**To:** William H. Flathead III

**From:** Claudio Facchinetti

**Subject:** Security issue: Buffer overflow vulnerability

**Date:** October 13, 2021

---

The objective of this memo is bring to your attention a security issue that has been discovered in webserver software.

As discussed in previous meetings it looks like someone gained access to some files which were only accessible by the root users; it also looks like the web server started to send a lot of spam.

It turns out that the suspects were true: the server has a buffer overflow vulnerability which turns out to be the cause of such events.

From further investigations it turned out that the webserver actually had two buffer overflow vulnerability, one at line `88` and one at line `313`.

Both the vulnerabilities occur because strings are managed without taking into consideration the sizes; in particular the first vulnerability does occur because it is assumed that the total size of an HTTP header would be less than 1024 characters, which is not necessarily true. The second vulnerability does occur because in generating the response it is not checked whether the final size of the message is within boundaries. This is actually a problem because it directly appends the requested URI, which is under the attacker's control.

This vulnerabilities might simply make the webserver crash if they are flooded with a large enough payload, however there is a worst case scenario which is exactly what we are observing. Making a long story short, with a specially crafted header value I managed to exploit the first vulnerability discussed to make the webserver expose a shell on a partiular port to which it is possible to connect to from anywhere.

The worst thing about this is that this shell allows an attacker to run commands as root: for example it is possible to create a new user with root privileges in order to gain a persistent access to the system. This is probably how the attacker managed to gain access to reserved documents and made the webserver send spam.

I have developed a patch for both vulnerabilities and this patch is already live. The first vulnerability has been patched by limiting the size of an HTTP header value to a maximum value of 1KB: this limit can be easily increased, but 1KB should be more than enough for our applications.

The second vulnerability has been patched in a more radical way: the result is no longer stored before it is sent but it is rather immediately sent to the client, so that there are no more memory issues.

Moving to the suggestions to take the system back to controlled state I would suggest to perform a check on the accounts that are present on the webserver machine and change all the passwords, including the one of the root user. By performing a reboot all the sessions would get terminated, requiring a login with new credentials that would be given only to authorised people.

While this blocks the possibility of existing compromised accounts being used, this does not protect against new attacks: I suggest to change the way the webserver is compiled in order to enable all the `gcc` flags which allow to have built-in protection against such attacks. It would be advisable that the webserver would not run as root user.

As a last thing I would suggest to check for any daemon that starts at boot-time, in order to stop also the spam activity. A fresh installation of the operating system would also be advisable, but the service to not be available for an uncertain amount of time.

It is not possible to estimate the amount of money lost because we do not know exactly what has been stolen or leaked; it is however possible to state that every possible file present in the webserver machine could have been copied and / or shared. As a final suggestion I would consider adding a logging solution, so that at least we can have a rough estimate in case something similar to this would happen again.