Office Application Startup: Outlook Forms

Adversaries may abuse Microsoft Outlook forms to obtain persistence on a compromised system. Outlook forms are used as templates for presentation and functionality in Outlook messages. Custom Outlook forms can be created that will execute code when a specifically crafted email is sent by an adversary utilizing the [same custom Outlook form

Once malicious forms have been added to the user's mailbox, they will be loaded when Outlook is started. Malicious forms will execute when an adversary sends a .specifically crafted email to the user

Procedure example:

Ruler is a tool to abuse Microsoft Exchange services. It is publicly available on GitHub and the tool is executed via the command line. The creators of Ruler have also released a defensive tool, NotRuler, to detect its usage

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

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

Mitigation:

For the Outlook methods, blocking macros may be ineffective as the Visual Basic engine used for these features is separate from the macro scripting engine. Microsoft has released patches to try to address each issue. Ensure KB3191938 which blocks Outlook Visual Basic and displays a malicious code warning, KB4011091 which disables custom forms by default, and KB4011162 which removes the legacy Home Page feature, are applied to systems.

If you find any evidence of either of these attacks, remediation is simple, just delete the rule or form from the mailbox. You can do this with the Outlook client or using remote PowerShell to remove rules.

Detection:

Microsoft has released a PowerShell script to safely gather mail forwarding rules and custom forms in your mail environment as well as steps to interpret the output. SensePost, whose tool Ruler can be used to carry out malicious rules, forms, and Home Page attacks, has released a tool to detect Ruler usage.

Collect process execution information including process IDs (PID) and parent process IDs (PPID) and look for abnormal chains of activity resulting from Office processes. Non-standard process execution trees may also indicate suspicious or malicious behavior.