This script is a response to social engineering attacks that leverage a compromised mailbox. An early IOC on an account takeover is the creation of inbox rules that allow an attacker to use a mailbox without the owner’s knowledge.

There are several common ways an attacker will leverage a mail account:

* Resetting passwords in payroll systems, to reroute the victim's direct deposit to a Green Dot card
* Launching untargeted phishing attacks using the victims mailing list
* Monitoring communications for large financial transactions, then taking over the conversation to redirect them to another account

Regardless of the goal, attackers will likely create mail rules once they have compromised an account. They will typically move incoming mail to Deleted Items or lesser-used folders, such as RSS subscriptions. This allows an attacker to monitor a mailbox, receive mail on behalf of the user, without the victim being aware.

Though Microsoft has built-in alerts for mail *forwarding,* there are no alerts available today for rules intended to *hide* mail.

Fortunately, users do not create mail rules often, and usually do so from their desktop. Attackers will usually use O365 webmail, so *any* email rule created from the O365 web interface is a red flag.

What does this script do?

* Login to exchange online administration on your behalf
* Retrieve any new inbox rules created through the O365 web interface
* Emails you with any new rules found

Setup

* Choose a machine for deployment.
  + The machine should be always on, and preferably administered by a single user. The script requires a credential file that can be decrypted by any user on the PC. A dedicated VM is most appropriate for this task.
* Download the powershell scripts.
* Execute "UpdatePassword.ps1"
  + You will be prompted for credentials. Enter your office 365 email address and password.
  + This will create a file "temp.cred" under your user profile, that contains your credentials encrypted with DPAPI.
  + For more info, see <https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.security/convertfrom-securestring?view=powershell-6>
  + **This step must be repeated whenever the user changes their password.**
* Create a scheduled task. We recommend the following settings:
  + General:
    - Run whether user is logged on or not, and store credentials
    - **This needs to be refreshed whenever the user changes their password.**
  + Triggers: Execute every 5 minutes. Choose a start date, recur "daily", but repeat task every 5 minutes for 1 day.
    - Machine generated alternative text:
      Edit Trigger 
      Begin the task: 
      o 
      One time 
      o 
      o 
      Monthly 
      On a schedule 
      Stat: 2018-11-21 
      Synchronize across time zones 
      Recur every: 
      Advanced settings 
      Delay task for up to (random delay): 
      Repeat task every: 5 minutes 
      Stop all running tasks at end of repetition duration 
      Stop task if it runs longer than: 
      for a duration of: 
      Synchronize across time zones 
      Cancel 
    - The tool retrieves rules created in the last 30 minutes; new rules don't show up in the log immediately, so this has been the best balance of ensuring detection and limiting repeated notifications.
  + Action: Start a program
    - Program/script:
      * Powershell
    - Add arguments:
      * C:\**path to script**\.ps1 EmailRuleAlert.ps1 "**office365emailaddress@domain.com**"
      * (substituting the path and email address with the actual path to the script and o365 email address)
    - String in:
      * **C:\path to script\**
      * (substituting the path with the actual path to the script)

Maintenance:

* Passwords for the Scheduled task must be updated when the user's password changes
* UpdatePassword.ps1 must be executed to update creds when the user's password changes

Usage:

* The script will emit emails when a new rule is created. Below is a redacted sample. The formatting from the messages is rough, so key elements have been called out.
* If you’re not certain whether the rule is legitimate, reach out to the sender through a known good phone number. Do not attempt to email the user, as if their inbox is compromised, you could be communicating with an attacker.

Search at

Friday, December 6, 2019 10:00:30

2019-12-06 15:30:30 2019-12-07 16:00:30

User Email Address

RunspaceId :

RecordType : ExchangeAdmin

CreationDate : 2019-12-06 15:41:58

UserIds : username@example.com

Operations : New-InboxRule

AuditData : {"CreationTime":"2019-12-06T15:41:58","Id":"","Operation":"New-Inbox

Rule","OrganizationId":"","RecordType":1,"ResultStatus":"True","User

Key":"","UserType":2,"Version":1,"Workload":"Exchange","ClientIP":"127.0.0.1",

Malicious rules may act on all emails, or use a list of suspicious keywords: “phish”, “spam”, and “adp” are common

"ObjectId":"PROD.OUTLOOK.COM\/Microsoft Exchange Hosted

Organizations\/organizatoin.onmicrosoft.com\/FirstName LastName\\This is a test rule","UserId":"username@

example.com","AppId":"","ClientAppId":"","ExternalAccess":false,"OrganizationName":"example.onmicr

osoft.com","OriginatingServer":"server (0.0.0.0)","Parameters":[{"Name":"AlwaysDeleteOutlo

okRulesBlob","Value":"False"},{"Name":"Force","Value":"False"},{"Name":"From","Value":"AlertTestOnly@example.com"},{"Name":"Name","Value":"This is a test rule"},{"Name":"DeleteMessage","Value":"True"

},{"Name":"StopProcessingRules","Value":"True"}],"SessionId":""}

ResultIndex : 1

ResultCount : 1

Malicious rules will typically delete messages or move them to an unusual folder

Identity :

IsValid : True

ObjectState : Unchanged