

# Microsoft 365 Security

Everything about Microsoft Security

MENU



## Hunting In On-Premises Exchange Server Logs

Posted on [October 7, 2022](#) | by [m365guy](#) | [2 comments](#)

This will be a high-level summary of the different logs that can be found on an On-Premises Exchange server, which can be useful during an IR. For each log, I'll try to explain what we can achieve with it. Not all logs are useful, so I've only picked the one's that I'm aware of and believe are useful.

### IIS logs

One of the useful logs on an Exchange server are the IIS logs. From hunting down **ProxyLogon** to **Webshell** activities. IIS logs can play a huge role in finding these suspicious activities. IIS logs are by default stored at the following location: **C:\inetpub\logs\LogFiles** and come with two folders. **W3SVC1** and **W3SVC2**. Both of these IIS log files contain all the GET and POST requests that are made. It also includes basic items such as IP and username, request date and time, service status and number of bytes received, as well as detailed items of target files.

This how the structure of the IIS log looks like with the all the fields.

```
#Software: Microsoft Internet Information Services 10.0
#Version: 1.0
#Date: 2022-10-06 07:13:03
#Fields: date time s-ip cs-method cs-uri-stem cs-uri-query s-port cs-username
c-ip cs(User-Agent) cs(Referer) sc-status sc-substatus sc-win32-status time-
taken
```

Let's take a quick example of an **GET** request that was made by an attacker. The two lines that I've marked in highlight is the Webshell activity.

```
2022-10-06 18:51:00 ::1 GET /Microsoft-Server-ActiveSync/default.eas &CorrelationID=<empty>;&afeReqId=8885aa6f-4867-463e-bff1-cfc937fdc1ad; 443 HealthMail
2022-10-06 18:51:06 10.0.0.11 GET /aspnet_netclient/4_0_30319/devilzShell.aspx dir=C%3A%5C&cmd=dsquery+*+-filter+%22%28adminCount%3D1%29%22&btnCommand=Exec
2022-10-06 18:51:06 10.0.0.11 GET /aspnet_netclient/4_0_30319/devilzShell.aspx img=bg 443 - 20.106.209.84 Mozilla/5.0+(Windows+NT+10.0;+win64;+x64)+AppleWe
2022-10-06 18:51:11 127.0.0.1 GET /mapi/emsmdb mailboxId=ddca3a8f-3ff9-4693-beed-ce1bdbb125f2@contoso.com&CorrelationID=<empty>;&afeReqId=82549bdc-4466-43
2022-10-06 18:51:17 127.0.0.1 GET /RPC/rpcproxy.dll ddca3a8f-3ff9-4693-beed-ce1bdbb125f2@contoso.com&CorrelationID=<empty>;&RequestId=0b5fd553-8f81-455b-a8
2022-10-06 18:51:17 ::1 GET /OAB/ &CorrelationID=<empty>;&afeReqId=faffef8a-e00b-4c51-984f-300d8fee0644; 443 CONTOSO\HealthMailbox5030916 ::1 AMPProbe/Loca
```

This how the entire result looks like:

```
2022-10-06 18:51:06 10.0.0.11 GET
/aspnet_netclient/4_0_30319/devilzShell.aspx dir=C%3A%5C&cmd=dsquery+*+-
filter+%22%28adminCount%3D1%29%22&btnCommand=Execute 443 - 20.106.209.84
Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+AppleWebKit/537.36+
(KHTML,+like+Gecko)+Chrome/106.0.0.0+Safari/537.36+Edg/106.0.1370.34
https://20.62.174.61/aspnet_netclient/4_0_30319/devilzShell.aspx?
dir=C%3A%5C&cmd=nltest+%2Fdomain_trusts+%2Fall_trusts&btnCommand=Execute 200
0 0 432
```

This is how we can interpret the data.

Date time	2022-10-06 18:51:06
s-ip	10.0.0.11 (This is the internal IP of the server)
cs-method	GET
cs-uri-stem	/aspnet_netclient/4_0_30319/devilzShell.aspx
cs-uri-query	dir=C%3A%5C&cmd= <u>dsquery</u> +*+ <u>filter</u> +%22%28adminCount%3D1%29%22&btnCommand=Execute
s-port	443
c-ip	20.106.209.84 (IP address of the attacker that initiated this request)
cs(User-Agent)	Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+AppleWebKit/537.36+ (KHTML,+like+Gecko)+Chrome/106.0.0.0+Safari/537.36+Edg/106.0.1370.34
cs(Referer)	https://20.62.174.61/aspnet_netclient/4_0_30319/devilzShell.aspx? <u>dir=C%3A%5C&amp;cmd=nltest+%2Fdomain_trusts+%2Fall_trusts&amp;btnCommand=Execut</u>
sc-status	200

At this example, we are having a different Webshell. However, this time we are initiating a **POST** request.

This is how the full POST request looks like:

```
2022-10-06 19:22:30 10.0.0.11 POST
/aspnet_netclient/4_0_30319/POWERshell.aspx - 443 - 20.106.209.84
Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+AppleWebKit/537.36+
(KHTML,+like+Gecko)+Chrome/106.0.0.0+Safari/537.36+Edg/106.0.1370.34
https://20.62.174.61/aspnet_netclient/4_0_30319/POWERshell.aspx 200 0 0 767
```

This is how we can interpret the data:

## Exchange Setup logs

During an installation of Exchange, there will be a new folder created in the **C:\** drive. By default, the Exchange setup logs are located at: **C:\ExchangeSetupLogs**.

The Setup log tracks the progress of every task during the Exchange installation and configuration. The file contains information about the status of the prerequisite and system readiness checks before installation starts, the application installation progress, and the configuration changes that are made to the system.

**2022-10-01** – Exchange was installed which contains a specific version 15.1.1713.5.

This means that on this date, we saw that Exchange Server 2016 CU12 was installed.

**2022-10-04** – Exchange has been upgraded to version 15.1.2507.6.

Here we can see that Exchange has been upgraded to CU23.

## Exchange PowerShell cmdlet History

Commands that were ran in Exchange PowerShell will be logged in the following location: **C:\Program Files\Microsoft\Exchange Server\V15\Logging\CmdletInfra\LocalPowerShell\Cmdlet**. The filenames look similar to something like **powershell.exe\_19516\_Cmdlet\_2022100116-1**.

This is how the structure of the Exchange PowerShell history logs look like:

```
#Software: Microsoft Exchange Server
#Version: 15.01.1713.001
#Log-type: Rps Cmdlet Logs
#Date: 2022-10-01T14:45:19.673Z
#Fields:
DateTime,StartTime,RequestId,ClientRequestId,MajorVersion,MinorVersion,BuildVersion,RevisionVersion,ServerHostName,ProcessId,ProcessName,ThreadId,CultureInfo,Organization,AuthenticatedUser,ExecutingUserSid,EffectiveOrganization,UserServicePlan,IsAdmin,ClientApplication,Cmdlet,Parameters,CmdletUniqueId,UserBudgetOnStart,ContributeToFailFast,RunspaceSettingsCreationHint,ADViewEntireForest,ADRecipientViewRoot,ADConfigurationDomainControllers,ADPreferredGlobalCatalogs,ADPreferredDomainControllers,ADUserConfigurationDomainController,ADUserPreferredGlobalCatalog,ADUserPreferredDomainControllers,ThrottlingInfo,DelayInfo,ThrottlingDelay,IsOutputObjectRedacted,CmdletProxyStage,CmdletProxyRemoteServer,CmdletProxyRemoteServerVersion,CmdletProxyMethod,ProxiedObjectCount,CmdletProxyLatency,OutputObjectCount,ParameterBinding,BeginProcessing,ProcessRecord,EndProcessing,StopProcessing,BizLogic,PowerShellLatency,UserInteractionLatency,ProvisioningLayerLatency,ActivityContextLifeTime,TotalTime,ErrorType,ExecutionResult,CacheHitCount,CacheMissCount,GenericLatency,GenericInfo,GenericErrors,ObjectGuid,ExternalDirectoryOrganizationId,ExternalDirectoryObjectId,NonPiiParameters
```

I've decided to take a snippet of an history log file. It contains which user ran which commands and so on. This can be very useful if someone cleaned the MExchangeManagement event logs. Since all the history logs will be still there on disk.

As we can see here, there is a user that is exporting all the mailboxes to the C:\Temp directory.

At the second example, we can see that a SMTP forwarding rule is created to forward the e-mails from Leon Edwards to an external domain.

Last example, we can see that a role assignment was initiated to assign a user to the Recipient Management role in Exchange.

## Exchange CosmosQueue Logs

Exchange CosmosQueue logs are like the audit logs in Exchange. It shows more of the operational activities that were performed in Exchange. This can include examples such as creating a new Database Availability Group (DAG) or removing a Mailbox database, putting the Exchange server in maintenance mode, and so on. All the logs are located at: **C:\Program Files\Microsoft\Exchange Server\V15\Logging\CosmosQueue** and have a similar filename such as **audit20221004-4**.

This is how the structure of the Exchange CosmosQueue logs look like:

```
#Software: Microsoft Exchange
#Version: 15.01.1713.001
#Log-type: audit
#Date: 2022-10-04T13:29:22.251Z
#Fields:
```

```
Timestamp, Server, TenantId, RecordType, Data, UserKey, RecordId, Operation, Workload  
, ResultStatus, Version, Scope
```

Here we can a snippet of some of the operational tasks in Exchange being logged:

This includes also the associated user that performed this operational activity.

## Exchange Control Panel – Activity Logs

Exchange Control Panel is like the admin panel for Exchange. Administrative tasks can be performed in this panel, and this activity is logged as well. All the activity logs of ECP are stored at the following location: **C:\Program Files\Microsoft\Exchange Server\V15\Logging\ECP\Activity** and it has similar filename as **ECPActivity\_9296\_20221004-1**.

This is how the structure of the ECP activity logs look like:

```
#Software: Microsoft Exchange Server  
#Version: 15.0.0.0
```



```
#Log-type: ECP Activity Context Log
#Date: 2022-10-04T09:20:35.883Z
#Fields: TimeStamp, ServerName, EventId, EventData
```

This is how the logs may look like:

Let's take a closer look at some of the ECP activity logs. I will include a couple of examples.

Here we can see that the user Testing was creating a new user mailbox in ECP.

```
2022-10-01T10:47:36.658Z, EXCHANGE, Request, S:PSA=
<PII>Testing@contoso.com</PII>; S:FE=EXCHANGE.CONTOSO.COM; S:URL=https://exchan
ge.contoso.com:444/ecp/UsersGroups/NewMailboxOnPremises.aspx?
pwmcid=3&ReturnObjectType=1 (https://exchange.contoso.com/ecp/UsersGroups/NewM
ailboxOnPremises.aspx?
pwmcid=3&ReturnObjectType=1); S:Bld=15.1.1713.5; S:ActID=3bbbf798-1dc1-40f8-
a56b-
0611baa1065d; Db1: BudgUse.T[]=7.00040006637573; I32:ADS.C[DC]=1; F:ADS.AL[DC]=1.
592; I32:ATE.C[DC.contoso.com]=0; F:ATE.AL[DC.contoso.com]=0; S:WLM.Bal=2.147484
E+09; Db1:WLM.TS=259
```

A new mailbox database was created by Testing.

```
2022-10-04T18:23:44.745Z, EXCHANGE, Request, S:PSA=
<PII>Testing@contoso.com</PII>; S:FE=EXCHANGE.CONTOSO.COM; S:URL=https://exchan
ge.contoso.com:444/ecp/DBMgmt/NewDatabase.aspx?
pwmcid=16&ReturnObjectType=1 (https://exchange.contoso.com/ecp/DBMgmt/NewDatab
```

```
ase.aspx?pwmcid=16&ReturnObjectType=1);S:Bld=15.1.1713.5;S:ActID=d9a2ef05-ef68-4e47-a4d9-d2a80d6180ac;Db1:WLM.TS=40
```

New Database Availability Group (DAG) was created in ECP.

```
2022-10-04T18:11:54.303Z,EXCHANGE,Request,S:PSA=<PII>Testing@contoso.com</PII>;S:FE=EXCHANGE.CONTOSO.COM;S:URL=https://exchange.contoso.com:444/ecp/DBMgmt/NewDAG.aspx?pwmcid=2&ReturnObjectType=1(https://exchange.contoso.com/ecp/DBMgmt/NewDAG.aspx?pwmcid=2&ReturnObjectType=1);S:Bld=15.1.1713.5;S:ActID=7abb1d23-ee8d-44a5-a3ac-536cdbaefc2;Db1:WLM.TS=53
```

Share this:



## 2 COMMENTS

**Ban**

December 19, 2022 2:38 am

Thank you for amazing writeup

★ Like

Reply

**RSA**

July 12, 2024 3:27 am

Thanks a lot this was helpful

★ Like

Reply

## LEAVE A COMMENT

---

---

Search ...



BUY ME A COFFEE





#### POPULAR THIS WEEK

- [Hunting in On-Premises Exchange Server logs](#)
- [Investigating Certificate Template Enrollment Attacks - \(ADCS\)](#)
- [Everything about Service Principals, Applications, and API Permissions](#)
- [Practical Compromise Recovery Guidance for Active Directory](#)
- [History of Exchange with having wide permissions in AD](#)

## CATEGORIES

- [M365 Advanced Hunting](#)
- [Azure Sentinel](#)
- [Azure Active Directory](#)
- [KQL](#)
- [Microsoft Identity](#)
- [Windows OS](#)
- [Jupyter Notebooks](#)

## CONTACT



## RECENT POSTS

- [Investigating Certificate Template Enrollment Attacks – \(ADCS\)](#)
- [How one misconfiguration in ADCS can lead to full AD Forest compromise](#)
- [Investigating Ransomware Deployments that happened via Group Policy](#)
- [Hunting and Responding to ProxyShell Attacks](#)
- [Investigating ProxyLogon Attacks and how to mitigate it](#)

## TAGS

[Jupyter Notebooks](#)