# Global Information Assurance Certification Paper

# *Securing Outlook Web Access 2000*

Robert Newhall
January 27, 2003
GSEC Security Essentials Practical Assignment Version 1.4b Option 1

## Introduction

Microsoft Outlook Web Access 2000 (OWA) is a standard component of Microsoft Exchange 2000 and installs by default when Exchange is installed. It allows users to access their Exchange accounts including their mailboxes, calendars, and contact lists from a web browser. The obvious advantage to this is that no client software, other than a standard web browser, is required on the user end. This eliminates the need for any specific VPN or dial-up access for employees who travel or telecommute.

Often, increased features and flexibility of an application lead to reduced levels of security. This rule holds true for OWA as well. The default installation of OWA relies on insecure protocols, authentication methods, and Microsoft services. This paper will examine the security vulnerabilities inherent in OWA as well as methods for configuring OWA to mitigate these risks.

## OWA Architecture

Outlook Web Access for Exchange 2000 is significantly different from the previous incarnation of OWA in Exchange 5. It no longer uses Messaging Application Programming Interface (MAPI), Collaboration Data Objects (CDO), or Active Server Pages (ASP). Instead, OWA relies on HTTP and JavaScript to render content on the client browser side and on the Microsoft Web Storage System on the Exchange server side. [1]

There are three main components of OWA 2000. The first component is the web browser at the user end which sends HTTP (or HTTPS) requests and receives HTTP (or HTTPS) responses. (Incidentally, although some functionality is lost, the browser does not necessarily have to be Internet Explorer.)

The second component is a front-end Exchange Server that acts as both a web server to handle HTTP requests and a proxy server to forward requests from the client browser and to the server holding the Exchange mailboxes. The front-end server is a new component in Exchange 2000. In simplest terms, the front-end server is an Exchange 2000 server with no user data on it. It is created by setting a single toggle on a default Exchange 2000 Enterprise server installation. The front-end server can also handle SMTP and POP3 communications but only the HTTP protocol will be discussed here. Internet Information Service (IIS) is the component of front-end server that provides the web server functions. Any secure socket layer (SSL/HTTPS) encryption or decryption is handled on the front-end server.

The third component is a back-end Exchange server that manages the mailbox stores and processes the requests forwarded through the front-end server. A back-end Exchange server is the default installation of Exchange 2000 Server or Exchange 2000 Enterprise Server. It is only called a back-end in the context of a front-end/back-end architecture.

Multiple back-end servers can be set up for performance when a large number of mailboxes exist. In this case, the front-end server performs a lookup in the Global Catalog on Active Directory to determine which back-end server to contact. After making this determination, the front-end server passes the HTTP request from the client browser to the correct back-end server. Small changes are made to the header information to indicate that the request came through a front-end server. The back-end server processes the requests and sends the response through the front-end server to the client browser. The front-end server makes no change to the response except that the front-end server may encrypt the response if SSL is being used to communicate to the client browser.

Note that even if the client browser and front-end server are using SSL encryption, the communication between the front-end server and the back-end server is not encrypted by SSL. The communication between the front-end and back-end server is strictly in the HTTP protocol for OWA requests and responses. This has security implications that will be examined later. [2]

**Authentication**

Before a user can access a mailbox through the client browser, the user must first enter a correct username and password and be authenticated. OWA has two configuration options for authentication. The first option is that the user is required to be authenticated on both the front-end server and the back-end server. The front-end server receives the initial request with the username and password, attempts to authenticate the user, and passes on the user credentials to the back-end server only if the authentication was successful. Because the front-end server is just a proxy server and only slightly modifies the header information of requests, the back-end server must also perform its own authentication based on the user credentials. Fortunately, the user does not have to re-enter a username and password when the request reaches the back-end server. This is known as dual authentication and is considered the more secure option of the two. The drawback to dual authentication is that the front-end server must make RPC calls to the domain controller and Active Directory Global Catalog server. This may be a problem if the front-end server is installed in a perimeter net that prohibits RPC calls into the corporate intranet.

Dual authentication is configured on the front-end server by turning off anonymous access to the Exchange virtual website. When anonymous access is on, IIS uses a guest account (iusr_*machinename*) to allow anyone access to the website. Anonymous access is turned off in the Exchange management utility under the “Directory Security” options for the front-end server’s Exchange virtual

directory. When anonymous access is off, the front-end server must authenticate the user. IIS performs the authentication by either Basic Authentication or Integrated Authentication. IIS will pick the stronger authentication method (Integrated Authentication) if the client supports it.

No changes need to be made on the back-end server because even if anonymous access is on, NTFS permissions on the Exchange mailbox stores will prevent unauthorized access. Still, it is a good idea to turn off anonymous access on the back-end server as well.

The second authentication option on the front-end server is to forward the initial access request to the back-end server. This is known as pass-through authentication and can make it easier to set up a front-end server in a perimeter network. The need for RPC calls through a firewall is eliminated since the back-end server makes the RPC calls to the domain controller and authenticates the user. This is less secure because the front-end server must be set up to allow anonymous access. This not only gives anonymous access to the front-end server, making it a possible target for attack, but also gives the back-end server direct exposure to non-authenticated users.

Another consideration of OWA authentication is that OWA can only support any of the stronger authentication methods like NTLM, Kerberos, or HTTP 1.1 Digest authentication, if the client browser also supports it. If the client browser does not support stronger authentication methods then OWA can only perform HTTP 1.1 basic authentication. Since basic authentication sends usernames and passwords in clear text, user credentials could possibly be sent unencrypted across the Internet depending on where users are located.

Still another issue to consider is the link between the front-end server and the back-end server. The front-end server will pass the clear text username and password to the back-end server. The only functional protocol for OWA between these servers is HTTP. The use of an encryption scheme might be indicated depending on how easily the clear text user credentials could be sniffed in this part of the link. [3]

**Security Environment**

The security weaknesses of the default OWA configuration are numerous. The importance of eliminating these weaknesses will depend in large part upon the environment in which OWA will operate. Some OWA implementations will operate completely inside a corporate intranet where hacking may not be a large concern while others will be running across the Internet where exacting adherence to good security practices is required. Because the implementation of OWA with exposure to the Internet is less secure and more fruitful to discuss, it will be assumed from here forward that this is the case.

## Encryption

The first obvious vulnerability is the transmission of clear text usernames and passwords. OWA can only use HTTP 1.1 basic authentication so usernames and passwords are unencrypted. There are two potential links where a user credentials could be observed by a sniffer. The first link, between the web client and the front-end server, is the more unsecure since it travels across the Internet. It is mandatory to secure this link with secure socket layer (SSL) encryption. When this is done, communications will now take place over TCP port 443 instead of the standard web port 80. Web addresses will begin with HTTPS, instead of HTTP.

SSL encryption will require that an SSL certificate be loaded onto the front-end server. The SSL certificate can be purchased from a commercial certificate authority (such as Verisign) or it can be generated on the server itself from Microsoft Certificate Server. The advantage to purchasing a certificate from a commercial certificate authority is that most browsers are already set up to trust certificates from these sources. A server-generated certificate, although free, will cause browsers to pop-up warning messages that the certificate is not from a trusted source. The SSL encryption will work with either method, the issue is whether users can deal with the warning message. [4]

The second link that may require encryption is that between the front-end server and the back-end server. The need to secure this link may not be as critical as that between the client browser and the front-end server. It is likely that either front-end server is in a perimeter net and the back-end server is in the corporate intranet or both the front-end and back-end servers are in the corporate intranet. This link is completely HTTP-based so any user credentials sent across it will also be in clear text. Exchange 2000 cannot use SSL encryption to secure this link. Instead, IPSec can be installed on both the front-end and back-end servers to encrypt data between the two servers. IPSec can be used to block ports and encrypt traffic based on port number. On the front-end server, the outbound port 80 should be encrypted. On the back-end server, the inbound port 80 should be encrypted. For additional security, if the front-end server is communicating to the client browser using SSL encryption over port 443, then the inbound port 80 on the front-end server can be blocked in IPSec. [5]

## Firewall Configurations

There are three basic ways to set up a secure Internet-connected OWA installation. The first is to isolate the front-end server in a perimeter network. In this set up, the front-end server is sandwiched between the Internet firewall and corporate intranet firewall. The inbound traffic from the Internet first passes through the Internet firewall (often a router), into the front-end server, through the intranet firewall into the intranet, and onto the back-end server. On the Internet firewall, only port 443 needs to be open for the communication between the client browser and the front-end server. The thinking behind this setup is that if the

front-end server is compromised then the attacker must still get through the (tighter) intranet firewall to get to the corporate intranet. [6]

The issue with this kind of set is that a number of ports must be opened on the intranet firewall so that the front-end server can perform its authentication in the dual authentication mode. In order to authenticate the user, the front-end server must communicate over several protocols into the corporate intranet. Domain Name Service (DNS) (TCP/UDP port 53) must be open for the front-end server to resolve the name of the back-end server. (The requirement for DNS can be eliminated by making entries in the local host table on the front-end server. The host table is in this file: C:\winnt\system32\drivers\etc\hosts.) To talk to domain controllers and the Active Directory Global Catalog server, Kerberos (TCP/UDP port 88), LDAP (TCP/UDP port 389 and TCP port 3268), and Netlogon (TCP port 445) must all be allowed. IIS also uses RPC endpoint mapper (TCP port 135) and RPC service ports which are randomly chosen from TCP ports greater than 1024. [7] To eliminate the randomly chosen ports, the RPC service ports can be restricted to one port by setting a registry key on any server that will be contacted by the front-end server. [8]

Of course, port 80 must be open for the actual HTTP communication to the back-end server. If this communication is encrypted with IPSec, it must also be allowed through the firewall. The necessary port to open is UDP port 500 for Internet Key Exchange (IKE). IP protocol 50 should also be allowed for Encapsulating Security Payload (ESP). Also, IPSec will not work across a network address translation (NAT) server if one exists between the perimeter net and the corporate intranet. The obvious dilemma is whether it is worth it to isolate the front-end server in the perimeter net if so many ports must be opened to the corporate intranet. [9]

The second alternative that avoids so many open ports is to put the front-end server on the intranet side of the intranet firewall. This is the easiest configuration to set up. In this case, only port 443 needs to be open on the intranet firewall. None of the DNS, Kerberos, RPC, domain controller, or Global Catalog service ports needs to be open on the intranet firewall because the front-end server and the back-end server are on the same side of the intranet firewall. The intranet firewall can perform IP filtering to limit inbound connections on port 443 to only complete if they are destined for the front-end server. The downside to this architecture is that now a server on the corporate intranet is exposed to the Internet, albeit over a single port. If the front-end server is compromised manually by a hacker or autonomously by a worm, the damage can spread more easily than if the front-end server was isolated in the perimeter net. [10]

The third alternative is the most complicated and expensive but is also the most secure. This configuration applies a third machine called an Internet Security and Acceleration (ISA) server. An ISA server is an advanced firewall and can perform a number of security functions including IP filtering, port filtering, protocol

filtering, and application filtering. It can also log and report any suspicious behavior it sees. ISA runs on Windows 2000 Server and is placed either in the perimeter net or acts as the corporate intranet firewall itself. [11]

The advantage to an ISA server is that it can add another layer of protection over a standard intranet firewall through application filtering. This means that the ISA server will actually examine each HTTP packet and check it for correct syntax to try to defeat intrusion attempts based on sending malformed HTTP requests. ISA has a function called Web Filters that allows the inspection of inbound packets for worms or other deviant code. It can block these potentially destructive packets before they reach the corporate intranet. The Web Filters are enabled on the ISA server's built-in Web Proxy service. Requests from the Internet are first directed to this service and examined before being forwarded to the front-end server in the corporate intranet. The Web Proxy service is designed to handle SSL communication with both the client browser and the front-end server. It also provides more defense-in-depth by adding another layer between the front-end server and the Internet. [12]

The ports that must be open for this configuration are port 443 on the Internet firewall and port 443 on the intranet firewall (or on the ISA server if it is used as the intranet firewall).

It should also be noted that ISA server can act as an intrusion detection system by taking action if suspicious activity like a port scan or malformed HTTP is detected. This action can be in the form of stopping a service, running a program, or sending an email to an administrator's inbox or pager. [13]

**IIS Considerations**

OWA depends heavily on the notoriously insecure Internet Information Service (IIS). It must be running on both the front-end and back-end servers. Although, the front-end and back-end servers may be on the corporate intranet and be buffered from the Internet by several layers of firewalls, it is still a good idea to eliminate as many vulnerabilities as possible on the IIS service. The methods for doing so are outlined on the SANS Top 20 Vulnerabities page (IIS is number 1). Briefly, these methods include:

- Stay current on patches and hotfixes by using the Microsoft Hot Fix Checker (HFNetChk.asp)
- Run the Microsoft IIS Lockdown tool (free from Microsoft) http://www.microsoft.com/technet/security/tools/locktool.asp
- Delete the sample applications and web administration tool scripts from the inetpub\wwwroot\scripts directory (only .asp is necessary)
- Unless an ISA server is already doing so, set up the URLScan filter to screen out malicious HTTP packets, the URLScan tool is included with the IIS Lockdown tool [14]
- Disable any default services that will not be used such as SMTP and FTP

- Install IIS on a partition other than C:\ to prevent worms from navigating through the default directories
- Remove any unnecessary default virtual web sites
- Enable monitoring and logging in IIS [15]

**Services**

It is typically a good idea to secure a server by shutting off any services that will not be used. A server with many unnecessary services eagerly waiting to answer requests is an invitation for an attack. The following services are specific to Exchange 2000 and must be running on the back-end server: [16]

- Microsoft Exchange Information Store
- Microsoft Exchange Management
- Windows Management Instrumentation
- Microsoft Exchange Routing Engine
- IIS Admin Service
- SMTP
- World Wide Web Publishing Service
- Microsoft Exchange System Attendant, which also requires:
  - Event Log
  - NTLM Security Support Provider
  - RPC
  - RPC Locator
  - Server
  - Workstation
- IPSec Policy Agent (if IPSec will be used between the front-end and back-end servers for encryption)

It is even more important to shut off unnecessary services on the front-end server as it will be dealing with incoming HTTP requests from the Internet. The following services are necessary on the front-end server: [17]

- Microsoft Exchange Routing Engine
- IIS Admin Service
- World Wide Web Publishing Service
- RPC Locator
- IPSec Policy Agent (if IPSec will be used between the front-end and back-end servers for encryption)

Be careful when attempting to shut off unnecessary services. Some of the services above may be dependent on other services that are not mentioned. It is a good idea to keep careful notes and to shut off services one at a time while verifying that OWA still functions as expected.

### OWA Password Changes

The default installation of OWA allows users to change their passwords by providing a button on the HTML form. The password change function uses a .htr script to enable this capability. As there have been IIS vulnerabilities associated with .htr script mappings, it may be a good idea to remove .htr from the script mapping list. [18] When the .htr script mappings are removed the change password function will cease to work but the actual "Change Password" button will still appear to the user on the OWA form. It is a good idea to remove the button to avoid user confusion. To remove the button, add this key using the Registry Editor to both the front-end and back-end servers:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MSExchangeWEB\OWA

The settings for this key are:

Value name: DisablePassword
Type: REG_DWORD
Data :1

To see the change on the OWA web form, restart the MSExchangeIS service and the IIS Admin service on the front-end and back-end servers. [19]

### Known Vulnerabilities

Kiosks that do not require a username and password to login present a special opportunity for unauthorized access. After a user is authenticated in OWA, the user's credentials are cached locally in the client browser and are used throughout the session. If the user finishes working in OWA and walks away without closing the browser, another user can hit the "Back" button on the browser and full access the first user's OWA session. The browser must be closed to remove cached credentials. Third-party programs exist for closing the browser when the OWA session is completed. [20]

Attachments that are crafted with malicious intent in OWA can present a problem when opened in Internet Explorer. Since IIS uses scripts to render the HTTP-based email displays, a script can be embedded in the attachment that will be executed when the user opens the attachment. The script could be designed to perform actions on the user's mailbox. Microsoft disclosed this vulnerability in Security Bulletin MS01-030. Updated files that prevent this vulnerability are offered on the web page describing this bulletin (see references). [21]

A potential for a denial-of-service (DOS) attack exists in the default OWA 2000 installation. A request for a deeply-nested but non-existent folder can be submitted that could absorb a large amount of processing resources on the OWA server. The potential for this type of attack is reduced by the fact that the user launching the attack must first be authenticated in the domain and must have

access rights to the mailbox. This vulnerability is described in MS Bulletin MS01-049 along with a patch to prevent the problem. [22]

## Conclusion

Outlook Web Access is a highly useful tool for keeping traveling employees and telecommuters in touch with the corporate office. However, its default use of insecure protocols like HTTP and Basic Authentication and its reliance on Internet Information Services mandate that extra efforts be taken to secure it. The environment of each link between client browser and mailbox store should be considered. When this is done, common security measures like encryption, filtering, and patch-checking can be used to create an OWA installation that is difficult to crack.

## References


[1] Hunt, Bob, Solazzo, Carl, and Sebben, Paul. "Technet, Exchange Server 2000, Resource Kit, Part 5, Chapter 25: Outlook Web Access". http://www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechnol/exchange/exchange2000/reskit/part5/c25owa.asp (Jan 27 2003).

[2] Lemson, KC and Martin, Michele. "Using Microsoft Exchange 2000 Front-End Servers". October 2002. pp. 2-4,11. http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=AFAD8426-572E-40F8-99DA-EB7198F374C4 (Jan 27 2003).

[3] Lemson, KC and Martin, Michele. "Using Microsoft Exchange 2000 Front-End Servers". October 2002. pp.21-23. http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=AFAD8426-572E-40F8-99DA-EB7198F374C4 (Jan 27 2003).

[4] Lemson, KC and Martin, Michele. "Using Microsoft Exchange 2000 Front-End Servers". October 2002. p. 29. http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=AFAD8426-572E-40F8-99DA-EB7198F374C4 (Jan 27 2003).

[5] Microsoft Corporation. "Security Operations for Microsoft Exchange 2000 Server". Version 1.0 2002. pp. 51-52. http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/prodtech/mailexch/opsguide/default.asp (Jan 27 2003).

[6] Lemson, KC and Martin, Michele. "Using Microsoft Exchange 2000 Front-End Servers". October 2002. p. 32. http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=AFAD8426-572E-40F8-99DA-EB7198F374C4 (Jan 27 2003).

[7] McBee, Jim. "OWA 2000 Security and Scalability". Jan 2002. p. 2. http://www.exchangeadmin.com/Articles/Index.cfm?ArticleID=23139&pg=1 (Jan 27 2003).

[8] Microsoft Corp. "Microsoft Knowledge Base Article Q224196". Oct 2002. http://support.microsoft.com/default.aspx?scid=kb;EN-US;224196 (Jan 20 2003).

[9] Microsoft Corporation. "Security Operations for Microsoft Exchange 2000 Server". Version 1.0 2002. pp. 51-52. http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/prodtech/mailexch/opsguide/default.asp (Jan 27 2003).

[10] Lemson, KC and Martin, Michele. "Using Microsoft Exchange 2000 Front-End Servers". October 2002. pp. 39-40.

http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=AFAD8426-572E-40F8-99DA-EB7198F374C4 (Jan 27 2003).

[11] Microsoft Corp. "ISA Server 2000: Firewall Security Services with Microsoft Internet Security and Acceleration Server 2000". 2001. pp. 13-19. http://www.microsoft.com/isaserver/featurepack1/overview/default.asp (Jan 27 2003).

[12] Microsoft Corp. "ISA Server 2000: Firewall Security Services with Microsoft Internet Security and Acceleration Server 2000". 2001. p. 23. http://www.microsoft.com/isaserver/featurepack1/overview/default.asp (Jan 27 2003).

[13] Microsoft Corp. "ISA Server 2000: Firewall Security Services with Microsoft Internet Security and Acceleration Server 2000". 2001. p. 19. http://www.microsoft.com/isaserver/featurepack1/overview/default.asp (Jan 27 2003).

[14] SANS/FBI. "The Twenty Most Critical Internet Security Vulnerabilities". Ver. 3.21. Oct. 29. 2002. http://www.sans.org/top20/#W1 (Jan 27 2003).

[15] Parker, Michael. "Securing Web Based Corporate E-Mail Using Microsoft Exchange Outlook Web Access". July 26, 2001. http://www.sans.org/rr/email/corp_email.php (Jan 27 2003).

[16] Microsoft Corporation. "Security Operations for Microsoft Exchange 2000 Server". Version 1.0 2002. p. 29. http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/prodtech/mailexch/opsguide/default.asp (Jan 27 2003).

[17] Microsoft Corporation. "Security Operations for Microsoft Exchange 2000 Server". Ver. 1.0 2002. p. 29. http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/prodtech/mailexch/opsguide/default.asp (Jan 27 2003).

[18] Parker, Michael. "Securing Web Based Corporate E-Mail Using Microsoft Exchange Outlook Web Access". July 26, 2001. http://www.sans.org/rr/email/corp_email.php (Jan 27 2003).

[19] Microsoft Corp. "XWEB: How to Hide the Change Password Button on the Outlook Web Access Options Page Microsoft Knowledge Base Article – 297121". Jun. 11 2002. http://support.microsoft.com/default.aspx?scid=kb;en-us;297121 (Jan 27 2003).

[20] Messageware Inc. "OWA Security: SecureLogoff for Outlook Web Access 2000" Jan 18, 2002.

http://www.messagingsolutions.com/Securelogoff/OWA_Security.pdf (Jan 27 2003).

[21] Microsoft Corp. "Microsoft Security Bulletin MS01-030: Incorrect Attachment Aandling in Exchange OWA Can Excecute Script ". Jun 13 2001. http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS01-030.asp (Jan 27 2003).

[22] Microsoft Corp. "Microsoft Security Bulletin MS01-049: Deeply-nested OWA Request Can Consume Server CPU Availability". Sep. 26 2001. http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS01-049.asp (Jan 27 2003).