Webcam Fuzz Testing: Testing IoT Deployments

Matthew Elbert
University of Utah

Jeffrey Kitchen
University of Utah

Abstract

Your Abstract Text Goes Here. Just a few facts. Whet our appetites.

1 Introduction

1.1 Motivation

Fuzz testing is a method of testing in which random, potentially invalid inputs are sent to a program or service in order to test for vulnerabilities. It is often used as either a gray or black-box testing platform for security testing. Fuzz testing can be a very cheap and effective implementation of testing as many inputs can be generated, sent, and analyzed automatically, without the need for a human to monitor. Because the world has become more internet-connected than ever, fuzz testing can be an important tool for developers as network-facing interfaces can experience any input, and need to be tested for this randomness. However, many functions are very state-driven, and a true, random test may only test a single interface, and not the whole system. Because of this, many of these fuzz testers must be stateful in order to do a complete test. But because a truly random input would take infinite time to find all possible inputs, a mutation of correct inputs approach is often implemented.

1.2 Problem

For this project, we were tasked with designing, implementing and evaluating an automated tester for a networking service. This is meant to evaluate the value of fuzz testing on a network, with the goal of hopefully finding a previously undiscovered bug. A very basic and highly proliferated technology deployed on the internet is an HTTP server. But because of this prevalence, popular servers such as Apache have been thoroughly tested and documented. However, more and more devices are being connected, with an uptake in the Internet of Things (IoT) mentality. Therefore, we chose wireless IP cameras as the platform to fuzz test. These are some of the most widely available IoT devices for consumers. We found that the two devices we purchased ran basic HTTP servers, and we could use tools made for these applications.

2 Related Work

We looked at papers like SecFuzz [3] and SNOOZE [1] as well as tools like AutoFuzz [2]

3 Proposed Solution

Use solutions for HTTP fuzzing, e.g. Pathoc, to fuzz the HTTP servers running on webcams.

4 Implementation

We used Pathoc for crafted malice towards the mini_httpd and DLink servers.

5 Results

References

[1] BANKS, G., COVA, M., FELMETSGER, V., ALMEROTH, K., KEMMERER, R., AND VIGNA, G. Snooze: Toward a stateful network protocol fuzzer. In *of Lecture Notes in Computer Science* (2006), Springer, pp. 343–358.

[2] GORBUNOV, S., AND ROSENBLOOM, A. Autofuzz: Automated network protocol fuzzing framework. *IJCSNS 10*, 8 (2010), 239.

[3] TSANKOV, P., DASHTI, M. T., AND BASIN, D. Secfuzz: Fuzz-testing security protocols. In *Proceedings of the 7th International Workshop on Automation of Software Test* (Piscataway, NJ, USA, 2012), AST '12, IEEE Press, pp. 1–7.

Notes

[1] Remember to use endnotes, not footnotes!