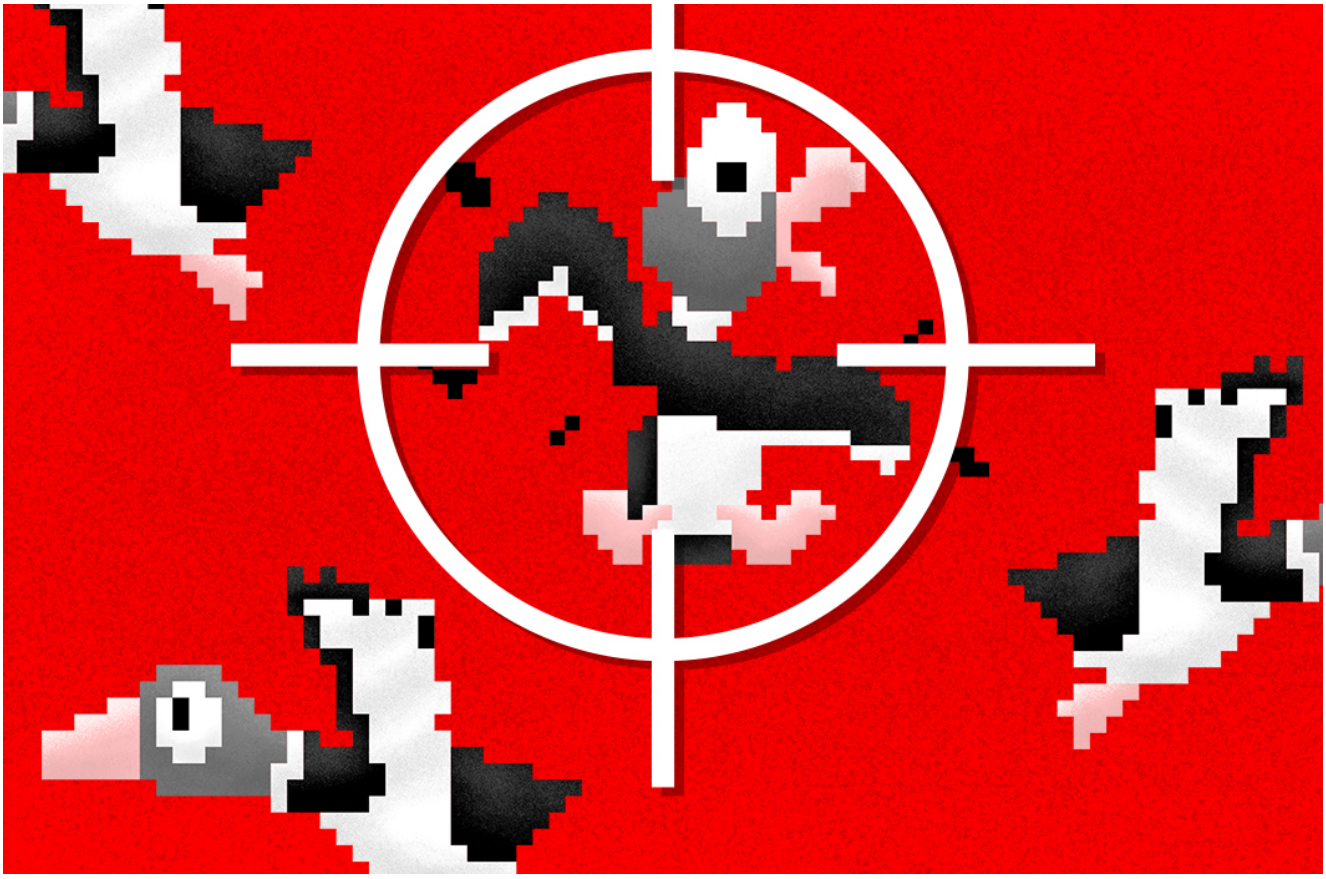
Duck Hunting w/Falcon Complete Pt. 1: QakBot Malware Overview

crowdstrike.com/blog/duck-hunting-with-falcon-complete-analyzing-a-fowl-banking-trojan-part-1/

Dylan Barker - Quinten Bowen - Ryan Campbell October 1, 2020



Adversaries constantly develop new tactics that enhance their capabilities to deploy malware across networked environments and monetize infected systems. This blog is Part 1 of a three-part series detailing research and observations by the CrowdStrike® Falcon Complete™ managed services team regarding one such malware variant, QakBot (aka QBot), and its behavior in recent campaigns.

In this blog we provide an overview of a recent QakBot campaign observed in the wild. Part 2 will feature an in-depth analysis of the evolution of QakBot tactics, techniques and procedures (TTPs) through June 2020. We will culminate the series in Part 3 by outlining the Falcon Complete team's strategy for the remote remediation of a QakBot-infected host.

Threat Background and Context

QakBot is an eCrime banking trojan that has the potential to severely impact an organization's ability to operate. QakBot can spread laterally throughout a network utilizing a worm-like functionality through brute-forcing network shares and Active Directory user group accounts, or via server message block (SMB) exploitation.

QakBot also employs a robust set of anti-analysis features to evade detection and frustrate analysis. Despite these evasion techniques, CrowdStrike Falcon® detects and prevents this malware from completing its execution chain.

QakBot has been observed for nearly a decade, and historically, it included traditional features of banking trojans and information stealers. However, it has since evolved and expanded its capabilities.

QakBot also shows no signs of slowing down — in fact, Falcon Complete observed a notable resurgence in its delivery volume, beginning in April 2020, with regular updates through the summer months. Recent campaigns have been delivered primarily via email, with attached ZIP archives containing a Visual Basic Script (VBS) downloader. In contrast, there have also been several tactical outliers, such as Microsoft Word DOC-based deliveries along with campaigns that included secondary malware payloads like Zloader.

The following section covers static and dynamic analysis of the QakBot DOC-based delivery campaign. This analysis includes an overview of techniques used by the threat actor to obfuscate, hinder and attempt to prevent analysis of malicious documents delivered by QakBot to ensure a successful infection of the victim.

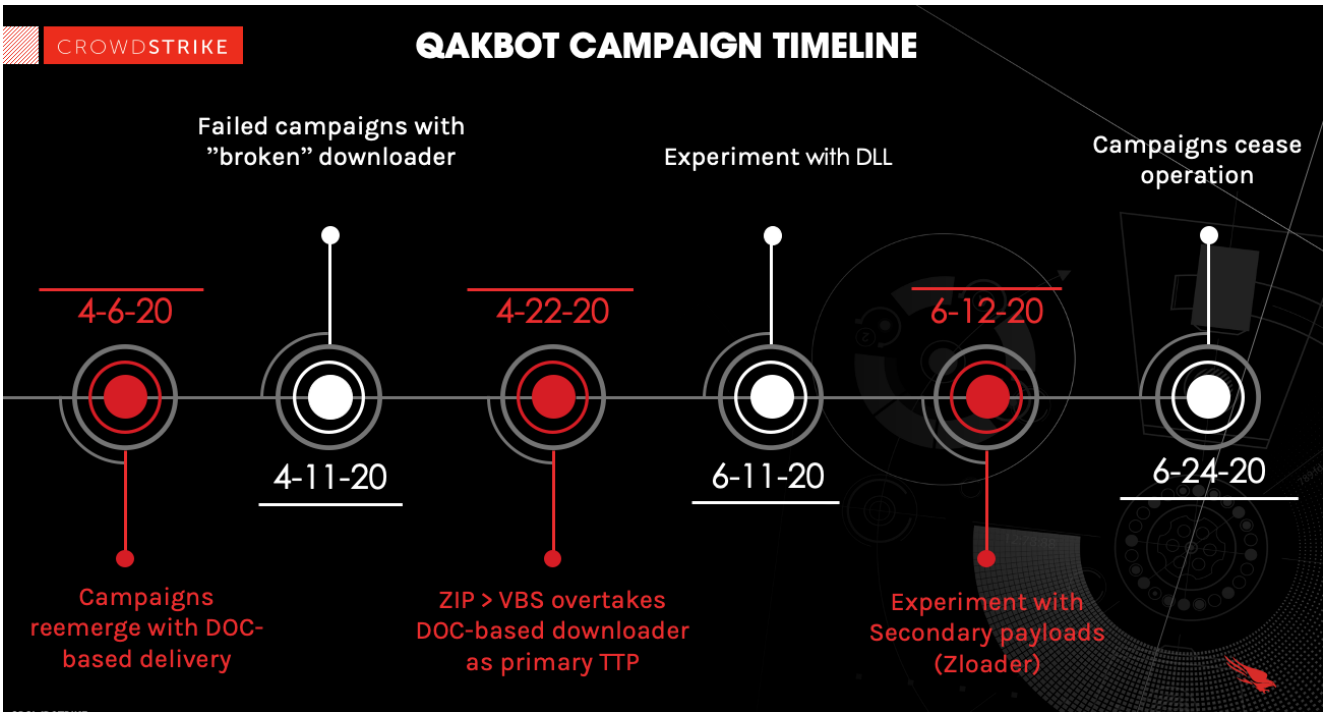


Figure 1. Timeline of QakBot Campaigns (click image to enlarge)

QakBot Introduces DOC-based Delivery

As QakBot surged in early April 2020, the operators leveraged a new tactic for delivery: a Microsoft Word document delivered via malspam that was weaponized with macros containing a malicious VB script. While operators shifted delivery tactics several times throughout the summer, the overall TTPs related to QakBot's execution chain remained largely the same, and analysis of this DOC file provides useful insights into the capabilities of QakBot's authors.

Static Analysis of Downloader DOC

Document Features

In early April, the team observed a document that was detected and blocked within a client environment.

- Filename: AGRMT_06052020_519.doc
- SHA256: b1e8b724380e6e041e3c2b4dfe5d4827fe0ae1bb0816b47d14d21d6f94194797

This was a typical phishing document that attempts to lure intended victims to enable macros that execute malicious code.

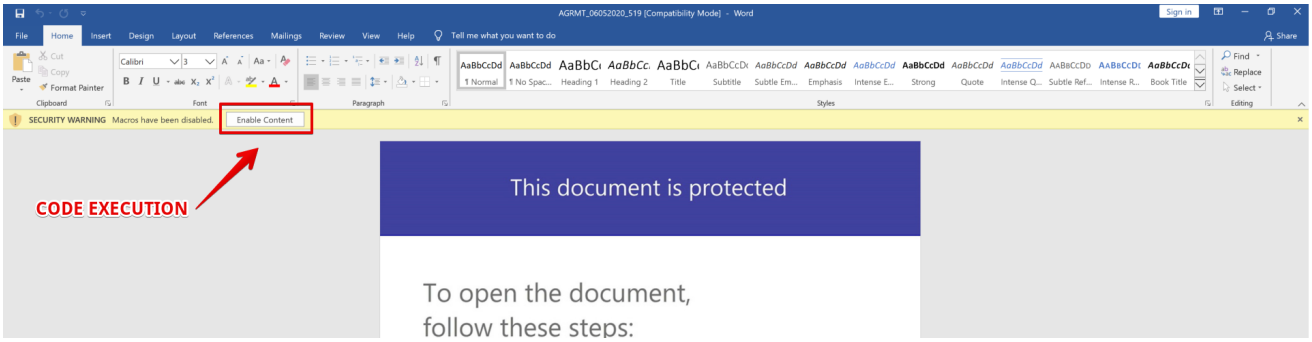


Figure 2. User Prompt (click image to enlarge)

The malicious macro included in the document is locked — an example of the many anti-analysis and sandbox-evasion features included in these campaigns.

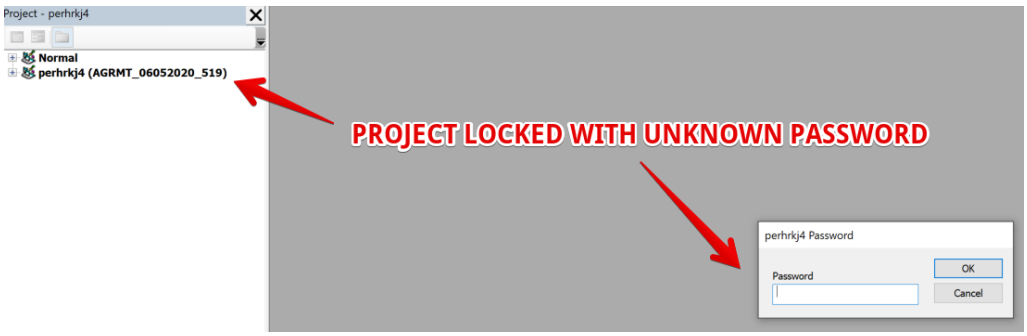


Figure 3. Locked Macro Modules (click image to enlarge)

The locked macros prevent sandboxes from inspecting the macro content and also impede manual analysis. Once this protection is bypassed, the contents of the Visual Basic for Applications (VBA) project are then available.

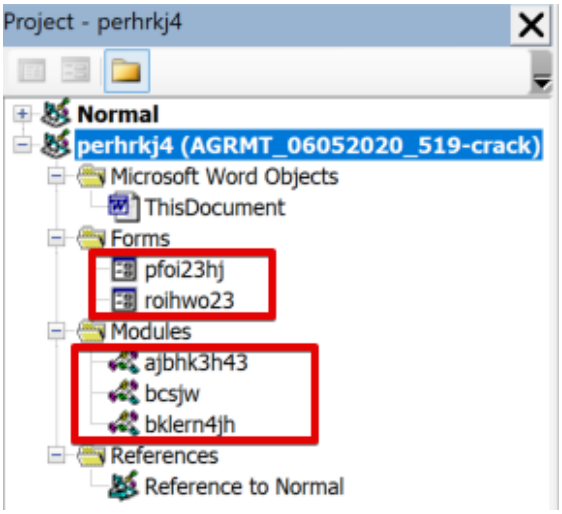


Figure 4. Macros Now Accessible

The macros are highly obfuscated and broken down into either UserForm objects or modules containing more typical Functions and Subs. The UserForm objects are structured in a variety of different text boxes, buttons and label objects that don't actually execute code themselves, but hold string data in their tags that is called by Functions in the macros code.

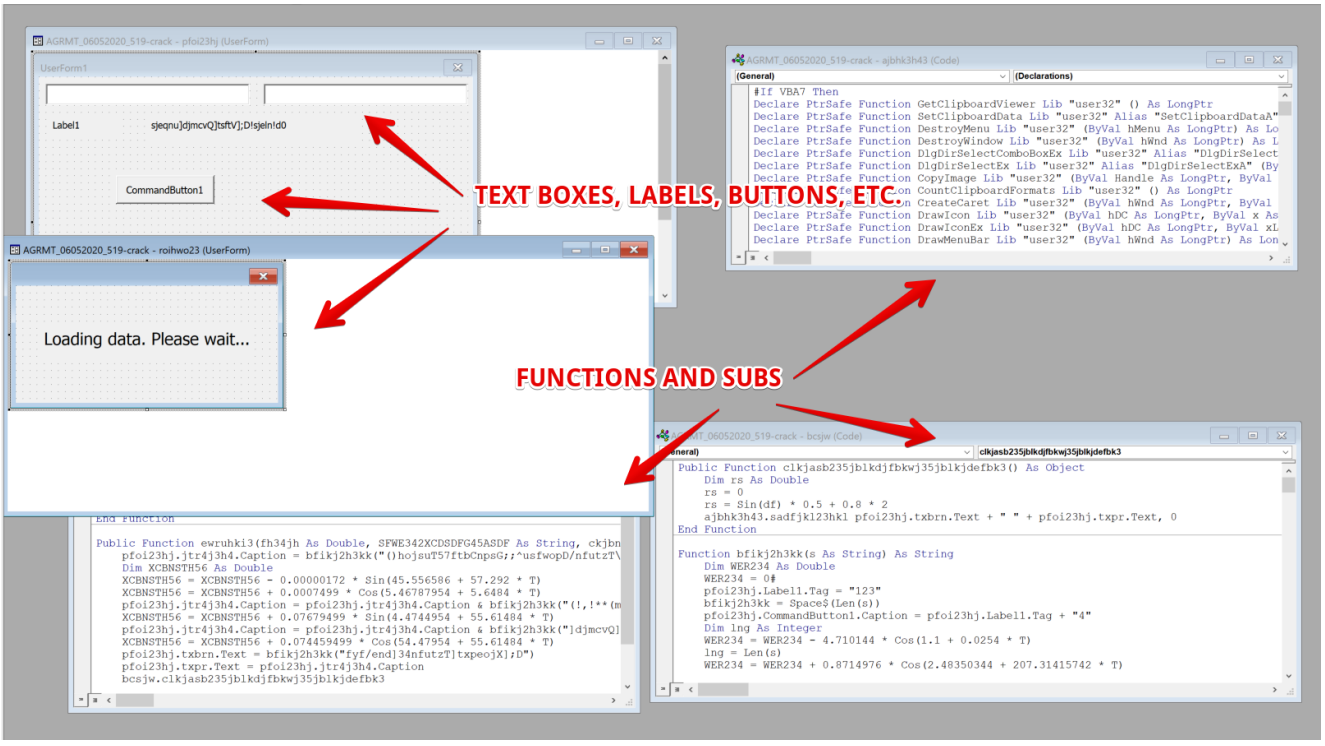


Figure 5. Modules and Objects (click image to enlarge)

Macro De-obfuscation

The macro contents are extracted to a text editor for continued static analysis and easier de-obfuscation. The macros are extensively obfuscated with a variety of anti-analysis techniques. This includes using several lines of garbage code, along with confusing variable assignments, string reversals and an interesting string encoding technique that leverages nested conversions.

The two components of greatest interest are the functions that include an encoding algorithm and a block of strings that contain the encoded URLs, from which the code will attempt to download the next-stage payload.

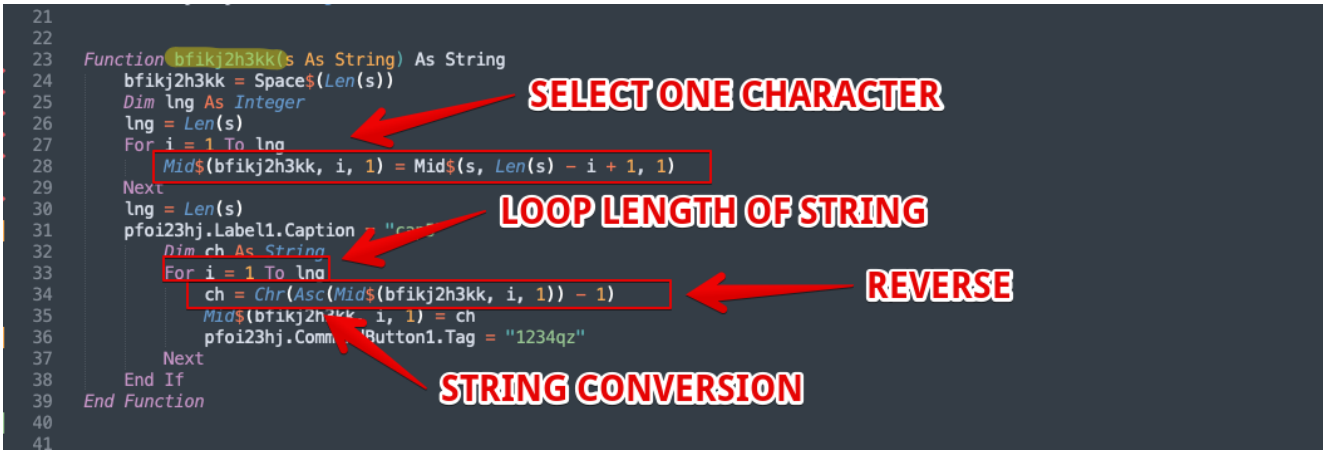


Figure 6. Encoding Algorithm (click image to enlarge)

The encoding algorithm leverages the VBA Mid function to perform operations on URL strings. The $Mid function sets the text block, sets the starting position of the text's index, and then instructs it to select the number of characters (in this case, 1). It is also wrapped in a For Loop that instructs the Function to loop over the length of the string in reverse order. Finally, it includes a series of string conversions following the loop operation. Each character is converted to ASCII and then subtracted by a value of 1, then converted back into a character. This new character string is actually Base64 encoded and must then be decoded to reveal the actual URL.

Figure 7. Obfuscated URLs (click image to enlarge)

For example, the URLs are decoded by taking the last character in each string, converting it to ASCII, subtracting by 1 and then looping backward the length of the string. Finally, the Base64 string is decoded. The first variable in Figure 9, "dkekr(1)," which holds the string

"o6HdvhEP5hEP5h{M5iHdkS4Mu:3Zv1H[ieIciO4MwpEd1SIb," decodes into the string "aHR0cDovL3NhbHdhZG0uY29tL3RjcGh4Lzg4ODg4ODgucG5n."

Figure 8. Decoded URL (click image to enlarge)

The VB code contains a total of six URLs that it will loop through until it receives a valid response to download the next stage payload. It is necessary to break down this obfuscation to successfully identify all six of these indicators of compromise (IOCs). If only network log data or sandbox results are used, the infection chain may stop at the first valid response and not loop through all six URLs. The complete list of IOCs may then be implemented in the appropriate network control such as a web proxy or firewall.

When executed, the final macro code as interpreted by CMD decodes into a classic PowerShell download cradle that fetches the initial QakBot payload. There is one last bit of obfuscation here as the script does contain two more encoded strings. One is the URL as seen above in Figure 8, and another is the full path to which the payload will initially be written: “C:\Users\Public\tmpdir\file”.

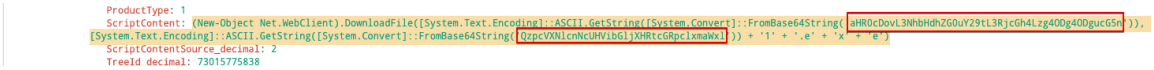


Figure 9. Download Cradle (click image to enlarge)

The cradle appends “1.exe” to the file named “file” and eventually writes itself to “C:\Users\Public” before continuing with the remainder of the execution chain if the appropriate anti-analysis checks are met. The execution chain for this delivery style is shown below in Figure 10. Please note that the examples in the following scenarios have CrowdStrike Falcon configured with DETECTIONS ONLY and PREVENTIONS OFF for illustrative purposes. A properly configured Falcon instance would prevent the activity presented here.

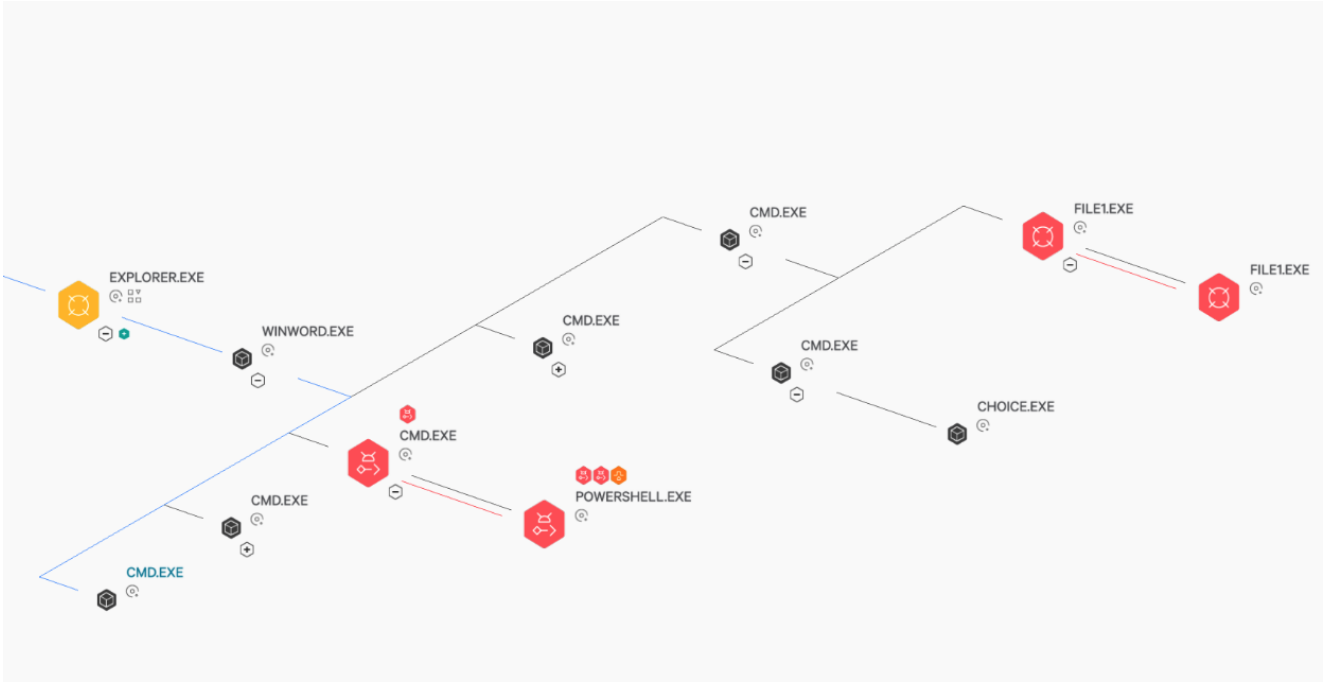


Figure 10. Process Tree as Displayed in Falcon (click image to enlarge)

The Falcon Complete team frequently responds to detections where Falcon has prevented a successful QakBot phishing attempt. In these cases that have proper implementation of preventions, the remediation simply consists of removing the original attachment written to the disk and any associated residual script artifacts. In these cases, Falcon interrupts the infection before binaries are executed or persistence is established.

Conclusion

QakBot has the potential to severely impact an organization due to its capability for lateral movement and data theft. As we have seen, QakBot employs a robust set of anti-analysis features and has recently surged in its operational volume within the threat landscape.

In this blog, we presented an analysis of a DOC-based QakBot downloader. The threat actors behind QakBot, tracked by CrowdStrike Intelligence as MALLARD SPIDER, have demonstrated the ability to rapidly re-tool, implement anti-analysis techniques and develop methods of advanced obfuscation in a short period.

Stay tuned for Parts 2 and 3 where we delve deeper into our analysis and offer recommendations for improving your defenses against QakBot and similar threats.

As QakBot evolves, so does the Falcon Complete team's ability to adapt, react to and remediate this threat to protect our client environments. The team provides the expertise to identify and remediate infections to help organizations recover from potentially devastating incidents. The Falcon Complete team focuses on stopping breaches so CrowdStrike clients can focus on their business goals and operations.

Appendix

Table 1 below contains a mapping of QakBot tactics to the MITRE ATT&CK® framework.

Tactic	Technique	Sub-Technique	ID
Initial Access	Phishing	Spear-Phishing Attachment	T1566.001
Execution	User Execution	Malicious Link, Malicious File	T1204.001, T1204.002
Execution	Command and Scripting Interpreter	PowerShell, CMD Shell, Visual Basic	T1059.001, T1059.003, T1059.005
Execution	Signed Binary Proxy Execution	Msiexec, Rundll32	T1218.007, T1218.011
Persistence	Boot or Logon Autostart Execution	Registry Run Keys / Startup Folder	T1547.001
Persistence	Scheduled Task/Job	Scheduled Task	T1053.005
Defense Evasion	Obfuscated Files or Information	None	T1027
Defense Evasion	Process Injection	Dynamic-link Library Injection	T1055.001
Defense Evasion	Virtualization/Sandbox Evasion	System Checks	T1497.001
Discovery	Virtualization/Sandbox Evasion	User Activity Based Checks	T1497.002
Discovery	Network Share Discovery	None	T1135
Credential Access	Brute Force	Password Guessing	T1110.001
Lateral Movement	Remote Services	SMB/Windows Admin Shares	T1021.002
Command and Control	Application Layer Protocol	Web Protocols	T1071.001

Table 1. MITRE ATT&CK Mapping

IOCs associated with QakBot analyses are shown in Table 2.

Indicator	Purpose
PicturesViewer.dll, PicturesViewer.exe, PaintHelper.dll, PaintHelper.exe, file1.exe	QakBot binary names

“[0-9]{6,9}\.zip”, “NUM_[0-9]{4,6}\.vbs”	Regular expression of observed filename convention of zip archives containing vbs to that launches QakBot downloader
“dfPEZd”, “ezQVN”, “wCdZgXH”	Scheduled Task tasknames
“””C:\windows\System32\WScript.exe”” “”C:\Users*\AppData\Local\Temp\Temp1_*.zip\NUM_*.vbs””	Command line example of initial execution
C:\Users*\Downloads\“[0-9]{6,9}\.zip”	Initial QakBot download path. Observed as an 8 or 9-character numeric name.
C:\Users*\AppData\Local\Temp\Temp1_“[0-9]{6,9}\.zip\NUM_[0-9]{4,6}\.vbs	Execution path of VB downloader script
%AppData%\lwob\esexydry.dll %AppData%\PicturesViewer.dll %APPDATA%\dasfdsfsdf.exe %APPDATA%\lwhoq\pozypua.dll %APPDATA%\IE\GGYJG27Z\dasfdsfs.df[1].exe C:\Users\Public\tmpdir	QakBot binary paths in home directories, observed as an alphabetical name under an alphabetical folder in %AppData%, or pre-named PicturesViewer, PaintHelper
HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run	Registry run key persistence

Table 2. IOCs Associated with QakBot

Additional Resources

- Read Part 2 and Part 3 of the Duck Hunting with Falcon Complete blog series.
- *Find out how CrowdStrike can help your organization answer its most important security questions: Visit the CrowdStrike Services webpage.*
- *Learn how any size organization can achieve optimal security with Falcon Complete by visiting the product webpage.*
- *Learn more about Falcon X™ threat intelligence by visiting the webpage.*
- *Learn about CrowdStrike's comprehensive next-gen endpoint protection platform by visiting the Falcon products webpage.*
- *Test CrowdStrike next-gen AV for yourself: Start your free trial of Falcon Prevent™.*