Threat Actors Delivers New Rozena backdoor with Follina Bug – Detection & Response

socinvestigation.com/threat-actors-delivers-new-rozena-backdoor-with-follina-bug-detection-response/

July 11, 2022

IOC

By

BalaGanesh

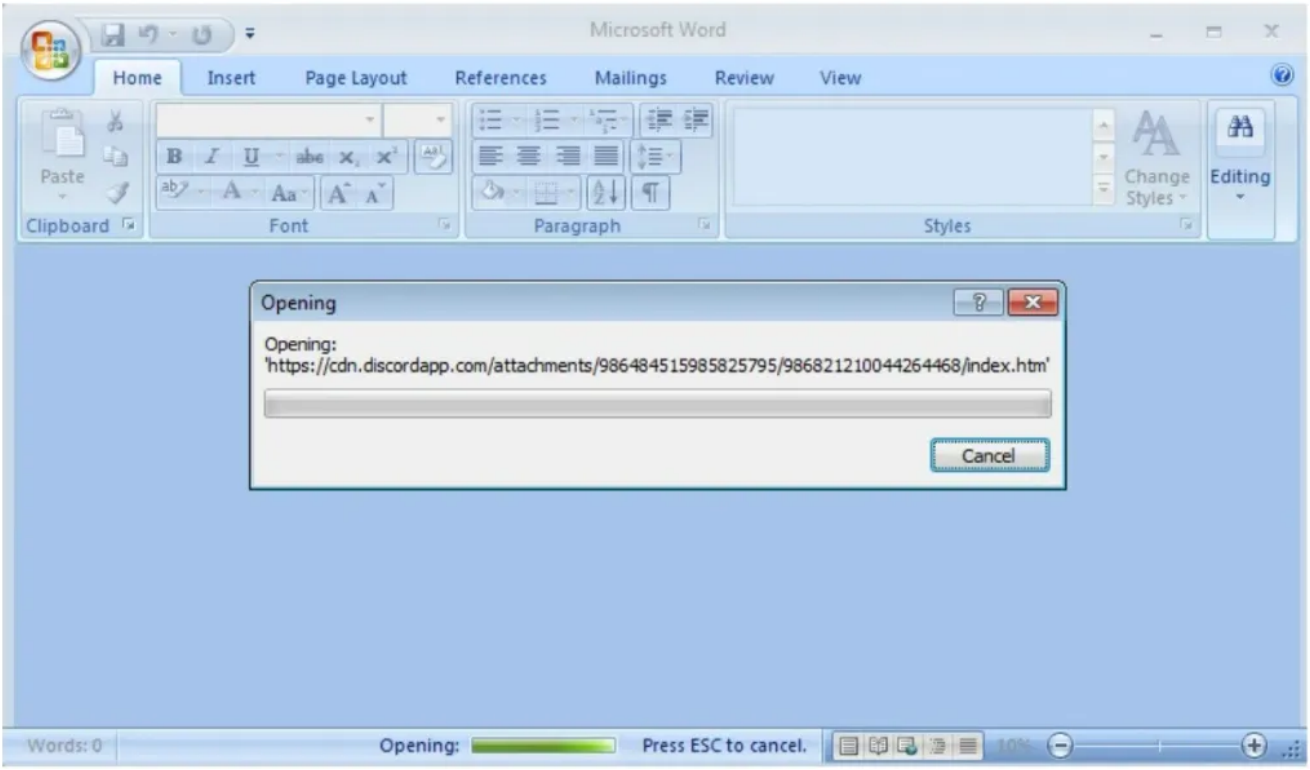
-

July 11, 2022

0

Fortinet FortiGuard Labs researchers observed a phishing campaign that is leveraging the recently disclosed Follina security vulnerability (CVE-2022-30190, CVSS score 7.8) to distribute the Rozena backdoor on Windows systems. The **Follina** issue is a remote code execution vulnerability that resides in the Microsoft Windows Support Diagnostic Tool (MSDT).

The Rozena backdoor is able to inject a remote shell connection back to the attacker's machine
The attack chain leverages a weaponized Office document that once clicked, starts connecting to an external Discord CDN URL ('hxxps://cdn[.]discordapp.com/attachments/986484515985825795/986821210044264468/index[.]htm) to download an HTML file (index.htm).



Then the HTML file invokes the msdt.exe tool with a PowerShell command which also invokes another web request to download the Rozena backdoor and save it as "Word.exe."

Also Read: Latest IOCs – Threat Actor URLs , IP's & Malware Hashes

"The PowerShell code will download one batch file cd.bat and start it with no window to hide. Then it invokes another web request to download Rozena and saves as "Word.exe" in the Windows Tasks folder." reads the **post** published by Fortinet FortiGuard Labs.

"the attacker decided to distract the victim. The original file has no content besides an external link in oleObject. To keep the victim from noticing anything odd the batch file downloads another Word document, 1c9c88f811662007.docx with a lot of pictures in it. To make it seem more real, this document is saved in directory C:\users\$env:USERNAME\Downloads, with a shorter name, 18562.docx."

The main feature of the Rozena backdoor is to inject shellcode that launches a reverse shell to the attacker's machine ("microsofto.duckdns[.]org"), in this way the attacker can take full control of the system.

Once the Rozena executable is run, it will create a process for a PowerShell command, experts pointed out that the decoded command has only one job to do, injecting the shellcode.

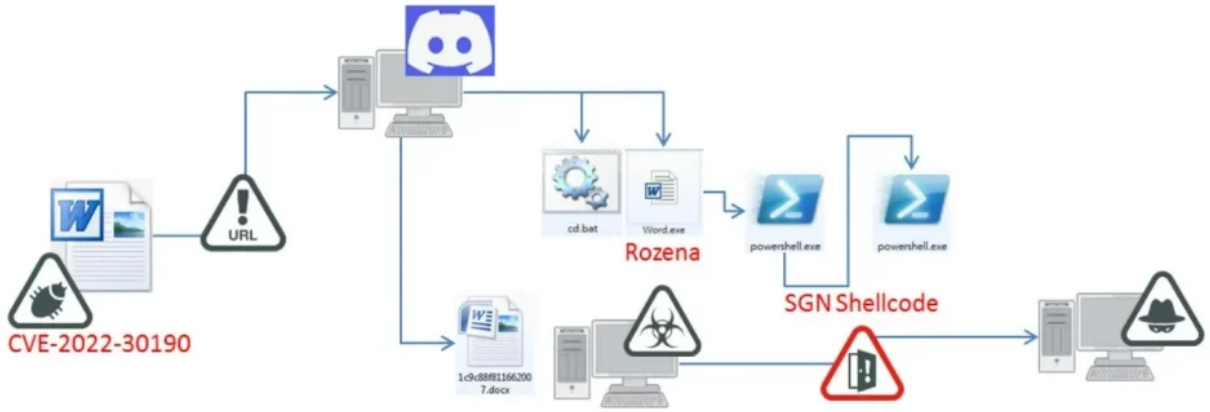


Figure 15. Attack scenario

Source: https://www.fortinet.com

Also Read: Threat Hunting Using Powershell and Fileless Malware Attacks

Indicator of Compromise:

SHA256:

432bae48edf446539cae5e20623c39507ad65e21cb757fb514aba635d3ae67d6

5d8537bd7e711f430dc0c28a7777c9176269c8d3ff345b9560c8b9d4daaca002

3558840ffbc81839a5923ed2b675c1970cdd7c9e0036a91a0a728af14f80eff3

27f3bb9ab8fc66c1ca36fa5d62ee4758f1f8ff75666264c529b0f2abbade9133

69377adfdfa50928fade860e37b84c10623ef1b11164ccc6c4b013a468601d88

CVE-2022-30190 is a high-severity vulnerability that lets a malicious actor deliver malware though an MS Word document. Microsoft already released a patch for it on June 14, 2022. In this blog we showed how an attacker exploits Follina and included details of Rozena and the SGN ShellCode. Users should apply the patch immediately and also apply FortiGuard protection to avoid the threat." concludes the report.

Also Read: Soc Interview Questions and Answers – CYBER SECURITY ANALYST

Detection & Response:

Splunk:

```
source="WinEventLog:*" AND ((((CommandLine="*msdt.exe*") AND
(CommandLine="*PCWDiagnostic*" OR CommandLine="*//*" OR CommandLine="*./*" OR
CommandLine="*/.*" OR CommandLine="*../*")) OR Image="*\\word.exe") OR
(Image="*\\powershell.exe" AND (CommandLine="*Invoke-Expression*" OR
CommandLine="*ExecutionPolicy Bypass*" OR CommandLine="*invoke*" OR
CommandLine="*NoProfile*" OR CommandLine="*JAB*")))
```

Qradar:

```
SELECT UTF8(payload) from events where LOGSOURCETYPENAME(devicetype)='Microsoft Windows
Security Event Log' and ((((("Process CommandLine" ilike '%msdt.exe%') and ("Process
CommandLine" ilike '%PCWDiagnostic%' or "Process CommandLine" ilike '%//%' or "Process
CommandLine" ilike '%./%' or "Process CommandLine" ilike '%/.%' or "Process CommandLine"
ilike '%../%')) or "Image" ilike '%\word.exe') or ("Image" ilike '%\powershell.exe' and
("Process CommandLine" ilike '%Invoke-Expression%' or "Process CommandLine" ilike
'%ExecutionPolicy Bypass%' or "Process CommandLine" ilike '%invoke%' or "Process
CommandLine" ilike '%NoProfile%' or "Process CommandLine" ilike '%JAB%')))
```

Elastic Query:

```
(((process.command_line:*msdt.exe* AND process.command_line:(*PCWDiagnostic* OR *\/\/* OR
*.\/* OR *\/.* OR *..\/*)) OR process.executable:*\\word.exe) OR
(process.executable:*\\powershell.exe AND process.command_line:(*Invoke\-Expression* OR
*ExecutionPolicy\ Bypass* OR *invoke* OR *NoProfile* OR *JAB*)))
```

Arcsight:

```
(((((((deviceCustomString1 CONTAINS "*msdt.exe*" OR destinationServiceName CONTAINS
"*msdt.exe*")) AND ((deviceCustomString1 CONTAINS "*PCWDiagnostic*" OR deviceCustomString1
CONTAINS "*//*" OR deviceCustomString1 CONTAINS "*./*" OR deviceCustomString1 CONTAINS
"*/.*" OR deviceCustomString1 CONTAINS "*../*" OR destinationServiceName CONTAINS
"*PCWDiagnostic*" OR destinationServiceName CONTAINS "*//*" OR destinationServiceName
CONTAINS "*./*" OR destinationServiceName CONTAINS "*/.*" OR destinationServiceName
CONTAINS "*../*"))) OR ((destinationProcessName ENDSWITH "\\word.exe" OR sourceProcessName
ENDSWITH "\\word.exe" OR deviceProcessName ENDSWITH "\\word.exe")))) OR
(((destinationProcessName ENDSWITH "\\powershell.exe" OR sourceProcessName ENDSWITH
"\\powershell.exe" OR deviceProcessName ENDSWITH "\\powershell.exe")) AND
((deviceCustomString1 CONTAINS "*Invoke-Expression*" OR deviceCustomString1 CONTAINS
"*ExecutionPolicy Bypass*" OR deviceCustomString1 CONTAINS "*invoke*" OR
deviceCustomString1 CONTAINS "*NoProfile*" OR deviceCustomString1 CONTAINS "*JAB*" OR
destinationServiceName CONTAINS "*Invoke-Expression*" OR destinationServiceName CONTAINS
"*ExecutionPolicy Bypass*" OR destinationServiceName CONTAINS "*invoke*" OR
destinationServiceName CONTAINS "*NoProfile*" OR destinationServiceName CONTAINS
"*JAB*")))))
```

CarbonBlack:

```
(((process_cmdline:*msdt.exe* AND process_cmdline:(*PCWDiagnostic* OR *\/\/* OR *.\/* OR
*\/.* OR *..\/*)) OR process_name:*\\word.exe) OR (process_name:*\\powershell.exe AND
process_cmdline:(*Invoke\-Expression* OR *ExecutionPolicy\ Bypass* OR *invoke* OR
*NoProfile* OR *JAB*)))
```

Crowdstike:

```
(((((CommandLine="*msdt.exe*") OR (CommandHistory="*msdt.exe*")) AND
((CommandLine="*PCWDiagnostic*" OR CommandLine="*//*" OR CommandLine="*./*" OR
CommandLine="*/.*" OR CommandLine="*../*") OR (CommandHistory="*PCWDiagnostic*" OR
CommandHistory="*//*" OR CommandHistory="*./*" OR CommandHistory="*/.*" OR
CommandHistory="*../*"))) OR ImageFileName="*\\word.exe") OR
(ImageFileName="*\\powershell.exe" AND ((CommandLine="*Invoke-Expression*" OR
CommandLine="*ExecutionPolicy Bypass*" OR CommandLine="*invoke*" OR
CommandLine="*NoProfile*" OR CommandLine="*JAB*") OR (CommandHistory="*Invoke-Expression*"
OR CommandHistory="*ExecutionPolicy Bypass*" OR CommandHistory="*invoke*" OR
CommandHistory="*NoProfile*" OR CommandHistory="*JAB*"))))
```

Fireeye:

```
(metaclass:`windows` (((args:`msdt.exe` args:[`PCWDiagnostic`,`//`,`./`,`/.`,`../`]) OR
process:`*\word.exe`) OR (process:`*\powershell.exe` args:[`Invoke-
Expression`,`ExecutionPolicy Bypass`,`invoke`,`NoProfile`,`JAB`])))
```

Graylog:

```
(((CommandLine.keyword:*msdt.exe* AND CommandLine.keyword:(*PCWDiagnostic* *\/\/* *.\/*
*\/.* *..\/*)) OR Image.keyword:*\\word.exe) OR (Image.keyword:*\\powershell.exe AND
CommandLine.keyword:(*Invoke\-Expression* *ExecutionPolicy\ Bypass* *invoke* *NoProfile*
*JAB*)))
```

Microsoft Defender:

```
DeviceProcessEvents | where ((((ProcessCommandLine contains "msdt.exe") and
(ProcessCommandLine contains "PCWDiagnostic" or ProcessCommandLine contains "//" or
ProcessCommandLine contains "./" or ProcessCommandLine contains "/." or ProcessCommandLine
contains "../")) or FolderPath endswith @"\word.exe") or (FolderPath endswith
@"\powershell.exe" and (ProcessCommandLine contains "Invoke-Expression" or
ProcessCommandLine contains "ExecutionPolicy Bypass" or ProcessCommandLine contains
"invoke" or ProcessCommandLine contains "NoProfile" or ProcessCommandLine contains
"JAB")))
```

Microsoft Sentinel:

```
SecurityEvent |  where EventID == 4688 | where ((((CommandLine contains 'msdt.exe') and
(CommandLine contains 'PCWDiagnostic' or CommandLine contains '//' or CommandLine contains
'./' or CommandLine contains '/.' or CommandLine contains '../')) or NewProcessName
endswith @'\word.exe') or (NewProcessName endswith @'\powershell.exe' and (CommandLine
contains 'Invoke-Expression' or CommandLine contains 'ExecutionPolicy Bypass' or
CommandLine contains 'invoke' or CommandLine contains 'NoProfile' or CommandLine contains
'JAB')))
```

RSA Netwitness:

```
((((CommandLine contains 'msdt\.exe') && (CommandLine contains 'PCWDiagnostic', '//',
'./', '/\.', '.\./')) || (Image contains 'word.exe')) || ((Image contains
'powershell.exe') && (CommandLine contains 'Invoke-Expression', 'ExecutionPolicy Bypass',
'invoke', 'NoProfile', 'JAB')))
```

Google Chronicle:

```
((((target.process.command_line = /.*msdt\.exe.*/ and (target.process.command_line =
/.*PCWDiagnostic.*/ or target.process.command_line = /.*\/\/.*/ or
target.process.command_line = /.*\.\/.*/ or target.process.command_line = /.*\/\..*/ or
target.process.command_line = /.*\.\.\/.*/)) or target.process.file.full_path =
/.*\\word\.exe$/)) or (target.process.file.full_path = /.*\\powershell\.exe$/ and
(target.process.command_line = /.*Invoke-Expression.*/ or target.process.command_line =
/.*ExecutionPolicy Bypass.*/ or target.process.command_line = /.*invoke.*/ or
target.process.command_line = /.*NoProfile.*/ or target.process.command_line =
/.*JAB.*/)))
```

Aws OpenSearch:

```
(((process.command_line:*msdt.exe* AND process.command_line:(*PCWDiagnostic* OR *\/\/* OR
*.\/* OR *\/.* OR *..\/*)) OR process.executable:*\\word.exe) OR
(process.executable:*\\powershell.exe AND process.command_line:(*Invoke\-Expression* OR
*ExecutionPolicy\ Bypass* OR *invoke* OR *NoProfile* OR *JAB*)))
```

Source/Credits: https://www.fortinet.com/blog/threat-research/follina-rozena-leveraging-discord-to-distribute-a-backdoor

ht://securityaffairs.co/wordpress/133051/hacking/follina-bug-rozena-backdoor.html

LEAVE A REPLY

Please enter your comment!

Please enter your name here

You have entered an incorrect email address!

Please enter your email address here