© Weiss

“Jitterbug 2.0”

– Stealthy Real-Time Keyloggers –

SPRING 5: SIDAR Graduierten-Workshop über Reaktive Sicherheit

Benjamin Michéle, 7. July 2010, Bonn, Germany
ben@sec.t-labs.tu-berlin.de

Agenda

- Introduction
- Compromising keyboards
- Jitterbug
- Motivation
- Current status

Introduction

- Ring 0, -1, -2, (-3) root kits
  - Run on platform
- Malicious peripherals?
  - Keyboard, firewire devices, hard disks

Compromising Keyboards

- K. Chen owns Apple keyboard @BH2009 [1]
- Focus on hack
- Problem of data retrieval
  - Need for physical access undesirable
  - Mentions Blaze et al.'s Jitterbug paper (next slide)
  - Not tested/implemented

- [1] http://www.blackhat.com/html/bh-usa-09/bh-usa-09-archives.html#Chen

Jitterbug

- “Keyboards and Covert Channels” by Blaze et al. [2]
- Add delays between keystrokes to encode information over interactive connections like ssh
- Extra hardware between keyboard and PS/2 port
  - Stores interesting keys like passwords
  - Exports one bit at a time with each new keystroke
- Bit is encoded by time between keystrokes δ_i
  - $\delta_i \bmod w = 0 \quad \rightarrow \text{bit}=0$
  - $\delta_i \bmod w = w/2 \quad \rightarrow \text{bit}=1$

- [2] http://www.usenix.org/events/sec06/tech/shah.html

Motivation

- Needed a project for students :)
- Combination of both papers feasible?
- Go one step further
  - Real-time keylogger
  - Encode each keystroke in timing delay
  - Eavesdrop on chat conversations, etc.
    - No physical access needed
    - Not detectable
    - Persistent root kit

Current status

- Flashing works, own tool for linux
- Key logging works

- Open problems
  - Jitterbug data export highly error prone
  - Many error sources → timers, etc.
  - Driver polls keyboard in fixed (big) intervals
    - Missing synchronization
    - Too slow for data rates > 1bit/keystroke

Questions?

Thank you!