Detect and remediate the Outlook rules and custom forms injections attacks. - Microsoft Defender for Office 365

By chrisda

Archived: 2026-04-05 22:58:28 UTC

Summary Learn how to recognize and remediate the Outlook rules and custom Forms injections attacks in Office 365.

After an attacker gains access to your organization, they try to establish a foothold to stay in or get back in after they're discovered. This activity is called *establishing a persistence mechanism*. There are two ways that an attacker can use Outlook to establish a persistence mechanism:

- By exploiting Outlook rules.
- By injecting custom forms into Outlook.

Reinstalling Outlook, or even giving the affected person a new computer doesn't help. When the fresh installation of Outlook connects to the mailbox, all rules and forms are synchronized from the cloud. The rules or forms are typically designed to run remote code and install malware on the local machine. The malware steals credentials or performs other illicit activity.

The good news is: if you keep Outlook clients patched to the latest version, you aren't vulnerable to the threat as current Outlook client defaults block both mechanisms.

The attacks typically follow these patterns:

The Rules Exploit:

1. The attacker steals a user's credentials.
2. The attacker signs in to that user's Exchange mailbox (Exchange Online or on-premises Exchange).
3. The attacker creates a forwarding Inbox rule in the mailbox. The forwarding rule is triggered when the mailbox receives a specific message from the attacker that matches the conditions of the rule. The rule conditions and message format are tailor-made for each other.
4. The attacker sends the trigger email to the compromised mailbox, which is still being used as normal by the unsuspecting user.
5. When the mailbox receives a message that matches the conditions of rule, the action of the rule is applied. Typically, the rule action is to launch an application on a remote (WebDAV) server.
6. Typically, the application installs malware on the user's machine (for example, PowerShell Empire).
7. The malware allows the attacker to steal (or steal again) the user's username and password or other credentials from local machine and perform other malicious activities.

The Forms Exploit:

1. The attacker steals a user's credentials.
2. The attacker signs in to that user's Exchange mailbox (Exchange Online or on-premises Exchange).
3. The attacker inserts a custom mail form template into the user's mailbox. The custom form is triggered when the mailbox receives a specific message from the attacker that requires the mailbox to load the custom form. The custom form and the message format are tailor-made for each other.
4. The attacker sends the trigger email to the compromised mailbox, which is still being used as normal by the unsuspecting user.
5. When the mailbox receives the message, the mailbox loads the required form. The form launches an application on a remote (WebDAV) server.
6. Typically, the application installs malware on the user's machine (for example, PowerShell Empire).
7. The malware allows the attacker to steal (or steal again) the user's username and password or other credentials from local machine and perform other malicious activities.

Users are unlikely to notice these persistence mechanisms and they might even be invisible to them. The following list describes the signs (Indicators of Compromise) that indicate remediation steps are required:

- **Indicators of the Rules compromise**:
    - Rule Action is to start an application.
    - Rule References an EXE, ZIP, or URL.
    - On the local machine, look for new process starts that originate from the Outlook PID.
- **Indicators of the Custom forms compromise**:
    - Custom forms present saved as their own message class.
    - Message class contains executable code.
    - Typically, malicious forms are stored in Personal Forms Library or Inbox folders.
    - Form is named IPM.Note.[custom name].

You can use either of the following methods to confirm the attack:

- Manually examine the rules and forms for each mailbox using the Outlook client. This method is thorough, but you can only check one mailbox at a time. This method can be very time consuming if you have many users to check, and might also infect the computer that you're using.
- Use the Get-AllTenantRulesAndForms.ps1 PowerShell script to automatically dump all the mail forwarding rules and custom forms for all the users in your organization. This method is the fastest and safest with the least amount of overhead.

1. Open the users Outlook client as the user. The user may need your help in examining the rules on their mailbox.
2. Refer to Manage email messages by using rules article for the procedures on how to open the rules interface in Outlook.
3. Look for rules that the user didn't create, or any unexpected rules or rules with suspicious names.

4. Look in the rule description for rule actions that start and application or refer to an .EXE, .ZIP file or to launching a URL.

5. Look for any new processes that start using the Outlook process ID. Refer to Find the Process ID.

1. Open the user Outlook client as the user.

2. Follow the steps in, Show the Developer tab for the user's version of Outlook.

3. Open the now visible developer tab in Outlook and select **design a form**.

4. Select the **Inbox** from the **Look In** list. Look for any custom forms. Custom forms are rare enough that if you have any custom forms at all, it is worth a deeper look.

5. Investigate any custom forms, especially forms marked as hidden.

6. Open any custom forms and in the **Form** group, select **View Code** to see what runs when the form is loaded.

The simplest way to verify a rules or custom forms attack is to run the Get-AllTenantRulesAndForms.ps1 PowerShell script. This script connects to every mailbox in your organization and dumps all the rules and forms into two .csv files.

You need to be a member of the Global Administrator* role in Microsoft Entra ID or the Organization Management role group in Exchange Online, because the script connects to every mailbox in the organization to read rules and forms.

Important

* Microsoft strongly advocates for the principle of least privilege. Assigning accounts only the minimum permissions necessary to perform their tasks helps reduce security risks and strengthens your organization's overall protection. Global Administrator is a highly privileged role that you should limit to emergency scenarios or when you can't use a different role.

1. Use an account with local administrator rights to sign in to the computer where you intend to run the script.

2. Download or copy the contents of the **Get-AllTenantRulesAndForms.ps1** script from GitHub to a folder that's easy to find and run the script from. The script creates two date stamped files in the folder: `MailboxFormsExport-yyyy-MM-dd.csv` and `MailboxRulesExport-yyyy-MM-dd.csv`.

   Remove lines 154 to 158 from the script, because that connection method no longer works as of July 2023.

3. Connect to Exchange Online PowerShell.

4. Navigate in PowerShell to the folder where you saved the script, and then run the following command:

```
.\Get-AllTenantRulesAndForms.ps1
```

- **MailboxRulesExport-*yyyy-MM-dd*.csv**: Examine the rules (one per row) for action conditions that include applications or executables:
  - **ActionType (column A)**: The rule is likely malicious if this column contains the value `ID_ACTION_CUSTOM`.
  - **IsPotentiallyMalicious (column D)**: The rule is likely malicious if this column contains the value `TRUE`.
  - **ActionCommand (column G)**: The rule is likely malicious if this column contains any of the following values:
    - An application.
    - An .exe or .zip file.
    - An unknown entry that refers to a URL.
- **MailboxFormsExport-*yyyy-MM-dd*.csv**: In general, the use of custom forms is rare. If you find any in this workbook, open that user's mailbox and examine the form itself. If your organization didn't put it there intentionally, it's likely malicious.

If you find any evidence of either of these attacks, remediation is simple: just delete the rule or form in the mailbox. You can delete the rule or form using the Outlook client or using Exchange PowerShell.

1. Identify all devices where the user has used Outlook. They all need to be cleaned of potential malware. Don't allow the user to sign on and use email until all devices have been cleaned.

2. On each device, follow the steps in Delete a rule.

3. If you're unsure about the presence of other malware, you can format and reinstall all the software on the device. For mobile devices, you can follow the manufacturers steps to reset the device to the factory image.

4. Install the most up-to-date versions of Outlook. Remember, current version of Outlook blocks both types of this attack by default.

5. Once all offline copies of the mailbox have been removed, do the following steps:

   - Reset the user's password using a high quality value (length and complexity).
   - If multi-factor authentication (MFA) isn't turned on for the user, follow the steps in Setup multi-factor authentication for users

   These steps ensure that the user's credentials aren't exposed via other means (for example, phishing or password reuse).

Connect to the required Exchange PowerShell environment:

- **Mailboxes on on-premises Exchange servers**: Connect to Exchange servers using remote PowerShell or Open the Exchange Management Shell.

- **Mailboxes in Exchange Online**: Connect to Exchange Online PowerShell.

After you connect to the required Exchange PowerShell environment, you can take the following actions on Inbox rules in user mailboxes:

- **View Inbox rules in a mailbox**:
  - **View a summary list of all rules**

    ```
    Get-InboxRule -Mailbox laura@contoso.onmicrosoft.com
    ```

  - **View detailed information for a specific rule**:

    ```
    Get-InboxRule -Mailbox laura@contoso.onmicrosoft.com -Identity "Suspicious Rule Name" | Format-L
    ```

  For detailed syntax and parameter information, see Get-InboxRule.

- **Remove Inbox rules from a mailbox**:
  - **Remove a specific rule**:

    ```
    Remove-InboxRule -Mailbox laura@contoso.onmicrosoft.com -Identity "Suspicious Rule Name"
    ```

  - **Remove all rules**:

    ```
    Get-InboxRule -Mailbox laura@contoso.onmicrosoft.com | Remove-InboxRule
    ```

  For detailed syntax and parameter information, see Remove-InboxRule.

- **Turn off an Inbox rule for further investigation**:

  ```
  Disable-InboxRule -Mailbox laura@contoso.onmicrosoft.com -Identity "Suspicious Rule Name"
  ```

  For detailed syntax and parameter information, see Disable-InboxRule.

The Rules and Forms exploits are only used by an attacker after they've stolen or breached a user's account. So, your first step to preventing the use of these exploits against your organization is to aggressively protect user accounts. Some of the most common ways that accounts are breached are through phishing or password spray attacks.

The best way to protect user accounts (especially admin accounts) is to set up MFA for users. You should also:

- Monitor how user accounts are accessed and used. You may not prevent the initial breach, but you can shorten the duration and the effects of the breach by detecting it sooner. You can use these Office 365 Cloud App Security policies to monitor accounts and alert you to unusual activity:
  - **Multiple failed login attempts**: Triggers an alert when users perform multiple failed sign in activities in a single session with respect to the learned baseline, which could indicate an attempted breach.

    - **Impossible travel**: Triggers an alert when activities are detected from the same user in different locations within a time period that's shorter than the expected travel time between the two locations. This activity could indicate that a different user is using the same credentials. Detecting this anomalous behavior necessitates an initial learning period of seven days to learn a new user's activity pattern.

    - **Unusual impersonated activity (by user)**: Triggers an alert when users perform multiple impersonated activities in a single session with respect to the baseline learned, which could indicate an attempted breach.

- Use a tool like Office 365 Secure Score to manage account security configurations and behaviors.

Fully updated and patched versions of Outlook 2013, and 2016 disable the "Start Application" rule/form action by default. Even if an attacker breaches the account, the rule and form actions are blocked. You can install the latest updates and security patches by following the steps in Install Office updates.

Here are the patch versions for Outlook 2013 and 2016 clients:

- **Outlook 2016**: 16.0.4534.1001 or greater.
- **Outlook 2013**: 15.0.4937.1000 or greater.

For more information on the individual security patches, see:

- Outlook 2016 Security Patch
- Outlook 2013 Security Patch

Even with the patches and updates installed, it's possible for an attacker to change the local machine configuration to reenable the "Start Application" behavior. You can use Advanced Group Policy Management to monitor and enforce local machine policies on client devices.

You can see if "Start Application" has been re-enabled through an override in the registry by using the information in How to view the system registry by using 64-bit versions of Windows. Check these subkeys:

- **Outlook 2016**: `HKEY_CURRENT_USER\Software\Microsoft\Office\16.0\Outlook\Security\`
- **Outlook 2013**: `HKEY_CURRENT_USER\Software\Microsoft\Office\15.0\Outlook\Security\`

Look for the key `EnableUnsafeClientMailRules`:

- If the value is 1, the Outlook security patch has been overridden and the computer is vulnerable to the Form/Rules attack.
- If the value is 0, the "Start Application" action is disabled.
- If the registry key isn't present and the updated and patched version of Outlook is installed, then the system isn't vulnerable to these attacks.

Customers with on-premises Exchange installations should consider blocking older versions of Outlook that don't have patches available. Details on this process can be found in the article Configure Outlook client blocking.

- Malicious Outlook Rules by SilentBreak Security Post about Rules Vector provides a detailed review of how the Outlook Rules.
- MAPI over HTTP and Mailrule Pwnage on the Sensepost blog about Mailrule Pwnage discusses a tool called Ruler that lets you exploit mailboxes through Outlook rules.
- Outlook forms and shells on the Sensepost blog about Forms Threat Vector.
- Ruler Codebase
- Ruler Indicators of Compromise
