DEMYSTIFY ORACLE APPS 11i SECURITY IN THE DMZ

ERIC HERNANDEZ

ZANETT

Why Use Oracle Applications Over the Internet?

Internet is so pervasive in the business world that most companies must take full advantage of it to survive. Intranet applications are no longer sufficient and efficient. Intimate interaction between business entities such as vendors, customers, partners, and so on can be simplified over the internet. For example, a distributor can enable its self service ordering system application over the internet by deploying a web server solution. It allows the distributor's customers to place any order anytime and anywhere in the world from a desktop computer in Japan to an internet handheld device in Argentina. However, some companies have not fully embraced this business concept because they are paralyzed by the possible risks involved that may have been hyped by the news media. Compared to intranet-only applications, internet based applications have a significant number of possible attackers that would be very difficult to locate and punish. How would a company properly protect its internal network from the public internet? How can external attacks be detected and prevented? How should the application servers be secured? How is the transmission of data between the US and Argentina be protected? What needs to be deployed to protect the company's valuable database? Since some of these companies may not feel comfortable about the answers they get to these questions, they deploy systems that do not take full advantage of the latest Internet technology and may end up incurring some type of business process inefficiency as a result.

Oracle Certified Topology of 11i in the DMZ

Fortunately, preventing and minimizing risks for internet enabled applications have been greatly simplified with the use of Firewalls, Intrusion Detection System (IDS), Reverse Proxy (RP) server, SSL encryption, and using Oracle certified Internet application products such as iRecruitment, iSupplier, and so on. For simplicity's sake, network jargons and terminology will be minimized. Data transmission between the external users (see Figure 1) and the application is secured using SSL cryptography with a corresponding opened port in the firewall. A firewall can be either hardware or software based which is configured to block unauthorized access to a network. Firewalls can be used to divide a network into segments. One of these segments is known as the Demilitarized Zone (DMZ) which separates public network from the internal network. In Figure 1 copied from Steven Chan's OTN blog site, there are 4 firewalls. The first firewall starting from the Internet side of the diagram is the first layer of defense from any external attacks. Most sophisticated attackers may be able to get into "DMZ 1" but they will have to go through 3 more firewalls to get to the ultimate prize, the database. Even the most advanced external attackers will have a difficult time penetrating network security when an IDS has been deployed between the internet and the first firewall. Network attacks would be easily identified by their network signatures. When an IDS determines an attack is on-going, it can alert a network administrator or block the source IP address of the attacker. Even if the external attacker is able to compromise the first firewall, the only thing available in the first DMZ is an RP server. The RP chosen is from Apache Software Foundation because of its reputation, widely used in the industry, and certified with Oracle E-Business Suite in DMZ configuration. The RP server will be configured to determine whether an Internet request should be allowed to go through or not. Only the RP server is allowed to send requests to the external web server. Another benefit of an RP server is that it allows the external web server to run Apache on non-privilege ports; thus, avoiding the need for Apache to be started and owned by root. Because of RP, would be external attackers will not be able to communicate directly with the external web server. With RP, SSL encryption and decryption processing is offloaded from the external web server providing better performance for 11i Internet users. An external web server is used to limit the number of privileges or responsibilities to be available over the internet. Therefore, even if the most highly sophisticated external attackers are able to hijack the external web server, they're access to the data is still limited.

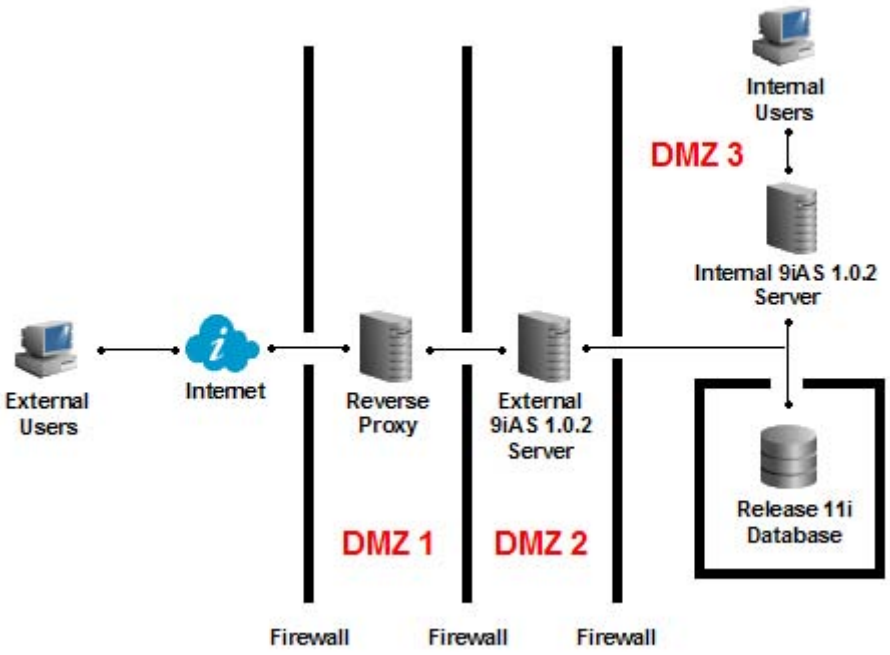

Figure 1 – Reverse Proxy with an External Web Server

There are other topologies that are fully supported by Oracle as listed below. A configuration that involves the RP and External Web Server is what we recommend when a load balancer is not involved because of the advantages mentioned earlier. If an alternative topology is configured that is not on this list, it would be supported only on a best-effort basis. Taken from Metalink Note: 287176.1, Figure 2 below shows a similar topology as Figure 1 but without the reverse proxy server. Depending on your business budget requirement, configuring an external web server only setup is one of the two least expensive setups. The security it provides is adequate but can still be improved. Since there's only one DMZ, the external web server is in jeopardy if security is breached. The external web server may have to be rebuilt from ground up depending on the severity of the damage. In this set up, 11i's Apache will have to run as root in order to use privileged port. This is a significant risk because any Apache weakness may be abused and be able to run as root directly in the external web server. The processing of SSL encryption and decryption will put a burden on the server. 11i components and services running on the same server will be affected. Consequently, phone calls from users will start coming in because of the poor performance.

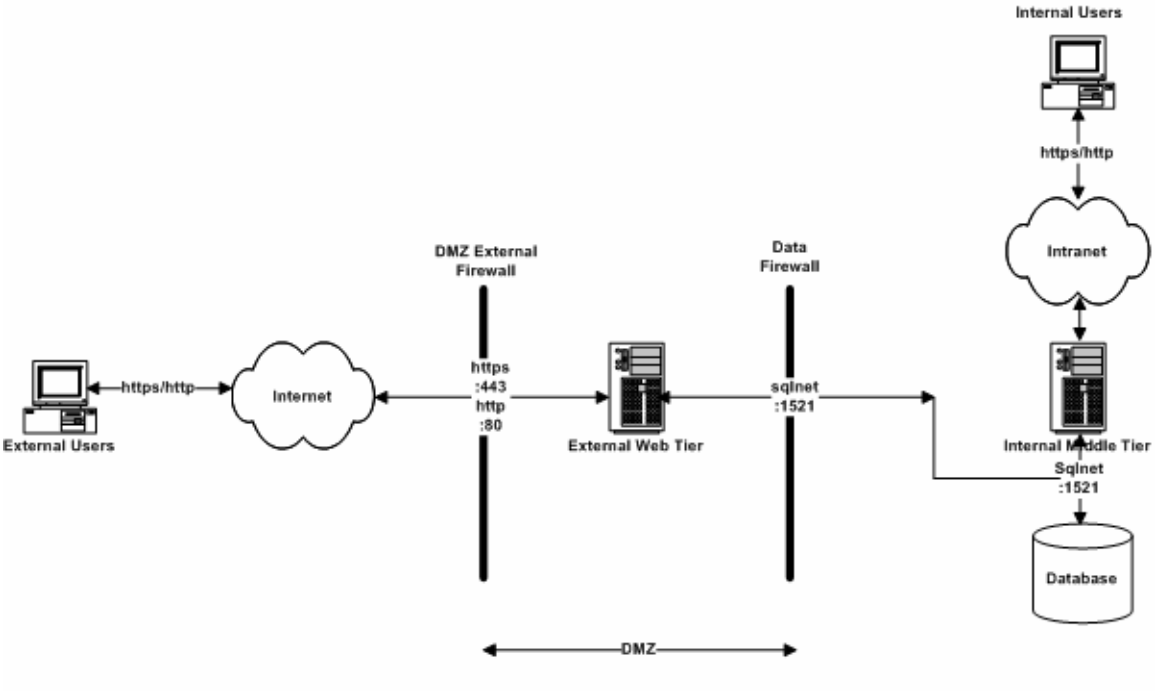

Figure 2 – External Web Server Only

Figure 3 below is also similar to Figure 1 without a physical external web server. However, the external web tier still exists along side with the internal web tier using the shared file system technology in the same physical internal server. Both external and internal tiers have different HTTP listeners and jserv processes. Obviously, the main disadvantage of this configuration is that if the internal application server crashes, both external and internal applications won't be accessible. Another disadvantage is if the external middle tier that resides in the same server as the internal is attacked with DoS from the Internet, then it will have a severe impact on the internal users' performance. The advantage of having a reverse proxy server is that there are no 11i Apache having to run as root. Only the non-11i Apache in the RP has to run as root. Additionally, if security is breached, fixing a damaged RP is easier, faster, and cheaper than having to deal with damaged 11i components in an external web server as illustrated in Figure 2. Finally, one significant purpose of an RP is to hide the identity of the 11i servers. A hacker can only attack once it has identified its target.

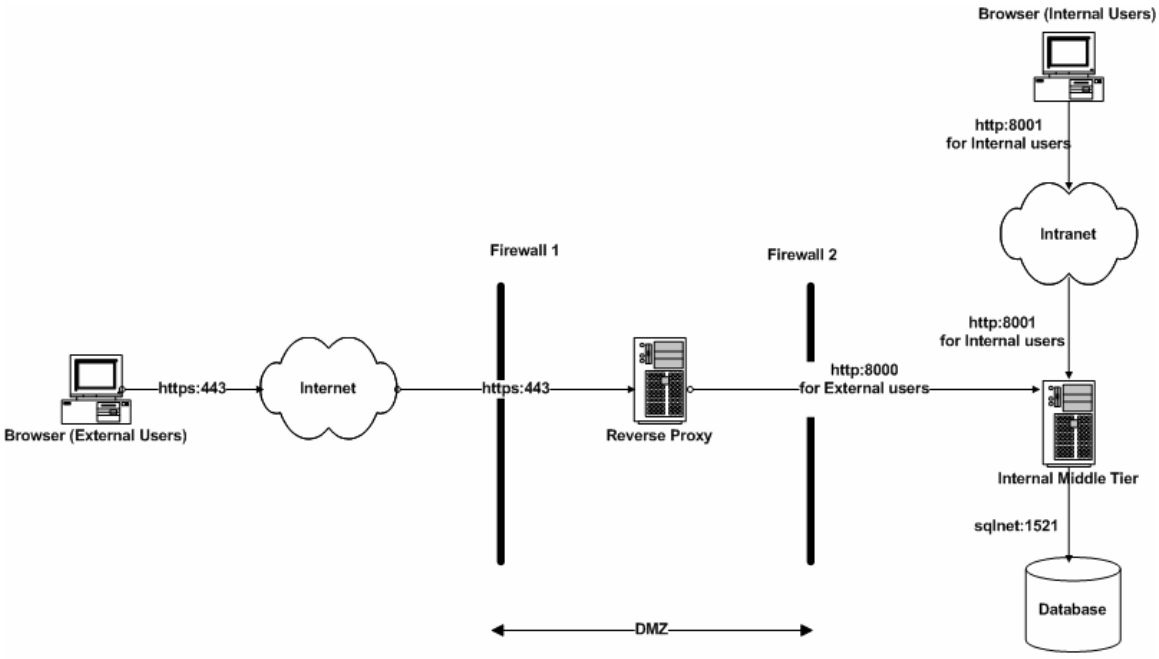

Figure 3 – Reverse Proxy Server Only

If redundancy is needed, Figure 4 would be highly recommended. If redundancy is needed with the least expensive setup, then Figure 5 would be the obvious choice.

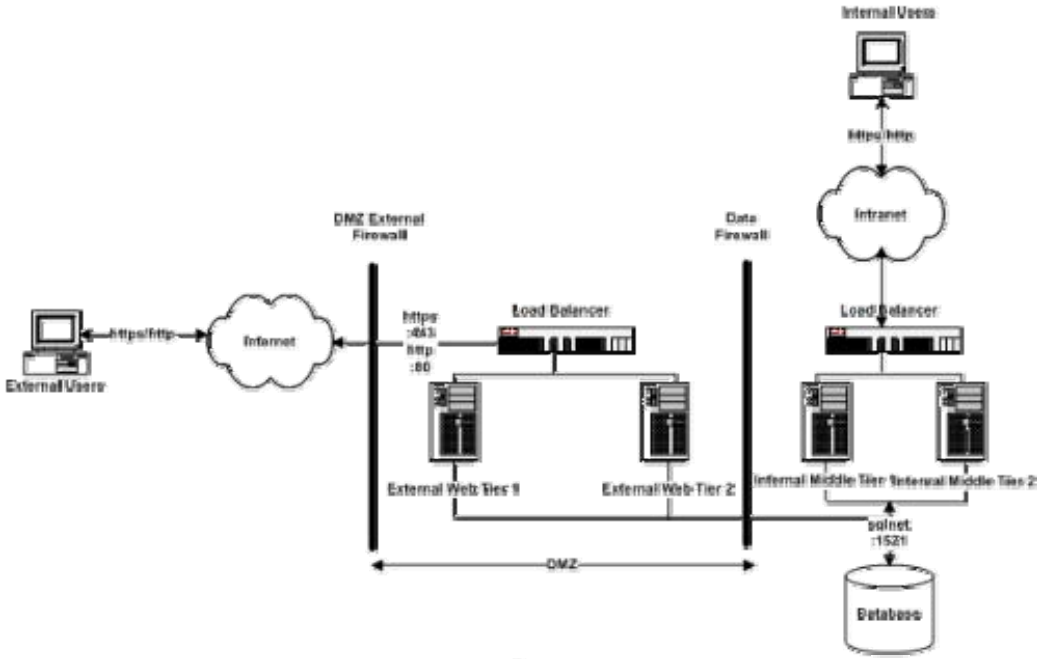

Figure 4 – Hardware Load Balancers with an External Web Server

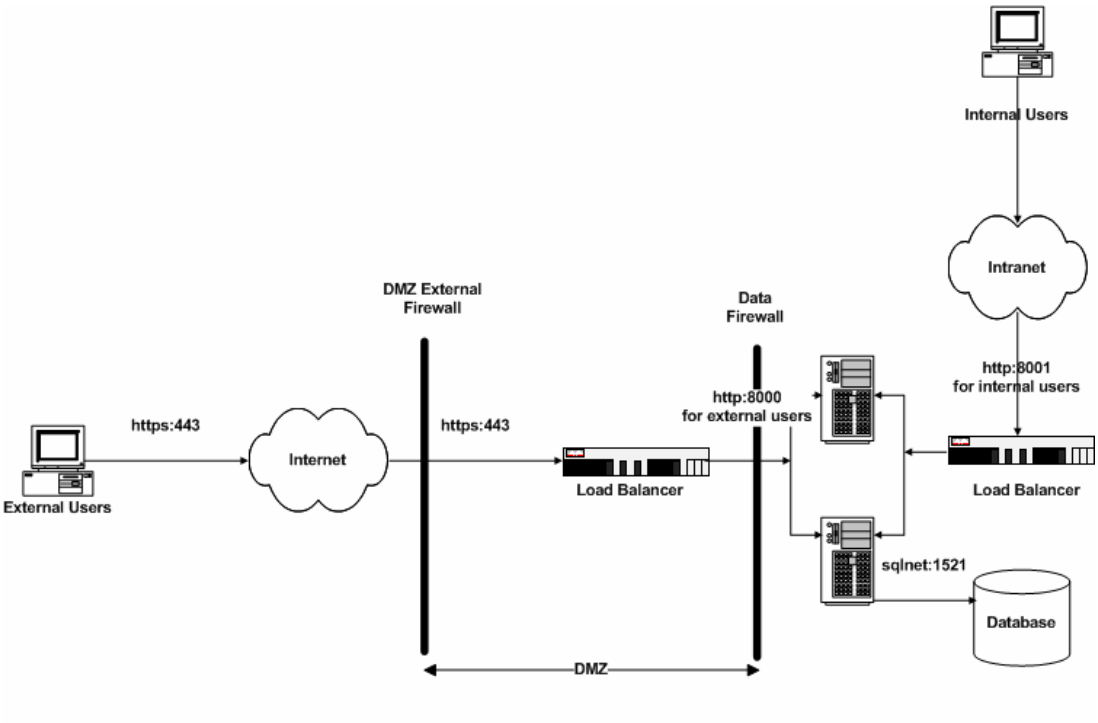

Figure 5 – Hardware Load Balancers without an External Web Server

Configure and Secure 11i iRecruitment in the External Web Server

iRecruitment is the selected 11i product to be demoed in this white paper and made available over the Internet using the reverse proxy with an external web server configuration shown in Figure 1. During the preparation stages, working side-by-side with a network administrator is a must. Past experiences have shown numerous delays caused by an Apps DBA not being on the same page as the network administrator or vice versa. Before proceeding on with the actual Oracle 11i internet product configuration, the communications between servers on specific ports must be verified.

Once the hardware and proper communications are in place, the following detailed steps must be completed on an existing Oracle Applications System to configure iRecruitment for public internet use. A list of certified Oracle E-Business Suite modules for external deployment is listed in Metalink Note: 287176.1. Please do not implement a module that is not certified for external deployment without creating an Oracle SR first. In this demonstration, a reverse proxy and external web server with a three DMZ setup will be used as shown in Figure 1.

The steps below make the following assumptions with regards to the fully qualified domain name (FQDN) of the servers and website name:

FQDN of Database Server: db.example.com
FQDN of Internal Application Server: intweb.example.com
FQDN of External Web Server: extweb.example.com
FQDN of Reverse Proxy Server: rp.example.com
Web Site Name: irecruitment.example.com

Preparation

1. Apply Patches to 11i of intweb.example.com, if applicable
    a. Patches Required for DMZ Configuration using 11i10
        i. 3240000
        ii. 4204335
        iii. 3942483
        iv. 5478710
    b. Patches Required for DMZ Configuration using 11i9
        i. 3072811
        ii. 4334965
        iii. 3942483
        iv. 5478710
    c. iRecruitment Patches
        i. 4242220
        ii. iRecruitment is used in this demonstration. If iRecruitment is not the module being implemented, please review the patches necessary in Appendix A of Metalink Note: 287176.1
    d. Rapid Clone Patches
        i. In order to minimize cloning issues, ensure the latest rapid clone prerequisites have been met

2. Clone Oracle Apps from intweb.example.com to extweb.example.com using Rapid Clone
    a. Sharing file systems between the external web server and internal middle tiers is not supported in any deployment option
    b. Obviously, if you have multiple external web servers, then sharing file systems is allowed

Configuring Oracle 11i in DMZ

1. Update Hierarchy Type
    a. In order to have two Oracle Applications Mid-Tier behave differently, extweb.example.com and intweb.example.com, the hierarchy type must be set to SERVRESP
        i. SERVRESP provides the ability to limit the privileges and functionality available in extweb.example.com
    b. Stop all services in extweb.example.com and intweb.example.com
    c. Execute the following SQL to change the profile options hierarchy type values to SERVRESP

sqlplus <apps_schema>/<apps_pwd> @FND_TOP/patch/115/sql/txkChangeProfH.sql SERVRESP

    d. Output of the sql will indicate if the execution is successful
    e. After a successful execution, run autoconfig in all nodes

2. Update Node Trust Level
    a. Start all services in intweb.example.com only
    b. In order to take advantage of the new SERVRESP hierarchy type, extweb.example.com must be tagged with a trust level value of "External" by doing the following:
        i. Login to Oracle E-Business Suite as sysadmin user using the internal URL
        ii. Select System Administrator Responsibility
        iii. Select Profile / System
        iv. From the "Find system profile option Values" window, select "extweb.example.com" as the server that you want to make external
        v. Query for "Node Trust Level".
            1. See Figure 6 below
        vi. Set the value of this profile option to External at the server level (not site level). The site-level value should remain Normal.
            1. See Figure 7 below

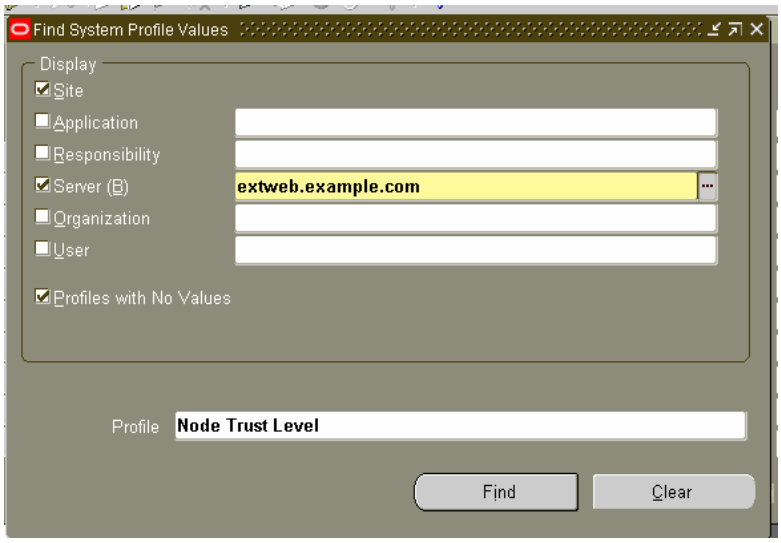

Figure 6 – Query for "Node Trust Level" Profile for extweb.example.com Server

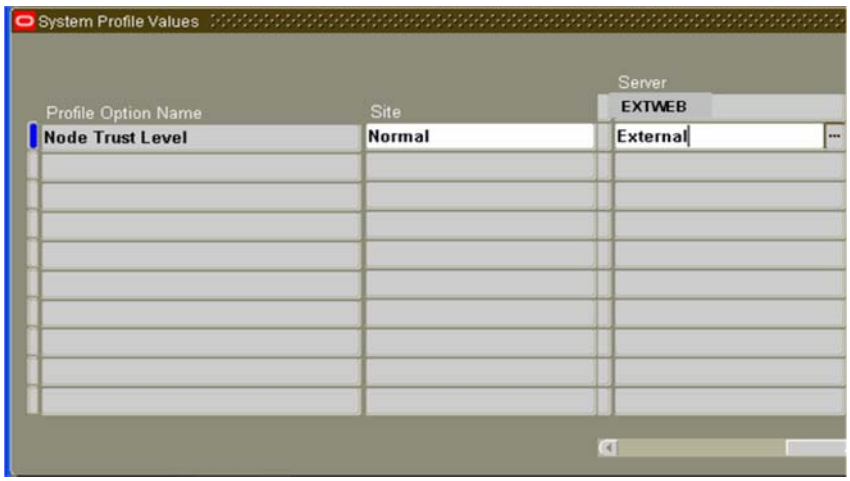

Figure 7 – Setting "Node Trust Level" Profile to "External" for "extweb.example.com" Server

3. Update Home Page Mode to Framework
    a. Login to Oracle E-Business Suite as sysadmin user using the internal URL
    b. Select System Administrator Responsibility
    c. Select Profile / System
    d. From the 'Find system profile option Values' window, query for %HOME%MODE%. You will see a profile option named '**Self Service Personal Home Page Mode**' , set the value of this profile option to **Framework Only.**
4. Configuring iRecruitment Responsibilities for extweb.example.com
    a. Login to Oracle E-Business Suite as sysadmin user using the internal URL
    b. Select System Administrator Responsibility
    c. Select Profile / System
    d. From the 'Find system profile option Values' window, select the iRecruitment responsibility that you want listed below to make external one at a time
        i. iRecruitment External Site Visitor
        ii. iRecruitment External Candidate
        iii. iRecruitment Employee Site Visitor
        iv. iRecruitment Employee Candidate
        v. iRecruitment Agency
    e. Query for "Responsibility Trust Level"
        i. See Figure 8 below
    f. Set the value of this profile option for the chosen responsibility to **External** at responsibility level (not site level). The site-level value should remain **Normal**.
        i. See Figure 9 below

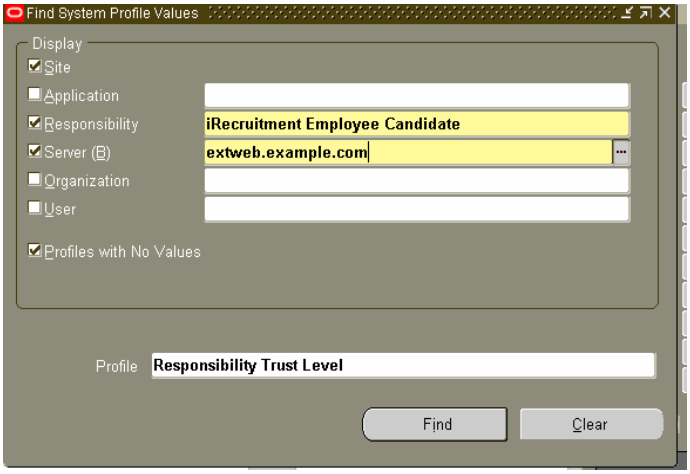

Figure 8 – Query for "Responsibility Trust Level" Profile for "extweb.example.com" Server and "iRecruitment Employee Candidate" Responsibility

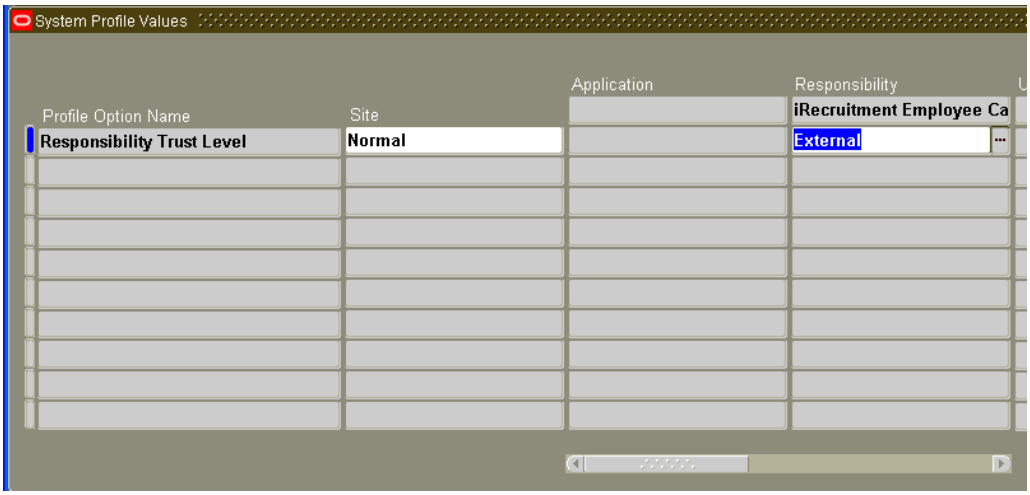

Figure 9 – Setting "Reponsibility Trust Level" Profile to "External" for "iRecruitment Employee Candidate" Responsibility

5. Ensure that authentication is not needed to get to the iRecruitment Visitor's home page
    a. Login to Oracle E-Business Suite as sysadmin user using the internal URL
    b. Select System Administrator Responsibility
    c. Navigate to Security > User > Define
    d. Query GUEST user name
    e. Add the following responsibilities to the GUEST account
        i. iRecruitment Employee Candidate
        ii. iRecruitment External Candidate

6. Ensure that the iRecruitment Visitors don't get the "The iRecruitment Application is not currently installed. Please contact your Oracle Representative" by doing the following
    a. Set the profile option 'IRC: Installed' to Yes at the site level.

Build Apache and Configure Reverse Proxy Server

1. Download Apache 2.0.54 src code from http://httpd.apache.org/download
    a. Download .tar.gz and its corresponding .md5

2. Download modsecurity-1.8.7 from http://www.modsecurity.org/download
    a. Download .tar.gz and its corresponding .md5

3. Verify the integrity of the file by executing MD5 checksum
    md5sum -c httpd-2.0.54.tar.gz.md5
    md5sum -c modsecurity-1.8.7.tar.gz.md5
        i. Ensure that the command does not produce an output

4. Unpack the Apache and mod_security tar files
    tar xzvf httpd-2.0.54.tar.gz
    tar xzvf modsecurity-1.8.7.tar.gz

5. Configure Apache
    cd httpd-2.0.54

    ```
    ./configure --prefix /dmz  --enable-ssl --enable-setenvif --enable-proxy --enable-proxy_http \
    --enable-headers --enable-rewrite --enable-so --disable-charset-lite --disable-include \
    --disable-env --disable-status  --disable-autoindex  --disable-asis --disable-cgi \
    --disable-negotiation  --disable-imap --disable-actions --disable-userdir --disable-alias
    ```

        i. prefix determines where Apache will be installed later

6. Ensure that mod_proxy does not proxy a request to the external web tier before the URL firewall based on mod_rewrite has a chance to reject it. From the downloaded source directory, do the following steps to change the NULL value to aszSucc

    cd modules/proxy
    vi mod_proxy.c

    Change the following parameter's value from:

    ap_hook_translate_name(proxy_trans, NULL, NULL, APR_HOOK_FIRST);

    to:

    ap_hook_translate_name(proxy_trans, **aszSucc** , NULL, APR_HOOK_FIRST);

7. Compile Apache from the main directory

    cd ../..
    make

8. Check that the expected modules are included and no others

    ./httpd -l
        i. httpd –l command should produce the following ouput

    Compiled in modules:
    core.c
    mod_access.c

mod_auth.c
mod_log_config.c
mod_headers.c
mod_setenvif.c
mod_proxy.c
proxy_http.c
mod_ssl.c
prefork.c
http_core.c
mod_mime.c
mod_dir.c
mod_rewrite.c
mod_so.c

9. Install Apache as root in /dmz

```
umask 022
make install
chown –R root:sys  /dmz
```

10. Install mod_security
    Change directory to modsecurity-1.8.7
    cd apache2
    /dmz/bin/apxs -cia mod_security.c

11. Edit httpd.conf and remove the following directives
    a. UserDir
    b. Alias
    c. AliasMatch
    d. RedirectMatch
    e. ScriptAlias
    f. IndexOptions FancyIndexing VersionSort
    g. AddIconByEncoding
    h. AddIconByType
    i. AddIcon
    j. DefaultIcon
    k. ReadmeName
    l. HeaderName
    m. IndexIgnore
    n. LanguagePriority
    o. ForceLanguagePriority

12. Start Apache
    /dmz/bin/apachectl start

13. Test Apache
    a. Go to http://<your site name>/index.html.en
        i. Since mod_negotiation or mod_dir is not included in the build of Apache, you'll have to include the language (such as "en")

14. Shutdown Apache
    /dmz/bin/apachectl stop

Set Up SSL for Reverse Proxy

1. Generate and install a test certificate

```
cd /dmz/conf
umask 022
mkdir ssl.key
mkdir ssl.crt

openssl req -new -x509 -days 30
-keyout ssl.key/server.key
-out ssl.crt/server.crt
-subj '/CN=Test-Only Certificate'

chmod 600 ssl.key/server.key
```

2. Start Apache with SSL

```
/dmz/bin/apachectl startssl
```

3. Try accessing Apache with https://<your site name>/index.html.en and http://<your site name>/index.html.en
    a. Since only a test certificate has been configured, please expect to be warned by the browser that you're dealing with an unrecognized Certificate Authority (CA)
    b. For your real deployment, purchase your SSL certificate from a CA

Key Apache Modules

Configuring security with Apache requires taking full advantage of Apache distributed modules listed below:

- mod_rewrite
    - Uses the URL Rewriting Engine as a URL firewall
        - Used for URL manipulation
            - If a URL is accessing via http, it will be rewritten as https to maintain security
            - If a URL being accessed from Apache is not a known URL, it will be rejected by the ReWriteRule
    - The most pertinent Apache Directive in httpd.conf is ReWriteRule
- mod_ssl
    - provides strong cryptography using the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols
    - The most pertinent Apache Directive in httpd.conf are
        - SSLCertificateFile
        - SSLCertificateKeyFile
- mod_proxy and mod_proxy_http
    - hides the Oracle Apps external web tier
    - The most pertinent Apache Directive in httpd.conf are
        - ProxyPass
        - ProxyPassReverse
- mod_security
    - Behaves as the Web Application Firewall
    - It discovers and blocks requests that are suspicious and intentionally malformed to launch an attack
        - Rejects bad requests before anything else happens

In order to expedite your configuration process to secure Apache, two files have been provided in the Appendices located at the end of this document that contains the security features discussed above. Appendix A and B shows a functioning reverse proxy httpd.conf and security.conf, respectively, with all of the necessary security

configurations. You will have to make modification to these files because they were created with the following assumptions:

1. the reverse proxy will be accessed via the hostname irecruitment.example.com
2. the E-Business Suite external webtier is called extweb.example.com
3. the server admin is webmaster@example.com
4. the apache proxy was configured and installed to /dmz
5. iStore is not a product that is used

Enable URL Firewall

Oracle Apps middle tier comes with a disabled URL firewall. The file name is url_fw.conf and is located in $IAS_ORACLE_HOME/Apache/Apache/conf. httpd.conf has a commented entry for url_fw.conf. The URL firewall file contains a whitelist of URLs. Any URL that is requested from Apache that is not matched in the whitelist is refused. Consequently, attackers can only get to areas that are already secure.

In order to enable URL Firewall, do the following

1. copy url_fw.conf from $IAS_ORACLE_HOME of an APPL TIER and paste in the /dmz/conf of the Reverse Proxy Tier
2. Edit httpd.conf and uncomment the only Include directive for url_fw.conf
3. Restart Apache

Allow iRecruitment requests through the URL firewall

1. Edit the /dmz/conf/url_fw.conf
2. Uncomment the following

```
RewriteRule  ^/OA_HTML/IrcVisitor\.jsp$  - [L]
RewriteRule ^/pls/[^/]*/irc_web.show_vacancy$ - [L]
RewriteRule ^/OA_HTML/JobPositionSeeker\.xsl$ - [L]
RewriteRule ^/OA_HTML/IRCRESUMEUK1\.xsl$ - [L]
RewriteRule ^/OA_HTML/IRCRESUMEUK2\.xsl$ - [L]
RewriteRule ^/OA_HTML/IRCRESUMEUS1\.xsl$ - [L]
RewriteRule ^/OA_HTML/IRCRESUMEUS2\.xsl$ - [L]
RewriteRule ^/OA_HTML/IRCRESUMEUS3\.xsl$ - [L]
```

3. Restart Apache

If you desire to have the default 11i page to be the iRecruitment Visitor page, do the following

1. Edit the /dmz/conf/url_fw.conf
2. Uncomment the following

```
RewriteRule ^/$ /OA_HTML/IrcVisitor.jsp [R,L]
```

3. Comment the following

```
RewriteRule ^/$ /OA_HTML/AppsLocalLogin.jsp [R,L]
```

4. Restart Apache

Appendices

Appendix A

#Fully functioning httpd.conf for an Apache configured for Reverse Proxy
Configuration for a Reverse Proxy (with http/80 https/443, mod_security and url firewall)

User nobody

```
Group #-1

ServerAdmin webmaster@example.com
ServerName irecruitment.example.com:80
UseCanonicalName Off

Timeout 300
KeepAlive On
MaxKeepAliveRequests 100
KeepAliveTimeout 15

<IfModule prefork.c>
StartServers 5
MinSpareServers 5
MaxSpareServers 10
MaxClients 150
MaxRequestsPerChild 300
</IfModule>

Listen 80

# Load our only DSO
LoadModule security_module modules/mod_security.so

#
# Files (internal verbosity)
HostnameLookups Off
LogLevel warn
ServerRoot "/dmz"
PidFile logs/httpd.pid
ErrorLog logs/error_log
LogFormat "%h %l %u %t \"%r\" %>s %b" common
CustomLog logs/access_log common

# How verbose to be (externally)
ServerTokens Prod
ServerSignature Off

BrowserMatch "Mozilla/2" nokeepalive
BrowserMatch "MSIE 4\.0b2;" nokeepalive downgrade-1.0 force-response-1.0
BrowserMatch "RealPlayer 4\.0" force-response-1.0
BrowserMatch "Java/1\.0" force-response-1.0
BrowserMatch "JDK/1\.0" force-response-1.0

# Local files serving; we don't really need any; we'll proxy all requests
DocumentRoot "/dmz/htdocs"
DirectoryIndex index.html index.html.en

# The below 2 virtualhosts will inherit all general settings - unless overwritten
# by similar rules in the VH container.

<VirtualHost *:80>
  ServerName irecruitment.example.com:80

  ## Include mod_security directives
  Include conf/security.conf
```

```
  ## Redirect to https or proxy a subset of iStore's URLS
  RewriteEngine   On
  RewriteLogLevel 1
  RewriteLog      logs/rewrite_log

  ## Unless you run iStore, redirect all HTTP requests to HTTPS.
  RewriteRule ^/(.*) https://irecruitment.example.com/$1 [R,L]

  ## If you run iStore:
  # Comment out the above redirect and enable the below block of lines
  #Include conf/url_fw_iStore_http.conf
  # If the rewrite engine has not bee switched off by a rule in url_fw.conf
  #RewriteRule ^/(.*) https://irecruitment.example.com/$1 [R,L]
  #
  #ProxyRequests     Off
  #ProxyPreserveHost On
  #ProxyPass         / http://extweb.example.com:8000/
  #ProxyPassReverse  / http://extweb.example.com:8000/
  ## end iStore HTTP block
</VirtualHost>

# Include the SSL configuration (in a <VirtualHost *:443> container)

SSLRandomSeed startup builtin
SSLRandomSeed connect builtin
#SSLRandomSeed startup file:/dev/random 512
#SSLRandomSeed startup file:/dev/urandom 512
#SSLRandomSeed connect file:/dev/random 512
#SSLRandomSeed connect file:/dev/urandom 512

Listen 443

# Some MIME-types for downloading Certificates and CRLs
AddType application/x-x509-ca-cert .crt
AddType application/x-pkcs7-crl .crl

SSLPassPhraseDialog builtin

#SSLSessionCache none
SSLSessionCache dbm:logs/ssl_scache
SSLSessionCacheTimeout 300
SSLMutex file:logs/ssl_mutex

<VirtualHost _default_:443>

  # General setup for the virtual host
  ServerName irecruitment.example.com:443

  SSLEngine on
  SSLOptions +StrictRequire

  SSLProtocol -all +TLSv1 +SSLv3
  SSLCipherSuite HIGH:MEDIUM:!aNULL:+SHA1:+MD5:+HIGH:+MEDIUM

  # Server Certificate:
```

```
  SSLCertificateFile conf/ssl.crt/server.crt
  # Server Private Key:
  SSLCertificateKeyFile conf/ssl.key/server.key
  # SSL Protocol Adjustments:
  SetEnvIf User-Agent ".*MSIE.*" \
    nokeepalive ssl-unclean-shutdown downgrade-1.0 force-response-1.0
  # Per-Server Logging:
  #CustomLog logs/ssl_request_log "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"
  CustomLog logs/ssl_request_log common

  ## Include mod_security directives
  Include conf/security.conf

  ## Add the URL Firewall here
  RewriteEngine   On
  RewriteLogLevel 3
  RewriteLog      logs/rewrite_log

  # Allowed if a match is found, rejected otherwise
  Include conf/url_fw.conf

  ProxyRequests     Off
  ProxyPreserveHost On
  ProxyPass         / http://extweb.example.com:8000/
  ProxyPassReverse  / http://extweb.example.com:8000/

</VirtualHost>


# The error document for "410 Gone" used by the mod_rewrite based URL Firewall
ErrorDocument 410 "<HTML><HEAD><TITLE>410 Gone</TITLE></HEAD><BODY bgcolor=red>\
<H1>Gone</H1><BLOCKQUOTE>\
Access to the requested URI has been blocked by the URL Firewall.<p>\
If you believe that you have reached this page while performing valid operations \
within the application, please send mail to webmaster@example.com explaining \
what you were doing when you got this error.</BLOCKQUOTE><HR></BODY></HTML>"

# The error document for "400 Bad Request" used by mod_security
ErrorDocument 400 "<HTML><HEAD><TITLE>400 Bad Request</TITLE></HEAD><BODY bgcolor=red>\
<H1>Gone</H1><BLOCKQUOTE>\
Access to the requested URI has been blocked by security module.<p>\
If you believe that you have reached this page while performing valid operations \
within the application, please send mail to webmaster@example.com explaining \
what you were doing when you got this error.</BLOCKQUOTE><HR></BODY></HTML>"

#end of httpd.conf
```

Appendix B

```
#Fully functioning security.conf used by the httpd.conf above

<IfModule mod_security.c>
# Turn the filtering engine On or Off (everything is dynamic to a proxy)
SecFilterEngine DynamicOnly

# Log only matched (rejected) requests; set to Off to avoid logging
SecAuditEngine    RelevantOnly
```

```
SecAuditLog       logs/sec_audit_log

# Debug (default is 0) - do not enable this on a production system
SecFilterDebugLevel 0
SecFilterDebugLog  logs/sec_debug_log

# Should mod_security inspect POST payloads
SecFilterScanPOST On

# Default action is "400 - Bad Request"
SecFilterDefaultAction "deny,log,status:400"

# Character ranges... (avoid null bytes)
SecFilterForceByteRange 1 255

# Avoid directory traversal ../
SecFilter "\.\./"

# URL encodings
SecFilterCheckURLEncoding On

# Unicode character encodings - only enable if your EBS site uses UTF-8 encoding!
#SecFilterCheckUnicodeEncoding On

# Don't Allow TRACE method - or any of the DAV methods
SecFilterSelective REQUEST_METHOD "!(GET|HEAD|POST)" "deny,log,status:405"

# Only accept request encodings we know how to handle. We exclude GET requests
# from this because some (automated) clients supply "text/html" as Content-Type
SecFilterSelective REQUEST_METHOD "!^GET$" chain
SecFilterSelective HTTP_Content-Type "!(^$|^application/x-www-form-urlencoded$|^multipart/form-data)"
# If you use Forms use this instead (also accept application/octet-stream):
#SecFilterSelective HTTP_Content-Type "!(^$|^application/octet-stream$|^application/x-www-form-
urlencoded$|^multipart/form-data)"

# Require Content-Length to be provided with every POST request
SecFilterSelective REQUEST_METHOD "^POST$" chain
SecFilterSelective HTTP_Content-Length "^$"
# Test Pattern "666666" - Remove after testing!
SecFilter "666666"
</IfModule>
```

References

- Metalink Note 287176.1
  - DMZ Configuration with Oracle E-Business Suite 11i
  - Review the note for items that pertains to you

- Metalink Note 373837.1
  - Oracle iRecruitment Implementation and User Guide