To: William H. Flathead III

From: Matteo Franzil and Claudio Facchinetti

Subject: Snort explanation

Date: November 25, 2021

We am writing this memo to explain the last of our testbed projects here at Frobozz. This time, we chose to experiment a bit with intusion prevention systems and created a test network with Snort installed. In case you didn't know, Snort is an open-source software that can either sniff the traffic (*IDS*) or sit in between and act prompty (*IPS*). At the core it allows the creation of *rules*, from very basic to elaborate, but it is a very sophisticated piece of software that allows basically any alteration and verification of incoming and outgoing traffic.

In the first part we managed to setup the simple network and started to look at the various packets flowing and noticed that a lot of unsecured packets were flowing through the network without any kind of control. We noticed that the `FileClient` applications were sending requests for files using a specified syntax and we tried the effectiveness of Snort simply making it log any request of an xml file. We also noticed that some files were not intended to be made available outside of a specific network, therefore we also created a Snort rule to prevent this behaviour.

After this we simulated some possible UDP flooding attack using the same tool presented in previous memos: while this was in place it was nearly impossible for any client to access the files on the `FileServer`. Again we mitigated this with a Snort rule which matched traffic and limited the amount of packets sent to the server if it realises that something suspicious is going on: with this in place the clients manage to reach the server in a reasonable time. The delay is due to the fact that the links are completely flooded. After this we also noticed that some of the machines were able to bypass Snort completely: we fixed it and also proposed some enhancements that could be done to enforce the overall security of the communications. With this we also pointed out that the Snort configuration should be adapted to the security improvements done.

As a last thing we discovered the source of the vulnerability in `FileServer` application which causes a remote attacker to gain remote code execution. In order to prevent this we firstly identified the conditions under which this gets exploited and then created a Snort rule to defend against this: be careful when assigning usernames, passwords, and file names since any string which matches those conditions gets dropped by Snort. After patching this we also discovrered that the rule could be bypassed using the "ASCII encoding" feature of the `FileServer`; to prevent this from happening we loaded a preprocessor on Snort to decode ASCII characters before applying any rule.

At the current time it is impossible for us to say if there has been any kind of violation of the network since there is no packet filtering / logging at all but it would be naive to believe that none of the vulnerabilities / security issues has been exploited. In order to make this change and have in future the possibility to at least understand how some vulnerability has been exploited or simply an attacked performed our suggestion is to setup an NIDS as soon as possible so that critical services can be protected with specialized rules. This does not only allow for analysis after the attack has been performed but also offers a rather easy way to understand that something is wrong and react immediately.