How I Accidentally Discovered CVE-2017-17087

Scott Court

October 5, 2018

Who am I?

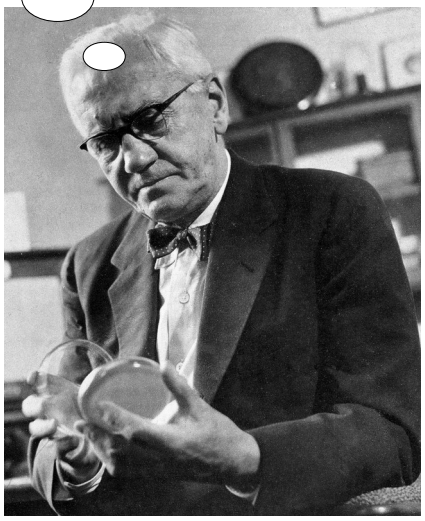
```
scott@Scott-Desktop:~$ ssh cucumberlinux.com

scott@cucumberlinux.com:~$ finger scott
Login: scott                            Name: Scott Court
Directory: /home/scott                  Shell: /bin/bash
On since Wed May 11 2016 10:00 (EST) on tty1

Major: CSEC
Year: Third year student
Preferred Scripting Language: Bash
Preferred Programming Language: C
Hobbies:
        Cucumber Linux
        No need for other hobbies, this takes up all my free time
```

How it Began – November 2, 2017

- I set out that morning to analyze CVE-2017-1000382, a different (but related) security vulnerability in Vim.

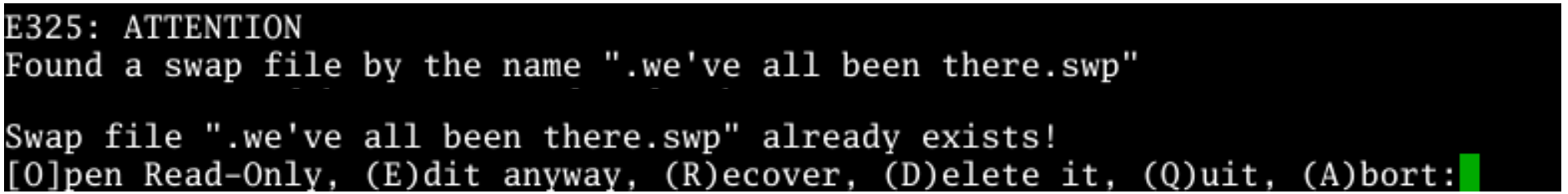
When I woke up just after dawn on November 2, 2017, I certainly didn't plan to discover my first CVE, ... But I guess that was exactly what I did.

- This vulnerability (discovered by Hanno Boeck of the Fuzzing Project) allowed for (among other things) a remote attacker to obtain Wordpress database credentials if a system administrator edited wp-config.php in Vim.



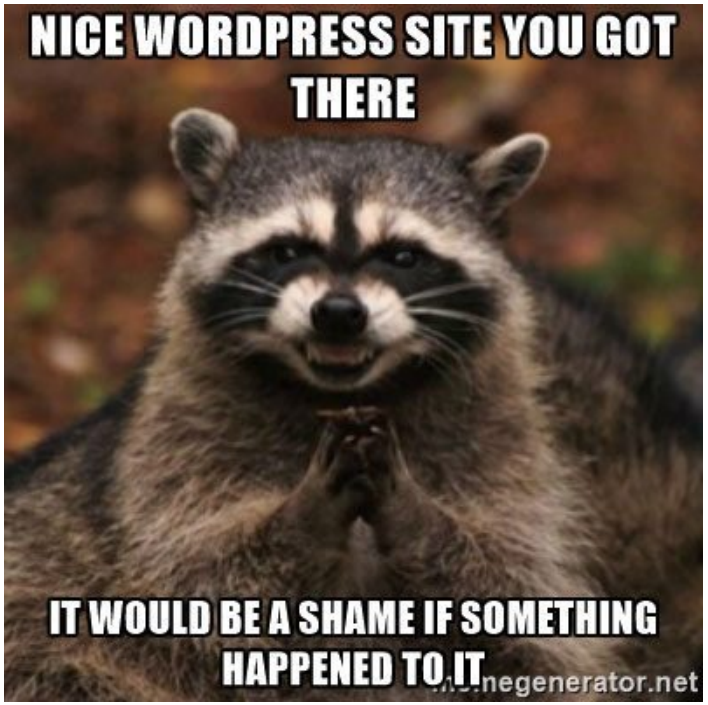
How did CVE-2017-1000382 Work?

- **It exploits the way Vim handles .swp files.**
  - Every time you edit a file in Vim, a corresponding .swp file is created (i.e. “foo” gets the swap file “.foo.swp”).
  - These files store the content of the file being edited, as well as some additional information.
  - This allows a user to recover the edits he was making to a file if Vim exits unexpectedly (i.e. if an SSH connection drops).
  - Once the user exits Vim properly, the .swp file is deleted.



How did CVE-2017-1000382 Work?

- **In Wordpress, all of the database credentials are stored in plain text in the file wp-config.php.**
  - This file is located in the public_html directory.
  - But it's a *.php script, so everything inside of the <?php ?> block gets parsed out before it is sent to the client so It's Secure™, right?
- **Now, the system administrator goes to edit the file in Vim.**
  - Vim creates the .wp-config.php.swp file.
  - But this is not a *.php script, so the <?php ?> tags don't get parsed out.
  - Now the database credentials are publicly visible at http://example.com/.wp-config.php.swp
  - To make matters worse, if the Vim session crashes, this file sticks around for a while.



Proof of Concept

But Something Else was Wrong…

- **The description for CVE-2017-1000382 in the MITRE CVE Dictionary was:**
  - "VIM version 8.0.1187 (and other versions most likely) ignores umask when creating a swap file…"
- **It occurred to me that the issue described there had absolutely nothing to do with the vulnerability.**
  - The original file (wp-config.php) was accessible by httpd already, so even if the .wp-config.php.swp file had exactly the same permissions as the original file, this would have still been an issue.
  - Also, it seemed like the umask should be ignored, as honoring the umask would result in creating .swp files that are more accessible than the original file in most cases anyway.
  - So I decided to run a `ls -la` on my public_html directory to investigate further, and this is what I found:

```
root@Scott-BackupLaptop:html# ll -a | grep config.php
-rw-r-----  1 webmaster users     4096 Oct  4 16:24 .wp-config.php.swp
-rw-rw----  1 apache    apache    2848 Oct  4 15:54 wp-config.php
```

This was a Problem

- **This alarmed me, as the .wp-config.php.swp file was now readable to members of the users group, whereas the original file was not accessible to the users group.**
  - Every user on the system is a member of this group by default, so now the credentials were also being disclosed locally.
- **At first, I was unsure if this was a distinct vulnerability from CVE-2017-1000382, so I phoned a friend to discuss it further.**
  - After discussing the details at length, we both agreed that it was a separate vulnerability since it was possible to fix one without fixing the other.

So I'd Discovered a Vulnerability

- **I had never discovered a vulnerability before, so I was slightly unsure of how to proceed:**
  - Should I disclose it immediately or try to come up with a fix first?
  - How should I disclose it (full disclosure or coordinated disclosure)?

- **Fortunately, I was able to come up with a fix for both my vulnerability and CVE-2017-1000382 very quickly:**
  - Each user got a dedicated swap file directory: ~/.vim/swap
  - This directory was chmod'ed 700 so that only that user had access to its contents.
  - Vim was reconfigured to store the swap files there instead of the directory containing the original file.

Proof of Concept #2

Time for Disclosure

- **Since I had a fix ready and the Vim developers didn't seem terribly concerned about CVE-2017-1000382, I opted to full disclose both the vulnerability and my proposed fix right away.**
  - The disclosure was posted to the Vim development mailing list as well as the oss-security list.
- **Then I requested a CVE ID and was assigned CVE-2017-17087 by MITRE.**

Response from Bram Moolenaar (the Vim BDFL)

- **Bram did not view CVE-2017-1000382 as a problem with Vim, saying:**
  - "Why would a web server expose and serve such a file? That clearly is the problem, not that Vim happens to create swap files."
  - He further claimed that this is something system administrators should be aware of and know to protect against on their own.
- **After I disclosed CVE-2017-17087, Bram didn't initially view that as a problem either, stating:**
  - "Why is [the user's] primary group one that all users on the system are a member of? That is asking for trouble."
  - Fortunately, there was enough backlash over this comment that he was finally forced to do something about it.

The “Solution”

- **Bram did not want to adopt my solution for CVE-2017-17087, claiming it was “convenient” to have the .swp files stored in the same directory for a couple of reasons:**
  - This allowed a user to detect if another user was already editing the file.
    - This is not a reliable way to determine if the file is being edited.
  - This allowed for a user to recover another user’s session if it crashed.
    - This is a bad idea anyway.
  - If a file was being edited on an external drive, having the .swp file there allowed the session to be recovered if the drive was plugged into another system.
- **Instead, Bram opted to chmod the .swp file so that the group access was the same as the other access.**
  - This did not fix CVE-2017-1000382.
  - It also still allowed for exploitation of CVE-2017-17087 in the case of group blacklisting.

The Status Now

- **CVE-2017-17087 was fixed upstream using Bram's solution in Vim 8.0.1263.**
  - On Cucumber Linux, we continue to use my solution.
  - This fix was not backported by any major distributions, so most stable distributions remain vulnerable.
- **CVE-2017-1000382 was never fixed upstream, and as such it remains exploitable on every Linux distribution except Cucumber Linux.**

What I Learned

- **Bram Moolenaar is particularly unpleasant and difficult to work with.**
  - He doesn't seem to care about security either.
  - This is a sentiment shared many.
- **Neovim had fixed both these vulnerabilities several years ago using an approach very similar to mine.**
  - Neovim also supports a lot of cool features that Vim does not.
  - In other words: Neovim > Vim

- **If you discover a CVE, every employer you ever talk to will want to hear all about it, so make sure you have a good story to tell.**

Further Reading

- https://security.cucumberlinux.com/security/details.php?id=166
- https://security.cucumberlinux.com/security/details.php?id=120
- https://groups.google.com/forum/#!msg/vim_dev/sRT9BtjLWMk/BRtSXNU4BwAJ
- https://www.openwall.com/lists/oss-security/2017/11/27/2

Questions?

Email: scott@cucumberlinux.com

This presentation can be found online at
https://cucumberlinux.com/~scott/presentations/