



Calendaring in Teams & Exchange (on-prem)

Ralf Leistner

ralf.leistner@microsoft.com



Agenda

Architecture and requirements

Autodiscover v2

Teams Calendar App

Meeting planning for Delegates

Presence status based on calendar items



Where Teams and Exchange interact

Calendar
App
data

Delegate
Meeting
Planning

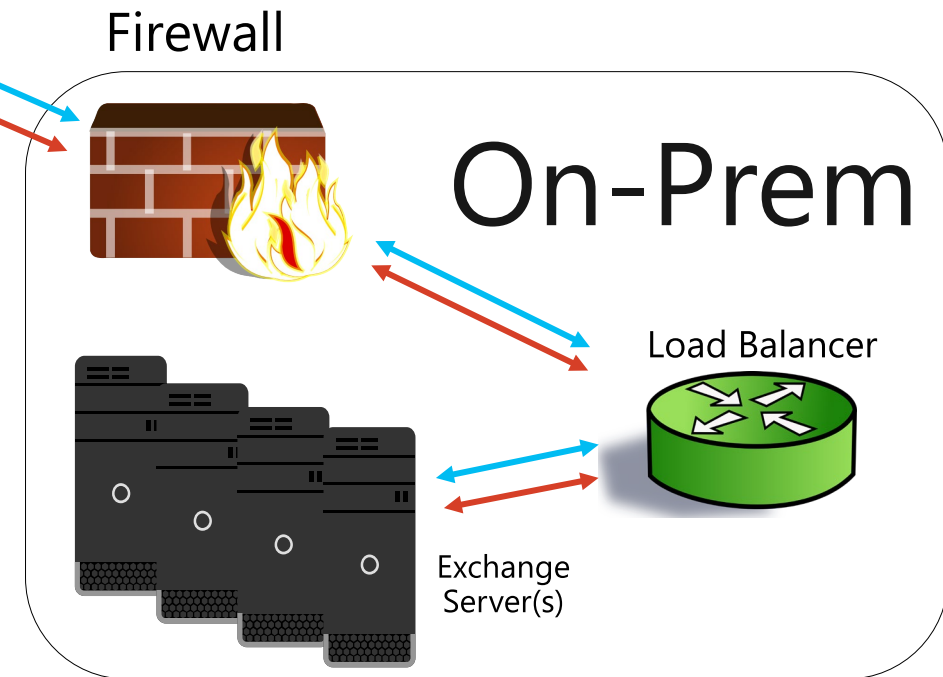
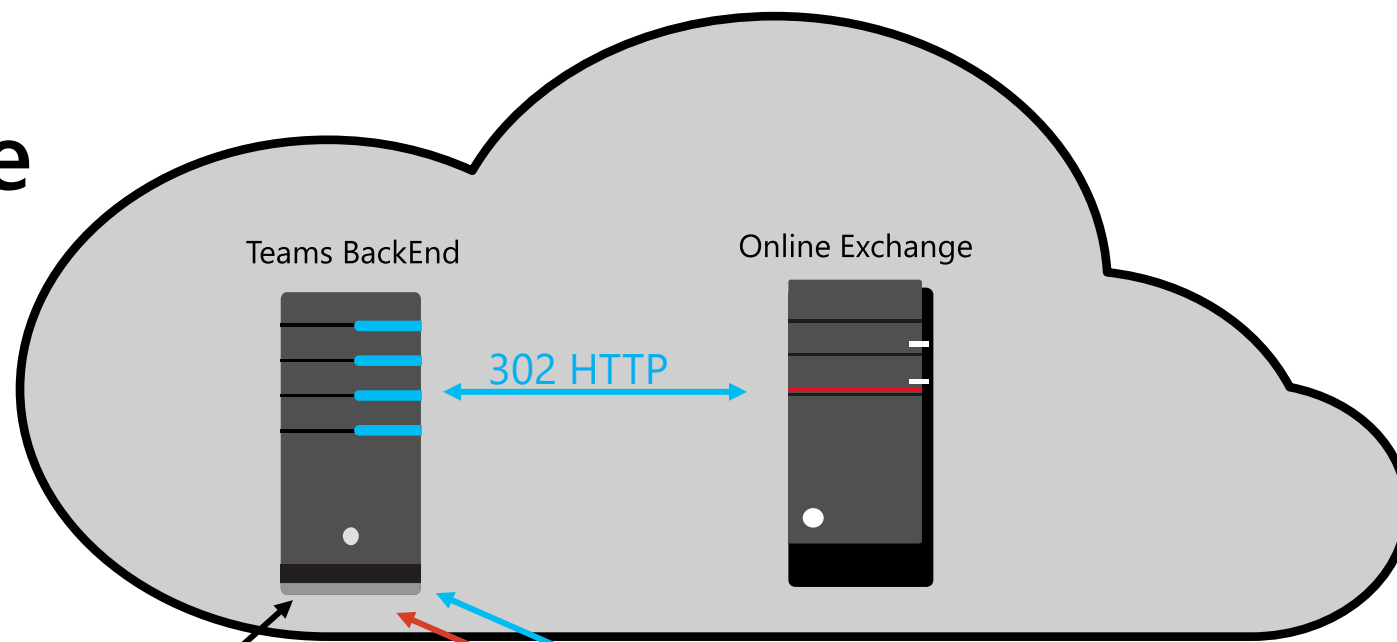
Presence
Status
Auto-Update

Autodiscover v2

Architecture



Microsoft 365



- Autodiscover
- Exchange Web Services / REST



Requirements - Teams interact with Exchange on-prem

- Exchange 2016 or higher (2013 for the Delegate scenario) on latest CU
- AAD-Connect to sync identities
- Teams licenses assigned
- oAuth to be configured between on-prem and Azure/M365
 - Using the Exchange HCW is preferred, manual steps documented (Classic Hybrid!!!)
- Exchange EWS/REST API and Autodiscover published
- EWS namespace to be configured as SPN in AAD (Outlook MAC only)
- Fully configured Partner Application (Delegate scenario only)

<https://docs.microsoft.com/en-us/skypeforbusiness/deploy/integrate-with-exchange-server/oauth-with-online-and-on-premises>



Knowledge Check



Key Point:

How many daily active users do we have in Teams (Oct. 2020)?



A

17Mio

B

73Mio

C

115Mio

D

542Mio



Teams Autodiscover

- Autodiscover to retrieve EWS and REST API locations
- Teams uses Autodiscover v2 JSON (anonymous)
- Is done by Teams Backend
- Process:
 1. Teams sends JSON to Exchange Online
 2. Exchange Online checks the recipient type
 - a) If it's a mailbox, return:
<https://outlook.office365.com/EWS/Exchange.asmx> (Done!)
 - b) If it's mail user, calculate a DNS Autodiscover endpoint (based on ExternalEmailAddress) -> i.e. autodiscover.contoso.com
 3. Send HTTP 302 redirect including the endpoint
 4. Teams sends JSON to the on-prem endpoint
 5. Exchange on-prem returns EWS/REST URL



Troubleshoot Autodiscover I

- Don't trace the Teams client
- AutodiscoverV2 is anonymous -> you can test yourself
- Use the PowerShell or Browser and a customer E-Mail-Address:

```
PS C:\> Invoke-RestMethod -Uri  
"https://outlook.office365.com/autodiscover/autodiscover.json?Email=jdoe@contoso.com&Protocol=EWS"
```

In the Browser:

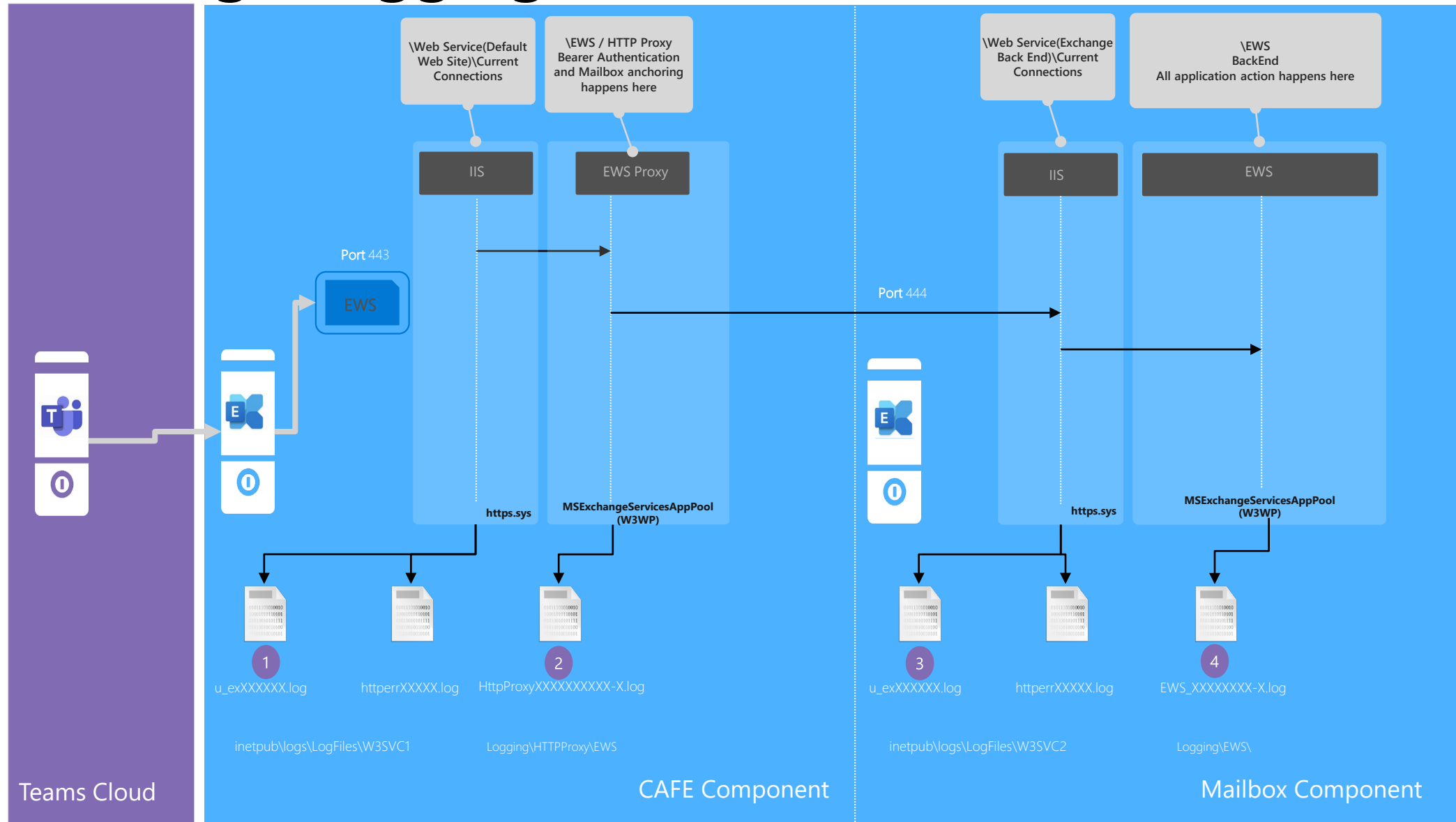
<https://outlook.office365.com/autodiscover/autodiscover.json?Email=jdoe@contoso.com&Protocol=EWS>

<https://outlook.office365.com/autodiscover/autodiscover.json?Email=jdoe@contoso.com&Protocol=REST>

- Collect a fiddler trace while testing
- A failure (esp. a timeout) doesn't indicate a problem -> restricted access to O365 IP-Ranges?
- Check the Autodiscover Logs to verify the request reaches the server
- Check the IIS logs for Teams requests
- Collect a network trace and check for Teams request, i.e. TLS handshake errors etc.



Exchange Logging (i.e. EWS)



Teams Cloud

CAFE Component

Mailbox Component



Troubleshoot Autodiscover II

Autodiscover returns unexpected EWS/REST URLs:

- a) The user's mailbox is on-prem but you receive:
"https://outlook.office365.com/EWS/Exchange.asmx"

This can happen if for the user exists a duplicate mailbox in Exchange Online.

- b) Something that looks like an "internal" URL:
"https://ex-srv1.contoso.local/EWS/Exchange.asmx"

The EWS "ExternalUrl" attribute is not populated on **ALL** Exchange servers.

- c) The user's mailbox is hosted on an EMEA Server, but you receive an APAC or NAM belonging namespace.

AutodiscoverV2 is NOT site aware (as of today). We're working into that



The Teams Calendar App

- Teams uses Exchange Web Services to retrieve calendar data
- Reminder: the Teams client does NOT connect to Exchange

How the process works:

1. The EWS URL is received by Autodiscover
2. Teams authenticates using OAuth on Exchange on-prem
3. It sends EWS requests to Exchange Web Services
4. Exchange sends back the calendar data
5. Teams Middle-Tier populates the App using the data



Troubleshooting Calendar App issues

- Verify if the URL has is in public DNS and reachable from external
 - Start fiddler and paste the URL into a browser to get 401 (again: timeout doesn't indicate a problem), check is the returned data origins from Exchange
- Try the Remote Connectivity Analyzer, new Teams test since this week!
- Verify that oAuth works correctly
 - Test-OAuthConnectivity
 - Check IOC, SPNs, AuthServers
- Try the Team Connectivity Analyzer Test
<https://testconnectivity.microsoft.com/tests/teams>

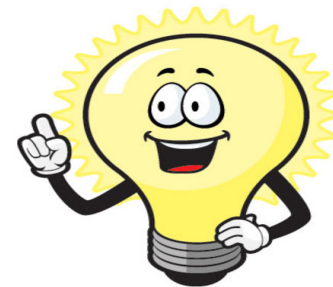
```
Raw | JSON | XML
HTTP/1.1 401 Unauthorized
Server: Microsoft-IIS/10.0
request-id: aa749f76-760c-41f1-98fd-5f7d5446caed
X-WSSecurity-Enabled: True
X-WSSecurity-For: None
X-OAuth-Enabled: True
WWW-Authenticate: Negotiate
WWW-Authenticate: NTLM
X-Powered-By: ASP.NET
X-FEServer: EX16
Date: Mon, 07 Dec 2020 13:20:45 GMT
Content-Length: 0
Proxy-Support: Session-Based-Authentication
```

Synchronization, Notification, Availability, and Automatic Replies

These tests walk through many basic Exchange Web Services tasks to confirm they're working. This is useful for IT administrators who want to troubleshoot external access using Entourage EWS or other Web Services clients.



Knowledge Check



Key Point:

How many times did users run the Teams Connectivity Analyzer scenario since it launched in Dec 2020?

A 136

B 484

C 778

D 1115





The EWS Allow- and BlockList Myth

- Covered in all UC Wiki and public articles
- By default, there's no limits
- Avoid sending action plans to configure something else as the defaults
- 99% of all customer don't use the lists
- There's more cases where "user agents" are filtered outside of Exchange
- Maintaining the EWS Lists is not a security feature

```
[PS] C:\temp>Get-OrganizationConfig | fl *EWS*

EwsAllowEntourage      :
EwsAllowList           :
EwsAllowMacOutlook     :
EwsAllowOutlook        :
EwsApplicationAccessPolicy :
EwsBlockList           :
EwsEnabled              :

[PS] C:\temp>Get-CASMailbox tom | FL *EWS*

EwsEnabled              :
EwsAllowOutlook        :
EwsAllowMacOutlook     :
EwsAllowEntourage      :
EwsApplicationAccessPolicy :
EwsAllowList           :
EwsBlockList           :
```

User Agent List:

Calendar: MicrosoftNinja/1.0 Teams/1.0 (ExchangeServicesClient/0.0.0.0) SkypeSpaces/1.0a\$*+

Delegate: SchedulingService

Presence: Microsoft.Skype.Presence.App/1.0

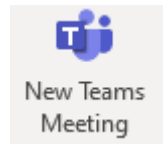


The Teams Delegate Meeting scenario

- Teams uses Exchange Web Services to retrieve a list of delegates and their permissions
- Limited to the Outlook Team Add-in, not supported in OWA on-prem
- Delegates need to be setup using the Outlook Delegation wizard

How the process works:

1. Delegate hits the "Teams Meeting" button in Outlook in manager's calendar
2. The Add-in connects to Teams Middle-tier
3. It does an Autodiscover to receive the EWS URL
4. Teams connects to /EWS and authenticates using oAuth on Exchange on-prem
5. It sends EWS a GetDelegate SOAP request for the "manager's" mailbox
6. Exchange sends back the delegate list including their folder permission
7. Teams Middle-Tier provides the meeting data to the Add-in



Microsoft Teams meeting

Join on your computer or mobile app

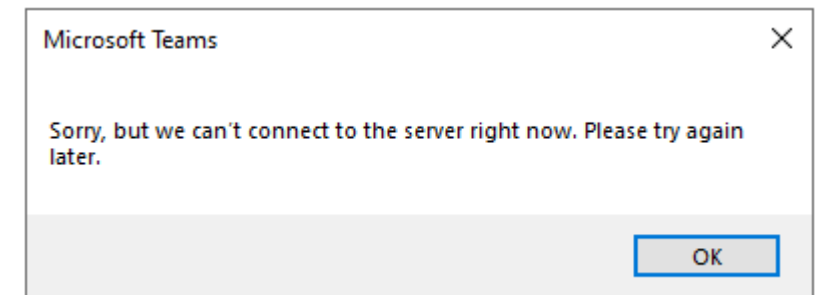
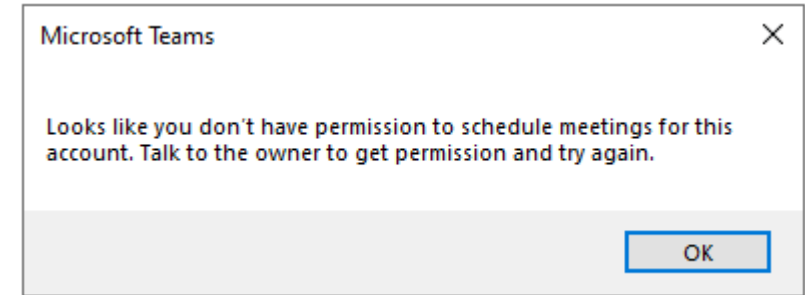
[Click here to join the meeting](#)

Or call in (audio only)



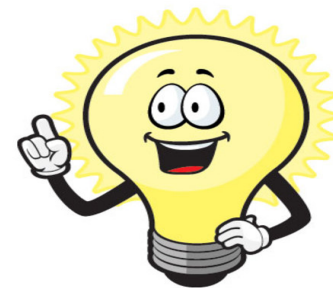
Troubleshooting Delegate issues

- Verify if the Teams calendar app works correctly
- Use SOAPe client to manually send a GetDelegate and check the answer
- Clientside fiddler you'll get "StatusCode": 403
- Adding calendar permissions is not sufficient
- Check EWS logs for the GetDelegate call? Is it a 200?
- Check PartnerApp configuration
- Check EWS Proxy and IIS FrontEnd logs
- Collect a network trace
- Eliminate (3rd party) layer 7 devices (user agent filtering) 😊





Knowledge Check



Key Point:

Gues the number of hits for the "How Exchange and Microsoft Teams interact" doc page in February 2021 so far?

<https://docs.microsoft.com/en-us/MicrosoftTeams/exchange-teams-interact>



A 19.575

B 7840

C 11.463

D 1217



The Presence Status based on calendar

- Users want status to be “In a Meeting” automatically
- Teams BackEnd uses REST API vDir to retrieve the data from Exchange
- Yes really, same data that is already in the Calendar app
- The clients pulls data every 6min from the Presence service
- Two modes:
 1. Pull mode, once per hour
 2. Push mode based on subscription
- Limited functionality in on-prem, Pull mode only. Subscription not available in Exchange on-prem.
- **Delegates need to be setup using the Outlook Delegation wizard**

How the process works:

1. Teams MT does an Autodiscover to receive the REST API URL
2. Teams connects to /api and authenticates using oAuth on Exchange on-prem
3. It tries to create a subscription -> fails RPC endpoint not found
4. It sends a get calendar request



Troubleshooting Presence Status issues

- Verify if the /api is returned by Autodiscover and is accessible/published, maybe use Fiddler
- Update the REST CAFE web.config (fixed in Exchange Dec. 2020 CUs)

C:\Program Files\Microsoft\Exchange Server\V15\FrontEnd\HttpProxy\Rest\web.config

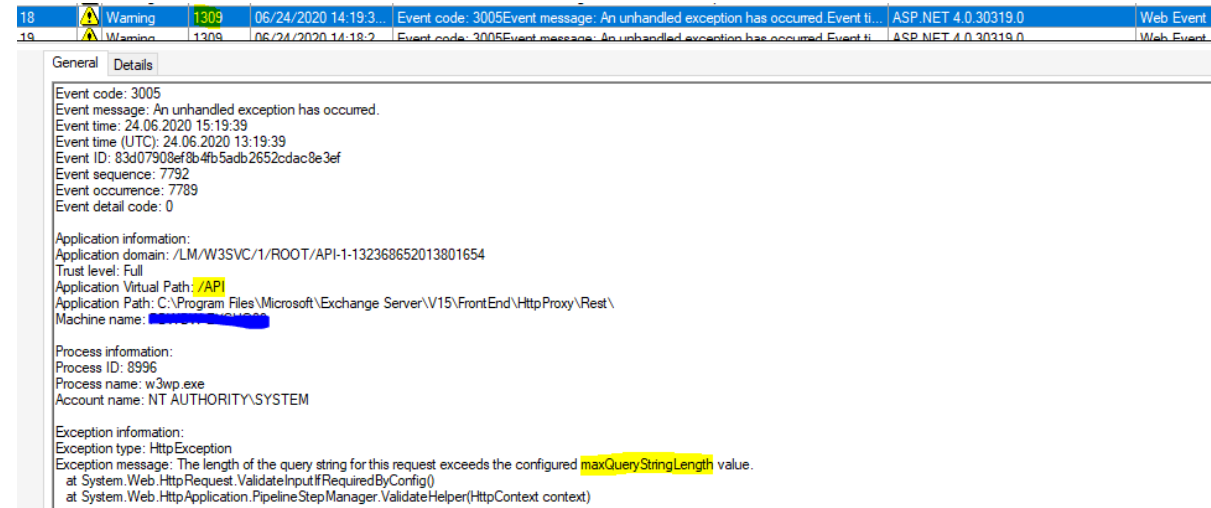
from:

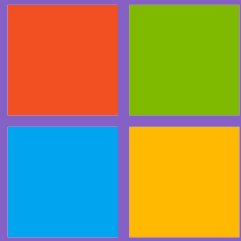
```
<httpRuntime maxRequestLength="2097151" maxUrlLength="2048" requestPathInvalidCharacters="&lt;,>,*,%,\,?" requestValidationMode="2.0" />
```

to:

```
<httpRuntime maxRequestLength="2097151" maxUrlLength="2048" maxQueryStringLength="4096" requestPathInvalidCharacters="&lt;,>,*,%,\,?" requestValidationMode="2.0" />
```

- Start looking to the REST logs, again filter by user agent string
- Check REST Proxy and IIS FrontEnd logs
- Eliminate (3rd party) layer 7 devices (user agent filtering) 😊





Microsoft

© 2021 Microsoft Corporation. All rights reserved. Microsoft, Windows, and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries. The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.