Global Information Assurance Certification Paper

Securing Web Based Corporate E-Mail Using Microsoft Exchange Outlook Web Access

GSEC Practical Assignment Version 1.2e

Michael Parker
July 26th, 2001-07-26

In today's hyper-connected environment, it is essential that the workforce remain in touch with their home office and clients. This is often at odds with the fact that a significant portion of the workforce is also a mobile one. To overcome this problem, a number of solutions have been employed such as cellular phones, personal digital assistants (PDA's) such as Palm Pilots, Blackberry's and iPaq devices. Each of these has restrictions on the amount of data that can be communicated. It is evident that the best system for communicating anything from brief "fyi's" to long-term corporate strategy is still a reliable e-mail system. As such, many organizations have moved towards exposing their internal e-mail systems to the Internet in order to maintain those lines of communication and an immediate transmission of data. Of course, doing so unleashes a significant number of security risks that must be addressed before even the first e-mail is opened at any unsecured location. Failure to do so threatens the host organization's mail server, network, corporate partners and even a corporation's very existence.

One such solution is Microsoft's Outlook Web Access (OWA), an extension of the Exchange 5.5 mail system, which allows a client access to basic e-mail features, public folders, and a personal calendar through a standard Web browser. As it is based on the notoriously insecure IIS web server, it may be supposed that this system cannot possibly be secured, but with the correct combination of technology, policy and user education, this system can indeed be secured sufficiently enough to withstand rigorous scrutiny.

What is OWA?

Microsoft Outlook Web Access is a Messaging Application Programming Interface (MAPI) application that is comprised of a variety of binary, HTML, and Active Server Page script files that reside on an IIS web server. This web server uses collaboration data objects (CDO) to act as a broker and translator between the end-users web browser and the organization's Exchange mail server.

Before continuing, it is important to understand the process of establishing an OWA session. The following steps outline what happens when a client, using a web browser requests a mail session via Outlook Web Access to an internal Exchange mail server (but not the authentication process).

1. A user requests an OWA session by clicking on a hyperlink or typing in a URL.
2. The web browser sends the request with header information to the IIS server running OWA.

3. IIS determines the appropriate language and then processes the OWA ASP.
4. These scripts use CDO to open the users mailbox in the user's Microsoft Exchange Server information store.
5. The CDO rendering library (Cdohtml.dll) converts the requested email messages into HTML and IIS sends the HTML to the browser.
6. The Web browser renders the HTML.

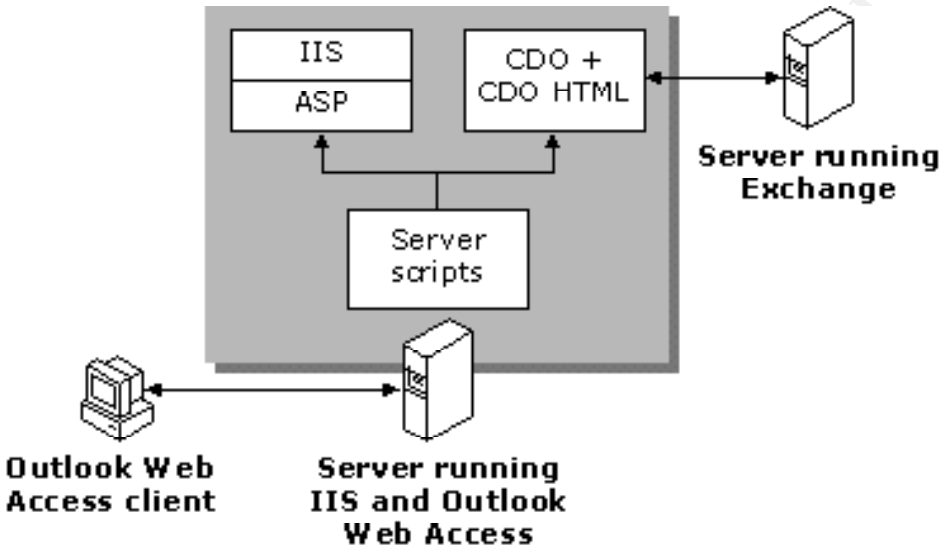


Figure 1: The interaction between the Outlook Web Access client, the IIS/Outlook Web Access server, and the Microsoft Exchange Server computer[1]

Securing the IIS Web Server

As previously mentioned, OWA is installed on top of Microsoft's Internet Information Server(IIS) which is notorious for the number of vulnerabilities it presents to even an unsophisticated hacker. To minimize your risk, it is exceedingly important that all patches and hot fixes are applied and monitored. Further, develop a baseline server configuration that can be carefully monitored for changes (preferably via third party software). To help you in this, Microsoft has developed a security checklist specifically for IIS 5.0[2]. This next section will deal with specific points of that document that you should be aware of.

The installation of OWA requires that the server it is to be installed on be either;

1. A member server within it's own domain, with an established two-way trust between the OWA server and the domain that the mail system belongs or,
2. The OWA server is a member of a corporate domain, residing in the DMZ, with secure, trusted entry points opened up through the corporate firewall to allow

[1] "Planning and Deploying Outlook Web Access 5.5", page 6
[2] See Howard, 2000

communication between the PDC, the Exchange server and the OWA server.

It is the authors' opinion that the latter scenario can be more strictly controlled, and as such will be the continued focus of this paper. *(Author's note - It is good practice to install IIS on a partition other than the default C:\. as several worms and or viruses search for default installs on the C:\, such as the recent sadmind/IISWorm for instance[3]. Installing on a separate partition lessens the likelihood of such an attack).*

- **Remove unnecessary web sites**

Since the OWA server is basically a dedicated web server, the number of required virtual web sites is minimal. The default web site should be disabled or removed and a new virtual web site should be created with a different name to eliminate any unnecessary risks associated with the default settings and files included with them. In addition, remove the IISHelp, IISAdmin and IISSamples sites for the same reason. Best practice would be not to allow remote administration of the OWA server.

- **Disable unnecessary services**

IIS installs FTP and SMTP services by default. Further, the ACL's for these two default folders (C:\inetpub\FTProot and C:\inetpub\mailroot) are set to allow "Everyone" Full Control. OWA doesn't require these services so it is a good idea to remove them altogether. Also, remove or simply do not install Index Server, Internet Services Manager (HTML) and Front Page Extensions. These services are not required, and installing them provides further avenues for a hacker to exploit.

- **Disable unnecessary script mappings**

OWA requires the ability to use .asp and .htr scripts (depending on how OWA is configured). As such, all other unnecessary script mappings should be removed, including .idc, .stm, .shtm, .shtml, .printer, .htw, .ida and .idq. If the organization is not going to enable the client the ability to remotely change their password via OWA, then it is also possible to remove the .htr script mapping. This is significant as the .htr file extension has been associated with numerous vulnerabilities in several version of IIS[4].

- **Implement logging and monitoring**

To complete the secure server configuration, you should also include monitoring and logging on your web server so that any unauthorized activity can be recognized and due action taken. Perhaps the best practice in this regard would be to implement a host based intrusion system[5] and configure it with the with appropriate responses in the event of compromise. Possible scenarios would be frequent password attempts within a limited time period, failed authentication attempts or any type of anomalous behaviour that could be deemed an attack-in-progress. In conjunction with a network based intrusion detection system this monitoring becomes formidable indeed and adds to the concept of "defence-in-depth[5]".

[3] "CERT® Advisory CA-2001-11 sadmind/IIS Worm"
[4] "Microsoft Security Bulletin (MS00-031)" and "Microsoft Security Bulletin (MS99-058)" to name a couple.
[5] Northcutt, p.16

Permissions Required for OWA Access

Prior to any access to their mailbox the user should (in theory) have a valid account in the domain where his mailbox resides. In addition, Microsoft recommends that IIS be granted anonymous access to the web server.

Create a local group on the OWA server and populate it with the groups that require access to OWA. It is a local group and should be populated with global groups from the corporate domain as opposed to entering specific usernames. See Appendix A for a complete list of required permissions.

Securing OWA

Microsoft recommends several architectural configurations, however most of the recommended scenarios seem to be better suited for deployment within an organizations Intranet, where the environment is under much tighter control. Deployment on the Internet means that administrators must now consider such scenarios as sniffers, insecure communication channels and terminals such as airport kiosks and internet café's to name a few. As such, the configuration options for a secure OWA deployment become much more limited.

Perhaps the most secure topology to be used when deploying OWA is with a dedicated OWA server located in the DMZ (Appendix B). Not only is this better for performance reasons, it also allows us to further isolate potential intrusions and/or damage as the OWA server would be afforded some protection from the external world by a firewall, and the backend Exchange server (and the rest of the network for that matter) would be protected by a second firewall between it and the OWA server.

Communications between the unsecured browser, the OWA server, the corporate Primary Domain Controller (PDC) and the Exchange server takes place over a series of channels. Securing these channels is critical in ensuring that the data being transmitted remains confidential, and is being used appropriately. In an Intranet environment it is possible to get away with using NTLM authentication, as the OWA component can be installed with IIS on the same Exchange server (indeed the only way NTLM authentication can be used is under those very circumstances). However, in the Internet environment, where the type of browser being used is not tightly controlled, it is more appropriate to use Basic/Clear Text Authentication. Unfortunately this also means that credentials and potentially sensitive data would be passed back and forth in the clear, and susceptible to interception unless guarded by another mechanism, hence you should also incorporate the use of Secure Socket Layer (SSL). Forcing the client to use SSL ensures that all data will be encrypted prior to transmission across the Internet. Furthermore, if the organization is using a dedicated OWA server as suggested, then the web server need not grant any access other than that required for the OWA server itself. SSL encrypted data uses port 443 to transfer

data, hence, the only port to the OWA server that needs to be opened is port 443. After the client types in the URL or clicks on a hyperlink[6], they can be taken to a secure page (as indicated by "https:\\" at the beginning of the URL). Prior to showing the user's mailbox they will first need to be authenticated to the domain, which will take place in the form of a popup window challenge requiring a username and a password. Once the client enters their information, it is transmitted to the PDC via Netbios for authentication. To enable this you need to configure the firewall between the OWA server and the corporate network to allow Netbios ports 135, 137, 138 and 139 into the network. As there are numerous methods of using Netbios to map out a network, it is also highly recommended that you specify rules on the firewall that limit access by the OWA servers' IP. In other words, make those ports available to the OWA server only and reject all others.

Once the client is authenticated, and this information is communicated back to IIS, a call is made to the Exchange server from the OWA server via a random RPC port. You will again need to define new rules through the firewall to allow the appropriate ports access to your network, and again it is highly recommended that this be restricted by IP address in the same manner as the Netbios ports mentioned earlier. Microsoft requires that all ports above 1024 be made available, but from a security perspective, this is clearly unacceptable. It is possible (and recommended) to limit the ports required/allowed. This is achieved by making modifications to the registry on the Exchange server;

1. Using Registry Editor on the Microsoft Exchange Server computer, add the following entry for the Microsoft Exchange Server directory in the following registry key:

   HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services \MSExchangeDS\Parameters:
   Entry: **TCP/IP port**
   Type: **REG_DWORD**
   Data: *port number to assign*

   For example, in the port number "dword:000004C9(1225)" the decimal number 1225 (4C9 in hexadecimal format) is for the directory.

2. Add the following entry for the information store in the following registry key:

   HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services \MSExchangeIS\ParametersSystem:
   Entry: **TCP/IP port**
   Type: **REG_DWORD**
   Data: *port number to assign*

   For example, in the port number "dword:000004CA(1226)" the decimal number

[6] The author does not recommend adding a visible link as this may provide too much of a temptation to would be hackers. Better to have the clients memorize and type in the URL themselves.

1226 (4CA in hexadecimal format) is for the information store.

3. Quit Registry Editor [7].

User Education & Policy

When the mail system is exposed to the Internet for the benefit of clients, the administrator may feel akin to the mother bird pushing her young out of the nest. This is because we are aware that even after all our efforts to secure the mail environment, we still have to place a substantial amount of faith in our users not to leave the door wide open for a potential hacker.

This sensation can be lessened by adequate user education. It is our duty to make sure that the user is aware of the potential hazards of not logging out of their mail session and closing the browser to remove traces of any session cookies and sensitive usernames and passwords. We have to make them aware that a file saved to the desktop of a kiosk terminal may not be the most appropriate behaviour, and to recognize the acceptable limits of a remote corporate e-mail system. Possible solutions could include personal knowledge shares with users who will have access to the system (either in groups or individually depending on the scale of the project), a detailed, yet readable e-mail that outlines potential hazards and finally, and an established policy of acceptable behaviour in regards to the remote use of e-mail. This policy should outline all the duties and responsibilities of the client and the penalties for not adhering to them, and should be brought to their attention on a reasonable, repeatable basis.

Conclusion

In spite of the potential hazards of exposing a corporate mail system to the Internet, it is indeed possible to secure this system against possible abuse through a combination of technology and training and monitoring. It has often been stated that no system can be completely secured, and this is no less true here. However, by applying multiple layers of security, it is certainly possible to provide a remote email service and maintain a balance between both client demands and acceptable levels of security and risk.. A secure base web server is key to this implementation, which should subsequently be enhanced by a combination of encryption, firewall and intrusion detection technologies. In addition, a strong user education program and policy combines to provide an Internet mail system that adheres to the practice of “defence-in-depth[8]”.

[7] Planning and Deploying Outlook Web Access 5.5, page 11
[8] Northcutt, p.16

References

"CERT® Advisory CA-2001-11 sadmind/IIS Worm". CERT/CC. URL: http://www.cert.org/advisories/CA-2001-11.html (May 8, 2001).

Colde, J. & Winters, S. "Host Perimeter Defence – SANS GIAC Level One 2000-2001", SANS GIAC Certification Program. January 2001.

Howard, Michael, "Secure Internet Information Services 5 Checklist". Microsoft Technet. URL: http://www.microsoft.com/technet/security/iis5chk.asp, (June 29, 2000).

Microsoft Corporation, "Planning and Deploying Outlook Web Access 5.5". 1999 URL: http://www.microsoft.com/exchange/techinfo/planning/55/OWA55_DeployPlan.doc. (July 23, 2001).

Microsoft Corporation, "Outlook Web Access Performance & Extensibility", URL: http://www.microsoft.com/TechNet/prodtechnol/office/maintain/optimize/owaperf.asp

Microsoft Corporation, "XWEB: Exchange Server 5.5 Outlook Web Access Logon Process". Article ID: Q263236. Revised June 22, 2001. URL:http://support.microsoft.com/support/kb/articles/q263/2/36.asp. (July 25, 2001).

Microsoft Corporation, "XWEB: Requirements for Outlook Web Access". Article ID: Q239569. Revised June 22, 2001. URL:http://support.microsoft.com/support/kb/articles/Q239/5/69.ASP (July 28, 2001).

Microsoft Technet, "Microsoft Security Bulletin (MS00-031)". Revised July 17, 2000. URL: http://www.microsoft.com/technet/security/bulletin/ms00-031.asp. (July 23, 2001).

Microsoft Technet, "Microsoft Security Bulletin (MS99-058) Frequently Asked Questions", December 21, 1999. URL: http://www.microsoft.com/technet/security/bulletin/fq99-058.asp. (July 24, 2001).

Microsoft Technet, "Microsoft Security Bulletin (MS00-078)". October 17, 2001 URL: http://www.microsoft.com/technet/security/bulletin/MS00-078.asp. (July 25, 2001).

Northcutt, S., "Information Assurance Foundations – SANS GIAC Level One 2000-2001", SANS GIAC Certification Program. January 2001.

Appendix A

Grant the following permissions and rights on the OWA server;

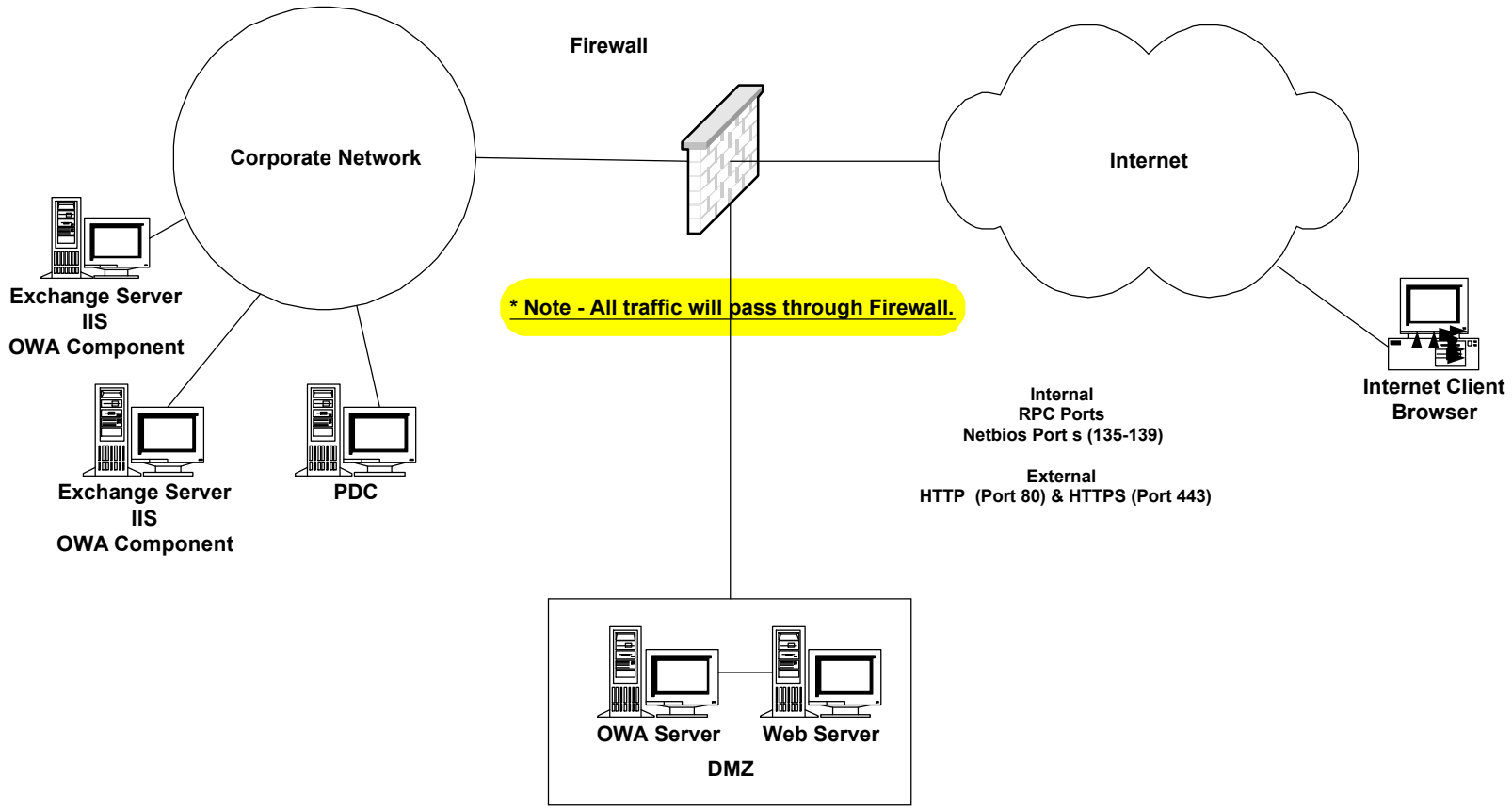
Log On Locally
- IUSR_<servername>

Access computer from the network
- Authenticated Users
- IUSR_<servername>
- IWAM_<servername>

NTFS security on c:\exchsrvr\webdata
- Everyone Read, Execute and List Folder Contents
- Administrators Full Control
- Domain Admins Full Control

NTFS security on c:\exchsrvr\webtemp
- Administrators Full Control
- Domain Admins Full Control
- OWA Local Group Everything but Full Control
- Everyone Read, Execute and List Folder Contents

Appendix B



1. Client can type in **www.domain.com/owa** or **https://owa.domain.com** Connection to this site is secured using SSL over port 443.

2. Client is challenged and must enter in their NT Username and Password. Credentials are sent to PDC over Netbios ports 135-139.

3. If authenticated, client is redirected to OWA logon page (../exchange/usa/logon.asp) where they must enter in their mailbox.

4. OWA component creates a MAPI session with the Exchange server using dynamically assigned ports >1024. A rule on the firewall only allows these ports through from this IP, and limits the range available.

5. Exchange looks up user, and accesses mailbox, provides mail and public folders. Information is passed via MAPI session back to OWA server.

6. OWA component translates MAPI session info into HTML and passes back to client via SSL session.

Last Updated: July 25th, 2017

Upcoming Training

SANS Prague 2017	Prague, Czech Republic	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Boston 2017	Boston, MA	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Salt Lake City 2017	Salt Lake City, UT	Aug 14, 2017 - Aug 19, 2017	Live Event
Community SANS Omaha SEC401*	Omaha, NE	Aug 14, 2017 - Aug 19, 2017	Community SANS
SANS New York City 2017	New York City, NY	Aug 14, 2017 - Aug 19, 2017	Live Event
SANS Chicago 2017	Chicago, IL	Aug 21, 2017 - Aug 26, 2017	Live Event
SANS Virginia Beach 2017	Virginia Beach, VA	Aug 21, 2017 - Sep 01, 2017	Live Event
SANS Adelaide 2017	Adelaide, Australia	Aug 21, 2017 - Aug 26, 2017	Live Event
Virginia Beach 2017 - SEC401: Security Essentials Bootcamp Style	Virginia Beach, VA	Aug 21, 2017 - Aug 26, 2017	vLive
Community SANS Pasadena SEC401 @ NASA	Pasadena, CA	Aug 23, 2017 - Aug 30, 2017	Community SANS
Mentor Session - SEC401	Minneapolis, MN	Aug 29, 2017 - Oct 10, 2017	Mentor
SANS San Francisco Fall 2017	San Francisco, CA	Sep 05, 2017 - Sep 10, 2017	Live Event
SANS Tampa - Clearwater 2017	Clearwater, FL	Sep 05, 2017 - Sep 10, 2017	Live Event
Mentor Session - SEC401	Edmonton, AB	Sep 06, 2017 - Oct 18, 2017	Mentor
SANS Network Security 2017	Las Vegas, NV	Sep 10, 2017 - Sep 17, 2017	Live Event
Community SANS Albany SEC401	Albany, NY	Sep 11, 2017 - Sep 16, 2017	Community SANS
Mentor Session - SEC401	Ventura, CA	Sep 11, 2017 - Oct 12, 2017	Mentor
Community SANS Columbia SEC401	Columbia, MD	Sep 18, 2017 - Sep 23, 2017	Community SANS
Community SANS Dallas SEC401	Dallas, TX	Sep 18, 2017 - Sep 23, 2017	Community SANS
Community SANS Boise SEC401	Boise, ID	Sep 25, 2017 - Sep 30, 2017	Community SANS
Baltimore Fall 2017 - SEC401: Security Essentials Bootcamp Style	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	vLive
Community SANS New York SEC401	New York, NY	Sep 25, 2017 - Sep 30, 2017	Community SANS
Rocky Mountain Fall 2017	Denver, CO	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS London September 2017	London, United Kingdom	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Baltimore Fall 2017	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Copenhagen 2017	Copenhagen, Denmark	Sep 25, 2017 - Sep 30, 2017	Live Event
Community SANS Sacramento SEC401	Sacramento, CA	Oct 02, 2017 - Oct 07, 2017	Community SANS
SANS DFIR Prague 2017	Prague, Czech Republic	Oct 02, 2017 - Oct 08, 2017	Live Event
Community SANS Charleston SEC401	Charleston, SC	Oct 02, 2017 - Oct 07, 2017	Community SANS
Mentor Session - SEC401	Arlington, VA	Oct 04, 2017 - Nov 15, 2017	Mentor
SANS October Singapore 2017	Singapore, Singapore	Oct 09, 2017 - Oct 28, 2017	Live Event