Global Information Assurance Certification Paper

An Additional Layer of Defense for Microsoft Outlook Web Access

Chris Sawall
GSEC Practical Assignment Version 1.4b Option 1
28 February 2003

Abstract
This document presents an alternative method of configuring secure access to a Microsoft Outlook Web Access (OWA) server. This document will not describe how to install OWA or how to set up the server securely. That topic is covered in great detail by other documents and FAQs that are readily available on the Internet and within the posted papers of previous GSEC certified graduates. There are many approaches to securing an OWA server. Most involve using a combination of a front-end (OWA) and a back-end (Exchange) server with the OWA server sitting on a secured DMZ exposed to the Internet. This approach requires opening multiple ports from the OWA server through the firewall to the Exchange server(s) located on the internal company network.

This document will cover the steps necessary to set up a secure reverse-proxy server using Apache and RSA SecurID on a Red Hat Linux x86 platform. Following these steps will minimize the number of ports necessary to access the OWA server. Further, it will show how to securely configure a Linux installation, ultimately providing additional security at minimal cost. The audience for this document is administrators with a basic knowledge of Firewalls and Red Hat Linux. It is assumed that the OWA and RSA ACE/Server infrastructure exists in the enterprise. This document can be leveraged to successfully set up a functioning reverse-proxy server even if an ACE/Server is not deployed.

Getting Started
This documentation uses the following software:

- Red Hat 8.0 (http://www.redhat.com/)
- Apache 1.3.27 (http://www.apache.org/dist/httpd/)
- Mod_SSL 2.8.12 (http://www.modssl.org/source/)
- OpenSSL 0.9.7 (http://www.openssl.org/source/)
- RSA WebAgent 5.1 (http://www.rsasecurity.com/go/apacheagent/index.html)

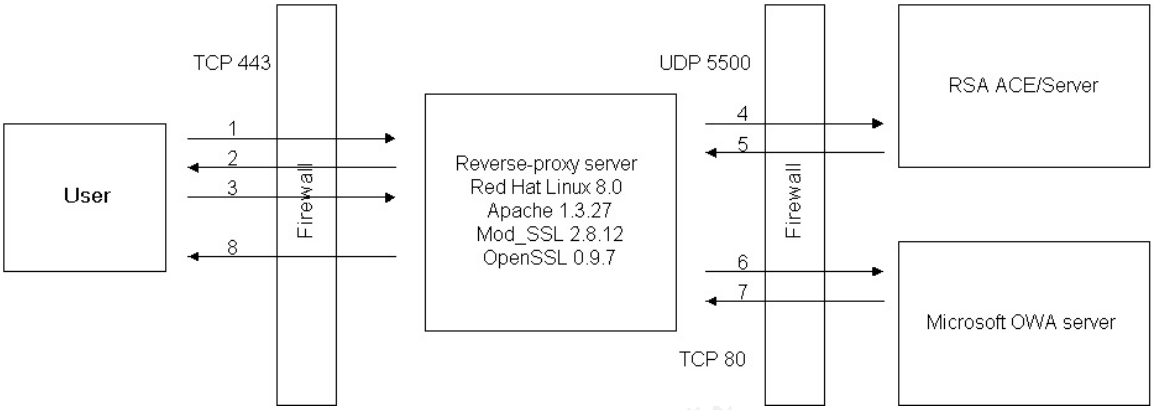
RSA's website states that the RSA ACE/Agent for Apache is only supported with Apache 1.3.26 with mod_ssl 2.8.10. While it is true that the ACE/Agent for Apache will not work with any of the Apache 2.x versions, it will work with higher versions of Apache 1.3.x and mod_ssl 2.8.x as demonstrated in this document.

RSA's website also states that only Red Hat 7.3 is supported. This document will demonstrate that the RSA ACE/Agent for Apache will also work on Red Hat 8.0. Further testing has shown that it will also work on various versions of Mandrake and Solaris, but that is beyond the scope of this paper.

The best location to place the reverse-proxy server is on a secured DMZ of a firewall. The ports that will need to be opened on the firewall are:

- From the Internet to the reverse-proxy server - TCP port 443
- From the reverse-proxy server to the internal OWA server - TCP port 80
- From the reverse-proxy server to the internal ACE/Server - UDP port 5500

Figure 1:



1. User makes initial request to website.
2. Reverse-proxy server returns RSA SecurID login page.
3. User enters their username and PASSCODE.
4. Reverse-proxy server sends data to ACE/Server.
5. ACE/Server acknowledges credentials and allows connection.
6. Reverse-proxy server connects to OWA server.
7. OWA server sends web page to reverse-proxy server.
8. Reverse-proxy server sends OWA web page to end-user.

It will be necessary to define two separate IP addresses on the reverse proxy server. The first one will be the IP address physically defined to the server. This will allow the server to communicate with the internal network. The second address will be the registered address that is available to the general public. This will have to be translated through the firewall. Both should be configured in DNS respective to the direction from which they will be accessed.

For the purposes of this paper, the following IP addresses will be used:

- Reverse-proxy - 192.168.1.100 / 255.255.255.0
- MS OWA server - 10.250.1.150 / 255.255.255.0
- RSA ACE/Server - 10.250.1.200 / 255.255.255.0
- Primary DNS - 10.250.1.30 / 255.255.255.0
- Secondary DNS - 10.250.1.31 / 255.255.255.0
- Tertiary DNS - 10.250.2.30 / 255.255.255.0

Basic Red Hat Installation

There are many available options when installing Red Hat Linux. A start to finish installation of Red Hat with minimal services and applications is described in this document. Servers should be set up with what is minimally required to run. This

reduces the risk of negatively impacting applications and lowers the security risk associated with running unnecessary applications or services that may become exploited.

For installation, only the first three Red Hat CDs are needed. Directions on how to download ISO images of these CDs can be found at http://www.redhat.com/download/howto_download.html.

- Boot the system from Disk One.
- At the "boot:" prompt, type "expert" and hit enter.
- Answer the first series of questions about drivers, language, media type, etc.
- Test the media if necessary, but it does take quite some time. After this stage, the Installer starts.
- Welcome Screen → Next
- Mouse Configuration - Select the mouse type → Next
- Installation Type - Choose Custom → Next
- Disk Partitioning Setup - Choose Automatically Partition → Next
- Automatic Partitioning - Choose to remove all partitions on the system and leave "Review (and modify if necessary) the partitions created" checked → Next
- Say "Yes" to the warning about removing all Partitions
- If everything looks good → Next
- Boot Loader Configuration - Leave defaults → Next
- Network Configuration -
    - Select Edit in the Network Devices window
        - Deselect "Configure using DHCP"
        - Enter IP address and netmask (i.e. 192.168.1.100/255.255.255.0)
        - Select OK
    - Set Hostname manually, be sure to use the fully qualified domain name
    - Fill in Miscellaneous Settings
        - Gateway
        - Primary, Secondary and Tertiary DNS servers
- Firewall Configuration
    - Select High security level
    - Leave Customize selected
    - Do not check ETH0 as a trusted interface because it will also be the interface that users from the Internet will be accessing
    - Allow Incoming
        - Choose "WWW (HTTP)" and "SSH"
    - Other Ports
        - Enter in "443" so that the firewall will allow HTTPS
- Additional Language Support → Next
- Choose the appropriate Timezone → Next

- Set the ROOT password; be sure to select a fairly complicated password. Tips on creating a secure password can be found at http://csrc.nist.gov/organizations/fissea/presentations/2000/passwrd-guide.doc.
    - Add a user account.
- Authentication Configuration → Next
- Package Group Selection
    - Deselect X Window System
    - Deselect GNOME Desktop Environment
    - Select Editors
        - Edit this selection and deselect all options except VIM
    - Deselect Graphical Internet
    - Edit Text-based Internet
        - Deselect all options and then select LYNX
    - Deselect Office/Productivity
    - Deselect Sound and Video
    - Deselect Graphics
    - Select Development Tools
        - Edit this selection and deselect all optional packages
    - Deselect Printing Support
- About to Install → Next
- Installation will begin
    - Insert Disk 2 and 3 when asked
- Boot Disk Creation - Leave the defaults → Next
- At the Insert Floppy warning, insert a blank floppy disk and choose "Make Boot Disk."
- Remove the floppy when finished and select Exit
    - The CD-ROM will eject upon installation shutting down.

Securing the Server

The first time the server is booted, KUDZU may run. This is the Red Hat hardware detection wizard software. KUDZU will be disabled later. Once the system is booted, the administrator will be able to access the system remotely via SSH. SSH has the same functionality as telnet, but the connection is secured via an encrypted tunnel. One of the better freeware SSH clients is PuTTY by Simon Tathom [1]. PuTTY can be downloaded from http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html.

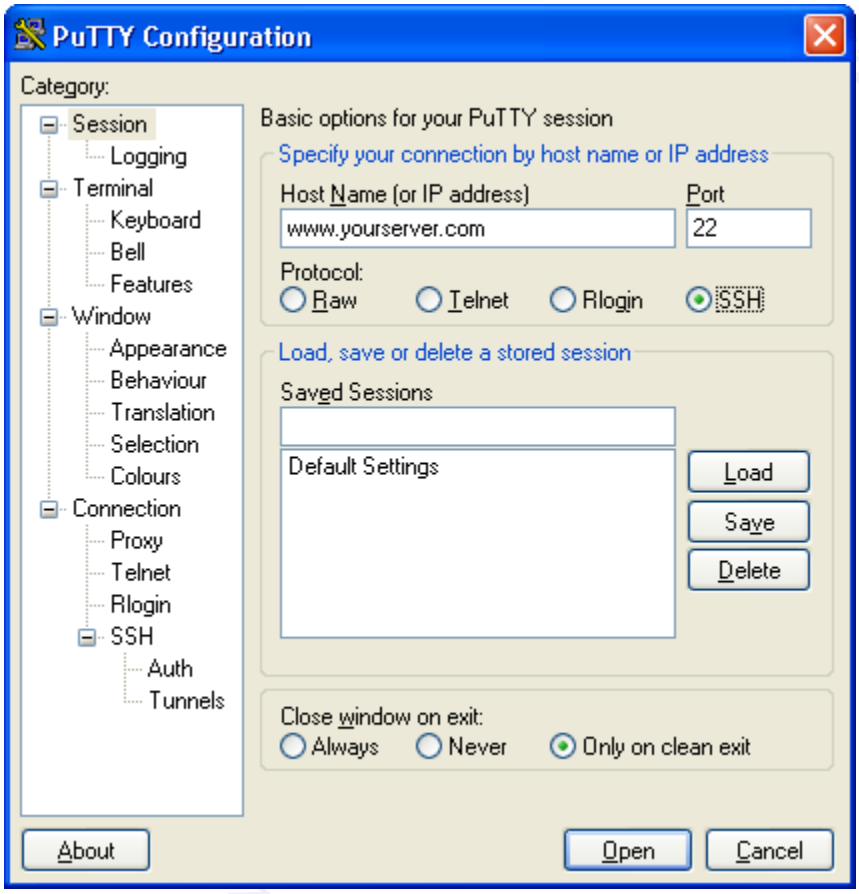
Start PuTTY and enter the server name or IP address into the Host Name field. Select the radio button for SSH and select Connect. See Figure 2 for an example.

Once a connection has been established, there will be a prompt to accept the fingerprint from the server. Accept the fingerprint. Enter the username and password that was created during the Red Hat installation. Once authenticated,

"su" to root. One way to accomplish this is by typing "su -" so that when users are switched, the session will inherit root's environment.
One of the first things to do is begin disabling services. A list of services and at what run level the services are running can be attained by typing:

```
chkconfig --list
```

Figure 2.



There are many services running that are not needed for Apache to run. Disable the following:

```
chkconfig --level 345 netfs off
chkconfig --level 2345 isdn off
chkconfig --level 345 kudzu off
chkconfig --level 345 nfslock off
chkconfig --level 2345 pcmcia off
chkconfig --level 345 sendmail off
```

Disable Sendmail unless it is required, perhaps for the purpose of emailing alerts. If this is the case, Sendmail should be secured. Visit http://www.sendmail.net/000705securitygeneral.shtml for details. If the RSA ACE/Agent for Apache is going to be installed, the Portmap service needs to remain running. If the RSA ACE/Agent for Apache is not going to be installed, the Portmap service should be disabled, as described above.

Reboot the system. After the system is back up, run the netstat command to see what services are running and what ports are open.

```
netstat -atuvp
```

As described in Bob Toxen's book, Real World Linux Security [2]:

> The -a flag indicates that even ports that are not connected to an associated port on another system should be listed. This is important in order to list servers that do not have clients presently. The -t and -u flags specify that only TCP and UDP services should be listed. The -v flag add verbosity. The -p flag says to list the name of the active program using each port; it requires root permission.

Edit the /etc/hosts file to create an entry for the server. Ensure that there is an entry similar to:

127.0.0.1 localhost.localdomain.com localhost
192.168.1.100 yourserver.yourdomain.com yourserver

Finally, IPTABLES will need to be configured to allow connections to the ACE/Server (if being used). View the current IPTABLES configuration:

```
cat /etc/sysconfig/iptables
```

```
# Firewall configuration written by lokkit
# Manual customization of this file is not recommended.
# Note: ifup-post will punch the current nameservers through the
#       firewall; such entries will *not* be listed here.
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:RH-Lokkit-0-50-INPUT - [0:0]
-A INPUT -j RH-Lokkit-0-50-INPUT
-A RH-Lokkit-0-50-INPUT -p tcp -m tcp --dport 443 --syn -j ACCEPT
-A RH-Lokkit-0-50-INPUT -p tcp -m tcp --dport 80 --syn -j ACCEPT
-A RH-Lokkit-0-50-INPUT -p tcp -m tcp --dport 22 --syn -j ACCEPT
-A RH-Lokkit-0-50-INPUT -i lo -j ACCEPT
-A RH-Lokkit-0-50-INPUT -p udp -m udp -s 10.250.1.30 --sport 53 -d 0/0 -j ACCEPT
-A RH-Lokkit-0-50-INPUT -p udp -m udp -s 10.250.1.31 --sport 53 -d 0/0 -j ACCEPT
-A RH-Lokkit-0-50-INPUT -p udp -m udp -s 10.250.2.30 --sport 53 -d 0/0 -j ACCEPT
-A RH-Lokkit-0-50-INPUT -p tcp -m tcp --syn -j REJECT
-A RH-Lokkit-0-50-INPUT -p udp -m udp -j REJECT
```

Red Hat creates the /etc/sysconfig/iptables file according to configuration choices made during installation. Add a line after the DNS entries to allow UDP port 5500. In the example above, this is after the eighth line. Add the line and save the configuration by typing the following from the command line.

```
iptables -I RH-Lokkit-0-50-INPUT 8 -p udp -m udp -s 10.250.1.200 --sport 5500 -d 0/0
-j ACCEPT
iptables-save > /etc/sysconfig/iptables
```

If the /etc/sysconfig/iptables file is viewed, it will now contain the new rule allowing UDP port 5500.

```
# Generated by iptables-save v1.2.6a on Fri Feb 21 14:55:05 2003
*filter
:INPUT ACCEPT [97:5208]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [80:8311]
:RH-Lokkit-0-50-INPUT - [0:0]
-A INPUT -j RH-Lokkit-0-50-INPUT
-A RH-Lokkit-0-50-INPUT -p tcp -m tcp --dport 443 --tcp-flags SYN,RST,ACK SYN -j ACCEPT
-A RH-Lokkit-0-50-INPUT -p tcp -m tcp --dport 80 --tcp-flags SYN,RST,ACK SYN -j ACCEPT
-A RH-Lokkit-0-50-INPUT -p tcp -m tcp --dport 22 --tcp-flags SYN,RST,ACK SYN -j ACCEPT
-A RH-Lokkit-0-50-INPUT -i lo -j ACCEPT
-A RH-Lokkit-0-50-INPUT -s 10.250.1.30 -p udp -m udp --sport 53 -j ACCEPT
-A RH-Lokkit-0-50-INPUT -s 10.250.1.31 -p udp -m udp --sport 53 -j ACCEPT
-A RH-Lokkit-0-50-INPUT -s 10.250.2.30 -p udp -m udp --sport 53 -j ACCEPT
-A RH-Lokkit-0-50-INPUT -s 10.250.1.200 -p udp -m udp --sport 5500 -j ACCEPT
-A RH-Lokkit-0-50-INPUT -p tcp -m tcp --tcp-flags SYN,RST,ACK SYN -j REJECT --reject-with
icmp-port-unreachable
-A RH-Lokkit-0-50-INPUT -p udp -m udp -j REJECT --reject-with icmp-port-unreachable
COMMIT
# Completed on Fri Feb 21 14:55:05 2003
```

Getting Ready to Install Apache and the RSA Web Agent

The next step is to create a common directory where the various applications can be downloaded and extracted.

```
mkdir /home/download
cd /home/download
```

Since a GUI was not installed on the system, a non-graphical tool to download the web applications will be utilized.

```
lynx http://www.apache.org/dist/httpd/apache_1.3.27.tar.gz
lynx http://www.modssl.org/source/mod_ssl-2.8.12-1.3.27.tar.gz
lynx http://www.openssl.org/source/openssl-0.9.7.tar.gz
lynx http://www.rsasecurity.com/go/apacheagent/index.html

gzip -d apache_1.3.27.tar.gz
gzip -d mod_ssl-2.8.12-1.3.27.tar.gz
gzip -d openssl-0.9.7.tar.gz
gzip -d WebAgent5.1.tar.gz
tar xvf apache_1.3.27.tar
tar xvf mod_ssl-2.8.12-1.3.27.tar
tar xvf openssl-0.9.7.tar

mkdir /home/download/rsaweb
mv WebAgent5.1.tar ./rsaweb/
cd /home/download/rsaweb
tar xvf WebAgent5.1.tar
```

Installing Apache

The first step is to configure OpenSSL.

```
cd /home/download/openssl-0.9.7
./config
make
```

The next step is to configure Mod_SSL for use with Apache.

```
cd /home/download/mod_ssl-2.8.12-1.3.27
./configure --with-apache=../apache_1.3.27 --with-ssl=../openssl-0.9.7 \
        --prefix=/usr/local/apache
```

Configure Apache.

```
cd /home/download/apache_1.3.27
SSL_BASE=/home/download/openssl-0.9.7 ./configure       \
        --with-layout=Apache --prefix=/usr/local/apache  \
        --enable-module=ssl --enable-module=so           \
        --enable-module=proxy
make
make certificate
```

Follow the onscreen instructions to create a test certificate. Creating a permanent certificate will be covered later.

- Choose RSA as the type.

- Spell out the STATE, for example, enter Missouri, not MO.
- Spell out the CITY, for example, enter Saint Louis, not St. Louis.
- Common name means the fully qualified domain name of the website, something similar to yourserver.yourdomain.com.
- Certificate length defaults to 365 days (1 year). Enter any length of time since this is a test certificate. A permanent certificate is only good for one to two years.
- Encrypt the key when asked.
- Set the pass phrase and do not forget it.

Finally, install Apache and test the installation.

```
make install
/usr/local/apache/bin/apachectl startssl
```

Upon starting the Apache service, it will prompt for a pass phrase. Enter the pass phrase used when creating the certificate. This is a security mechanism to ensure that the appropriate party is starting the SSL web server.

The functionality should now exist to browse to https://yourserver.yourdomain.com/ with any Internet web browser. The application will prompt to accept the certificate. This is because a non-trusted third party issued the certificate. The user should notice a yellow sign with an exclamation point icon. The information next to this sign reiterates the fact that it is a non-trusted certificate. For testing purposes, this is fine. Accept the certificate by clicking Accept. The web server should return the SSL/TLS test web page.

Requesting and Installing a Permanent Certificate

To obtain a permanent certificate, it will be necessary to create a Certificate Signing Request (CSR). For the purposes of this documentation, a certificate from Thawte (http://www.thawte.com/) was used. The following information on creating a CSR to request a certificate was obtained from Thawte [3].

Change into the parent directory where Apache stores its SSL certificates.

```
cd /usr/local/apache/conf
```

The folders necessary for certificate and key storage are created with the Apache install. The key folders that will be used to store the files are ssl.key, ssl.csr and ssl.crt.

Run the following command to create a private key. The certificate will be useless without the private key file. There are two ways to go about creating this key; one can encrypt the key or create a plaintext key. Encrypting the key will require the entering of a pass phrase every time the Apache server is started, this is the more secure alternative. However, if encryption of the key for administrative purposes is not chosen, ensure that the key can only be accessed by the administrative account.

To generate a private key that requires a pass phrase:

```
openssl genrsa -des3 -out private.key 1024
```

To generate a private key that does not require a pass phrase:

```
openssl genrsa -out private.key 1024
```

Run the following command to generate a CSR file:

```
openssl req -new -key private.key -out public.csr
```

If running this operation generates errors and the need exists to specify the location of the OpenSSL configuration file, run the command as follows [4]:

```
openssl req -config /usr/share/ssl/openssl.cnf -new -key private.key -out public.csr
```

While creating this key, OpenSSL will ask a series of questions, similar to the questions asked when the test certificate was created. Follow the same rules for the STATE, CITY and Common name.

The private and public files have now been created. Move the files into the appropriate directories and secure the private.key.

```
chmod 400 private.key
mv private.key ./ssl.key/
mv public.csr ./ssl.csr/
```

The next step is to go to Thawte and purchase the certificate. Thawte provides detailed directions on obtaining a permanent certificate and the information that is required.

http://www.thawte.com/guides/StepByStepEnrolmentSSLSGC.doc

The permanent certificate should arrive via email within a couple of days. The certificate will be in the following syntax format [5]:

```
-----BEGIN CERTIFICATE-----
MIAGCSqGSIb3DQEHAqCAMIACAQExADALBgkqhkiG9w0BBwGggDCCAmowggHXAhAF
UbM77e50M63v1Z2A/5O5MA0GCSqGSIb3DQEOBAUAMF8xCzAJBgNVBAYTAlVTMSAw
HgYDVQQKExdSU0EgRGF0YSBTZWN1cml0eSwgSW5jLjEuMCwGA1UECxMlU2VjdXJl
IFNlcnZlciBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0eTAeFw0wMTA4MDIwMDAwMDBa
Fw0wMzA4MDIyMzU5NTlaMIGQMQswCQYDVQQGEwJVUzERMA8GA1UECBMIVmlyZ2lu
aWExETAPBgNVBAcUCFJpY2htb25kMSAwHgYDVQQKFBdDYXZhbGllciBUZWxlcGhv
bmUsIExMQzEcMBoGA1UECxQTSW5mb3JtYXRpb24gU3lzdGVtczEbMBkGA1UEAxQS
Ym9uZGluZy5jYXZ0ZWwuY29tMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC8
x/1dxo2YnblilQLmpiEziOqb7ArVfI1ymXo/MKcbKjnY2Axc45IelP3LHz+/N0Z/
M4N0Noa9eJPiIpRuVMToegtQ9TQuxqRqmrE6tECJja5g0hLRwl/WnHSxg6YbRRsF
xB0H9HShyf9bYmBJ8FYCu6HpWT7p5SVCfj5wjhGxnQIDAQABMA0GCSqGSIb3DQEB
BAUAA34AiwLwK76NzWNPn5Zwfdvpok8Nd3adgUL3yi0MUOb6Ws1EaddUKUqpNiNs
E+cFEpf0WForA+eRP6XraWw8rTN8102zGrcJgg4P6XVS4l39+l5aCEGGbauLP5W6
K99c42ku3QrlX2+KeDi+xBG2cEIsdSiXeQS/16S36ITclu4AADEAAAAAAAAA
-----END CERTIFICATE-----
```

Copy this text to a plain text file that can be moved to the server. It is best to use a program like Notepad or Vi since programs like Microsoft Word sometimes add invisible characters. Copy the file to the server and name it appropriately, for example public.crt, and place the file in the SSL certificate folder, /usr/local/apache/conf/ssl.crt/.

The easiest way to do this is to SSH to the server. Create and edit a new file by typing:

```
vi public.crt
```

Enter into Insert mode by hitting “i” and paste the contents of the certificate. Hit Escape to exit insert mode and type “:wq” to save the file and exit Vi.

The last step is to configure the httpd.conf file so that Apache can locate the certificates and keys. It is recommended that a backup of the configuration file is made before making any modifications.

```
cd /usr/local/apache/conf
cp httpd.conf httpd.orig
vi httpd.conf
```

To find the appropriate variables to change, scroll through the configuration file until getting to the Virtual Host section, or search for the variables by typing “/SSLCertificate” and hitting enter. Set the following variables to match the

names of the certificates that have just been copied to the server and save the configuration file.

SSLCertificateFile /usr/local/apache/conf/ssl.crt/public.crt
SSLCertificateKeyFile /usr/local/apache/conf/ssl.key/private.key

Reinitialize Apache and it will read the new configuration file.

```
/usr/local/apache/bin/apachectl stop
/usr/local/apache/bin/apachectl startssl
```

Test accessing the server again by going to https://yourserver.yourdomain.com/. It should return the SSL/TLS test web page and there should no longer be a prompt to accept the certificate. This is because the web browser now trusts the source of the third-party certificate.

Since there will only be encrypted traffic allowed to this website, disable Apache from listening on TCP port 80. Edit the httpd.conf file as shown below. Search for "Port 80" and comment it out with a pound sign (#). Directly below that, under "<IfDefine SSL>", comment out "Listen 80".

```
# Port: The port to which the standalone server listens. For
# ports < 1023, you will need httpd to be run as root initially.
#
#Port 80

##
##  SSL Support
##
##  When we also provide SSL we have to listen to the
##  standard HTTP port (see above) and to the HTTPS port
##
<IfDefine SSL>
#Listen 80
Listen 443
</IfDefine>
```

After restarting the Apache service, Apache will only respond to web requests via HTTPS.

Configuring Apache to Act as a Reverse-Proxy Server

Configuring Apache to act as a reverse proxy server is very simple. Web traffic should be allowed from the reverse-proxy server to the back-end web server (OWA) on the internal network on TCP port 80.

The configuration file only requires an additional two statements, ProxyPass and ProxyPassReverse [6]. Place these variables under the *SSL Virtual Host Context* section of the httpd.conf file to keep it close to the rest of the SSL configuration settings. Do not turn on any of the other proxy settings or variables in the *Proxy Settings* section.

Using the predefined IP addresses for this document, the new statements should look something like this:

ProxyPass / http://10.250.1.150/
ProxyPassReverse / http://10.250.1.150/

ProxyPass allows remote servers to be mapped from one local server. To serve our purpose, anyone going to https://yourserver.yourdomain.com/ will now actually see the web pages from http://10/250.1.150/ (the internal OWA server).

ProxyPassReverse helps for the return traffic. It will make the return traffic appear that it is coming from https://yourserver.yourdomain.com/.

Save the httpd.conf file and restart the Apache web server.

```
/usr/local/apache/bin/apachectl stop
/usr/local/apache/bin/apachectl startssl
```

Now test the server again. Now browse to https://yourserver.yourdomain.com/, the test SSL/TLS web page should no longer be seen. The default web page for http://10.250.1.150/ should now be displayed. Note that the back-end web server is not referenced anywhere within the HTML headers.

Adding RSA SecurID Authentication to the Reverse-Proxy

Since the preferred location of the reverse-proxy server is on a secured DMZ of the firewall, allow access from the reverse-proxy server to the RSA ACE/Server on UDP port 5500.

Key points to remember:

- The reverse-proxy server must be resolvable by an internal DNS server. The RSA web agent issues cookies to the clients, which are based on a domain suffix. The reverse-proxy server must resolve to something like yourserver.yourdomain.com.
- The /etc/hosts file on the reverse-proxy server must have the proper address listed for the server. If it has any other IP address, such as 127.0.0.1 listed, the RSA web agent will NOT work.
- It is important to remember that the current version of RSA web agent (version 5.1) is only supported by RSA with Apache 1.3.26.

When adding the reverse-proxy server to the ACE/Server, add the server name that is fully resolvable by the internal DNS or add it to the hosts file of the ACE/Server. On the ACE/Server the Agent Host (the reverse-proxy server) will need to be defined as a Unix Server.

Copy the sdconf.rec file from the RSA ACE/Server to a floppy and take it to the reverse-proxy server. Copy the sdconf.rec file to the reverse-proxy server and do the following:

```
mkdir /floppy
mount -t auto /dev/fd0 /floppy
mkdir /var/ace
chmod 774 /var/ace
cp /floppy/sdconf.rec /var/ace/.
chmod 644 /var/ace/sdconf.rec
umount /floppy
```

Change permissions as directed above in order for the RSA ACE/Server to communicate properly with the reverse-proxy server.

It is important to remember that RSA does not support the use of Web Agent 5.1 with Apache 1.3.27. However, it is critical to remain current with software packages such as Apache because newer versions fix vulnerable flaws in older versions. RSA has changed the way the Installer script works with Web Agent 5.1. When the Installer runs, it checks the version of Apache and if it is not the correct version, the installation aborts.

This can be modified by editing the "install" file with the /home/download/rsaweb folder. Scroll through the file until getting to the section titled "Check to see if this is a supported webserver platform." There will be a line that has a version of Apache listed that will look like:

1.3.26|1.3.22) ;;

Change 1.3.26 to read 1.3.27 and save the file. Running the installation should not abort.

Before continuing with the installation of the RSA web agent, stop the Apache web service.

```
/usr/local/apache/bin/apachectl stop
cd /home/download/rsaweb
./install
```

During the installation:

- Choose Apache
- Agree to the EULA (after reading it)
- Accept the conditions (type Accept)

- Accept the default directory location of sdconf.rec ([/var/ace])
- Accept the default path to the Apache install ([/usr/local/apache])
- Accept the default path to the httpd.conf file ([/usr/local/apache/conf/httpd.conf])
- Accept the default path to the httpd binary ([/usr/local/apache/bin/httpd])

All files for the RSA web agent will install into /usr/local/apache/rsawebagent.

Setup will begin immediately following installation. Follow the onscreen instructions. All of the default parameters are acceptable, however, for an easier time with the use of OWA:

- On the first screen, change "Expiration time for idle cookie in minutes" from 15 to 45. This will increase the time before the end user is prompted to re-authenticate. If the time is too short and a user takes an extremely long time to write an email, it may fail because their time expired.
- On the second setup screen, change "Use separate Page for username and PASSCODE" from *enabled* to *disabled*. This makes logging in simpler for the end user.
- On the second setup screen, change " Attempt to prevent Browser to cache protected pages" from *enabled* to *disabled*. This is necessary for users to download attachments.

If following the changes above, the two set up screens should look similar to the following:

Figure 3 (Screen 1):

```
SETUP [ default ]

 1.     Expiration time for idle cookie in minutes (1-1440)  [45]
 2.     Expiration time for cookie in minutes (1-1440)       [60]
 3.     SSL Port number to be used.  (80 443 ) [443]
 4.     WebID URL/URI name  [/webauthentication]
 5.     Directory for Web authentication templates
            [/usr/local/apache/rsawebagent/Templates]

 0.     Reset to previous values
```

Figure 4 (Screen 2):

```
CONFIGURATION [ default ]

 1.     Agent protection of this web server                    [enabled]
 2.     Use RSA ACE/Server 5.0 name locking feature            [enabled]
 3.     Use separate Page for username and PASSCODE            [disabled]
 4.     Require SSL to access protected resource              [enabled]
 5.     Redirect to SSL when accessing protected resource     [enabled]
 6.     Attempt to prevent Browser to cache protected pages   [disabled]
 7.     Auto Submit (avoid having to click Continue)          [enabled]
 8.     Use Java Script Popup window to authenticate in frames [disabled]
 9.     Ignore browser's IP Address to sign  the cookie       [disabled]
10.     Cookie valid for current Domain                       [disabled]
11.     Cookie valid across multi-domain                      [disabled]

 0.     Reset to previous values
```

Note that RSA recommends using separate username and PASSCODE web pages. This is discussed in the documentation that comes with the web agent install.
The next step is to test authentication to the ACE/Server in order to verify that the reverse-proxy is communicating properly with the RSA ACE/Server.

```
cd /usr/local/apache/rsawebagent
./acetest
```

The following will be displayed:

- Enter USERNAME: {enter the RSA SecurID username}
- Enter PASSCODE: {enter the PIN followed by the TOKENCODE}

This test will create two files, “securid” and “sdstatus.{x}”, in the /var/ace directory. The “securid” file is the node verification file. By default, it does not have the correct permissions for subsequent authentication tests. Change the permissions on this file.

```
chmod 440 /var/ace/securid
```

Run a second test authentication to verify that it still works.

```
./acetest
```

Monitor the ACE/Server Activity Log to see if the test is successful and there are no node verification failures on the RSA ACE/Server. Test to see if authentication works with Apache.

```
/usr/local/apache/bin/apachectl startssl
```

The following error may occur when starting the web server, after installing the RSA web agent:

“[DATE][warn] module mod_rsawebagent.c is already added, skipping”

Everything will still start and work properly; this message is just a nuisance. To get rid of this error, edit the RSA web agent configuration file.

```
vi /usr/local/apache/rsawebagent/rsawebagent.conf
```

The following is the default configuration:

```
#
# RSA Web Agent configuration information
# This file is included by the current httpd.conf file
#
# Load and add the web agent module in the configuration
LoadModule rsawebagent_module /usr/local/apache/rsawebagent/mod_rsawebagent.so
AddModule mod_rsawebagent.c
#
# RSA Web Agent installation directory
#
<IfModule mod_rsawebagent.c>
RSAWebAgentInstallPath /usr/local/apache/rsawebagent
VAR_ACEPath /var/ace
</IfModule>
```

To correct the error, place a pound sign (#) in front of the following line to comment it out.

AddModule mod_rsawebagent.c

Stop and start the Apache service. The previous error should no longer occur.

```
/usr/local/apache/bin/apachectl stop
/usr/local/apache/bin/apachectl startssl
```

Browse to https://yourserver.yourdomain.com with an Internet browser. When the web site is displayed, it should show a new web page for RSA SecurID authentication. Enter the username and PASSCODE in the appropriate fields and click submit. It should come back and say that authentication was successful and then present the content from the back-end OWA server.

Conclusion

As businesses become more dependent on the Internet and users require access to company resources such as email, securing that access becomes critical. There is a fine line between securing that access and making it unusable.

There are two lines of thought with respect to security. The first is that, if implemented correctly, security always comes at a high cost. The second is that implementing security measures will render a system or environment unusable.

It is often difficult to find the balance between proper security and usability. OWA is not inherently secure and no matter what steps are taken to secure it, there are still “doorways” that must remain open for it to function properly.

Many companies place servers on the Internet without giving much thought to security. This approach allows authorized users access to what they need, but it

also puts the company at risk of intrusion. This could cost the company much more than if they had initially invested in security up front.

All of the software required to configure the reverse-proxy server as presented in this document, except the certificate, is freeware and is readily available on the Internet. The certificate will vary in cost depending on the type of certificate purchased. This paper shows that taking additional steps up front to secure a resource does not necessarily require a large monetary investment, just a little time and planning.

References

[1] Tatham, Simon. "PuTTY FAQ." 16 February 2003 URL: http://www.chiark.greenend.org.uk/~sgtatham/putty/faq.html (17 February 2003).

[2] Toxen, Bob. Real World Linux Security. Upper Saddle River: Prentice Hall, 2003.

[3] Thawte. "Key and CSR Generation Instructions." Software Support - Apache+mod_ssl Key and CSR Generation. URL: http://www.thawte.com/html/SUPPORT/server/softwaredocs/modssl.html (20 February 2003).

[4] Apache. "FAQ." 11 February 2003 URL: http://www.apache-ssl.org/#FAQ (12 February 2003).

[5] Verisign. "Installing a Secure Site Server Pro (Global) ID Or Commerce Site Pro ID on an Apache Server with modssl." URL: http://www.verisign.com/support/install/apache/v00Mod.html#global (18 February 2003).

[6] Apache Software Foundation. "Apache module mod_proxy." Apache HTTP Server Version 1.3.
URL: http://httpd.apache.org/docs/mod/mod_proxy.html (12 February 2003).

Red Hat. "How to Download Red Hat Linux." URL: http://www.redhat.com/download/howto_download.html (08 February 2003).

National Institute of Standards and Technology (NIST). "Password Guidelines: Tips on how to create a secure password!" URL: http://csrc.nist.gov/organizations/fissea/presentations/2000/passwrd-guide.doc (08 February 2003).

Sendmail. "Securing Sendmail." URL: http://www.sendmail.net/000705securitygeneral.shtml (09 February 2003).

Thawte. "Thawte's Step by Step Guide to Certificate Enrolments." URL: http://www.thawte.com/guides/StepByStepEnrolmentSSLSGC.doc (20 February 2003).

Sawall, Chris. "HOWTO: Reverse Proxy using Apache with SSL and RSA SecurID® Authentication." Version 1.5. 03 December 2002. URL: http://tech.stlsawall.com/docs/rproxy.htm (10 February 2003).

Apache Software Foundation. "Security Tips for Server Configuration." Apache HTTP Server Version 1.3. URL: http://httpd.apache.org/docs/misc/security_tips.html (08 February 2003).

OpenSSH. "OpenSSH FAQ." 03 April 2002 URL: http://www.openssh.org/faq.html (08 February 2003).

RSA Security. "RSA ACE/Agent for Web." URL: http://www.rsasecurity.com/products/securid/techspecs/webagent.html (19 February 2003).

Andreasson, Oskar. "Iptables Tutorial 1.1.11". 2001 URL: http://www.iptables.org/documentation/tutorials/blueflux/iptables-tutorial.html (18 February 2003).

Last Updated: October 18th, 2017

Upcoming Training

SANS vLive - SEC401: Security Essentials Bootcamp Style	SEC401 - 201710,	Oct 23, 2017 - Nov 29, 2017	vLive
San Diego Fall 2017 - SEC401: Security Essentials Bootcamp Style	San Diego, CA	Oct 30, 2017 - Nov 04, 2017	vLive
SANS Seattle 2017	Seattle, WA	Oct 30, 2017 - Nov 04, 2017	Live Event
SANS San Diego 2017	San Diego, CA	Oct 30, 2017 - Nov 04, 2017	Live Event
SANS Gulf Region 2017	Dubai, United Arab Emirates	Nov 04, 2017 - Nov 16, 2017	Live Event
Community SANS Vancouver SEC401^	Vancouver, BC	Nov 06, 2017 - Nov 11, 2017	Community SANS
Community SANS Colorado Springs SEC401~	Colorado Springs, CO	Nov 06, 2017 - Nov 11, 2017	Community SANS
SANS Miami 2017	Miami, FL	Nov 06, 2017 - Nov 11, 2017	Live Event
SANS Sydney 2017	Sydney, Australia	Nov 13, 2017 - Nov 25, 2017	Live Event
SANS Paris November 2017	Paris, France	Nov 13, 2017 - Nov 18, 2017	Live Event
SANS San Francisco Winter 2017	San Francisco, CA	Nov 27, 2017 - Dec 02, 2017	Live Event
Community SANS St. Louis SEC401	St Louis, MO	Nov 27, 2017 - Dec 02, 2017	Community SANS
SANS London November 2017	London, United Kingdom	Nov 27, 2017 - Dec 02, 2017	Live Event
Community SANS Portland SEC401	Portland, OR	Nov 27, 2017 - Dec 02, 2017	Community SANS
SANS Khobar 2017	Khobar, Saudi Arabia	Dec 02, 2017 - Dec 07, 2017	Live Event
Community SANS Ottawa SEC401	Ottawa, ON	Dec 04, 2017 - Dec 09, 2017	Community SANS
SANS Austin Winter 2017	Austin, TX	Dec 04, 2017 - Dec 09, 2017	Live Event
SANS Munich December 2017	Munich, Germany	Dec 04, 2017 - Dec 09, 2017	Live Event
SANS vLive - SEC401: Security Essentials Bootcamp Style	SEC401 - 201712,	Dec 11, 2017 - Jan 24, 2018	vLive
SANS Bangalore 2017	Bangalore, India	Dec 11, 2017 - Dec 16, 2017	Live Event
SANS Cyber Defense Initiative 2017	Washington, DC	Dec 12, 2017 - Dec 19, 2017	Live Event
SANS Cyber Defense Initiative 2017 - SEC401: Security Essentials Bootcamp Style	Washington, DC	Dec 14, 2017 - Dec 19, 2017	vLive
Community SANS Nashville SEC401^	Nashville, TN	Jan 08, 2018 - Jan 13, 2018	Community SANS
Community SANS Hawaii SEC401	Honolulu, HI	Jan 08, 2018 - Jan 13, 2018	Community SANS
SANS Security East 2018	New Orleans, LA	Jan 08, 2018 - Jan 13, 2018	Live Event
Mentor Session - SEC401	Memphis, TN	Jan 09, 2018 - Mar 13, 2018	Mentor
Northern VA Winter - Reston 2018	Reston, VA	Jan 15, 2018 - Jan 20, 2018	Live Event
SANS Amsterdam January 2018	Amsterdam, Netherlands	Jan 15, 2018 - Jan 20, 2018	Live Event
Mentor Session - SEC401	Minneapolis, MN	Jan 16, 2018 - Feb 27, 2018	Mentor
Las Vegas 2018 - SEC401: Security Essentials Bootcamp Style	Las Vegas, NV	Jan 28, 2018 - Feb 02, 2018	vLive
SANS Las Vegas 2018	Las Vegas, NV	Jan 28, 2018 - Feb 02, 2018	Live Event