	id	summary
0	CVE-2022-1620	NULL Pointer Dereference in function vim_regexec_string at regexp.c:2729 in GitHub repository vim/vim prior to 8.2. NULL Pointer Dereference in function vim_regexec_string at regexp.c:2729 allows attackers to cause a denial of service (application crash) via a crafted input.
1	CVE-2022-1619	Heap-based Buffer Overflow in function cmdline_erase_chars in GitHub repository vim/vim prior to 8.2. This vulnerabilities are capable of crashing software, modify memory, and possible remote execution
2	CVE-2018-25033	ADMesh through 0.98.4 has a heap-based buffer over-read in stl_update_connects_remove_1 (called from stl_remove_degenerate) in connect.c in libadmesh.a.
3	CVE-2022-1616	Use after free in append_command in GitHub repository vim/vim prior to 8.2.4895. This vulnerability is capable of crashing software, Bypass Protection Mechanism, Modify Memory, and possible remote execution
4	CVE-2022-24735	Redis is an in-memory database that persists on disk. By exploiting weaknesses in the Lua script execution environment, an attacker with access to Redis prior to version 7.0.0 or 6.2.7 can inject Lua code that will execute with the (potentially higher) privileges of another Redis user. The Lua script execution environment in Redis provides some measures that prevent a script from creating side effects that persist and can affect the execution of the same, or different script, at a later time. Several weaknesses of these measures have been publicly known for a long time, but they had no security impact as the Redis security model did not endorse the concept of users or privileges. With the introduction of ACLs in Redis 6.0, these weaknesses can be exploited by a less privileged users to inject Lua code that will execute at a later time, when a privileged user executes a Lua script. The problem is fixed in Redis versions 7.0.0 and 6.2.7. An additional workaround to mitigate this problem without patching the redis-server executable, if Lua scripting is not being used, is to block access to `SCRIPT LOAD` and `EVAL` commands using ACL rules.
5	CVE-2022-24736	Redis is an in-memory database that persists on disk. Prior to versions 6.2.7 and 7.0.0, an attacker attempting to load a specially crafted Lua script can cause NULL pointer dereference which will result with a crash of the redis-server process. The problem is fixed in Redis versions 7.0.0 and 6.2.7. An additional workaround to mitigate this problem without patching the redis-server executable, if Lua scripting is not being used, is to block access to `SCRIPT LOAD` and `EVAL` commands using ACL rules.
6	CVE-2022-27406	FreeType commit 22a0cccb4d9d002f33c1ba7a4b36812c7d4f46b5 was discovered to contain a segmentation violation via the function FT_Request_Size.
7	CVE-2022-27404	FreeType commit 1e2eb65048f75c64b68708efed6ce904c31f3b2f was discovered to contain a heap buffer overflow via the function sfnt_init_face.
8	CVE-2022-27405	FreeType commit 53dfdcd8198d2b3201a23c4bad9190519ba918db was discovered to contain a segmentation violation via the function FNT_Size_Request.
9	CVE-2022-24765	Git for Windows is a fork of Git containing Windows-specific patches. This vulnerability affects users working on multi-user machines, where untrusted parties have write access to the same hard disk. Those untrusted parties could create the folder `C:\.git`, which would be picked up by Git operations run supposedly outside a repository while searching for a Git directory. Git would then respect any config in said Git directory. Git Bash users who set `GIT_PS1_SHOWDIRTYSTATE` are vulnerable as well. Users who installed posh-gitare vulnerable simply by starting a PowerShell. Users of IDEs such as Visual Studio are vulnerable: simply creating a new project would already read and respect the config specified in `C:\.git\config`. Users of the Microsoft fork of Git are vulnerable simply by starting a Git Bash. The problem has been patched in Git for Windows v2.35.2. Users unable to upgrade may create the folder `.git` on all drives where Git commands are run, and remove read/write access from those folders as a workaround. Alternatively, define or extend `GIT_CEILING_DIRECTORIES` to cover the _parent_ directory of the user profile, e.g. `C:\Users` if the user profile is located in `C:\Users\my-user-name`.