RSAConference2022

Session ID: HTA-W08

The SaaS RootKit: A New Attack Vector to Create Hidden Forwarding Rules in O365

Maor Bin, CEO & Co-Founder
Adaptive Shield

Disclaimer

Presentations are intended for educational purposes only and do not replace independent professional judgment. Statements of fact and opinions expressed are those of the presenters individually and, unless expressly stated to the contrary, are not the opinion or position of RSA Conference LLC or any other co-sponsors. RSA Conference does not endorse or approve, and assumes no responsibility for, the content, accuracy or completeness of the information presented.

Attendees should note that sessions may be audio- or video-recorded and may be published in various media, including print, audio and video formats without further notice. The presentation template and any media capture are subject to copyright protection.

DISCOVERY

An attack vector due to a vulnerability within Microsoft's OAuth application registration.

Through this vulnerability, **one can leverage Exchange's legacy API to create hidden forwarding rules in M365 mailboxes**.

Agenda

- Background: A Look into the Inbox Rules of Microsoft 365
  - Hidden Forwarding Rules Discovery
- The Next Evolution in Hidden Forwarding Rules: The SaaS Rootkit
- Demo part 1
- OAuth and 3rd Party Apps Access
- Demo part 2
- Discussion
- Mitigation Strategies
- Summary

What Are Inbox Rules in Microsoft 365?

Actions that occur based on preset conditions within your Microsoft mailbox.

Example use cases:

- Auto-mark the importance level of incoming messages
- Automatically delete outgoing emails
- **Automatically forward incoming emails**

Inbox Forwarding Rules

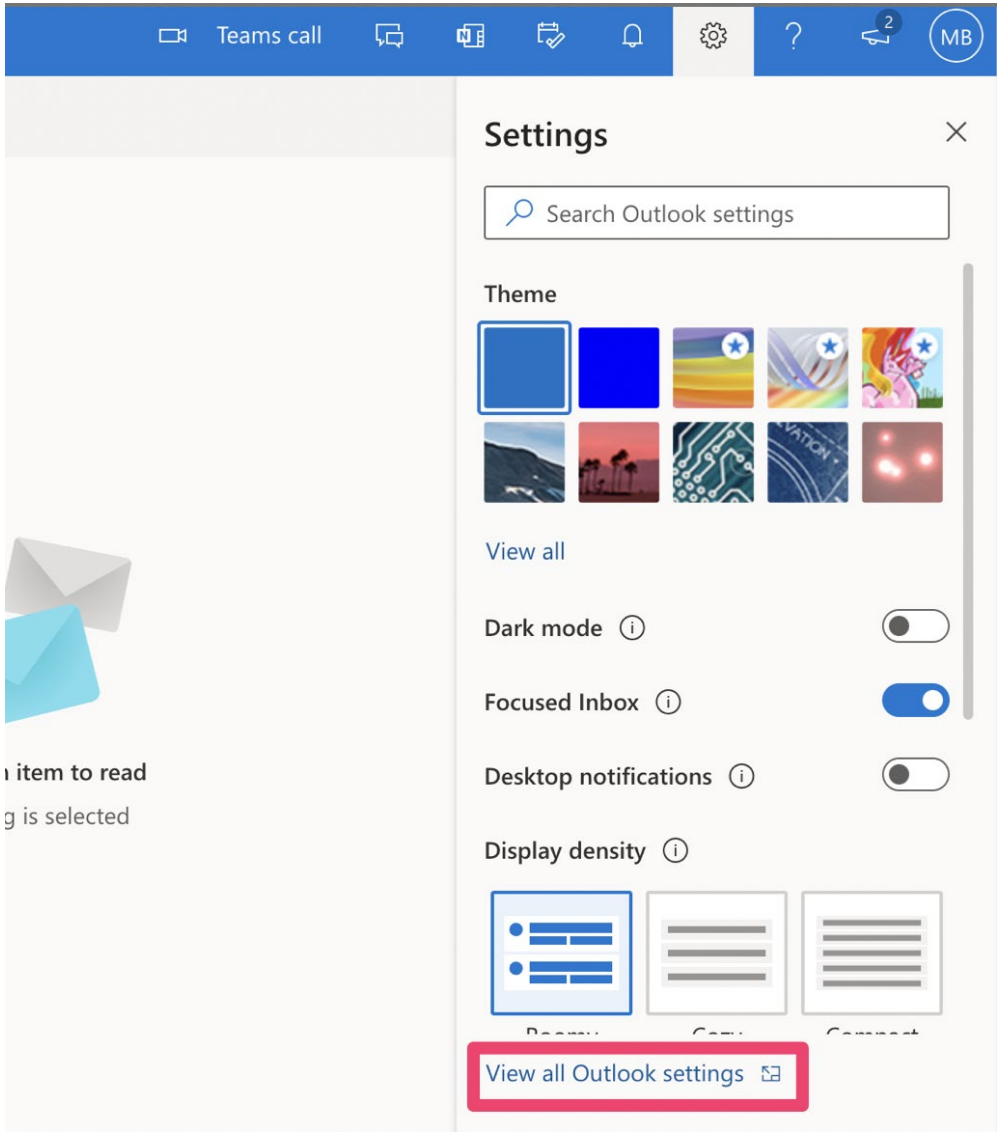
Why are there forwarding rules?

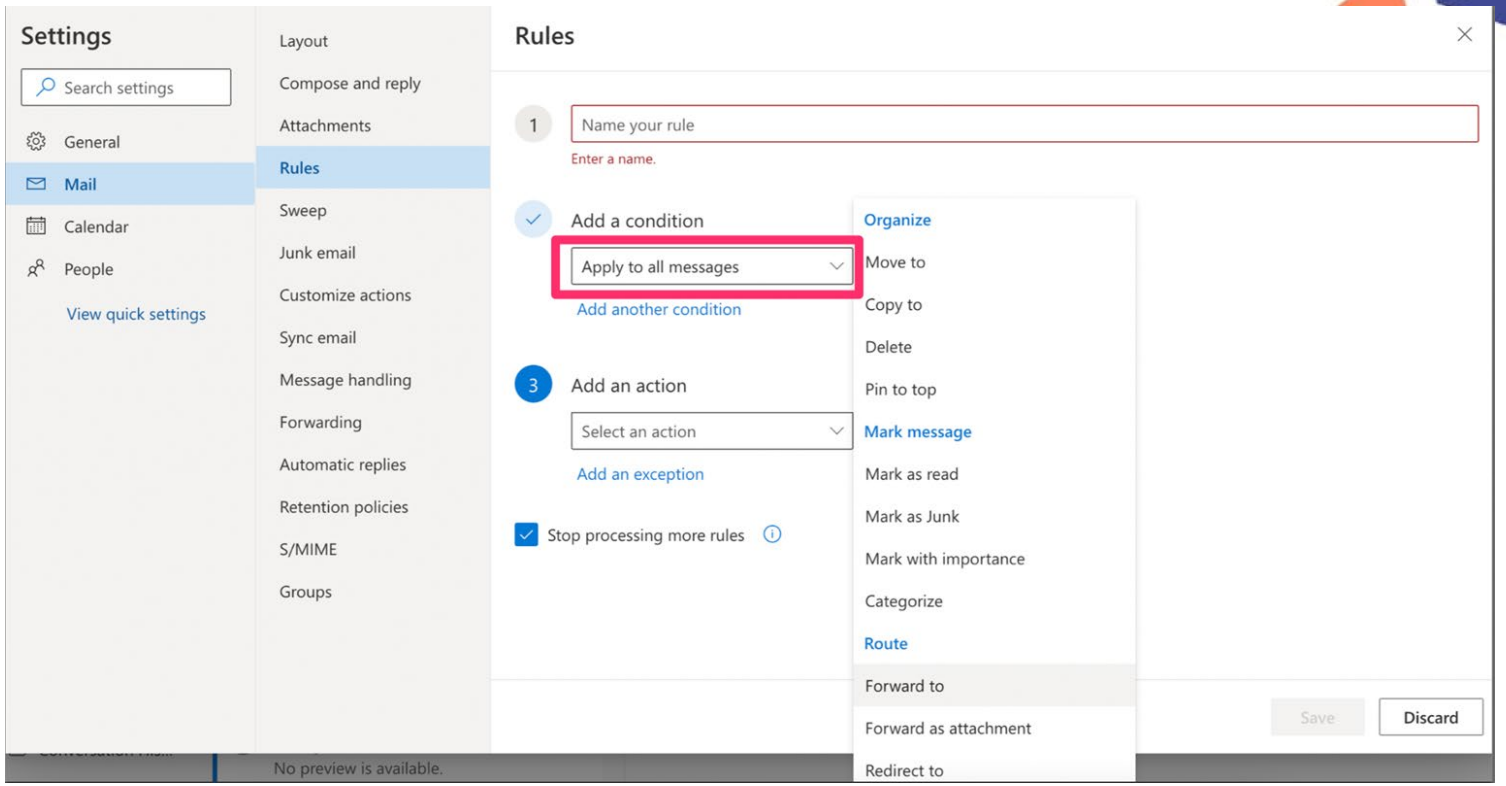
A company wants to set up email forwarding for a specific user's mailbox

How can they be configured?

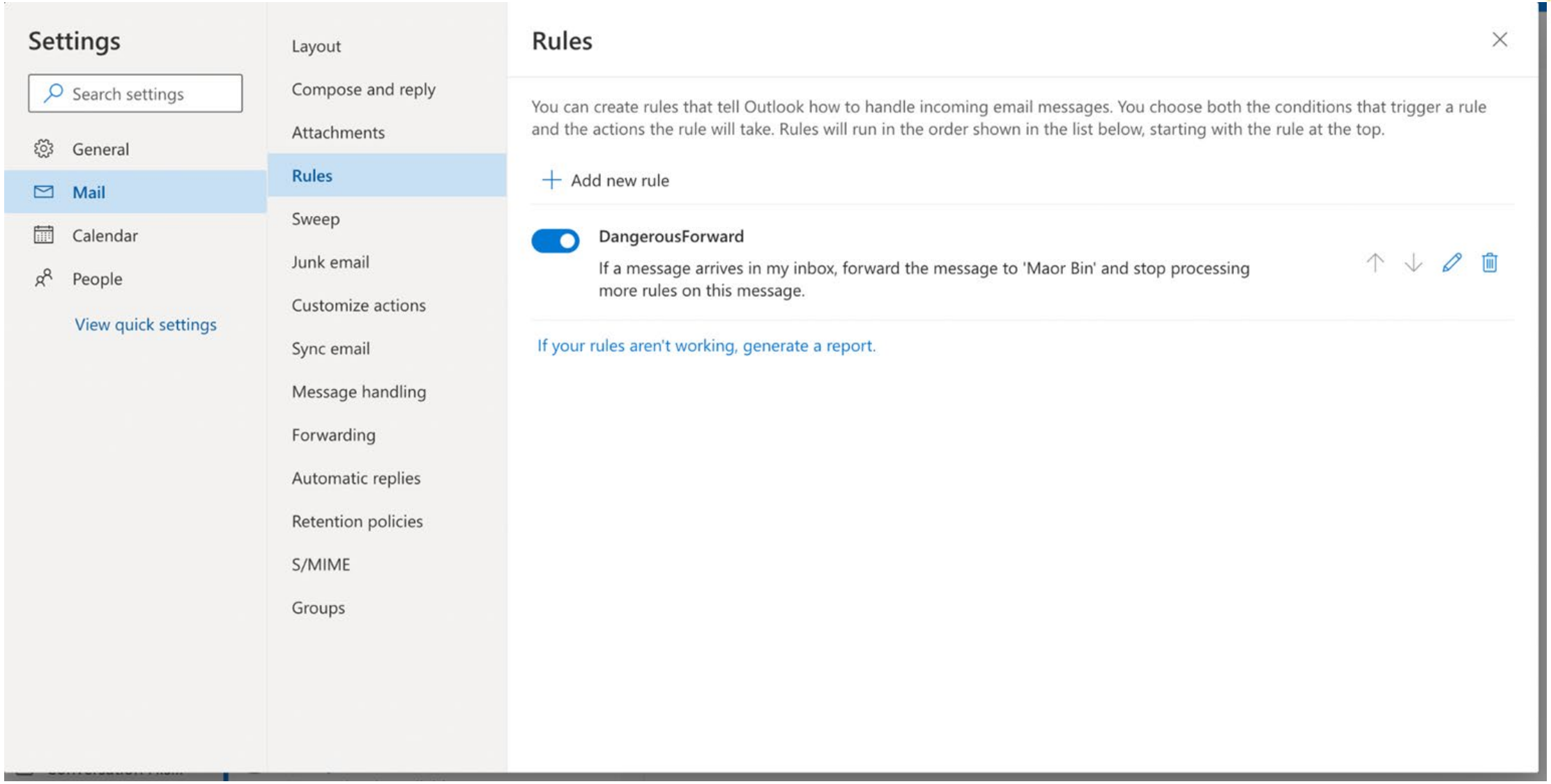
- Admins usually use **ForwardingSMTPAddress** or **ForwardingAddress**
- Users can set up **Mail-Flow Rules** or **Inbox Rules:** Inbox Rules can trigger different forwarding rules based on different attributes of the user's inbox

How Do User's Create an Inbox Rule?





The Inbox Rule Is Created! The Frontend Experience



The Inbox Rule Is Created! The Backend Experience

An example of a raw inbox forwarding rule

(IPM.Rule.Version2.Message)

Inbox (Hidden Contents): Display Name Not Found

Actions Folder Search Property Table Tools

Received	Submitted	Message Class	Size	Message Flags	EID	Longterm EID
06:01:59 20.07.2018	06:01:59 20.07.2018	IPM.RuleOrganizer	2429	1097 (MSGFLAG_READ | ...	cb: 42 lpb: EF000000DC8...	cb: 70 lpb: 0
06:48:40 20.07.2018	06:48:40 20.07.2018	IPM.Rule.Version2.Mess	2104	1097 (MSGFLAG_READ | ...	cb: 42 lpb: EF000000DC8...	cb: 70 lpb: 0
02:38:24 19.07.2018	02:38:24 19.07.2018	IPM.Microsoft.Migratio...	1084	1097 (MSGFLAG_READ | ...	cb: 42 lpb: EF000000DC8...	cb: 70 lpb: 0
06:33:18 20.07.2018	06:33:18 20.07.2018	IPM.MessageManager	1258	1088 (MSGFLAG_ASSOCI...	cb: 42 lpb: EF000000DC8...	cb: 70 lpb: 0
08:10:46 19.07.2018	08:10:46 19.07.2018	IPM.ExtendedRule.Mess	1682	1097 (MSGFLAG_READ | ...	cb: 42 lpb: EF000000DC8...	cb: 70 lpb: 0
02:38:24 19.07.2018	02:38:24 19.07.2018	IPM.Configuration.Table...	1354	1097 (MSGFLAG_READ | ...	cb: 42 lpb: EF000000DC8...	cb: 70 lpb: 0

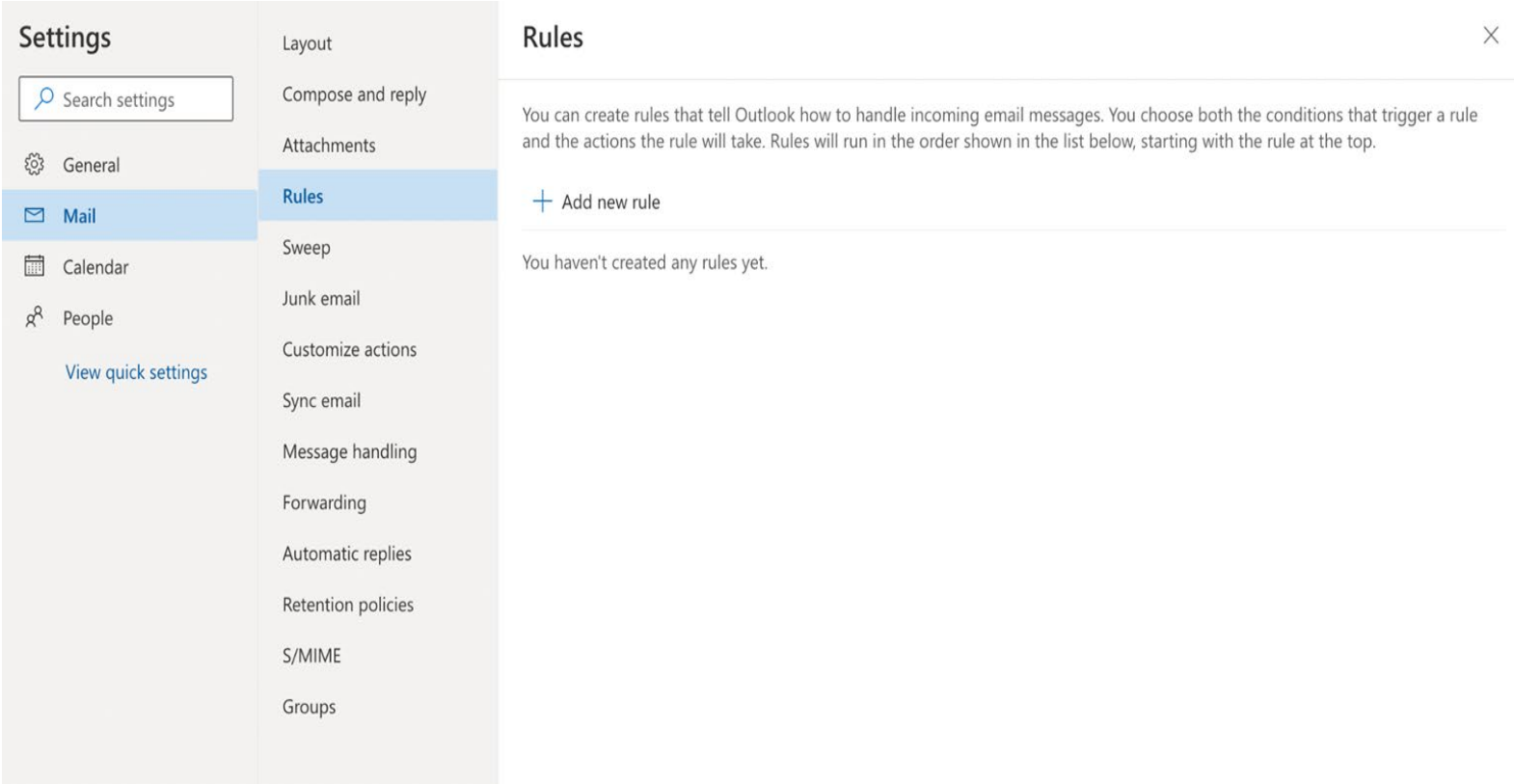
Name	Other Names	Tag	Type	Value	Value (alternate view)	Smart
PR_REPLICA_VERSION		0x664B0014	PT_I8	0x0F000005:0x834F0FC8	1080863934246752200	
PR_RTF_COMPRESSED	PidTagRtfCompressed, ptagRTFC...	0x1009000A	PT_ERROR	Err: 0x8004010F=MAPI_E_NOT_FO	Body: Open to view	
PR_RTF_IN_SYNC	PidTagRtfInSync, ptagRTFInSync	0x0E1F000B	PT_BOOLEAN	False		
PR_RULE_MSG_LEVEL	PidTagRuleMessageLevel, ptagRul...	0x65ED0003	PT_LONG	0	0x0	
PR_RULE_MSG_NAME	PR_RULE_MSG_NAME_A, PidTagR...	0x65EC001E	PT_STRING8	DangerousForward	cb: 16 lpb: 44616E6765726F757346...	
PR_RULE_MSG_PROVIDER	PR_RULE_MSG_PROVIDER_A, ptag...	0x65EB001E	PT_STRING8	RuleOrganizer	cb: 13 lpb: 52756C654F7267616E69...	
PR_RULE_MSG_PROVIDER_DATA	PidTagRuleMessageProviderData, ...	0x65EE0102	PT_BINARY	cb: 16 lpb: 0100000001000000BCB...	.....¼»»»«$å@	
PR_RULE_MSG_SEQUENCE	PidTagRuleMessageSequence, pta...	0x65F30003	PT_LONG	10	0xA	
PR_RULE_MSG_STATE	PidTagRuleMessageState, ptagRul...	0x65E90003	PT_LONG	1	0x1	Flags
PR_RULE_MSG_USER_FLAGS	PidTagRuleMessageUserFlags, pta...	0x65EA0003	PT_LONG	0	0x0	
PR_SEARCH_KEY	PidTagSearchKey, ptagSearchKey	0x300B0102	PT_BINARY	cb: 16 lpb: CC980A5F854A94438F3...	Ì..._J.C8{|°òK.	
PR_SENDER_ADDRTYPE	PR_SENDER_ADDRTYPE_A, PR_SE...	0x0C1E000A	PT_ERROR	Err: 0x8004010F=MAPI_E_NOT_FO		
PR_SENDER_EMAIL_ADDRESS	PR_SENDER_EMAIL_ADDRESS_A, ...	0x0C1F000A	PT_ERROR	Err: 0x8004010F=MAPI_E_NOT_FO		
PR_SENDER_ENTRYID	PidTagSenderEntryId, ptagSender...	0x0C19000A	PT_ERROR	Err: 0x8004010F=MAPI_E_NOT_FO		
PR_SENDER_NAME	PR_SENDER_NAME_A, PR_SENDE...	0x0C1A000A	PT_ERROR	Err: 0x8004010F=MAPI_E_NOT_FO		
PR_SENSITIVITY	PidTagSensitivity, ptagSensitivity	0x00360003	PT_LONG	0	0x0	Flags
PR_SENT_REPRESENTING_ADD...	PR_SENT_REPRESENTING_ADDRT...	0x0064000A	PT_ERROR	Err: 0x8004010F=MAPI_E_NOT_FO		
PR_SENT_REPRESENTING_EMAI...	PR_SENT_REPRESENTING_EMAIL_...	0x0065000A	PT_ERROR	Err: 0x8004010F=MAPI_E_NOT_FO		
PR_SENT_REPRESENTING_ENTR	PidTagSentRepresentingEntryId, p...	0x0041000A	PT_ERROR	Err: 0x8004010F=MAPI_E_NOT_FO		
PR_SENT_REPRESENTING_NAME	PR_SENT_REPRESENTING_NAME	0x0042000A	PT_ERROR	Err: 0x8004010F=MAPI_E_NOT_FO		

The Discovery of *Hidden* Inbox Forwarding Rules

Damian Pfammater found an undocumented method that can be used to hide these types inbox rules.

These hidden forwarding rules:

- Are functional
- Are **NOT** visible through common interfaces (Email clients, Admin Dashboard, or API)



For Reference: BLOG POST SEPTEMBER 17, 2018

Back to the Backend: How Can the User Hide an Inbox Rule?

Tamper this object:
PR_RULE_MSG_PROVIDER

(could be empty or malformed)

Inbox (Hidden Contents): Display Name Not Found

Actions Folder Search Property Table Tools

Received	Submitted	Message Class	Size	Message Flags	EID	Longterm EID
06:01:59 20.07.2018	06:01:59 20.07.2018	IPM.RuleOrganizer	2429	1097 (MSGFLAG_READ | ...	cb: 42 lpb: EF000000DC8...	cb: 70 lpb: 0
06:48:40 20.07.2018	06:48:40 20.07.2018	IPM.Rule.Version2.Mess	2104	1097 (MSGFLAG_READ | ...	cb: 42 lpb: EF000000DC8...	cb: 70 lpb: 0
02:38:24 19.07.2018	02:38:24 19.07.2018	IPM.Microsoft.Migratio...	1084	1097 (MSGFLAG_READ | ...	cb: 42 lpb: EF000000DC8...	cb: 70 lpb: 0
06:33:18 20.07.2018	06:33:18 20.07.2018	IPM.MessageManager	1258	1088 (MSGFLAG_ASSOCI...	cb: 42 lpb: EF000000DC8...	cb: 70 lpb: 0
08:10:46 19.07.2018	08:10:46 19.07.2018	IPM.ExtendedRule.Mess	1682	1097 (MSGFLAG_READ | ...	cb: 42 lpb: EF000000DC8...	cb: 70 lpb: 0
02:38:24 19.07.2018	02:38:24 19.07.2018	IPM.Configuration.Table...	1354	1097 (MSGFLAG_READ | ...	cb: 42 lpb: EF000000DC8...	cb: 70 lpb: 0

Name	Other Names	Tag	Type	Value	Value (alternate view)	Smart
PR_REPLICA_VERSION		0x664B0014	PT_I8	0x0F000005:0x834F0FC8	1080863934246752200	
PR_RTF_COMPRESSED	PidTagRtfCompressed, ptagRTFC...	0x1009000A	PT_ERROR	Err: 0x8004010F=MAPI_E_NOT_FO	Body: Open to view	
PR_RTF_IN_SYNC	PidTagRtfInSync, ptagRTFInSync	0x0E1F000B	PT_BOOLEAN	False		
PR_RULE_MSG_LEVEL	PidTagRuleMessageLevel, ptagRul...	0x65ED0003	PT_LONG	0	0x0	
PR_RULE_MSG_NAME	PR_RULE_MSG_NAME_A, PidTagR...	0x65EC001E	PT_STRING8	DangerousForward	cb: 16 lpb: 44616E6765726F757346...	
PR_RULE_MSG_PROVIDER	PR_RULE_MSG_PROVIDER_A, ptag...	0x65EB001E	PT_STRING8	RuleOrganizer	cb: 13 lpb: 52756C654F7267616E69...	
PR_RULE_MSG_PROVIDER_DATA	PidTagRuleMessageProviderData, ...	0x65EE0102	PT_BINARY	cb: 16 lpb: 0100000001000000BCB...	.....¼»»»«$å@	
PR_RULE_MSG_SEQUENCE	PidTagRuleMessageSequence, pta...	0x65F30003	PT_LONG	10	0xA	
PR_RULE_MSG_STATE	PidTagRuleMessageState, ptagRul...	0x65E90003	PT_LONG	1	0x1	Flags
PR_RULE_MSG_USER_FLAGS	PidTagRuleMessageUserFlags, pta...	0x65EA0003	PT_LONG	0	0x0	
PR_SEARCH_KEY	PidTagSearchKey, ptagSearchKey	0x300B0102	PT_BINARY	cb: 16 lpb: CC980A5F854A94438F3...	Ì.._J.C8{|°òK.	
PR_SENDER_ADDRTYPE	PR_SENDER_ADDRTYPE_A, PR_SE...	0x0C1E000A	PT_ERROR	Err: 0x8004010F=MAPI_E_NOT_FO		
PR_SENDER_EMAIL_ADDRESS	PR_SENDER_EMAIL_ADDRESS_A, ...	0x0C1F000A	PT_ERROR	Err: 0x8004010F=MAPI_E_NOT_FO		
PR_SENDER_ENTRYID	PidTagSenderEntryId, ptagSender...	0x0C19000A	PT_ERROR	Err: 0x8004010F=MAPI_E_NOT_FO		
PR_SENDER_NAME	PR_SENDER_NAME_A, PR_SENDE...	0x0C1A000A	PT_ERROR	Err: 0x8004010F=MAPI_E_NOT_FO		
PR_SENSITIVITY	PidTagSensitivity, ptagSensitivity	0x00360003	PT_LONG	0	0x0	Flags
PR_SENT_REPRESENTING_ADD...	PR_SENT_REPRESENTING_ADDRT...	0x0064000A	PT_ERROR	Err: 0x8004010F=MAPI_E_NOT_FO		
PR_SENT_REPRESENTING_EMAI...	PR_SENT_REPRESENTING_EMAIL_...	0x0065000A	PT_ERROR	Err: 0x8004010F=MAPI_E_NOT_FO		
PR_SENT_REPRESENTING_ENTR	PidTagSentRepresentingEntryId, p...	0x0041000A	PT_ERROR	Err: 0x8004010F=MAPI_E_NOT_FO		
PR_SENT_REPRESENTING_NAME	PR_SENT_REPRESENTING_NAME...	0x0042000A	PT_ERROR	Err: 0x8004010F=MAPI_E_NOT_FO		

Microsoft's Response to Damian

"[…] Our engineering team investigated the behavior that you described. They determined that it is not considered a security issue because it requires control of the account to create these rules. However, they are considering ways to improve the software in the future."

"[…] MSRC will not be tracking the issue and we won't have future updates about it […]"

In other words, Microsoft is saying:

It's not a bug, it's a feature

The Next Evolution in Hidden Forwarding Rules: an Attack Method Through SaaS

- Malware that lives as a SaaS app
- Maintains access to the victim's account
- With *rootkit* capabilities

= The SaaS Rootkit

The SaaS Rootkit

"A **rootkit** is a collection of computer software, typically malicious, designed to enable access to a computer or an area of its software that is not otherwise allowed and often **masks its existence** or the existence of other software."

"Rootkit detection is difficult because **a rootkit may be able to subvert** the software that is intended to find it"

For reference: Rootkit definition from Wikipedia

Demo (part 1): Vulnerability in Azure AD App Registration Process

- Create app that looks credible
- Entice user to accept and gain permissions
- Attacker can add deleted Exchange online scopes

Undocumented Resources - No Server Side Validation

```
ReadWriteConsistencyToken=""
AuthBearer=""
AppId=""

if test -z "$ReadWriteConsistencyToken" || test -z $AuthBearer || test -z $AppId
then
      echo "One or more of the vars is NULL"
else

  curl -s "https://graph.microsoft.com/v1.0/myorganization/applications/$AppId" \
    -X "PATCH" \
    -H "Prefer: return-content" \
    -H "ReadWriteConsistencyToken: $ReadWriteConsistencyToken" \
    -H "Authorization: Bearer $AuthBearer" \
    -H "x-ms-effective-locale: en.en-us" \
    -H "Content-Type: application/json" \
    -H "Accept-Language: en" \
    -H "Accept: */*" \
    -H "Referer: " \
    -H "User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.114 Safari/5
    --data-raw "{'id':'$AppId','requiredResourceAccess':[{'resourceAppId':'00000002-0000-0ff1-ce00-000000000000',
    'resourceAccess':[{'id':'dc890d15-9560-4a4c-9b7f-a736ec74ec40','type':'Role'}]}]}" \
    --compressed

  sleep 5

fi
```

3rd Party App Access Through OAuth 2.0

OAuth 2.0:

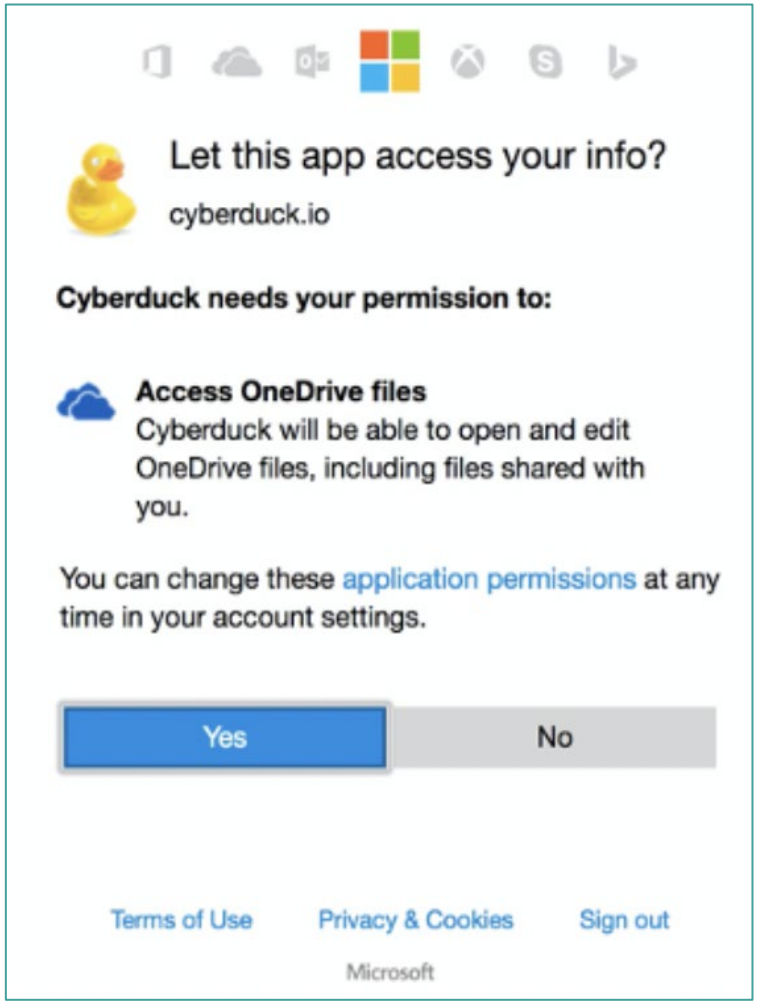
- has greatly simplified authentication & authorization
- offers a fine-grained delegation of access rights.

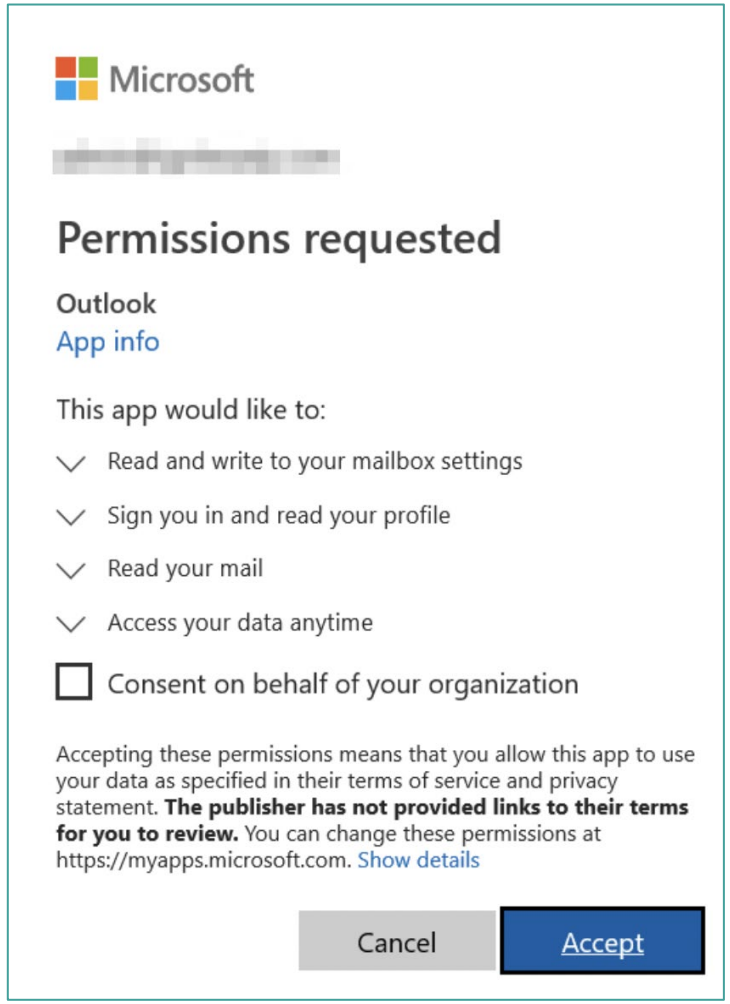
Represented in the form of scopes, an application asks for the user's authorization for specific permissions.

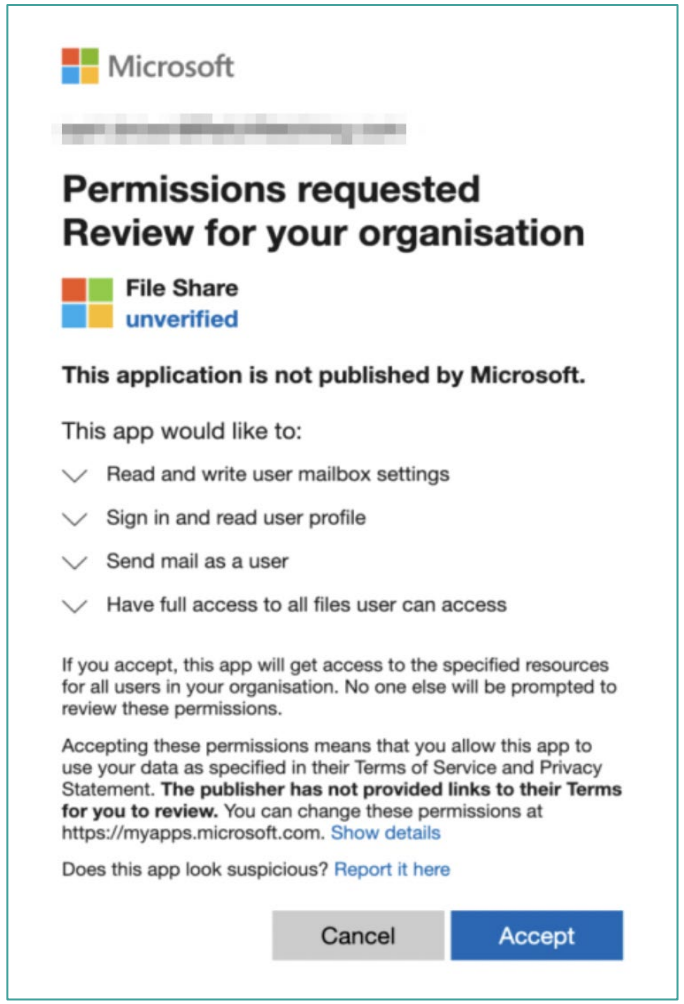
- an app can request one or more scopes.
- the user grants these apps permissions to execute code to perform logic behind the scenes within their environment.

These apps can be harmless or as threatening as an executable file

A Look at OAuth & 3rd Party App Access

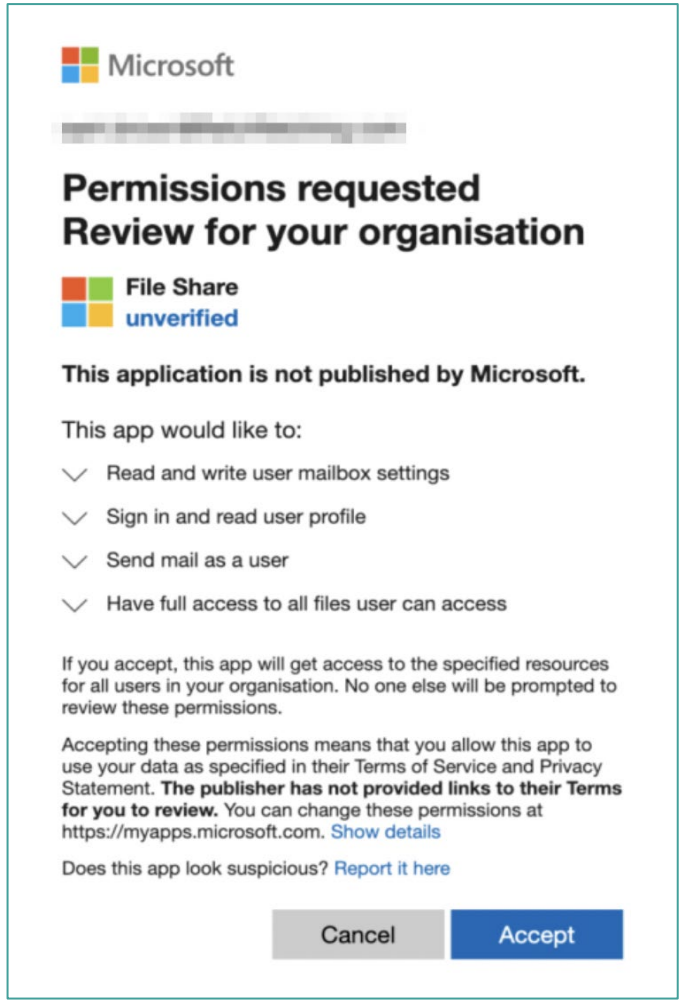






Remind You of Something?

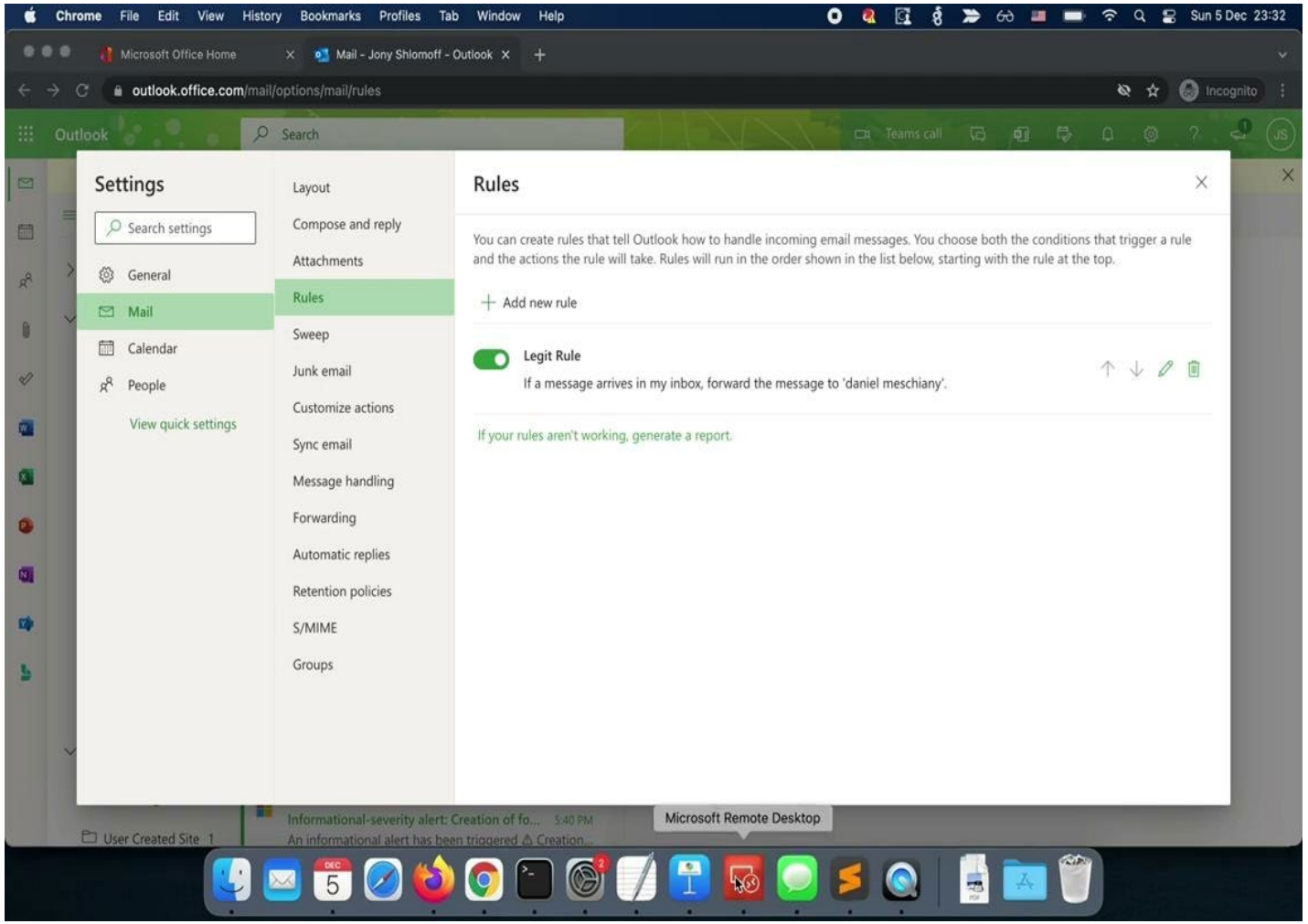
Potential SaaS Malware



Potential Endpoint Malware

Demo (part 2)
Abusing Exchange Online Legacy API

- Attacker can now create the hidden forwarding rules
- Hide the created rules
- This is no different than sending a malicious executable file. Rogue OAuth apps are equivalent to malware. Unfortunately, no EDR can detect it.



Discussion: What Just Happened?

- **We modified [or improved :)] the version of the Microsoft tool:** Get-AllTenantRulesAndForms.ps1 into GetAllRules.ps1 and Hide-Rule.ps1
- Get-AllTenantRulesAndForms.ps1 gaps:
  - Using user credentials authentication
  - Leaves out PR_RULE_MSG_PROVIDER (the field we need to tamper)
  - Unable to update / tamper objects (e.g. overwrite PR_RULE_MSG_PROVIDER)

Reverse engineering EWS DLL helped us finding the right functions and using them in our scripts:

```
// Microsoft.Exchange.WebServices.Data.Item
public void Update(ConflictResolutionMode conflictResolutionMode)
{
    this.Update(conflictResolutionMode, false);
}
```

```
using System;
namespace Microsoft.Exchange.WebServices.Data
{
    public enum ConflictResolutionMode
    {
        NeverOverwrite,
        AutoResolve,
        AlwaysOverwrite
    }
}
```

Microsoft's Answer to Us

"We have gone over the report in detail, including all of your additional files. Unfortunately it was determined that ***while the issue you reported is valid, it does not meet our the bar for immediate servicing***. In this case, we do think this can be improved upon, but due to the high requirements on the attacker, with the issue being post exploitation of an administrator, this would not be tracked by the security team for servicing.

That being said, this submission has been flagged for future review by the product team as an opportunity to improve the security of the affected product.

We do not have a timeline for when this review will occur, and will not be able to provide status for this issue moving forward. At this time, you are able to blog about/discuss this case and/or present your findings publicly about the current version."

How big of a problem is OAuth app access?

The SSPM Survey Report highlights the 340+ perspectives of security leaders today and the steps they are taking to secure their SaaS app stack.

3rd party app access is a top concern

What are the top concerns when adopting SaaS application in your company? (Select up to 3)

56% Lack of visibility into 3rd Party application access to the core SaaS stack

54% Lack of visibility into SaaS security settings

41% Inability to remediate SaaS Security misconfigurations

38% Lack of SaaS security knowledge

35% Lack of automation or tooling for SaaS security

32% Insufficient amount of SaaS security staff

How to Best Mitigate a SaaS Rootkit Attack

- Track activities and look for “New-InboxRule“ (or similar events) and compare them with users listed rules
- Continuously monitor 3rd party apps access
- Review new inbox rules with untrusted domains in the destination
- If possible, disable 3rd party apps registrations
- Continuously monitor new forwarding rule from untrusted domains

To Sum Up

Hidden forwarding rules are still a problem, even in a more dangerous fashion as it can show up through the trusted, Microsoft website.

Traditional controls were created to stop malware, but malware has evolved and has a new attack vector that can exploit any SaaS app, from M365 to Salesforce to G-Workspace, etc.

Utilize native security configurations to control the OAuth application installations across SaaS apps to protect users from malicious attacks like these.

RSAConference2022

Thank you!

Maor.Bin@adaptive-shield.com