Exploring Cyber Threats and Threat Actors in the Financial Sector: A Comprehensive Study

Srivastava Shivang¹, Chinnuswamy Tamizhselvan², Parameswaran Ganesan³, Rengabashyam Asha⁴

¹,²,³,⁴Graduate Student, Nanyang Technological University

Abstract:
This paper aims to discuss the recent activities of Financially motivated Threat actors and gather IOCs and Threat Intelligence based on the same. Common TTPs are mapped for 18 FIN threat actor groups along with known mitigations as per MITRE Attack Framework. In particular, FIN 7 is discussed in detail, including the lifecycle of Qakbot Malware and malwares are analyzed to gather IOCs using Static Analysis. Intrusion Detection Systems (Snort and YARA) are drafted for Qakbot. A comprehensive analysis on Diamond Model, Kill Chain and Pyramid of Pain is performed for Qakbot Malware and mitigations are mapped to MITRE ATTACK framework. Threat intelligence is gathered on the 1000 latest samples of Qakbot to deep dive into most commonly used delivery methods, malware file types and a timeline analysis is conducted. Advanced tools like OpenCTI and Cuckoo Sandbox are utilized to give an overall analysis on Financially motivated threat actors

1.0 Introduction
The financial sector is facing an ever evolving and complex threat landscape in the realm of cybersecurity. In recent years, there has been a rise in the frequency and sophistication of attacks on the financial and banking industry. The financial sector was the second most impacted sector based on the number of breaches last year.

According to the IBM cost of a data breach report 2023,
• The global average cost of a data breach in 2023 was $4.45 million, 15% more than in 2020.
• 51% of organizations are planning to increase security investments because of breach.
• The effect of extensive security AI and automation on the financial impact of a breach is USD1.76M

<table>
<thead>
<tr>
<th></th>
<th>2023</th>
<th>2022</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>United States USD 9.49 million</td>
<td>United States USD 9.44 million</td>
</tr>
<tr>
<td>2</td>
<td>Middle East USD 8.07 million</td>
<td>Middle East USD 7.46 million</td>
</tr>
<tr>
<td>3</td>
<td>Canada USD 5.13 million</td>
<td>Canada USD 5.14 million</td>
</tr>
<tr>
<td>4</td>
<td>Germany USD 4.67 million</td>
<td>United Kingdom USD 5.05 million</td>
</tr>
<tr>
<td>5</td>
<td>Japan USD 4.92 million</td>
<td>Germany USD 4.85 million</td>
</tr>
</tbody>
</table>

Figure 1: Data breach costs (Top five countries)
The selection of 17 industries has been included in the study for multiple years. Out of 17 industries, the financial industry suffers 14% of data breaches. Refer to the below diagram from the report.

![Pie Chart showing the distribution of data breaches across industries.]

<table>
<thead>
<tr>
<th>Industry</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Financial</td>
<td>14%</td>
</tr>
<tr>
<td>Services</td>
<td>12%</td>
</tr>
<tr>
<td>Technology</td>
<td>11%</td>
</tr>
<tr>
<td>Industrial</td>
<td>10%</td>
</tr>
<tr>
<td>Energy</td>
<td>8%</td>
</tr>
</tbody>
</table>

Five industries together accounted for 55% of organizations sampled in this year’s study.

More information on the data breaches in the year 2023 can be found in the IBM report. [1]

**Risks faced by financial sector:**

Based on the Cybersecurity and Financial system resilience report, Cybersecurity came up at the top of list as the potential risks and emerging threats that affects the U.S. economy. It was called out RaaS (Ransomware-as-a-Service) and sophisticated DDoS (Distributed-Denial-of-Service) attacks as the biggest risks to financial institutions ability to operate and safeguard customer data.

It was highlighted in the report,

“The rising number of advanced persistent threats increases the potential for malicious cyber activity within the financial sector. These threats may result in incidents that affect one or more participants in the financial services sector simultaneously and have potentially systemic consequences. Such incidents could affect the ability of targeted firms to provide services and conduct business as usual, presenting a unique challenge to operational resilience. These incidents can also threaten the confidentiality, integrity, and availability of the targeted firm’s data.” [1]

**Active Campaigns:**

In line with the above reports, our research has identified a lot of active campaigns against financial institutions. The below snapshots show active campaigns.
Our Focus:
The financial sector has witnessed a surge in cyber-attacks, necessitating a comprehensive analysis of the factors contributing to this trend. By examining the motivations of cybercriminals, the vulnerabilities inherent in the sector's digital transformation, and the sophisticated attack techniques employed, we can better understand the magnitude of the threats faced.

To highlight the significance of this research, we will explore recent high-profile attacks that have impacted the financial sector. These case studies will underscore the importance of proactive security measures and the potential consequences of failing to adequately protect financial institutions and their customers. Through this report, we aim to provide valuable insights into the evolving nature of cyber threats in the financial sector, emphasizing the importance of proactive cybersecurity measures such as threat monitoring and detection and fostering a collective effort to safeguard the integrity and stability of the financial ecosystem.

2.0 Our Research
2.1 APT groups
We have extensively looked at the Advanced Persistent Threats (APT) group which are motivated by financial gains, and we mapped the tactics and techniques of these groups in the MITRE ATT&CK framework.

<table>
<thead>
<tr>
<th>No</th>
<th>Threat Group</th>
<th>Introduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>FIN 4</td>
<td>FIN4 is a financially motivated threat group that has targeted confidential information related to the public financial market, particularly regarding healthcare and pharmaceutical companies, since at least 2013. [2]</td>
</tr>
<tr>
<td>2</td>
<td>FIN 5</td>
<td>FIN5 is a financially motivated threat group that has targeted personally identifiable information and payment card information. The group has been active since at least 2008 and has targeted the restaurant, gaming, and hotel industries. [2]</td>
</tr>
<tr>
<td>No</td>
<td>Threat Group</td>
<td>Introduction</td>
</tr>
<tr>
<td>----</td>
<td>--------------</td>
<td>--------------</td>
</tr>
<tr>
<td>3</td>
<td>FIN 6</td>
<td>FIN6 is a cybercrime group that has stolen payment card data and sold it for profit on underground marketplaces. This group has aggressively targeted and compromised point of sale (PoS) systems in the hospitality and retail sectors. [2]</td>
</tr>
<tr>
<td>4</td>
<td>FIN 7</td>
<td>FIN7 is a financially motivated threat group that has been active since 2013 primarily targeting the U.S. retail, restaurant, and hospitality sectors, often using point-of-sale malware. A portion of FIN7 was run out of a front company called Combi Security. [2]</td>
</tr>
<tr>
<td>5</td>
<td>FIN 8</td>
<td>FIN8 is a financially motivated threat group known to launch tailored spear phishing campaigns targeting the retail, restaurant, and hospitality industries. [2]</td>
</tr>
<tr>
<td>6</td>
<td>FIN 10</td>
<td>FIN10 is a financially motivated threat group that has targeted organizations in North America from 2013 through 2016. The group uses stolen data exfiltrated from victims to extort organizations. [2]</td>
</tr>
<tr>
<td>7</td>
<td>CARBANK</td>
<td>Carbanak is a cybercriminal group that has used Carbanak malware to target financial institutions since at least 2013. Carbanak may be linked to groups tracked separately as Cobalt Group and FIN7 that have also used Carbanak malware. [2]</td>
</tr>
<tr>
<td>8</td>
<td>SILENCE</td>
<td>Silence is a financially motivated threat actor targeting financial institutions in different countries. The group was first seen in June 2016. Their main targets reside in Russia, Ukraine, Belarus, Azerbaijan, Poland, and Kazakhstan. [2]</td>
</tr>
<tr>
<td>9</td>
<td>COBALT</td>
<td>Cobalt Group is a financially motivated threat group that has primarily targeted financial institutions since at least 2016. The group has conducted intrusions to steal money via targeting ATM systems, card processing, payment systems and SWIFT systems. Cobalt Group has mainly targeted banks in Eastern Europe, Central Asia, and Southeast Asia. [2]</td>
</tr>
<tr>
<td>10</td>
<td>APT38</td>
<td>APT38 is a North Korean state-sponsored threat group that specializes in financial cyber operations; it has been attributed to the Reconnaissance General Bureau. Active since at least 2014, APT38 has targeted banks, financial institutions, casinos, cryptocurrency exchanges, SWIFT system endpoints, and ATMs [2]</td>
</tr>
<tr>
<td>11</td>
<td>APT41</td>
<td>APT41 is a threat group that researchers have assessed as Chinese state-sponsored espionage group that also conducts financially motivated operations. Active since at least 2012, APT41 has been observed targeting healthcare, telecom, technology, and video game industries in 14 countries. [2]</td>
</tr>
</tbody>
</table>
| 12 | BLACKTECH    | BlackTech is a suspected Chinese cyber espionage group that has primarily targeted organizations in East Asia--particularly Taiwan, Japan, and Hong Kong--and the US since at least 2013. BlackTech has
Threat Group | Introduction
--- | ---
13 DARKVISHNYA | DarkVishnya is a financially motivated threat actor targeting financial institutions in Eastern Europe. In 2017-2018 the group attacked at least 8 banks in this region. [2]
14 EVILNUM | Evilnum is a financially motivated threat group that has been active since at least 2018 [2]
15 EXOTIC LILY | EXOTIC LILY is a financially motivated group that has been closely linked with Wizard Spider and the deployment of ransomware including Conti and Diavol. EXOTIC LILY may be acting as an initial access broker for other malicious actors and has targeted a wide range of industries including IT, cybersecurity, and healthcare since at least September 2021. [2]
16 GOLD SOUTHFIELD | GOLD SOUTHFIELD is a financially motivated threat group active since at least 2018 that operates the REvil Ransomware-as-a-Service (RaaS). GOLD SOUTHFIELD provides backend infrastructure for affiliates recruited on underground forums to perpetrate high value deployments. [2]
17 TA551 | TA551 is a financially motivated threat group that has been active since at least 2018. The group has primarily targeted English, German, Italian, and Japanese speakers through email-based malware distribution campaigns. [2]
18 WIZARD SPIDER | Wizard Spider is a Russia-based financially motivated threat group originally known for the creation and deployment of TrickBot since at least 2016. Wizard Spider possesses a diverse arsenal of tools and has conducted ransomware campaigns against a variety of organizations, ranging from major corporations to hospitals. [2]

Table 1: APT Groups (Financially Motivated)

The information on the Threat groups can be found in [2]

2.2 Heatmap

With many APT groups are financially motivated, we have researched on their Tactics, Techniques and Procedures (TTPs) of these groups mentioned in the Table 1 and created heatmaps for TTPs used by these groups.

What is TTPs?

Based on NIST, Tactics, Techniques and Procedures (TTPs) means

“The behavior of an actor. A tactic is the highest-level description of this behavior, while techniques give a more detailed description of behavior in the context of a tactic, and procedures an even lower-level, highly detailed description in the context of a technique.” [3]
2.2.1 MITRE ATT&CK Heatmap:

A MITRE ATT&CK heatmap is a visual representation that showcases the tactics, techniques, and procedures (TTPs) used by threat actors. It provides a structured way to understand and analyze cybersecurity threats and defenses by mapping observed behaviors. This heatmap can help organizations assess their security posture and develop strategies to defend against cyber threats. We have used python to generate the heatmap in Excel sheet. Below is the python script to generate the MITRE ATT&CK Heatmap.

```python
# MITRE ATT&CK Heatmap
import csv
import json
from collections import Counter
import pandas as pd
import matplotlib.pyplot as plt

# MITRE ATT&CK Heatmap in Excel sheet using python script

# Load the data from CSV file
with open('HeatmapData.csv', 'r') as file:
    reader = csv.reader(file)
    headers = next(reader) # Get the headers
    data = [row for row in reader] # Append headers for the selected columns

# Print the number of data points
print(f'Number of data points: {len(data)}

# Generate a heatmap
plt.figure()
plt.scatter(data[:,0], data[:,1], c=data[:,2], cmap='coolwarm')
plt.colorbar()
plt.title('Heatmap of Data Points')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.show()
```

**Figure 3:** Heatmap with Python Code
The figure presented below is the output generated by the Python script mentioned earlier. It represents a MITRE ATT&CK heatmap focusing on 18 distinct threat groups. In this heatmap, red signifies a high frequency of occurrence of tactics, techniques, and procedures (TTPs), while blue indicates a lower frequency. This visualization helps in quickly identifying the prevalence and distribution of TTPs among the different threat groups.

The visual representation in the figure below is a column heatmap that offers an organized view of tactics used in cyber threats. It arranges these tactics in descending order of frequency, with the most utilized tactics occupying the upper sections and the less frequently employed tactics located lower down in the heatmap. This arrangement provides a clear and intuitive way to understand the distribution and prevalence of tactics used by threat actors.
Based on the MITRE ATT&CK heatmap analysis, we have identified the top 13 techniques that are frequently employed by FIN threat actors in various cyberattacks. These techniques represent the most common strategies and tactics used by malicious actors to compromise systems and networks, highlighting critical areas that organizations should focus on to enhance their cybersecurity defenses.

Our research into the threat actor utilizing these prominent techniques and tactics has led us to identify the threat group FIN 7. This specific threat group extensively employs 11 out of the top 14 tactics and techniques in their malicious activities. The following figure visually represents the mapping of these eleven techniques, showcasing their significance in the operational playbook of the FIN 7 group.

**Figure 4: Column wise Heatmap (With Highest Priority)**

**Figure 5: Top techniques used by FIN groups.**

**Figure 6: Top tactics used by FIN groups.**

**Figure 7: Top techniques used by FIN 7**
Based on the previously outlined rationale and findings, we have made the informed decision to prioritize the investigation and detailed analysis of the threat actor known as FIN 7. This selection is based on various factors, including their extensive use of the top tactics and techniques, making them a significant player in the cybersecurity threat landscape. Further research into FIN 7's tactics, strategies, and characteristics will provide valuable insights and contribute to a better understanding of their activities, ultimately enhancing our ability to defend against their threats.

3.0 FIN 7

FIN7 is a financially motivated threat group that has been active since 2013 primarily targeting the U.S. retail, restaurant, and hospitality sectors, often using point-of-sale malware. A portion of FIN7 was run out of a front company called Combi Security. Since 2020 FIN7 shifted operations to a big game hunting (BGH) approach including use of REvil ransomware and their own Ransomware as a Service (RaaS), Darkside. FIN7 may be linked to the Carbanak Group, but there appears to be several groups using Carbanak malware and are therefore tracked separately.

3.1 Overview of FIN 7

FIN 7 group has been working since 2013. The below figure shows the FIN7 activities in the year 2020-2021. [4]

![Figure 8: FIN7 Activity in 2020-2021](image)

The distribution of relations for FIN 7 is shown in the below figure.

![Figure 9: Distribution of Relations](image)
3.2 Geo Victims and Target of FIN 7 team:

Based on the [thehackernews.com](http://thehackernews.com) article published in the year 2022, An exhaustive analysis of FIN7 has unmasked the cybercrime syndicate's organizational hierarchy, alongside unraveling its role as an affiliate for mounting ransomware attacks. It has also exposed deeper associations between the group and the larger threat ecosystem comprising the now-defunct ransomware DarkSide, REvil, and LockBit families.

The highly active threat group, also known as Carbanak, is known for employing an extensive arsenal of tools and tactics to expand its "cybercrime horizons," including adding ransomware to its playbook and setting up fake security companies to lure researchers into conducting ransomware attacks under the guise of penetration testing.

More than 8,147 victims have been compromised by the financially motivated adversary across the world, with most of the entities located in the U.S. Other prominent countries include China, Germany, Canada, Italy, and the U.K. FIN7's intrusion techniques, over the years, have further diversified beyond traditional social engineering to include infected USB drives, software supply chain compromise, and the use of stolen credentials purchased from underground markets.

3.3 Tools and Malwares and Vulnerabilities used by FIN7:

FIN7 uses various tools such as Powersploit, Mimikalz, Crack MapExec are some of the tools mainly used by this group.

![Figure 10: Tools used by FIN 7](image-url)

Similarly, multiple malwares were used by this group are listed down in the below figure.
Some of the vulnerabilities used by this group FIN 7 are shown in the below figure.

Some of the vulnerabilities used by the FIN 7 group are referenced in the below table.

<table>
<thead>
<tr>
<th>CVE</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVE-2021-42278</td>
<td>Active Directory Domain Services Elevation of Privilege Vulnerability [6]</td>
</tr>
<tr>
<td>CVE-2020-14882</td>
<td>Vulnerability in the Oracle WebLogic Server product of Oracle Fusion Middleware. Easily exploitable vulnerability allows unauthenticated attacker</td>
</tr>
</tbody>
</table>
CVE-2020-14882
CVE-2020-14750
CVE-2020-1472

<table>
<thead>
<tr>
<th>Attack</th>
<th>Brief Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attack Disguised as Brown-Forman Inc.</td>
<td>In June 2021, FIN7 attacked a law firm with a fake complaint that appeared to belong to Brown-Forman Inc., a prominent American company in the wine and spirits industry known for Jack Daniels whisky. This deceptive complaint served as bait to trick a law firm into downloading a version of the JSSLoader Remote Access Trojan (RAT) that was hidden within an Excel file attachment.</td>
</tr>
<tr>
<td>Clever Phishing Lure in the Form of a Gift Card Exchange</td>
<td>In 2020, one of its attacks, FIN7 sent out physical letters purportedly from Best Buy, with a $50 gift card and a USB drive, claiming to contain a list of items to spend on. The USB was identified as a “BadUSB Leonardo USB ATMEGA32U4” device, programmed to emulate a USB keyboard, allowing it to automatically inject malicious commands once plugged in.</td>
</tr>
</tbody>
</table>
Other than the above known attacks, FIN7 (AKA Carbanak) threat actor is linked to Black Basta. Black Basta is a ransomware operator and Ransomware-as-a-Service (RaaS) criminal enterprise that emerged in early 2022 and immediately became one of the most active RaaS threat actors in the world. This intrigued us and we wanted to explore the Black Basta, and we chose this as our incident for our research and started working on this incident.

4.0 Black Basta

4.1 Introduction on Black Basta

The Black Basta operator(s) use the double extortion technique, meaning that in addition to encrypting files on the systems of targeted organizations and demanding ransom to make decryption possible, they also maintain a dark web leak site where they threaten to post sensitive information if an organization chooses not to pay ransom.

Based on Unit 42 report, The ransomware is written in C++ and impacts both Windows and Linux operating systems. It encrypts users’ data using a combination of ChaCha20 and RSA-4096, and to speed up the encryption process, the ransomware encrypts in chunks of 64 bytes, with 128 bytes of data remaining unencrypted between the encrypted regions. [7]

![Figure 13: Black Basta Ransomware Info](image)

The black basta ransomware using QBot as an initial point of entry and to move laterally in compromised networks. QBot, also known as Qakbot, is a Windows malware strain that started as a banking trojan and evolved into a malware dropper. Along with other researchers, we noted that Black Basta infections began with Qakbot delivered by email and macro-based MS Office documents, ISO+LNK droppers and .docx documents exploiting the MSDTC remote code execution vulnerability, CVE-2022-30190. The Black Basta group was observed using Qakbot for both initial access and to spread laterally throughout the network.

The sample of Black Basta file can be downloaded from MalwareBazaar. The link to download the sample is provided here [Black Basta Malware Sample Download][8]. The Black Basta file information is as shown in the below figure:
There are many research conducted on the Black Basta, and the below figure shows Black Basta Attack Lifecycle as explained in Unit42 report.

**Figure 15: Black Basta Attack Lifecycle**

### 4.2 Qakbot/QBot

QBot is a modular information stealer also known as Qakbot or Pinkslipbot. It has been active for years since 2007. It has historically been known as a banking Trojan, meaning that it steals financial data from infected systems, and a loader using C2 servers for payload targeting and download.

#### 4.2.1 Reference

Qakbot/QBot reference available based on our research is from the year 2009 to the year 2023 (October). All along the Qakbot are used various attacks to deliver payloads, connect to C2 servers and in some cases, it helped lateral movement as well. The consolidated reference can be found in the [https://malpedia.caad.fkie.fraunhofer.de/details/win.qakbot](https://malpedia.caad.fkie.fraunhofer.de/details/win.qakbot) [9]

#### 4.2.2 Qakbot Malware sample

Qakbot/QBot malware sample can be downloaded from the following link.
[https://bazaar.abuse.ch/sample/3c35f7163318f296b2f63bae7dfdb1037ac0a383b16d2149a455970a8e139dau/](https://bazaar.abuse.ch/sample/3c35f7163318f296b2f63bae7dfdb1037ac0a383b16d2149a455970a8e139dau/)

The malware sample is a .zip folder which contains an “Adobe Acrobat Document”. The below figure shows the snapshot of the .pdf file.
4.2.3 Qakbot/QBot Infection chain
QBot’s infection chain is described in the following flow-chart.

Figure 17: Qakbot infection chain

4.2.4 Qakbot/QBot Initial Access
The initial infection chain starts by sending specially crafted emails to the target organizations. The method is less sophisticated than spear-phishing techniques but has additional attributes which add to its credibility. One of these is called “Hijacked Email Threads” – capturing archived email conversations and replying to the sender with the malicious content. Those conversations could be captured using Qbot’s Email Collector module. Some examples of crafted phishing emails are as shown below.
4.2.5 Qakbot/QBot Analysis

We have used remnux tool to start the analysis of the downloaded Qakbot malware sample (*.pdf) file. Below is the step-by-step analysis we have conducted on the .pdf file.

1. To understand the sample PDF related information’s, we have used “pdfid.py” tool. The below snapshot shows the command executed and information gathered from the pdf file.

```
remnux@remnux:~/Downloads$ ls
3c35f7163318f290b2f63bae77dfde10377c8a383b16d149e455970a8e139daa.pdf
3c35f7163318f290b2f63bae77dfde10377c8a383b16d149e455970a8e139daa.zip
7zz301.x64.exe
malware_samples
malware_samples.zip
Unit42-Wireshark-Tutorials-main
Unit42-Wireshark-Tutorials-main.zip
remnux@remnux:~/Downloads$ pdfid.py 3c35f7163318f290b2f63bae77dfde10377c8a383b16d149e455970a8e139daa.pdf
PDF Header: %PDF-1.4
obj 99
endobj 99
stream 35
destream 35
xref 1
trailer 1
startxref 1
/Page 9
/%Encrypt 0
/%Object 0
/JS 0
/JavaScript 0
/AA 0
/OpenAction 0
/AcroForm 0
/JBIG2Decode 0
/JavaScript 0
/Launch 0
/%EmbeddedFile 0
/XFA 0
/URI 2
/Colors > 2^24 0
```

2. We found there are “/URI” in the pdf and we used “strings” command to see the URI embedded in the pdf file. Below snapshot shows the command executed and found the URI path

```
remnux@remnux:~/Downloads strings 3c35f7163318f290b2f63bae77dfde10377c8a383b16d149e455970a8e139daa.pdf
```

```
%PDF-1.4
xref 35
destream 35
stream 35
```

```
https://ourloverlyday.us/xuenxavleu/xuenxavleu.gif
```

Figure 18: Qakbot Initial Access – Phishing emails.

Figure 19: pdfid.py information

Below snapshot shows the information gathered from the above link.

![URLhaus Database](image)

**Figure 21: Qakbot Information based on URL.**

From the above figure, we can understand, this PDF is related to Qakbot/QBot related and host is “Online” until 07th July 2023. Also, it was identified that the URL is used for Malware download as well. The related payloads associated with this URL are also available in the location. The below figure shows the information of payloads associated with this URL.
4.2.6 Qakbot Execution:

After the initial analysis, we start exploring the PDF file and how QBot is executing in our test environment. Below are the steps we have identified during our analysis.

1. html drops .zip via html smuggling.
2. zip contains iso file.
3. iso contains .lnk.
4. Lnk file launches calc.exe,
5. calc.exe sideloads windowscodecs.dll
6. windowscodecs.dll executes the malicious payload dll (102755.dll).

Stage 1: Analysis of HTML

1. We downloaded the malware sample file earlier, which contains HTML page. We have analyzed the HTML page and found the variable as shown in the below figure.
2. Based on ChatGPT query [12], we understood the variable started with the text “UEsDB”. This is a common starting sequence in the Base64 representation of a ZIP file. Below is the information gathered from ChatGPT.

- A ZIP file typically starts with a specific byte sequence known as a "magic number" that helps identify it. The magic number for a ZIP file is "PK" (0x50 0x4B in hexadecimal). The base64 representation of these bytes is "UEsD" in ASCII.
- If you have a ZIP file and you want to encode the starting sequence "PK" in base64, you can do so by encoding these bytes as follows:
  - Convert "PK" to its hexadecimal representation: 0x50 0x4B.
  - Encode these bytes in base64: "UEs=

"UEs=" is the base64 encoding of the ZIP file starting sequence "PK." Please note that this base64 encoding only represents the initial bytes of the ZIP file and not the entire file.

3. We downloaded the ZIP file from the text by decoding Base64 to file. Refer the below snapshot on how to download the ZIP file from the HTML page variable.
4. When we tried to extract the zip file, it required the password. We were able to find the password in the HTML page. Refer the below figure for extracted password.

![Figure 25: Stage 1- Identifying the Password](image)

Stage 2: ZIP contains iso file.
1. We were able to extract the content from zip file, using the identified password (abc321). Refer the below figure.

![Figure 26: Stage 2- Extracting the files from the ZIP folder.](image)

2. After extracting the ZIP file, we can extract the files using 7z using the following commands. The below figure shows the extraction of files from the ISO file.

![Figure 27: Stage 2- Extracting the files from the ISO.](image)
Stage 3: ISO contains .lnk

1. As you can see, the extracted folder has multiple files such as calc.exe, dll files, Lnk files. The below figure shows the extracted files.

<table>
<thead>
<tr>
<th>File Name</th>
<th>Library</th>
</tr>
</thead>
<tbody>
<tr>
<td>102755.dll</td>
<td></td>
</tr>
<tr>
<td>calc.exe</td>
<td></td>
</tr>
<tr>
<td>TXRTN_2636021.iso</td>
<td></td>
</tr>
<tr>
<td>TXRTN_2636021.lnk</td>
<td></td>
</tr>
</tbody>
</table>

*Figure 28: Stage 3- Extracted files from the ISO.*
Figure 29: Stage 3- Analysis of files.

Stage 4: .lnk files executes calc.exe.

1. When we analyzed the file “Txrtn_2636021.lnk”, we identified the “calc.exe” is being executed in the behind. Refer the below figure for analysis of .lnk file.

Figure 30: Stage 4- Analysis of Txrtn_2636021.lnk.
Stage 5: calc.exe sideloads windowscodecs.dll

1. When the calc.exe is executed, the windowscodecs.dll file as well. Refer the below figure how the process is executed in the process monitor.

   ![Figure 31: Stage 5- windowscodecs.dll sideloaded when calc.exe executed.](image)

2. Analyzing the windowscodecs.dll file, refer to the below figures for information.

   ![Figure 32: Stage 5- Analysis of windowscodecs.dll file](image)
3. The windowscodecs.dll file is registered using DllRegisterServer and its entries are included in the system registry. This enables other applications to recognize and utilize the functionality provided by the dll.

```c
=== EXPORTS ===

# module "dll helper.dll"
# flags=0x0 ts="2106-02-07 06:28:15" version=0.0 ord_base=1
# nFuncs=2 nNames=2

ORD ENTRY VA NAME
1 1030 DllInstall
2 1060 DllRegisterServer
```

Figure 33: Stage 5- dll registration in DllRegisterServer.

Stage 6: windowscodecs.dll executes the malicious payload dll (102755.dll)

1. When Windowscodecs.dll is loaded, it will execute the malicious payload 102755.dll as well.
2. Analyzing the 102755.dll file, refer to the below figures for information.

Figure 34: Stage 6- 102755.dll Analysis.
3. 102755.dll file will connect to the C2 to download the payloads. Refer the below figures for more information.

![Figure 35: Stage 6- 102755.dll connect with C2.](image)

5.0 Diamond Model

To analyze and understand the cyberthreats and incidents by Qakbot/Qbot, we can use the diamond model. The four components of diamond model Adversary, Infrastructure, Victim and Capability are explained below for Qakbot/Qbot malware.

![Figure 36: Qakbot Diamond Model](image)
Table 4: Mapping of TTPs to QakBot

<table>
<thead>
<tr>
<th>Tactic</th>
<th>Technique</th>
<th>Sub Technique</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TA0002 - Execution</td>
<td>T1204 - User</td>
<td>T1204.001 Malicious Link</td>
<td>QakBot leverages malicious hyperlinks to initiate the execution of its payload.</td>
</tr>
<tr>
<td></td>
<td>Execution</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TA0002 - Execution</td>
<td>T1204 - User</td>
<td>T1204.002 Malicious File</td>
<td>QakBot can be delivered through malicious file attachments, exploiting vulnerabilities upon execution.</td>
</tr>
<tr>
<td></td>
<td>Execution</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TA0005 - Defense Evasion</td>
<td>T1218</td>
<td>T1218.007 System Binary Proxy Execution: Msiexec</td>
<td>QakBot uses the Windows Installer (msiexec) for evasion purposes, blending into legitimate processes.</td>
</tr>
<tr>
<td>TA0004 - Privilege-Escalation</td>
<td>T1055</td>
<td>T1055 Process Injection</td>
<td>QakBot utilizes process injection to run malicious code within legitimate processes, evading detection.</td>
</tr>
<tr>
<td>TA0004 - Privilege-Escalation</td>
<td>T1574</td>
<td>T1574.002 DLL Side-Loading</td>
<td>QakBot may load malicious DLLs into legitimate processes to escalate privileges.</td>
</tr>
<tr>
<td>TA0004 - Privilege-Escalation</td>
<td>T1543</td>
<td>T1543.003 Create or Modify System Process: Windows Service</td>
<td>QakBot may establish persistence by creating a Windows service with malicious functionality.</td>
</tr>
</tbody>
</table>
6.0 Detection for Qakbot/QBot – Yara Rules

YARA is a popular open-source tool and a rule-based language used for identifying and classifying files based on patterns, attributes, and characteristics. YARA rules are essentially a set of defined patterns and conditions that help you search for and identify files or data that match specific criteria. They are widely used in cybersecurity for malware detection, threat hunting, and intrusion detection. Here's an overview of YARA rules: To detect the malicious dll of Qakbot/Qbot, we can create YARA rules as below.

```yaml
rule Detect_102755_DLL {
    meta:
        description = "YARA rule for detecting multiple strings in 102755.dll"
    strings:
        $string_list = "win_hook"
        $string_list = "network_udp_sock"
        $string_list = "network_tcp_list"
        $string_list = "network_tcp_socket"
        $string_list = "network_dns"
        $string_list = "screenshot"
        $string_list = "keylogger"
        $string_list = "win_registry"
        $string_list = "win_files_operation"
        $string_list = "Str_Win32_Winsock2_Library"
        $string_list = "Delphi_FormShow"
        $string_list = "Delphi_CompareCall"
        $string_list = "Delphi_Copy"
        $string_list = "Delphi_StrToInt"
        $string_list = "Delphi_DecodeDate"
        $string_list = "Borland"
        $string_list = "IsPE32"
        $string_list = "IsDLL"
        $string_list = "IsWindowsGUI"
        $string_list = "HasOverlay"
        $string_list = "HasDigitalSignature"
        $string_list = "borland_delphi_dll"
        $string_list = "Borland_Delphi_40_additio"n"
        $string_list = "Microsoft_Visual_Cpp_v50v60_MFC"
        $string_list = "Borland_Delphi_30_additio"n"
        $string_list = "Borland_Delphi_30_
        $string_list = "Borland_Delphi_Setup_Module"
        $string_list = "Borland_Delphi_40"
        $string_list = "Borland_Delphi_v40_v50"
        $string_list = "Borland_Delphi_v30"
        $string_list = "Borland_Delphi_DLL"
```
rule Detect_Calc_EXE {
    meta:
        description = "YARA rule for detecting multiple strings in calc.exe"
    strings:
        $string_list = "SEH_Save"
        $string_list = "SEH_Init"
        $string_list = "Check_OutputDebugStringA_iat"
        $string_list = "anti_dbg"
        $string_list = "screenshot"
        $string_list = "keylogger"
        $string_list = "win_registry"
        $string_list = "IsPE32"
        $string_list = "IsWindowsGUI"
        $string_list = "IsPacked"
        $string_list = "HasDebugData"
        $string_list = "HasRichSignature"

    condition:
        any of them
}

rule Detect_WindowsCodecs_DLL {
    meta:
        description = "YARA rule for detecting multiple strings in WindowsCodecs.dll"
    strings:
        $string_list = "anti_dbg"
        $string_list = "IsPE32"
        $string_list = "IsDLL"
        $string_list = "IsWindowsGUI"
        $string_list = "HasDebugData"
        $string_list = "HasRichSignature"
        $string_list = "Microsoft_Visual_Cpp_v50v60_MFC"

    condition:
        any of them
}
7.0 Understanding the Science behind the Qakbot

After the analysis we started researching the science behind the Qakbot. We tried to get knowledge on Qakbot from the database of Malware Bazaar through API.

1. We send a request to the API, specifically asking for the 1000 "latest" malware SHA-256 samples with the tag "Qakbot." This gives us access to the most recent instances of this malware strain. Each sample is identified by its unique SHA-256 hash. Refer the below figure for the Python code to import the data,

```python
import requests

API_ENDPOINT = "https://mb-api.abuse.ch/api/v1/
headers = {'API-KEY': '12ecb52a793113ef8478bc9f3180317c'} # Replace 'YOUR_API_KEY'
data = {
    'query': 'get_taginfo',
    'tag': 'Qakbot',
    'limit': 1000
}

response = requests.post(API_ENDPOINT, data=data, headers=headers)
if response.status_code == 200:
    results = response.json()
    if 'data' in results:
        sha256_hashes = [entry.get('sha256_hash') for entry in results['data']]
        with open('qakbot_hashes.txt', 'w') as file:
            for hash_value in sha256_hashes:
                file.write(hash_value + '
')
        print("SHA-256 hashes saved to 'qakbot_hashes.txt'.")
    else:
        print("No data found in the response.")
else:
    print("Request failed with status code {response.status_code}.")
```

2. Run the above code to get the 1000 Hash values (latest samples) to get the information about the samples and save them in excel namely First seen, Last seen, Delivery Method, File Type and the corresponding SHA value for the malware.

```python
import requests

API_ENDPOINT = "https://mb-api.abuse.ch/api/v1/
headers = {'API-KEY': '12ecb52a793113ef8478bc9f3180317c'} # Replace with your actual api key

data = {
    'query': 'get_taginfo',
    'tag': 'Qakbot',
    'limit': 1000
}

response = requests.post(API_ENDPOINT, data=data, headers=headers)
if response.status_code == 200:
    results = response.json()
    if 'data' in results:
        sha256_hashes = [entry.get('sha256_hash') for entry in results['data']]
        with open('qakbot_hashes.txt', 'w') as file:
            for hash_value in sha256_hashes:
                file.write(hash_value + '
')
        print("SHA-256 hashes saved to 'qakbot_hashes.txt'.")
    else:
        print("No data found in the response.")
else:
    print("Request failed with status code {response.status_code}.")
```
3. Downloaded intelligence about 1000 latest malware samples from Qakbot malware are in the format shown below.

<table>
<thead>
<tr>
<th>First Seen</th>
<th>Last Seen</th>
<th>File Type</th>
<th>Ivery M-</th>
<th>Hash</th>
</tr>
</thead>
<tbody>
<tr>
<td>2023-10-19 15:28:43</td>
<td>lnk</td>
<td>web_dow</td>
<td>9d6e3977e1e402ca1985e5bb4e194fd37131e019f4e98550e823e6a3a384d19</td>
<td></td>
</tr>
<tr>
<td>2023-10-1 2023-10-2</td>
<td>zip</td>
<td>web_dow</td>
<td>77ca608935e75d21d2b533262b3daa647b2856e743c8d94055117749df66e3ec9150</td>
<td></td>
</tr>
<tr>
<td>2023-07-07 18:32:30</td>
<td>lnk</td>
<td>web_dow</td>
<td>85fa78e2b92c3f4b0e7ae1ba4ada6e895df6f32e3c11ae8a6040e3a478bdab112</td>
<td></td>
</tr>
<tr>
<td>2023-08-25 01:11:58</td>
<td>z7</td>
<td>3f0d42916057ac4df762d6c63c6d8c77e2eb09610121d5eca24b6a2b027baf</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2023-08-18 10:30:44</td>
<td>pja</td>
<td>email_att</td>
<td>1ed3co63ca11665709a84333a9a59b5aaa1cfbb037b52b5899cfe089c12e8ac</td>
<td></td>
</tr>
<tr>
<td>2023-08-03 23:20:20</td>
<td>dll</td>
<td>7ee6095b8ae4e9fe1f7c507382e1aae77544300327385597b2c5771</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2023-01-11 10:43:21</td>
<td>pja</td>
<td>6656a4a4595029707f2012b0b4665767d1f39629a17557778e44596715b999</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2023-07-10 09:31:25</td>
<td>dll</td>
<td>4e2df5aee4f853039f072a41995e577482b2ca4219cd9173f75e0c4b44380</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2023-07-10 09:31:17</td>
<td>pja</td>
<td>cs295f76d653e106bca83837eb17bda2631649b6a7271f0840d0b810100c783a4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2023-07-10 09:31:11</td>
<td>zip</td>
<td>web_dow</td>
<td>7409699a9e6189e197d0d9e3bb379324dfcfef7bb57a8540c6b23e0909e6</td>
<td></td>
</tr>
<tr>
<td>2023-06-25 07:05:26</td>
<td>dll</td>
<td>76199db1cb2ee2e2e38ed3edf5e9e77e0b0c1941e40f866ca9af03b3a1343</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2023-06-23 19:35:13</td>
<td>pja</td>
<td>885300e8b12e3e50a12a49f4e88b7305b7c62ab106bc45a0e6e5c1c8a9620</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2023-06-23 19:34:52</td>
<td>pja</td>
<td>be256d7a70cb3eaa61238f36f3b58a21d616a3b9aab0d6098950d8086e0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2023-06-23 19:34:31</td>
<td>pja</td>
<td>db1983109f4e3e01706815e52e808732a2e99ba5bf6dfef6913c2f018241</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2023-06-23 19:34:05</td>
<td>pja</td>
<td>db816523944135159534b4c34528a7d73b797ae8b07352b488a4c03f9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2023-06-23 19:33:41</td>
<td>pja</td>
<td>22709b9b0b5cc6b6501b8s93509d69b14804311b54aa052cc54a1231fe56e655a</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2023-06-23 19:32:05</td>
<td>dll</td>
<td>4576c28a31de68f44401c3d3e3353b32cc7e6f2cb6c48a2e2cdd868a28ddba97</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2023-06-23 19:31:24</td>
<td>dll</td>
<td>8366c26e84062db2a3794666313e520eb4e4edc8e1e44e4a2e890d47b5f98e5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4. With the information gathered in excel, we plotted the common delivery method for Qakbot. The below figure shows the code used to plot the chart.

```
# -*- coding: utf-8 -*-

"Created on Sun Oct 29 03:15:36 2023"

@author: shiva

import pandas as pd
import matplotlib.pyplot as plt

# Reading the CSV
df = pd.read_excel('output.xlsx')

# Extracting the 'Delivery Method' column
delivery_methods = df['loc']

# Counting frequencies
delivery_counts = delivery_methods.value_counts().nlargest(5)

# Plotting the bar chart
plt.figure(figsize=(10, 6))
delivery_counts.plot(kind='bar', color='red')
plt.title('Top 5 Delivery Method of Qakbot')
plt.xlabel('Delivery Method of Qakbot')
plt.ylabel('Frequency')
plt.show()
```

The most used delivery method of Qakbot is “Web_download”. The below is the top delivery methods.

1. Web_download
2. email_attachment
3. email_link
4. other
5. multiple
Refer the below figure for the generated chart.

5. With the information gathered in excel, we plotted the delivery method for Qakbot. The below figure shows the code used to plot the chart.

```python
import pandas as pd
import matplotlib.pyplot as plt
from datetime import datetime

# Reading the CSV
df = pd.read_excel('output.xlsx')
# Extracting the 'Delivery Method' column
delivery_methods = df.iloc[:, 2]

# Counting frequencies
delivery_counts = delivery_methods.value_counts().nlargest(10)

# Plotting the bar chart
plt.figure(figsize=(10, 6))
delivery_counts.plot(kind='bar', color='blue')
plt.title('Top 5 File Types used to deliver (Over 1000 samples)')
plt.xlabel('Delivery Method of Quakbot')
plt.ylabel('Frequency')
plt.show()
```

The most common file types used to deliver Qakbot are represented in the below chart.
6. Timeline of samples first seen on malware bazaar for Qakbot (entry effect). It is important to note that we picked the latest 1000 samples. So, it is easy to see that after 2023-07, the sample is not being uploaded anymore, indicating likely that it is patched and isn’t functional, and the CCs are dow
7. Timeline of samples last seen on malware bazaar for Qakbot (trailing effect). It is important to note that we picked the latest 1000 samples. So, it is easy to see that after 2023-06/7, the sample is not being uploaded anymore, indicating likely that it is patched and isn’t functional, and the CCs are down.

```
df = pd.read_excel('output.xlsx')

dates = df['datetime']
values = [1] * len(dates)

plt.scatter(dates, values, color='skyblue', s=100)

plt.title('Timeline of Samples last Uploaded on Malware Bazaar (trailing effect)')
plt.xlabel('Date')
plt.ylabel('Occurrence (1=File uploaded)')
plt.show()
```
8.0 Mitigations for the identified Techniques

<table>
<thead>
<tr>
<th>Technique ID</th>
<th>Frequency</th>
<th>Technique</th>
<th>Tactics</th>
<th>Mitigations</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1210</td>
<td>10</td>
<td>Execution</td>
<td>Initial Access</td>
<td>Quarantine using Antivirus software.</td>
</tr>
<tr>
<td>T1205</td>
<td>10</td>
<td>Execution</td>
<td>Initial Access</td>
<td>Quarantine using Antivirus software.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Technique ID</th>
<th>Frequency</th>
<th>Technique</th>
<th>Tactics</th>
<th>Mitigations</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1207</td>
<td>10</td>
<td>Indicator Removal</td>
<td>Defense</td>
<td>Defense Evasion</td>
</tr>
<tr>
<td>T1201</td>
<td>10</td>
<td>Collocated Files or Information</td>
<td>Defense</td>
<td>Defense Evasion</td>
</tr>
<tr>
<td>T1203</td>
<td>10</td>
<td>Remote Services</td>
<td>Local Movement</td>
<td>Use multi-factor authentication on remote service where possible.</td>
</tr>
</tbody>
</table>
| T1205        | 10        | Schedule Task | Execution | Privilege Escalation | Tools to help the PowerShell framework contain PowerShell modules that can be used to escalate privileges to local and domain accounts. | Configure settings for scheduled tasks to force tasks to run under the context of the authenticated account instead of allowing them to run as SYSTEM. | Configure the Increase Scheduling Priority option to only allow the authenticated account to run them. | Restrict user account privileges and use Privilege Escalation vectors so that only authorized administrators can use scheduled tasks on remote systems. | Use Multi-factor Authentication on remote service. | Use Training.
The mitigation for various techniques were derived from mitre.org. The reference [2] can be found under section Error! Reference source not found.

9.0 Conclusion:
The QBOT malware family is highly active and still part of the threat landscape in mid-2023 due to its features and its powerful modular system. While initially characterized as an information stealer in 2007, this family has been leveraged as a delivery mechanism for additional malware and post-compromise activity.

However, with recent samples received in Malware Bazaar, the number of samples received is “null” and this explains most of the vulnerabilities used by this malware are already patched.

10.0 Appendix
We have utilized OpenCTI as threat intelligence platform to understand better on the Qakbot/Qbot malwares.

10.1 OpenCTI
OpenCTI is an open-source platform allowing organizations to manage their cyber threat intelligence knowledge and observables. It has been created to structure, store, organize and visualize technical and non-technical information about cyber threats.

Below are some of the snapshots of the OpenCTI platform. (We will demonstrate OpenCTI during our presentation)
10.2 Cuckoo

Cuckoo Sandbox is free software that automated the task of analyzing any malicious file under Windows, macOS, Linux, and Android. Below are some of the snapshots of the Cuckoo platform. (We will demonstrate Cuckoo during our presentation). [13]
Figure 39: Snapshots of Cuckoo Sandbox


