- many-to-1
  This option is the choice for this demonstration. Screen 3 shows the details: we enabled Wild Card certificate mapping, and mapped two email address, *tema@bigfood.com* to the local account *art* and *akazantsev@ncaccesscare.ork* to the account *jennifer.* The e-mail addresses on this screen are for description only, they are used by website operators to distinguish the rules. The last line shows that we will refuse web access by certificates, that are not explicitly mapped. Although this line is not strictly necessary, it will help the administrator to troubleshoot failed connections.

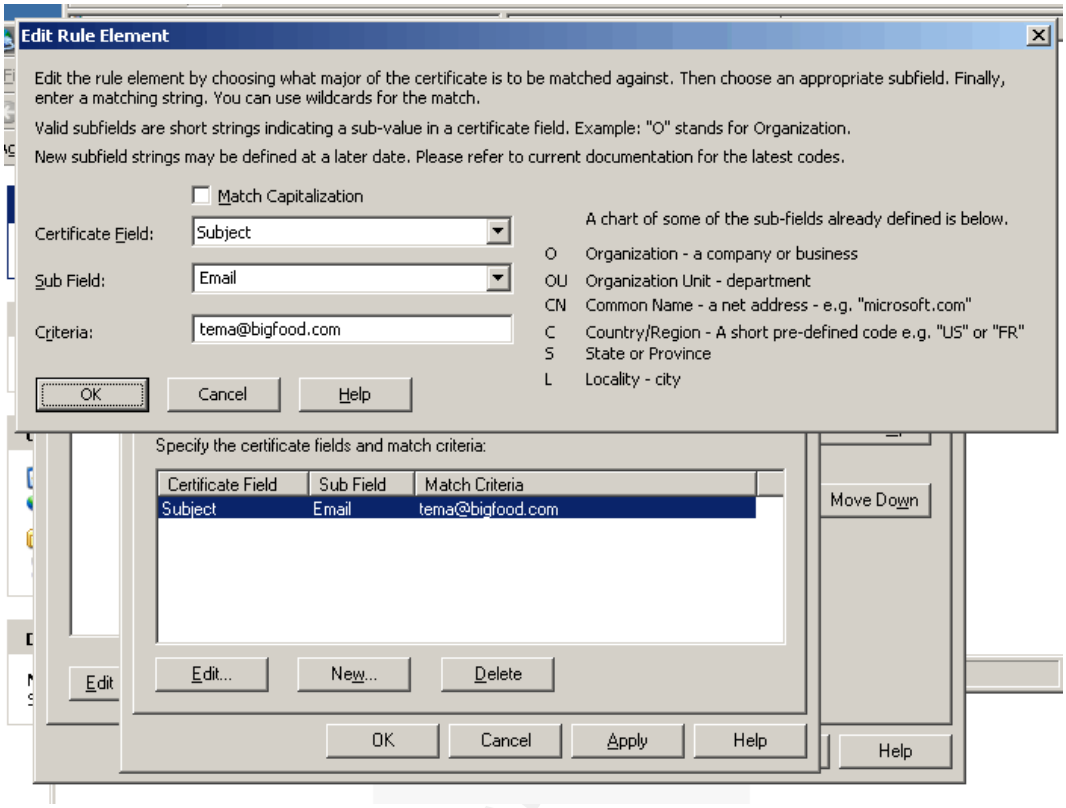


screen 3

Screen 4 shows the first tab of 'Edit rule' selection.

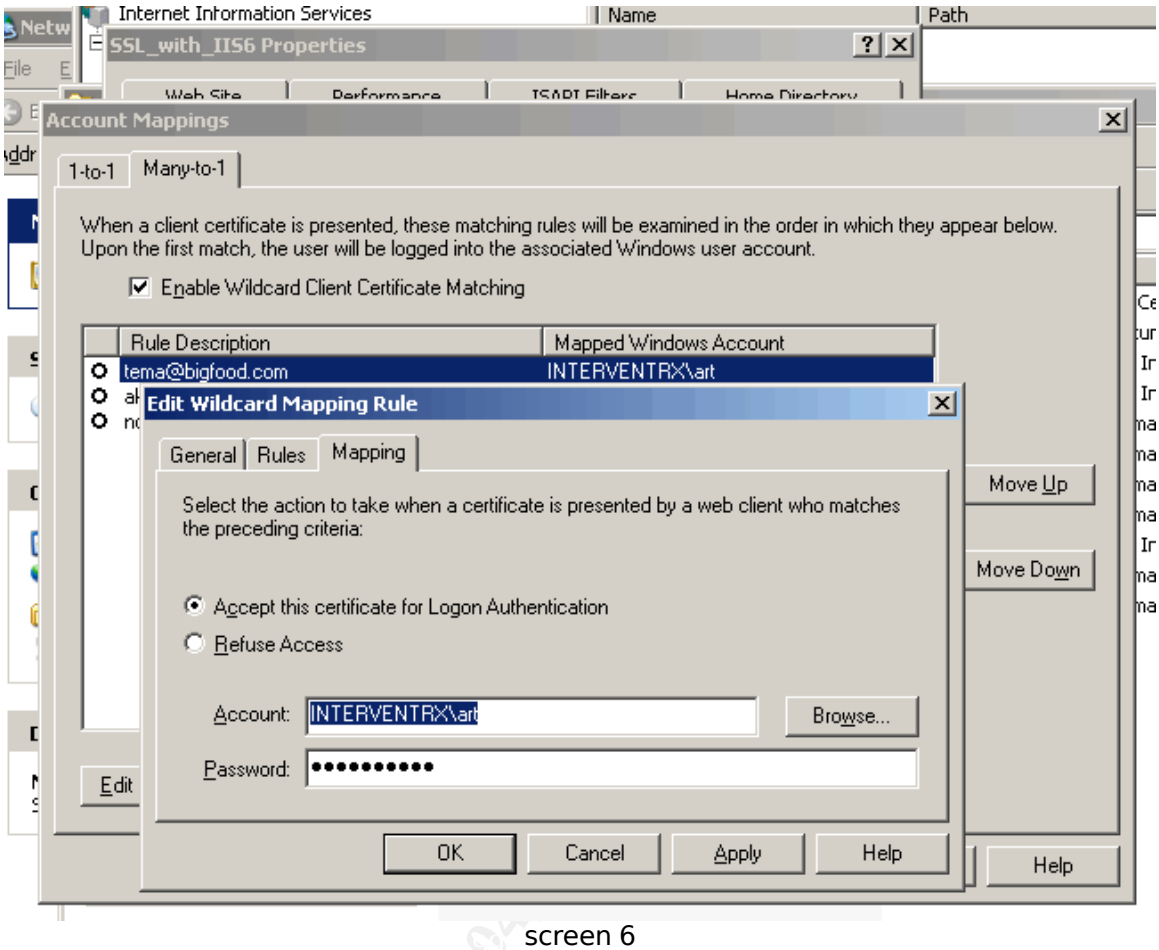


screen 4



screen 5

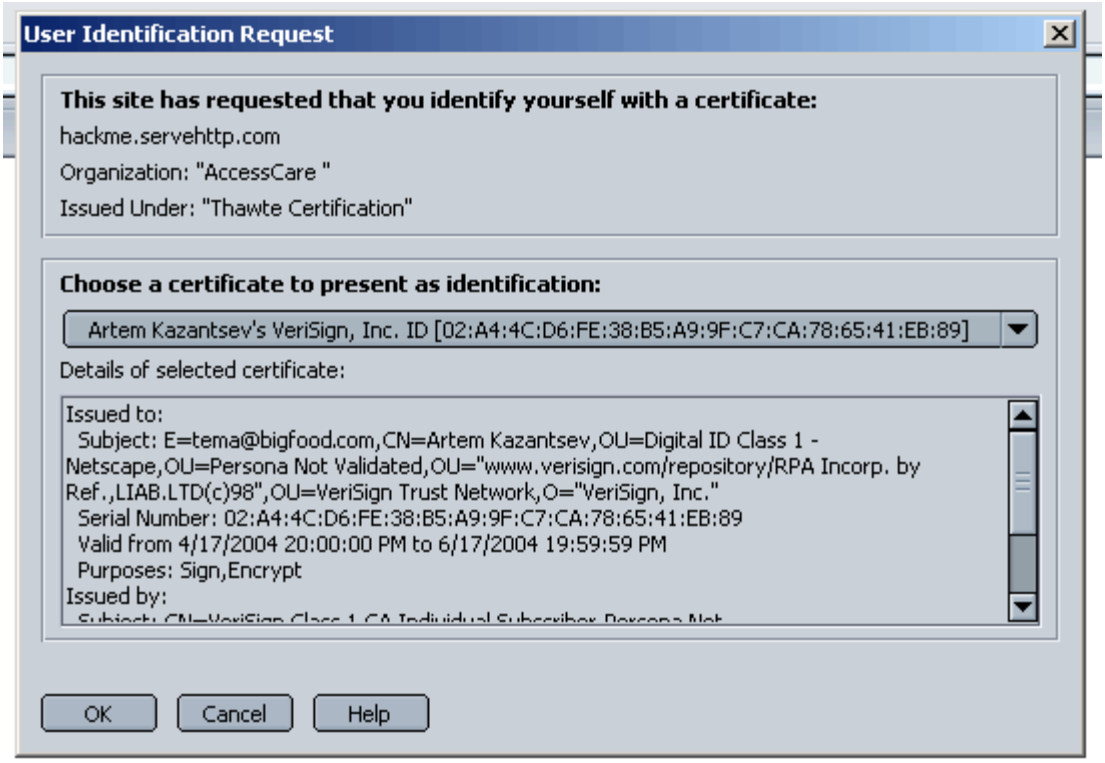
Screen 5 has the most importance to us because here we can select what Certificate fields can be used and what values can be assigned. In this example, we selected not to 'Match capitalization' (for compatibility reasons). Next, we selected 'Subject' (the other option could be 'Issuer'), Sub field- Email, Criteria – *tema@bigfood.com.* Note that any valid Certificate, issued by any CA from the 'Trusted IIS CTL' and that has *tema@bigfood.com* in the Subject, will be accepted. The implication of this rule for the website administrator is that she doesn't need to know anything about this user except for his e-mail address. In essence, this rule provides secure, encrypted website access to the user based purely on trust, offloading the burden of proof of identity to Certificate Authorities. No clear text passwords were sent to a user, no mail envelopes, yet, we managed to ensure that: a) the access is granted to the right person, b) the connection is encrypted, c) that simple ACL rules can be achieved just with one extra step (shown on screen 6).



screen 6

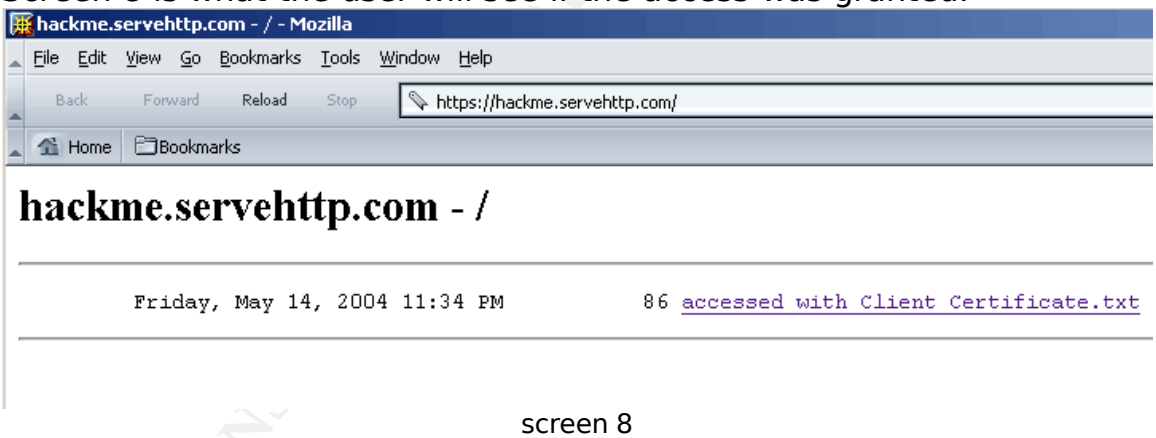
On screen 6 we mapped the Certificate to the local account *art* . Now, the presenter of the Certificate with e-mail address *tema@bigfood.com* will be matched with the user *art* and will have the same access rights to the local computer as user *art* does. For example, if NTFS permissions on the local machine do not allow the access of the user *art* to the folder 'Jennifer's secret data', the '*tema@bigfood.com*' Certificate holder will not be able to access this folder either.
The other consideration is whether to fill in the 'Password' or not. That would depend on the particular situation. In case when users of the site never login interactively it makes sense to generate long, strong, random passwords, and assign them to local user accounts. Keep the master password list encrypted and off the webserver. On the other hand, if the password is used every day, say, for Domain authentication, filling it in would be neither practical, nor secure (one should never share one's password with anybody, even with the system administrator).

Finally, the screen 7 shows the dialog box that Mozilla Browser presents while the Preference option Select Certificate is set to 'Manual'. (Server's web address is hackme.servehttp.com)



screen 7

Screen 8 is what the user will see if the access was granted.



screen 8

6. **Caveats and limitations**

Basic principles of security should not be set aside when one implements SSL authentication with Digital IDs. For example, SSL protocol doesn't protect data in storage, i.e. on the web server itself. It also doesn't prevent general website exploits, virus penetration and other attacks that could be mounted due to unpatched OS or other software.

But certain components of the described process are particularly important because they are the foundation blocks for the whole mechanism. Let´s discuss the most important ones:

- *Domain Name System (DNS)* – if one cannot adequately protect

one´s DNS server, any certificate, issued to the specific domain name cannot be trusted since the control of the domain itself is in question.

- *E-mail protection* – we established that most CAs issue client certificates, (and some vendors (http://www.freessl.com) – server certificates), via e-mail. For Client Digital IDs e-mail address is the most important field in the Subject, and for majority of low grade versions of Certificates, the only field, that a CA will assert. Therefore, protection of the email from interception or messages stored in Client´s account becomes a paramount.
- *Time control* – each certificate, issued by a well-known CA, has a time frame within which it is valid, usually, one year. Thus, by manipulating computer clock, an attacker can re-use expired and potentially untrusted certificates.
  The effect of a Client and Server's clocks being out off sync could become a nuisance, especially when Digital IDs are just installed. Client and Server´s software would automatically reject certificates, that are expired or not yet valid, and would not accept a perfectly good certificate because one of the computers' clocks is off. A remedy to this potential problem is the use of Internet Time synchronization, see http://www.boulder.nist.gov/timefreq/service/its.htm
- *Private Key protection* – what it is suppose to be – protect your private key!
- *Patch your OS (and software)* – no matter how stringent your verification mechanism. If your OS is not updated with the latest patches or the software is not upgraded, you are not protected from exploits and ultimately, you become an untrusted party. For instance, some vulnerabilities in Microsoft Windows could allow identity spoofing (see Microsoft Security Bulletin MS02-050 (Q329115)[7]) or deletion of existing certificates (see Microsoft Security Bulletin MS02-048 (Q323172)[8]. A buffer overflow in OpenSSL could lead to DOS attacks[9], thus undermining one of the three main security principles, availability. The list of known vulnerabilities in SSL could be found at CVE site mitre.org [10].

7. Other platforms

In the interest of space, the example of a configuration file of Apache Httpd server 2.0 with mod_ssl and OpenSSL , that runs on Linux, is included in Attachment 1. The content of two Client Certificates, used in all our examples, is presented in Attachment 2. The output was generated by OpenSSL 0.9.7a software.
The setup of IIS 5 on Windows Server 2000 is similar to the one described in part 5 for IIS 6. The only significant difference that we discovered was the fact, that “Trust List” in IIS 5 is optional, as well as the import of the Intermediate signing certificates.

8. Conclusion

We demonstrated that Client Certificates issued by well-established Certificate Authorities could be used successfully in authentication to websites. Small and medium healthcare organizations, law firms, consulting companies -- all could benefit from usage of Digital IDs. Trying out free Client Certificates in large companies should become a training ground before a full-fledge PKI solution is adapted. It is likely that with time, Digital IDs will become as widely used as the web itself, but even now they can be used effectively for web authentication tasks.

Reference

1. TLS version 1, Request for Comments 2246
   http://www.ietf.org/rfc/rfc2246.txt
2. How SSL works
   http://developer.netscape.com/tech/security/ssl/howitworks.html
3. mod_ssl manual, SSL introduction
   http://www.modssl.org/docs/2.8/ssl_intro.html
4. Advanced Encryption Standard (AES) Ciphersuites for Transport Layer Security (TLS)
   http://www.ietf.org/rfc/rfc3268.txt
5. “PKI Policy pitfalls” Information Security magazine, July 2001 p.68
6. Internet X.509 Public Key Infrastructure
   http://www.ietf.org/rfc/rfc2459.txt
7. Microsoft Security Bulletin MS02-050
   http://www.microsoft.com/technet/security/bulletin/MS02-050.mspx
8. Microsoft Security Bulletin MS02-048
   http://www.microsoft.com/technet/security/bulletin/MS02-048.mspx
9. Open SSL security advisory
   http://www.openssl.org/news/secadv_20040317.txt
10. Known vulnerabilities in SSL
    http://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=SSL

Attachment 1

Example of Apache 2.0 / mod_ssl configuration files

/etc/httpd/conf.d/ssl_test.conf

```
#
# This configuration file is a test of using SSL with
# Client x.509 authentication;
#
# current setup is Apache 2.0.49 running on Fedore Core2 test3
# with mod_ssl 2.0.49-4 and OpenSSL 0.9.7a-37 ;
# user mapping is used in two places and in two ways:
# by using SSLOptions +FakeBasicAuth
```

```
# and using environment variable SSL_CLIENT_S_DN_Email
#
# to illustrate the point, we will use two different client certificates,
# with different Subject, Email address and from different CAs,
# Verisign and Thawte

# Example 1
# directory /secure is set up in such a way that only trusted Certificate with the
# Email address akazantsev@ncaccesscare.ork be accepted

Alias /secure /var/www/secure
<Location "/secure">
# we will protect our testing environment
# since we are using local network for this demonstration:

order deny,allow
deny from all
allow from localhost
allow from 192.168.10.0/255.255.255.0

# enable ssl
SSLRequireSSL

# Deny low grade encryption:
SSLCipherSuite HIGH:MEDIUM

# here a client must  present a certificate, otherwise connection is
SSLVerifyClient  require

# We will accept only certificates with no deeper than 5 chain links;
# practically, examination of certificates employed in our tests shows that
# they use no more than 2 hops:
# Client Certificate (0) -- Intermediate Signing (1) -- Root CA (2)
# As a side effect, this option also will force the renegotiation of SSL
# session, because we use it in the per directory fashion.
SSLVerifyDepth       5

# ca-bundle.crt comes standard with the OpenSSL distribution
# and should be updated on the regular basis, because it
# contains CA root certificates as well as all revoked ones.
# OCSP is not implemented, use published CRLs from CAs
# to download manually. One may edit this file to further
# restrict CA pool to a small number preferred ones.
SSLCACertificateFile /usr/share/ssl/certs/ca-bundle.crt
SSLCACertificatePath /usr/share/ssl/certs/

# here we request that only certificates with
# Email address akazantsev@ncaccesscare.ork in the subject
# of the Client's certificate be accepted
SSLRequire          %{SSL_CLIENT_S_DN_Email} in {"akazantsev@ncaccesscare.ork"}
</Location>

# Example 2
# The  directory /ssl_users does not have restrictions on the Certificate
# subject,  therefore any Client certificate that is valid, issued by trusted
```

```
# CA, and has less than 5 chain links will get through.
# This would be the ideal place to publish the rules and procedures for
# users, who already obtained a certificate, but need an approval by the site
# administrator or a webmaster for the deeper access.
#
Alias /ssl_users/ /var/www/ssl_users/
<Directory /var/www/ssl_users/>
order deny,allow
deny from all
allow from localhost
allow from 192.168.10.0/255.255.255.0
SSLVerifyClient      require
SSLVerifyDepth       5
SSLCACertificateFile /usr/share/ssl/certs/ca-bundle.crt
SSLCACertificatePath /usr/share/ssl/certs/
SSLRequireSSL
</Directory>

# Example 3
# The directory /ssl_users/art was created for users,
# whose certificates should satisfy all rules from Example 2 plus their Client certificate
# entire Subject fied should be mapped to the predetermined password in the file
# /etc/httpd/conf/ssl.passwd
# that is controlled by the option +FakeBasicAuth
# (see also the file ssl.passwd)

Alias /ssl_users/art /var/www/ssl_users/art
<Directory /var/www/ssl_users/art>
order deny,allow
deny from all
allow from localhost
allow from 192.168.10.0/255.255.255.0
SSLVerifyClient      require
SSLVerifyDepth       5
SSLCACertificateFile /usr/share/ssl/certs/ca-bundle.crt
SSLCACertificatePath /usr/share/ssl/certs/
SSLOptions           +FakeBasicAuth
SSLRequireSSL
AuthName             "test of SSL user mapping"
AuthType             Basic
AuthUserFile         /etc/httpd/conf/ssl.passwd
require              valid-user
</Directory>
```

file /etc/httpd/conf/ssl.passwd

```
/O=VeriSign, Inc./OU=VeriSign Trust Network/OU=www.verisign.com/repository/RPA
Incorp. by Ref.,LIAB.LTD(c)98/OU=Persona Not Validated/OU=Digital ID Class 1 -
Netscape/CN=Artem Kazantsev/emailAddress=tema@bigfood.com:xxj31ZMTZzkVA
```

The password xxj31ZMTZzkVA should be the same for all enabled users. If we use this file as described in Example 3, only the user with the Certificate from Verisign and name Artem Kazantsev, Email address tema@bigfood.com will connect to the /ssl_users/art directory. Note the difference between Example 1 and 3, for the authentication in Example 1 we used only Email address as a unique identifier.

Attachment 2

Certificate 1

Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number: 665939 (0xa2953)
    Signature Algorithm: md5WithRSAEncryption
    Issuer: C=ZA, ST=Western Cape, L=Cape Town, O=Thawte, OU=Certificate Services, CN=Personal Freemail RSA 2000.8.30
    Validity
      Not Before: Jun 16 18:00:48 2003 GMT
      Not After : Jun 15 18:00:48 2004 GMT
    Subject: C=US, ST=North Carolina, L=Morrisville, O=Accesscare Inc./title=IT Manager, SN=Kazantsev, GN=Artem, CN=Artem Kazantsev/emailAddress=akazantsev@ncaccesscare.ork
    Subject Public Key Info:
      Public Key Algorithm: rsaEncryption
      RSA Public Key: (1024 bit)
        Modulus (1024 bit):
          00:f0:a4:84:4d:47:02:68:d1:bb:fe:8b:cb:55:1a:
          73:d3:61:a3:f6:91:3c:06:40:8c:82:18:d4:48:4c:
          e4:a9:fc:ef:96:1e:e1:32:e6:ae:14:ce:ee:17:d3:
          3c:2a:bf:9d:df:e3:a0:69:9e:79:b2:df:c5:3d:5d:
          03:73:e0:2e:b0:81:ab:5d:99:96:59:2b:24:60:55:
          0b:68:c8:20:1f:a1:65:66:1a:f4:96:af:a6:e1:94:
          54:0c:dd:91:8d:26:29:6d:99:4e:95:30:27:74:6a:
          8e:2e:29:33:4a:47:95:d3:3b:1d:fe:0b:e0:30:24:
          49:00:09:7e:12:04:e2:0d:1f
        Exponent: 65537 (0x10001)
    X509v3 extensions:
      Strong Extranet ID:
      Version: 1 (0x0)
      Zone: 1102, User: Artem Kazantsev
      X509v3 Subject Alternative Name:
      email:akazantsev@ncaccesscare.ork
      X509v3 Basic Constraints: critical
      CA:FALSE
  Signature Algorithm: md5WithRSAEncryption
    08:49:2c:25:2b:c8:ba:dd:09:21:db:78:06:5d:af:eb:dd:7c:
    eb:d7:e6:b5:4b:94:f7:19:a8:e9:ee:87:f1:8b:e9:5b:b3:42:
    fd:49:30:62:ed:fb:34:2d:2f:29:01:6b:32:af:06:09:a4:d9:
    32:4d:5c:a6:9e:2b:6b:5d:20:ba:f4:1b:03:93:ae:5a:21:69:
    3a:da:ea:f6:fb:e1:b6:5e:d5:e7:68:9d:bc:5f:3b:e4:45:e3:
    e1:bf:b4:a3:9a:e8:72:15:8e:0e:82:38:7c:fc:36:eb:ef:1f:
    78:8d:2b:ee:db:94:e2:7a:45:d3:78:5b:f3:92:e1:82:2b:e8:
    d5:74

Certificate 2

Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number:
      02:a4:4c:d6:fe:38:b5:a9:9f:c7:ca:78:65:41:eb:89
    Signature Algorithm: md5WithRSAEncryption
    Issuer: O=VeriSign, Inc., OU=VeriSign Trust Network,

OU=www.verisign.com/repository/RPA Incorp. By Ref.,LIAB.LTD(c)98, CN=VeriSign Class 1 CA Individual Subscriber-Persona Not Validated
Validity
Not Before: Apr 18 00:00:00 2004 GMT
Not After : Jun 17 23:59:59 2004 GMT
Subject: O=VeriSign, Inc., OU=VeriSign Trust Network, OU=www.verisign.com/repository/RPA Incorp. by Ref.,LIAB.LTD(c)98, OU=Persona Not Validated, OU=Digital ID Class 1 - Netscape, CN=Artem Kazantsev/emailAddress=tema@bigfood.com
Subject Public Key Info:
Public Key Algorithm: rsaEncryption
RSA Public Key: (2048 bit)
Modulus (2048 bit):
00:b3:75:06:a4:95:45:5e:e6:8f:fa:00:22:83:62:
bd:f6:1c:99:55:04:ae:46:f2:4d:92:62:c8:03:80:
32:29:b0:b6:12:96:83:e7:4b:64:15:4a:8b:5b:e6:
1f:3c:30:a2:ee:c9:ea:98:0c:db:fd:02:2f:b2:e3:
2d:10:c3:b3:18:c7:ea:c0:91:53:9d:61:5f:aa:e4:
93:15:25:d0:c5:1b:35:62:5a:b8:0b:bb:ff:b1:ea:
0b:51:a1:69:d1:27:6b:7a:44:33:81:76:fe:e8:4f:
66:f3:e5:72:fb:5e:c9:71:c6:f7:e9:c1:52:c1:dc:
6b:a9:fe:16:c1:ba:67:7f:49:d6:5b:e2:9d:5c:e6:
00:5e:00:1a:52:6b:e9:f1:98:83:04:ae:21:0c:3f:
31:a0:c5:33:84:aa:fa:63:18:a3:c3:be:79:62:62:
43:80:7c:6c:69:fc:f1:1c:d9:e1:a0:ab:70:fe:f0:
9e:6c:76:fa:a6:0e:19:f9:47:65:dc:c6:8a:5e:23:
28:d4:8c:55:29:3a:8c:b2:5d:01:5b:d5:ca:c9:4a:
f7:ca:57:09:c9:9b:76:21:f1:35:2e:82:ba:fe:0b:
cd:f7:1c:05:08:5f:8f:15:44:e4:19:80:fe:9c:00:
aa:31:4e:61:45:ef:05:c2:ca:3e:17:8d:48:7a:06:
b6:83
Exponent: 65537 (0x10001)
X509v3 extensions:
X509v3 Basic Constraints:
CA:FALSE
X509v3 Certificate Policies:
Policy: 2.16.840.1.113733.1.7.1.1
CPS: https://www.verisign.com/CPS
User Notice:
Organization: VeriSign, Inc.
Number: 1
Explicit Text: VeriSign's CPS incorp. by reference liab. ltd. (c)97 VeriSign

Netscape Cert Type:
SSL Client
X509v3 CRL Distribution Points:
URI:http://crl.verisign.com/class1.crl

Signature Algorithm: md5WithRSAEncryption
b0:4a:f0:4e:1e:81:f2:3f:f0:3c:53:0d:82:cb:8a:41:be:fb:
65:3f:12:ba:9b:5c:1c:b5:fa:74:48:ef:6d:4c:32:5f:9d:c3:
0e:0a:71:d8:a7:ed:b4:38:a9:3a:4e:4e:b4:2f:3d:fa:11:d4:
ed:87:11:4d:92:c6:8d:6a:5a:31:8a:5b:95:e5:11:e4:e8:ba:
94:24:07:dc:0e:e4:d2:6e:ae:04:51:d1:8b:45:c1:6b:27:0a:
b8:2d:54:39:10:0c:28:4b:b1:67:ec:89:11:78:a7:d4:bc:41:

52:ff:03:c4:23:03:71:86:0d:b1:b2:b2:8e:d5:1d:72:a9:cf:
c7:9a