

ENTRY // VStarCam Security Research #7 // Firmware Analysis

[\[- \]](#) [\[X \]](#)

Date: 2019-09-03 Author: Ian Rash Index: #7

Now that we are at the point of near 100% protocol coverage, we can start to think about some ways that we could potentially abuse the protocol, and the devices behind them. One thing I noticed after completely tearing down every packet in the connection process, was that all the information needed to impersonate the camera is sent to broadcast. This means that even when connected directly through LAN, the camera could potentially be impersonated by anyone on the same subnet. A couple things also hint at this.

1. The IP address of the camera can change, and the client must be able to respond to this change. * This means that chances are the protocol checks IP address very weakly, and might be vulnerable to someone using a different IP address. 2. The device itself is not very powerful, so corners were most likely cut in software to save money/processing power. It's likely that certain kinds of checks were overlooked, especially when security was obviously not the goal in mind with the camera. 3. The device may or may not parse all fields in the packets (DBR, BPS, BPA) so it might be easy for us to replay packets without having to know exactly what a field does.

Flaws in the camera programming, as well as the client protocol, also give out too much information. The camera (for seemingly no reason) sends the DBR, which contains all the info we need to impersonate, to broadcast. The client also initiates connection over broadcast (although later switching to direct communications later). Looking at these things, there is a very real possibility of a vulnerability here. We may need to swap fields out of the DBR, or change our MAC address to conform to the clients checks, but it should be possible.

TAGGED SKILLS:

[Hardware Security](#) [Reverse Engineering](#) [Linux Kernel](#)