attachments, OneNote files, and generic attack surface reduction

Sam Scholten, Detection Engineering

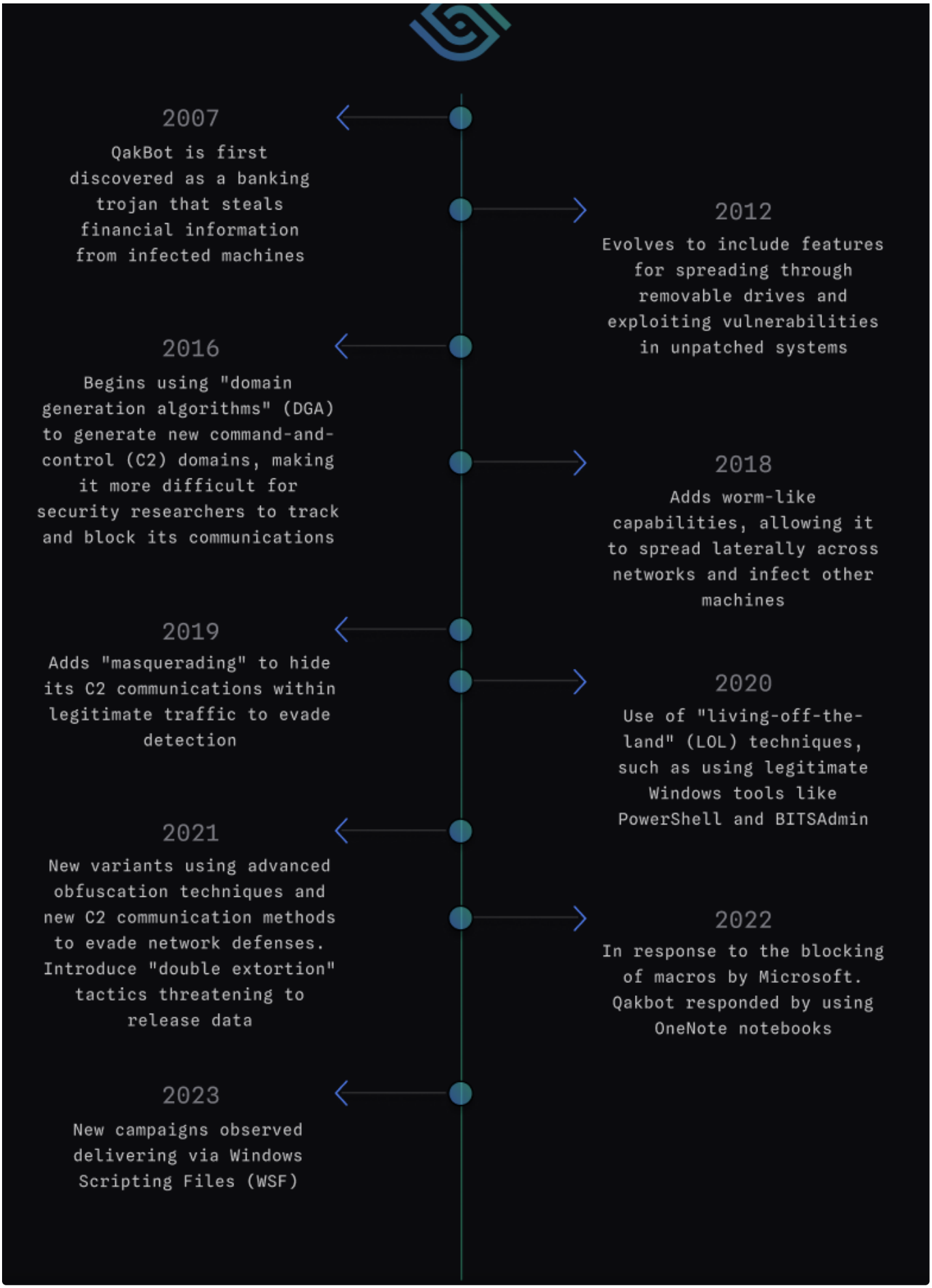
April 12, 2023

This post will cover a brief timeline of QakBot's evolution, and focus primarily on recently observed attack techniques. We'll discuss detection methodologies and share MQL rules that anyone can use to detect, prevent, and hunt for these threats in email environments today. If you're already running Sublime, you received these new protections automatically.

QakBot History and Evolution

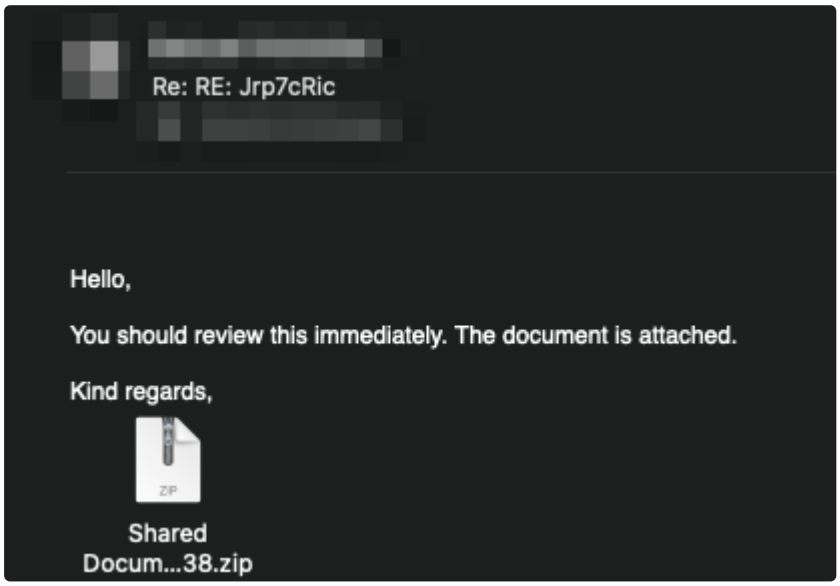
QakBot, also known as QBot and Pinkslipbot, has been active since 2007 and has been consistently and constantly evolving. Initially, QakBot started as a banking Trojan that utilized command and control (C2) servers for payload delivery. With modularity being a crucial component, QakBot's primary objective was to steal financial data and login credentials from victims. It was also capable of spying on financial operations and redirecting users to fake banking sites.

Over the years, QakBot has used many different techniques to infect users, including malspam campaigns with malicious attachments, hyperlinks, or embedded images to drop a second-stage payload. The



QakBot timeline

In early 2023, QakBot was observed using a new method of distribution through Windows Script Files (.wsf). In this scenario, the phishing email contains a zip file with a random name, which includes a wsf file and txt file, and a decoy pdf file.



sample email

The malicious attachment is delivered in the following sequence:

1. A .zip file containing multiple files, including a decoy .pdf file, a .txt file, and a .wsf file.
2. The .wsf file is used to execute the malicious code contained in the decoy .pdf file.
3. The .pdf file contains a script that downloads a .dll file and executes it on the infected machine.

This delivery method is unique in that it uses a .wsf file to execute the malicious code, rather than relying on macros or other scripting languages.

When the victim tries to open the .wsf file, javscript is executed to download the QakBot DLL file. The file is usually loaded into the

```
rundll32 C:\\ProgramData\\Z8w7V9.SmcisaK,Wind
```

sample

Using MQL to Detect The WSF Variant

Let's create an MQL rule that can detect this specific delivery method by looking for the following characteristics:

1. An inbound email with at least one attachment.
2. The attachment is an archive.
3. The archive file contains a .pdf file, .txt file, and .wsf file at a depth of 1.

The Rule:

Attachment: Archive with pdf, txt and wsf files

```
name: "Attachment: Archive with pdf, txt and wsf files"
description: |
  Detects a known Qakbot delivery method, zip file with pdf, txt and wsf file at
a depth of 1
type: "rule"
references:
  - "https://twitter.com/pr0xylife/status/1625528782240071681"
severity: "medium"
source: |
  type.inbound
  and any(attachments,
        .file_extension in~ $file_extensions_common_archives
        and length(distinct(filter(file.explode(.), .depth == 1
         and .flavors.mime in~ ("application/pdf", "text/plain")),
         .flavors.mime)) == 2
        and any(file.explode(.), .depth == 1 and .file_extension == "wsf")
  )
tags:
  - "Qakbot"
  - "Suspicious attachment"
```

Attachment: Archive with pdf, txt and wsf files (MQL)

Breaking Down the Rule:

The rule is inspecting inbound mail with at least 1 attachment. It uses Sublime's open-source static-files, specifically the **$file_extensions_common_archive** list, to determine if the file is an archive.

If an attachment is found with an archive , we use **file_extension** to check for archives and the **file.explode** function, which explodes the archive file. The rule then checks if the archive file contains a .pdf, .txt, and .wsf file at a depth of 1.

If all of these conditions are met, the rule tags the email as "Qakbot" and "Suspicious attachment" and assigns it a medium severity rating.

Test in Playground

Attack Surface Reduction

In addition to specific detections, it's important to consider the protections gained by a more generalized approach. Attack surface reduction (ASR) is a proactive security strategy that involves minimizing potential avenues of attack for malicious actors by limiting their opportunity to do harm.

One effective way to protect against Qakbot and other similar malware threats is by implementing attack surface reduction (ASR) techniques, such as the **rule** below. This rule utilizes MQL to scan email content for any links that may lead to an encrypted zip file, and then checks whether the zip file contains a disk image in IMG, ISO, or VHD format.

```
description: |
  A link in the body of the email downloads an encrypted zip that contains a dis
k image of the format IMG, ISO or VHD. This is a combination of file types used
to deliver Qakbot.
type: "rule"
references:
  - "https://twitter.com/pr0xylife/status/1592502966409654272"
  - "https://delivr.to/payloads?id=ca00292e-d5a2-43f9-b638-6c0b01b73353"
  - "https://www.trendmicro.com/en_us/research/22/j/black-basta-infiltrates-netw
orks-via-qakbot-brute-ratel-and-coba.html"
  - "https://www.cyfirma.com/outofband/html-smuggling-a-stealthier-approach-to-d
eliver-malware/"
severity: "medium"
authors:
  - twitter: "ajpc500"
source: |
  type.inbound
  and any(body.links,
      any(beta.linkanalysis(.).files_downloaded,
          any(
              file.explode(.), (
                  any(.flavors.yara, . == "encrypted_zip") and
                  any(.scan.zip.all_paths, any([".img", ".iso", ".vhd"], string
s.ends_with(.., .)))
              )
          )
      )
  )
  // first-time sender
  and (
      (
          sender.email.domain.root_domain in $free_email_providers
          and sender.email.email not in $sender_emails
      )
      or (
          sender.email.domain.root_domain not in $free_email_providers
          and sender.email.domain.domain not in $sender_domains
      )
  )
tags:
  - "Qakbot"
  - "Suspicious link"
  - "Malware"
  - "HTML smuggling"
```

Link to auto-downloaded disk image in encrypted zip (MQL)

OneNote Attack Surface Reduction

Macros. We can again leverage MQL to surface these attempts. The rule below was contributed by @Kyle_Parrish_/Kyle Parrish, a Sublime Community user.

```
name: "Attachment: Malicious OneNote Commands"
description: |
  Scans for OneNote attachments that contain suspicious commands that may indica
te malicious activity.
references:
  - "https://www.trustedsec.com/blog/new-attacks-old-tricks-how-onenote-malware-
is-evolving/"
  - "https://bazaar.abuse.ch/sample/aafc0ca9681c1f5c368b0f6da85b90e433f6d62fb34e
d2e968e53f83981a800f"
type: "rule"
authors:
  - twitter: "Kyle_Parrish_"
    name: "Kyle Parrish"
severity: "high"
source: |
  type.inbound
  and any(attachments,
      (
          .file_extension in~ ("one") or
          .file_extension in~ $file_extensions_common_archives
      )
      and any(file.explode(.),
      any(
          .flavors.yara, . == "onenote_file"
          and
           any(..scan.strings.strings,
             strings.ilike(.,
                 "*WshShell*",
                 "*ExecuteCmdAsync*",
                 "*CreateObject*",
                 "*Wscript.Shell*",
                 "*schtasks*",
                 "*CreateProcess*",
                 "*winmgmts*",
                 "*SetEnvironmentVariable*",
                 "*powershell*",
                 "*echo off*")
               )
           )
        )
    )
tags:
  - "Suspicious attachment"
  - "Malware"
```

This rule aims to identify potential threats in OneNote attachments by searching for specific suspicious commands. It first checks for OneNote files, as well as OneNote files inside archives, using Sublime's static-files list (**$file_extensions_common_archives**). The rule then uses MQL to scan for specific strings that may indicate malicious behavior, such as references to shell commands (e.g., Windows Script Host, scheduled tasks), PowerShell, and other malware indicators. This is another great example of reducing your attack surface, while not specifically aimed at Qakbot, but any malware looking to leverage similar delivery mechanisms.

Conclusion

Qakbot's unique delivery methods require a multi-layered approach for detection.

One effective technique for safeguarding against Qakbot and other similar malware threats is implementing Attack Surface Reduction (ASR) measures. By proactively reducing potential avenues of attack for malicious actors, ASR can significantly minimize the opportunity for harm.

All of the rules described above can be used as both detection rules to prevent new attacks going forward, as well as a Hunt rules to look for historical attacks. They've all been added to the core Sublime Rules Feed, which means all Sublime instances, both free and paid, receive these new protections by default.

Back to Blog