Remote Video Eavesdropping Using a Software Defined Radio Platform

Martin Marinov
St Edmund's College

A dissertation submitted to the University of Cambridge in partial fulfilment of the requirements for the degree of Master of Philosophy in Advanced Computer Science

University of Cambridge
Computer Laboratory
William Gates Building
15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom

Email: mtm46@cam.ac.uk

May 17, 2014

Declaration

I Martin Marinov of St Edmund's College, being a candidate for the M.Phil in Advanced Computer Science, hereby declare that this report and the work described in it are my own work, unaided except as may be specified below, and that the report does not contain material that has already been used to any substantial extent for a comparable purpose.

Total word count: ??,???

Signed:

Date:

Abstract

This dissertation presents a software toolkit for remotely eavesdropping video monitors using a Software Defined Radio (SDR) receiver. It exploits compromising emanations from cables carrying video signals. Analogue video is usually transmitted one line of pixels at a time encoded as a varying current. This generates a wideband electromagnetic wave that can be picked up by an SDR receiver. The presented software can map the received field strength of each pixel to a grayscale value in order to show a real-time false colour estimate of the original video signal.

The software significantly lowers the costs required for undertaking a practical attack compared to existing solutions. Furthermore, it allows for an additional digital post-processing which can aid in analysing and improving the results. It also provides mobility for a potential adversary, requiring only a commodity laptop and an USB SDR dongle. The attacker does not need to have any prior knowledge about the victim's video display. All parameters such as resolution and refresh rate can be estimated with the aid of the software.

The software comprises of a library written in C, a collection of plug-ins for various Software Define Radio (SDR) frontends and a Java based Graphical User Interface (GUI). It is designed to be a multi-platform application. All native libraries can be pre-compiled and packed into a single Java jar file which allows the toolkit to run on any supported operating system.

This report documents the digital processing techniques that have been employed in order to extract, detect and lock to a video signal. It also explains the architecture of the software system and the techniques used in order to achieve low latency and real-time interactivity. It demonstrates the usage of the system by performing a practical attack. It then gives some ideas about what could be improved further and some analysis of data that was collected during the development of the software.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Unintended

Chapter 2

Background

[A more extensive coverage of what's required to understand your work. In general you should assume the reader has a good undergraduate degree in computer science, but is not necessarily an expert in the particular area you've been working on. Hence this chapter may need to summarize some "text book" material.

This is not something you'd normally require in an academic paper, and it may not be appropriate for your particular circumstances. Indeed, in some cases it's possible to cover all of the "background" material either in the introduction or at appropriate places in the rest of the dissertation. ok]

Chapter 3

Related Work

[This chapter covers relevant (and typically, recent) research which you build upon (or improve upon). There are two complementary goals for this chapter:

1. to show that you know and understand the state of the art; and
2. to put your work in context

Ideally you can tackle both together by providing a critique of related work, and describing what is insufficient (and how you do better!)

The related work chapter should usually come either near the front or near the back of the dissertation. The advantage of the former is that you get to build the argument for why your work is important before presenting your solution(s) in later chapters; the advantage of the latter is that don't have to forward reference to your solution too much. The correct choice will depend on what you're writing up, and your own personal preference.]

[1]

Chapter 4

Architecture

4.1 Overview

4.2 Main Library

4.3 Plug-ins

4.4 JavaGUI

4.4.1 Multiplatform

4.4.2 Graphical Interface

Chapter 5

Digital Signal Processing

5.1 Overview

5.2 Multithreading

5.3 Demodulation

5.4 Re-sampling

5.5 Auto Gain

5.6 Low-pass

5.7 Frame Synchronization

5.8 Autocorrelation

5.9 Visualisation

5.10 Extended Bandwidth

Chapter 6

Practical attack

[For any practical projects, you should almost certainly have some kind of evaluation, and it's often useful to separate this out into its own chapter.]

Chapter 7

Summary and Conclusions

[As you might imagine: summarizes the dissertation, and draws any conclusions. Depending on the length of your work, and how well you write, you may not need a summary here.

You will generally want to draw some conclusions, and point to potential future work.]

Bibliography

[1] Markus G Kuhn. Compromising emanations: eavesdropping risks of computer displays. *University of Cambridge Computer Laboratory, Technical Report, UCAM-CL-TR-577*, 2003.