7 **Node Installer Local Privilege Escalation**

Share:

TIMELINE

deepsurface-robert submitted a report to Node.js. May 27th (2 ye

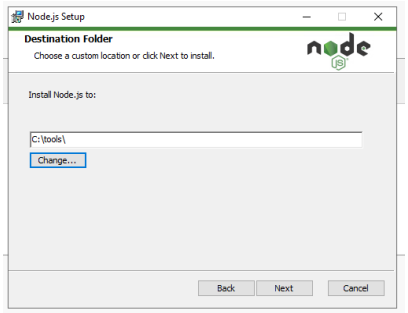
Node is vulnerable to local privilege escalation attacks under certain conditions on Windows platforms. More specifically, improper configuration of permissions in installation directory allows an attacker to perform two different escalation attacks: PATH and DLL hijacking.

To demonstrate this flaw, we first download the latest version of Node from https://nodejs.org/en/download/. At the time of writing, this was node version 14.17.

We follow the standard installation steps, except for the installation directory, which we change to `C:\tools`. This directory can either be created through the ins GUI, or through `mkdir C:\tools`.

Image F1318095: image1.png 16.45 KiB

Zoom in Zoom out Copy Download



We also select the option in a later step to "automatically install the necessary tools".

In the screenshot below, note the improper permissions, `BUILTIN\Users Allow *`, on the installation directory, which are inherited from the drive root. This gives a local user the ability to create arbitrary files in the installation directory.

Image F1318096: image4.png 32.53 KiB

Zoom in Zoom out Copy Download

```
PS C:\tools> get-acl . | fl

Path   : Microsoft.PowerShell.Core\FileSystem::C:\tools
Owner  : BUILTIN\Administrators
Group  : EC2AMAZ-K83CFT1\None
Access : NT AUTHORITY\SYSTEM Allow  FullControl
         BUILTIN\Administrators Allow  FullControl
         BUILTIN\Users Allow  ReadAndExecute, Synchronize
         BUILTIN\Users Allow  AppendData
         BUILTIN\Users Allow  CreateFiles
         CREATOR OWNER Allow  268435456
```

This unprotected directory has also been added to the system `PATH` variable, allowing an attacker to drop malicious executables in that directory and have them executed by other users in certain circumstances. (Note that you may have to start a new powershell instance to see the `PATH` change.)

Image F1318097: image5.png 47.69 KiB

Zoom in Zoom out Copy Download

```
PS C:\Users\Administrator> $env:path -split ";"
C:\Python39\Scripts\
C:\Python39\
C:\Windows\system32
C:\Windows
C:\Windows\System32\Wbem
C:\Windows\System32\WindowsPowerShell\v1.0\
C:\Windows\System32\OpenSSH\
C:\Program Files\Amazon\cfn-bootstrap\
C:\tools\
C:\ProgramData\chocolatey\bin
C:\Users\Administrator\AppData\Local\Microsoft\WindowsApps
C:\Users\Administrator\AppData\Roaming\npm
```

To fully demonstrate the implications of this vulnerability, first create a new unprivileged user. Then, as this user, drop a malicious exe into the `C:\tools` directory rename it to `npm.exe`. For testing purposes, you can simply do `cp node.exe npm.exe`. Note that the same could be done for `npx`.

Windows will search for a program with the `.exe` extension first, meaning that the malicious npm.exe will take precedence over `npm.cmd`.

Now, as the privileged user, try running `npm`. This should drop you into the node shell, demonstrating how an attacker could run a malicious executable.

Image F1318098: image2.png 12.51 KiB

Zoom in Zoom out Copy Download

```
PS C:\tools> npm
Welcome to Node.js v14.17.0.
Type ".help" for more information.
>
```

Aside from the `PATH` vulnerability, the insecure permissions configured could also allow an attacker to perform a DLL hijacking attack against the `node.exe`. Usin Process Monitor, we can confirm that node attempts to load a number of DLLs from the unprotected folder.

Image F1318099: image3.png 25.49 KiB

Zoom in Zoom out Copy Download

10:5...	node.exe	54...	CreateFile	C:\tools	SUCCESS	Desired Acc...
10:5...	node.exe	54...	CreateFile	C:\tools\dbghelp.dll	NAME NOT ...	Desired Acc...
10:5...	node.exe	54...	CreateFile	C:\tools\PHLPAPI.DLL	NAME NOT ...	Desired Acc...
10:5...	node.exe	54...	CreateFile	C:\tools\USERENV.dll	NAME NOT ...	Desired Acc...
10:5...	node.exe	54...	CreateFile	C:\tools\WINMM.dll	NAME NOT ...	Desired Acc...
10:5...	node.exe	54...	CreateFile	C:\tools\WINMMBASE.dll	NAME NOT ...	Desired Acc...

For more information on DLL hijacking attacks, see our blog post.

It is worth noting that a very similar problem was discovered in RabbitMQ and reported by the DeepSurface Security research team. The RabbitMQ team fixed this issue in May 2021. For more information, see: CVE-2021-22117.

Impact

A locally unprivileged attacker could perform a local privilege escalation attack through PATH and DLL hijacking.

5 attachments:
F1318095: image1.png
F1318096: image4.png
F1318097: image5.png
F1318098: image2.png
F1318099: image3.png

mcollina Node.js staff posted a comment. May 31st (2 ye

Thanks Robert for reporting this issue. I'm going to ask a little bit more patience while we triage this as we do not have many Windows developers on the team.

Have you got a pointer to what would be the fix for this issue?

deepsurface-robert posted a comment. May 31st (2 ye

Hi @mcollina,

No worries, thanks for looking over this report.

Regarding the fix, the goal would be to remove the `AppendData` and `CreateFiles` permissions from `BUILTIN\Users` in the installation directory. This can be done removing inheritance from the parent directory and regranting read + execute for normal users. For example, see rabbitMQ's patch and erlang's patch.

-Robert

mcollina Node.js staff changed the status to **Triaged**. Jun 5th (2 ye

Thanks, this is confirmed. We'll work on a solution and keep you up to date on our progress.

deepsurface-robert invited another hacker as a collaborator. Jun 5th (2 ye

deepsurface joined this report as a collaborator. Jun 5th (2 ye

kumarak39 Node.js staff posted a comment. Jun 11th (2 ye

@mcollina, I raised PR with the changes which explicitly configure ACL for the install directory (https://github.com/nodejs-private/node-private/pull/269).

The changes remove the ACL inherited from the parent and configure the install directory with the new permission list. All install paths are configured with the sar ACL and if one chooses to install at default path (`C:\"Program Files"`) or custom path (`C:\tools`) will have the same permission list. Please let me know your thoughts about it.

Here is the ACL list shows up on installing it at `C:\tools` :

Code 453 Bytes Wrap lines Copy Dow

```
Path   : Microsoft.PowerShell.Core\FileSystem::C:\tools
Owner  : NT AUTHORITY\SYSTEM
Group  : NT AUTHORITY\SYSTEM
Access : NT AUTHORITY\Authenticated Users Allow  ReadAndExecute, Synchronize
         NT AUTHORITY\SYSTEM Allow  FullControl
         BUILTIN\Administrators Allow  FullControl
         BUILTIN\Users Allow  ReadAndExecute, Synchronize
Audit  :
Sddl   : O:SYG:SYD:P(A;OICI;0x1200a9;;;AU)(A;OICI;FA;;;SY)(A;OICI;FA;;;BA)(A;OICI;0x1200a9;;;BU)
```

Please let me know your feedback about the changes. Also, suggest me if I missed something. I checked the ACL list of the default install path and most of the permissions are inherited from the parent and I think we don't need to explicitly add any service to the list.

deepsurface-robert posted a comment. Jun 12th (2 ye

Hi @kumarak39,

Thanks for the prompt response, that ACL looks good to me.

-Robert

mcollina Node.js staff posted a comment. Jun 14th (2 ye

Amazing. How could we attribute this discovery to you Robert?

deepsurface-robert posted a comment. Jun 15th (2 ye

Would it be possible to credit "Robert Chen from DeepSurface Security"?

Sure, what email address should we use?

deepsurface-robert posted a comment. Jun 15th (2 ye

`security@deepsurface.com` would be great.

danbev updated CVE reference to CVE-2021-22921. Jun 23rd (about 1 y

richardlau Node.js staff joined this report as a participant. Jul 1st (about 1 y

danbev closed the report and changed the status to **Resolved**. Jul 1st (about 1 y

This fix has now been released.

danbev requested to disclose this report. Jul 1st (about 1 y

deepsurface-robert agreed to disclose this report. Jul 1st (about 1 y

This report has been disclosed. Jul 1st (about 1 y

he Internet Bug Bounty rewarded deepsurface with a **$125** bounty. Jul 6th (about 1 y

Thank you for this report, we appreciate it.

he Internet Bug Bounty rewarded deepsurface-robert with a **$125** bounty. Jul 6th (about 1 y

Thank you for this report, we appreciate it.