Severe local 0-Day escalation exploit found in Steam Client Services

Enlarge / Breaking bugs are as described—a security flaw in Steam's client service allows easy execution of arbitrary code as LOCALSYSTEM.

reader comments

134 with 91 posters participating, including story author

Share this story

Earlier today, disgruntled security researcher Vasily Kravets released a zero-day vulnerability in the Windows version of the ubiquitous gaming service Steam. The vulnerability allows any user to run arbitrary code with LOCALSYSTEM privileges using just a very few simple commands.

The vulnerability lies within Steam Client Service. The service may be started or stopped by unprivileged users. This becomes a problem because, when run,Steam Client Service automatically sets permissions on a range of registry keys. If a mischievous—or outright malicious—user were to symlink one of these keys to that belonging to another service, it becomes possible for arbitrary users to start or stop *that* service as well. This becomes even more problematic when you realize that it's possible to pass arguments to services that run under extremely privileged accounts—such as msiserver, the Windows Installer service.

Enlarge / Following a demonstration I saw from Redditor /u/R_Sholes today, I used an unprivileged user account to write a file to C:\Windows\System32 as LOCALSYSTEM. That’s game over, for those of you playing along from home.
Jim Salter
The image walkthrough above follows a few simple steps:

1. Demonstrate that I cannot write to C:\Windows\System32\. System error message is in red.
2. Demonstrate that I cannot arbitrarily monkey around with registry keys under HKLM\CurrentControlSet\Services. System-error message is in red.
3. Delete the NSIS installer key for Steam (to give myself a target for shenanigans), then recreate it as a symlink to msiserver‘s registry key. Success in green.
4. Demonstrate that Steam’s NSIS key now points to Windows’ msinstaller key. Success in green.
5. Attempt to modify the msiserver key to run my shenanigans. I haven’t started Steam Client Services

yet, so this fails. System-error message in red.
6. Start Steam Client Services, *then* modify the msiserver key once Steam has helpfully opened it up for me. Success in green.
7. Start the newly-modified msiserver service. msiserver runs as LOCALSYSTEM, so it successfully creates a file under C:\Windows\System32. Success in green.

I did this test on a clean Windows VM; aside from Steam itself, the only code I needed to download was regln-x64.exe, a simple utility for the linking of registry keys, which requires no installation. Windows User Account Control was never triggered during this process, and the whole thing only took a few minutes. I did not have any Steam games installed, so I just monkeyed with the Steam installer.

A genuinely malicious user might use this procedure to directly pop a locally or remotely accessible shell with LOCALSYSTEMprivileges, after which they can do whatever they like with no further tricks necessary.

Professional driver on a closed course

Do not follow this procedure on a Windows machine you care about; this demonstration left both Steam's installer and the Windows Installer service broken, because this is a throwaway virtual machine.

The vulnerability demonstrated here is only 45 days old. Normally, publicly disclosing an exploit this quickly would be a big no-no in the Infosec community—the typical grace period for response is 90 days. In this case, it's difficult to point any blame to the researcher. Upon first reporting the bug via HackerOne, it was rejected as out-of-scope, with «Attacks that require the ability to drop files in arbitrary locations on the user's filesystem» as the reason given.

The attack does not require any file to be dropped anywhere or any special privileges. Although we downloaded regln-x64 to make the proof of concept prettier, I could have accomplished its task—symlinking registry keys—directly inside regedit.exe.

When the researcher argued with HackerOne's staff, a second HackerOne employee eventually reproduced the exploit, confirmed the report, and sent it off to Valve. But a few weeks later, a third HackerOne employee rejected it again. The employee reiterated «Attacks that require the ability to drop files in arbitrary locations on the user's filesystem» and added «Attacks that require physical access to the user's device» as reasons the vulnerability is supposedly out-of-scope.

Rejected

The second reason for rejection is no more valid than the first: a malicious "game" developer could easily create a free-to-play "game" that reproduces all the steps of this exploit. Such a bad actor could pop a shell with LOCALSYSTEM privileges and own the user's machine.

With this second rejection, Vasily decided there was no further recourse but public disclosure, and he informed HackerOne that he would disclose after July 30. He alleges that on August 2, yet another HackerOne employee forbid the disclosure of the vulnerability, despite HackerOne having closed it repeatedly as out-of-scope while Valve itself never weighed in one way or the other.

Ars has reached out to Valve about this story, and we will update with any response.

https://arstechnica.com/?p=1547805