To: Frobozz CEO
From: Vinci Nicolò
Subject: Buffer Overflow Vulnerabilities
Date: 15/10/2021

Security Flaws

There are two security flaws regard buffer overflow in webserver.c:

- The first is at line 313 where the string `path` is appended to the string `sendmessage`. However, `sendmessage` is a fixed buffer of char and the `path` string can be manipulated by the user through the URL of the request. So, a malicious user can craft a very long request URL to overflow the `sendmessage` buffer. For example, a request composed of 2000 characters will overflow the buffer.
- The second is in the function `*get_header`. A buffer called `hdrval` is instantiated at line 87 with a fixed size of 1024. Then, the buffer is filled with the content pointed by `hdrptr`. A malicious user can manipulate the pointer `hdrptr`, because it actually points to the value of a request header. For example, if a user performs a GET request, he can craft a very long field for the header `If-Modified-Since` to overflow the buffer.

The two flaws have been fixed computing dynamically the size of the two vulnerable buffers, checking the input provided by the user request.

A script `exploit.sh` has been developed to automatically exploit the vulnerabilities. The port and number of chars can be set dynamically when the script is launched. Also, you can choose whether to overflow the buffer through the request or the header.

Recovery plan:

1. The breach was really serious, because an attacker could be able to cause a Denial of Service of the webserver.c overflowing one of the two buffers. More importantly, an attacker could manipulate the stack in order to gain access directly to the server.
2. It should be enough to apply the proposal fixes and restart the webserver.c. However, if an attacker has already gained the access directly to the server, he may be able to do whatever he wants. For example, he may have disrupted any service inside the server or he may have stolen any sensible data. In this case, an estimate of the damage should be performed and then a custom recovery plan can be applied.
3. The webserver.c also presents other problems. For example, if a user performs a GET request with the URL `http://`, the web server will crash. It evaluates wrongly the request in the if statement at line 226: `(path = strchr(path + 7, '/'))`.