UNIVERSITY OF AMSTERDAM

SDIO: a new peripheral attack vector

Research Project 2

Thom Does
thom.does@os3.nl

Dana Geist
dana.geist@os3.nl

Introduction

- Secure Digital Input Output (SDIO)
  - Adds I/O functions to SD
  - PDAs, tablets, laptops
- SDIO presents similarities with USB
- BadUSB attack (2014)
  - Inject keystrokes
  - Rogue DHCP
- Seemingly no protections to prevent BadUSB-like attacks

Research Question

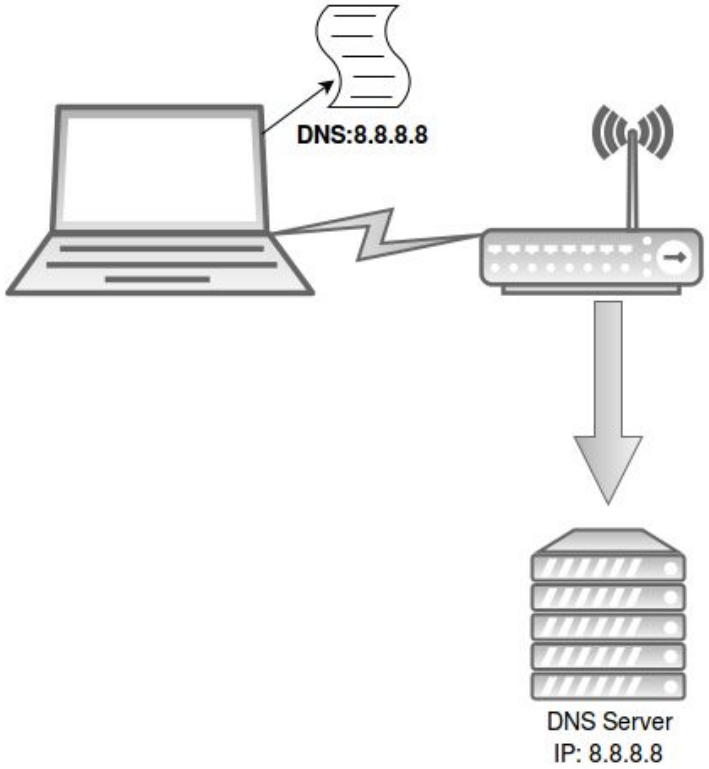
Could SDIO be used as a new attack vector on SDIO-aware hosts?

State of art

- No previous research on SDIO as an attack vector
- SD/SDIO specifications
  - Only simplified version available without license
- SD card hack (2013)
  - Several microSD cards were tested
  - Reversed engineered firmware
  - Developed novel applications for microcontroller

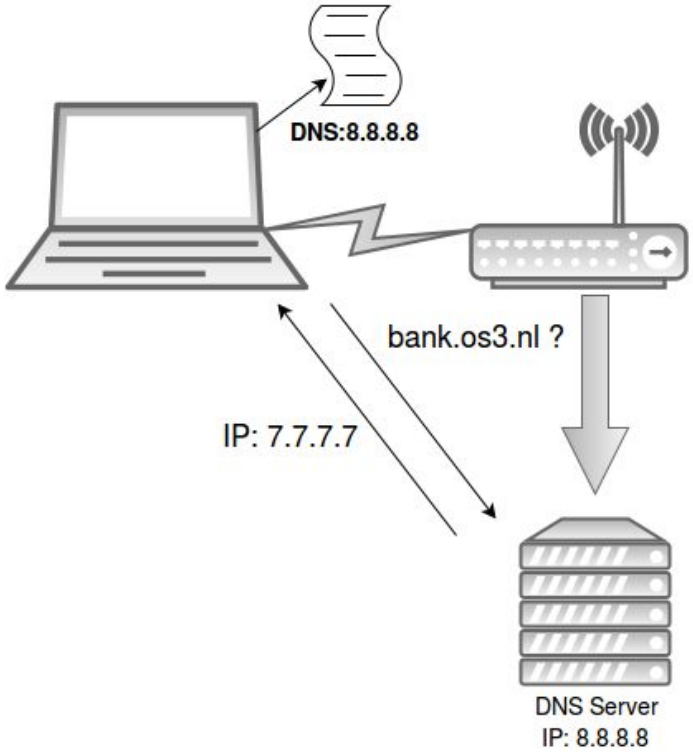
Attack path: WLAN SDIO card

Step 1



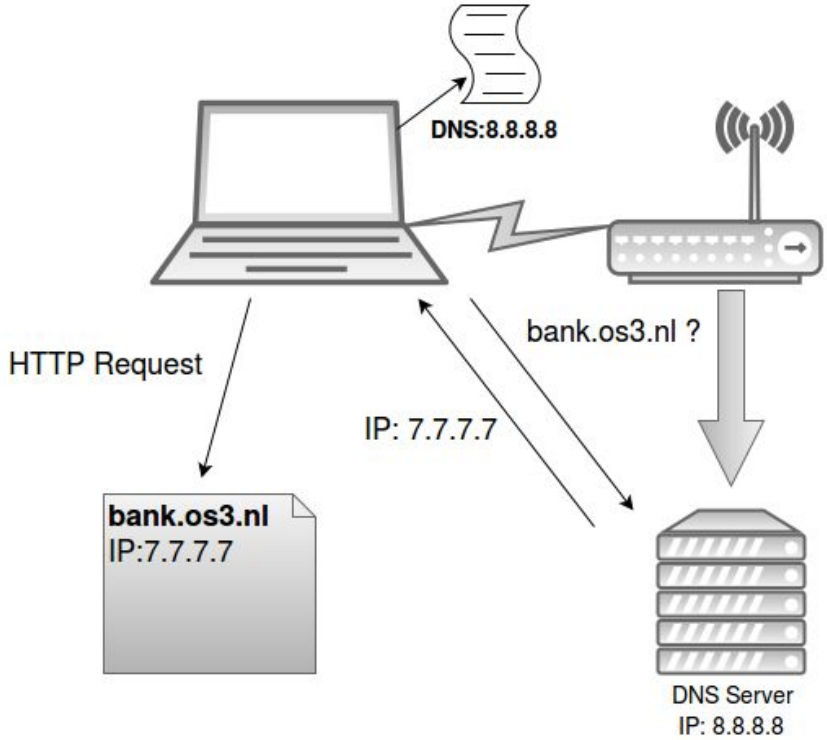
Attack path: WLAN SDIO card

Step 1



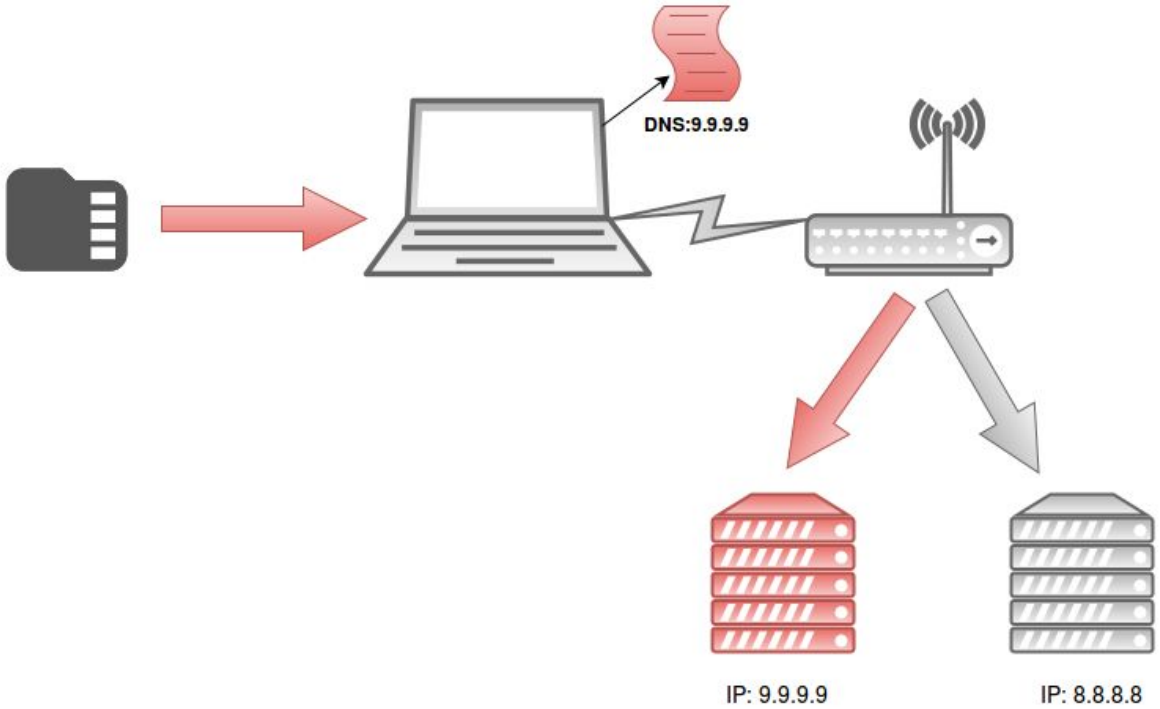
Attack path: WLAN SDIO card

Step 1



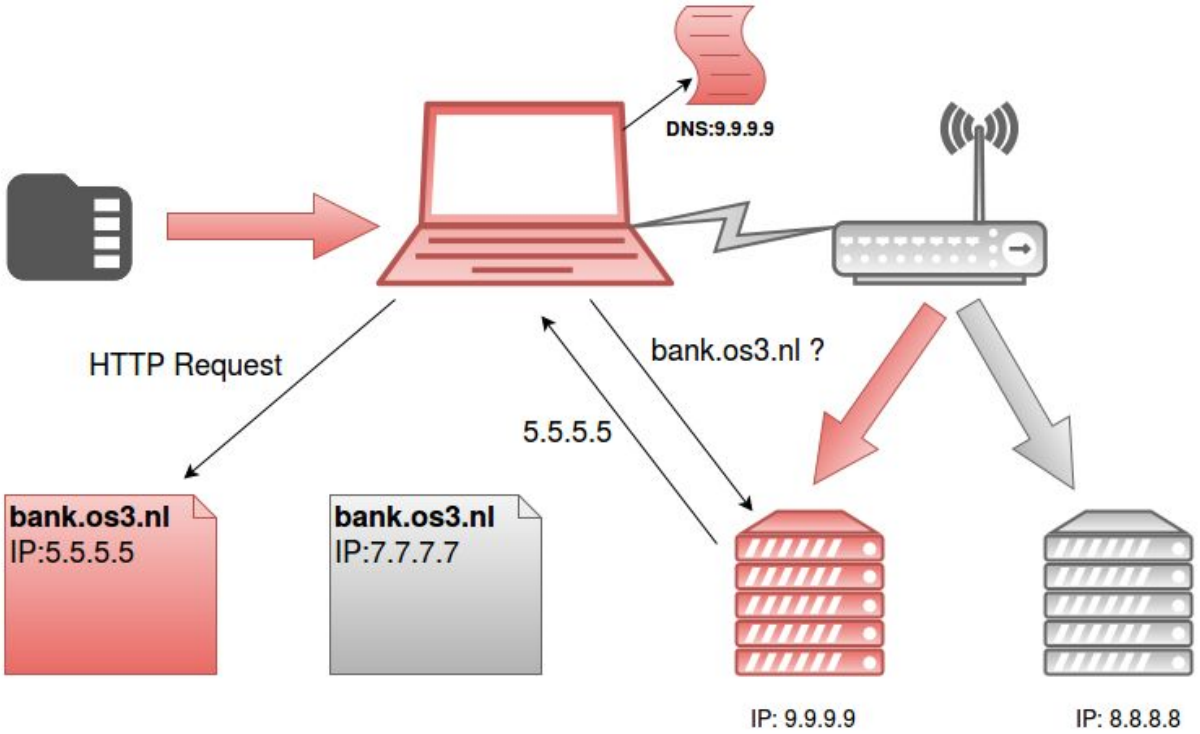
Attack path: WLAN SDIO card

Step 2

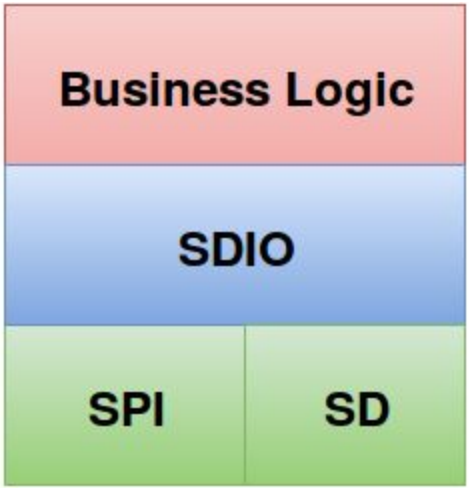


Attack path: WLAN SDIO card

Step 3

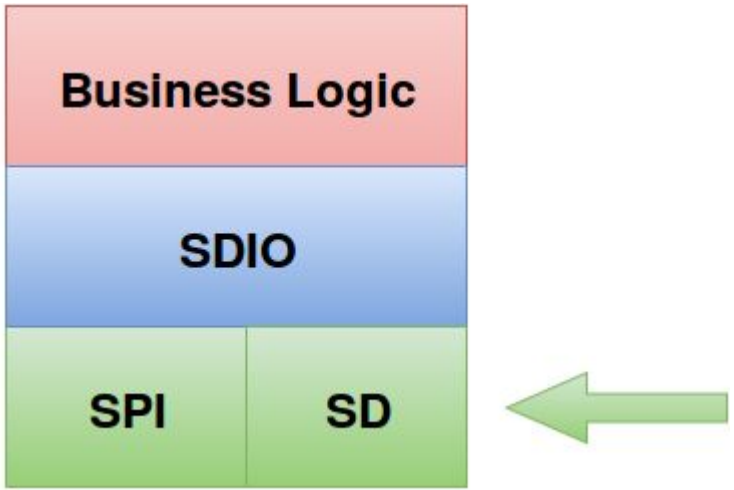


SDIO Stack



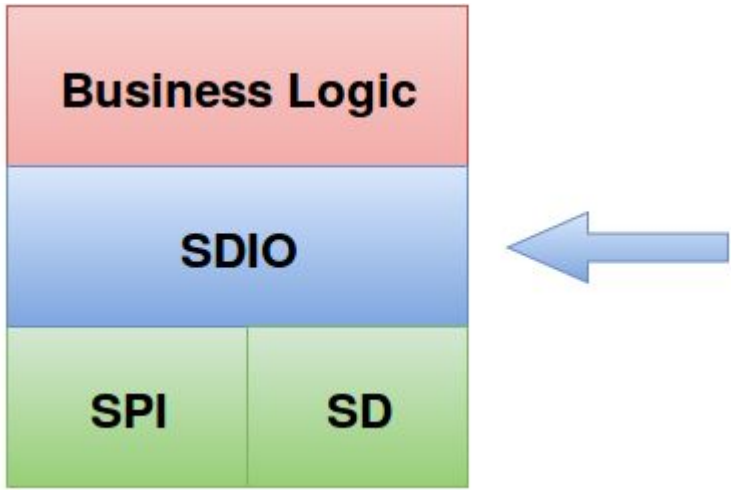
Physical Layer: SPI vs. SD

SPI	SD
Wide variety of applications	Used by SD cards and readers
Well-known “open” protocol	License required
Simple (one data line)	More complex (commands, data lines)
Supported natively by many MCUs	Special purpose MCUs or bitbanging
Fallback protocol for SD	Default for SD



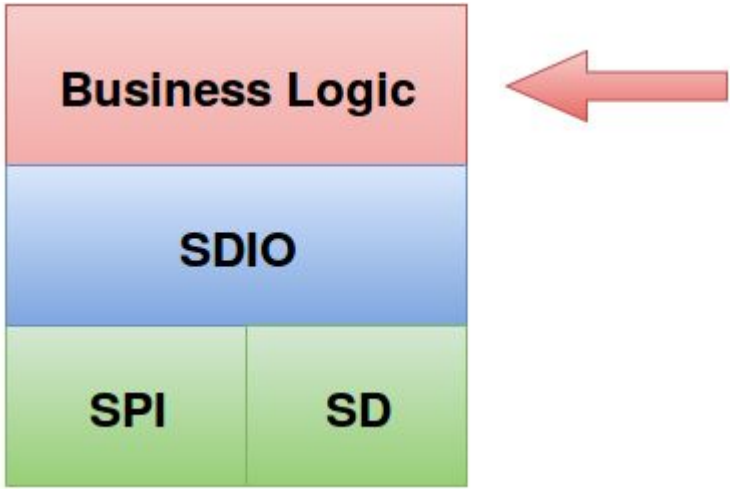
SDIO Layer

- Maintained by SD association
- Documentation requires licensing
- Defines SDIO commands
  - Formats
  - Initialization
  - Transfer modes
- Master-Slave based protocol
  - The card reader is the Master
  - The SDIO card is the slave

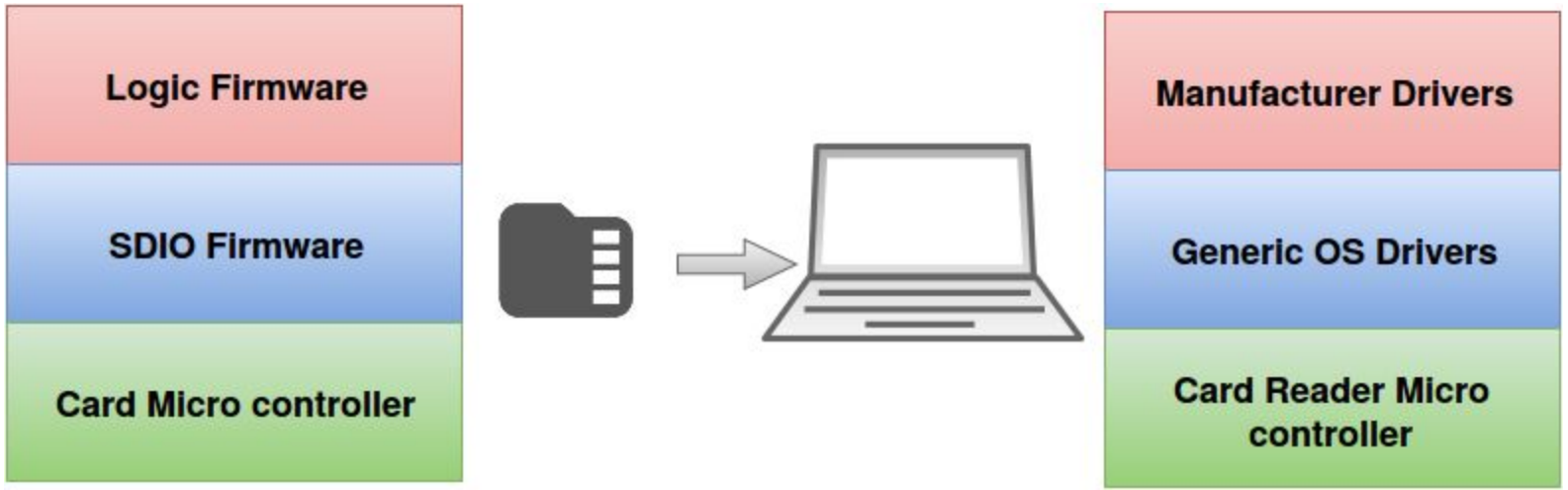


Business Logic Layer

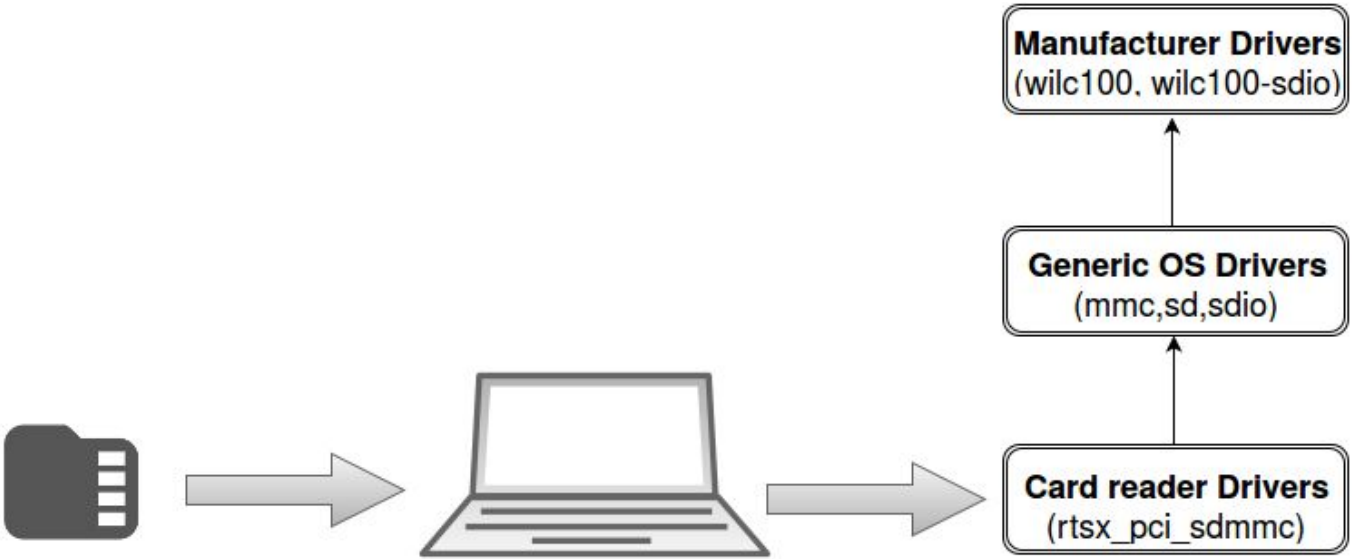
- Multiple manufacturers
- Standardized and manufacturer specific interfaces
  - Firmware
  - Drivers
- Each interface is an attack surface
  - WLAN, bluetooth, GPS
- Manipulate higher-level applications
  - DHCP-client, command injection, navigation system



SDIO Model


Logic Firmware
SDIO Firmware
Card Micro controller
Manufacturer Drivers
Generic OS Drivers
Card Reader Micro controller

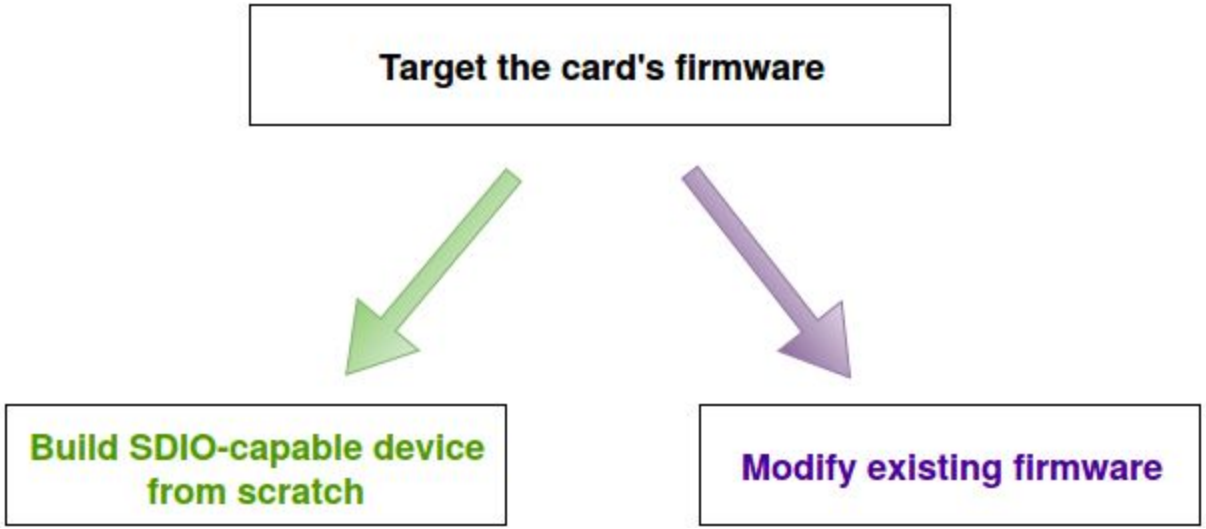
SDIO Model: Host’s drivers



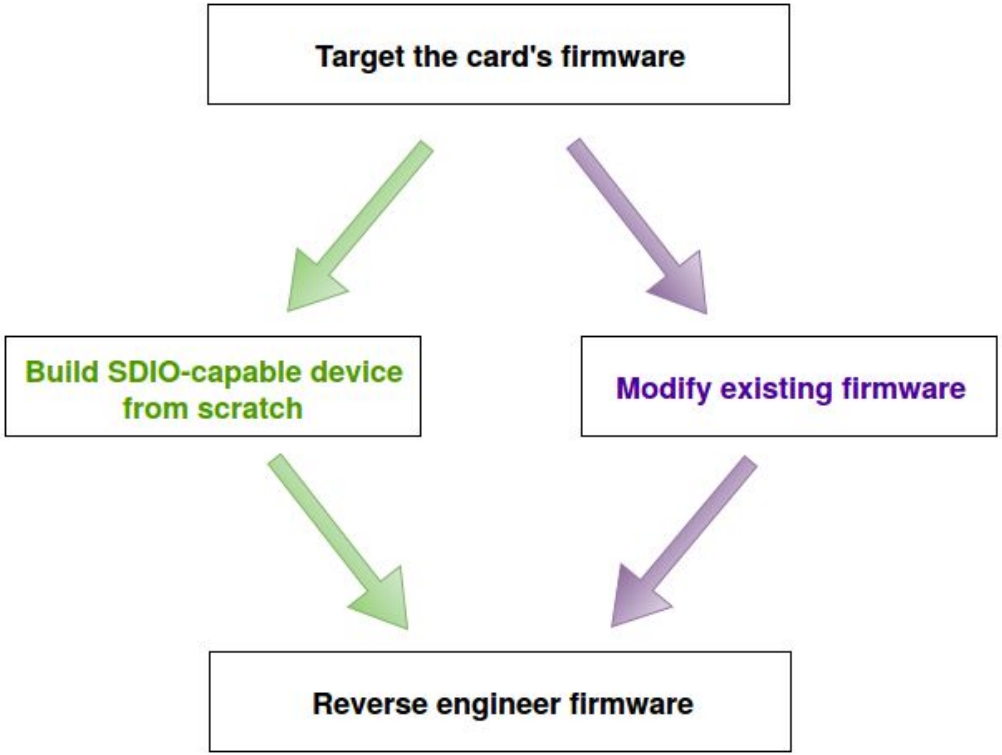
How can the host system be exploited?

Target the card's firmware

How can the host system be exploited?



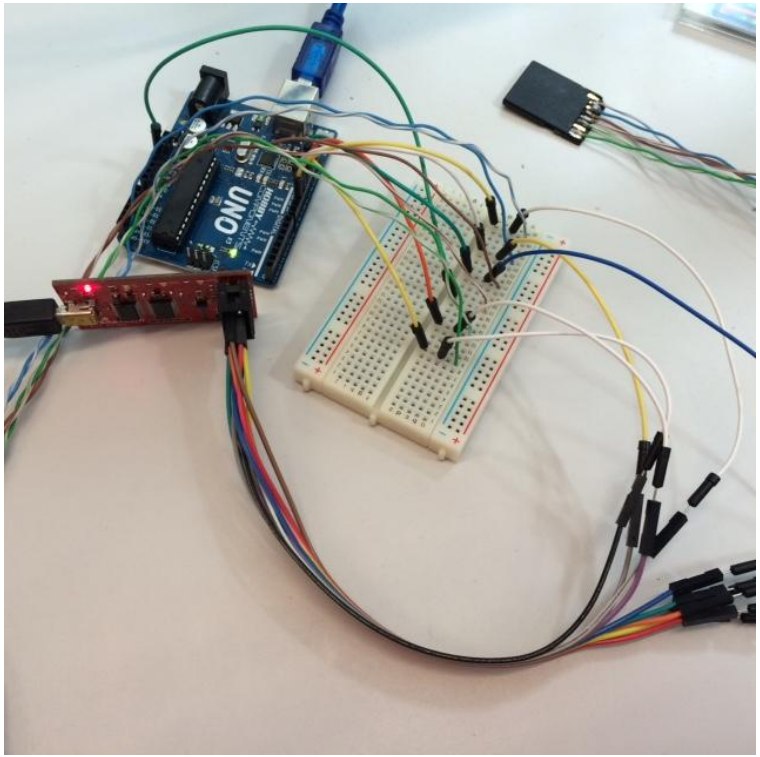
How can the host system be exploited?



Build SDIO device from scratch (SPI)

If host supports SPI

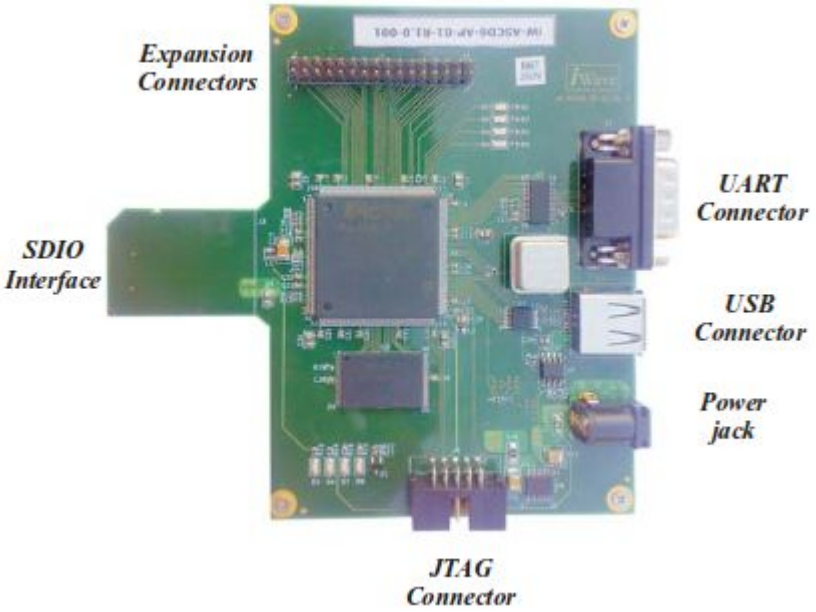
- Use low cost microcontrollers to implement protocol
- Build low cost sniffers to ease the development
- Use open source software to analyze the protocol
- Not all hosts support SPI



Build SDIO device from scratch (SD)

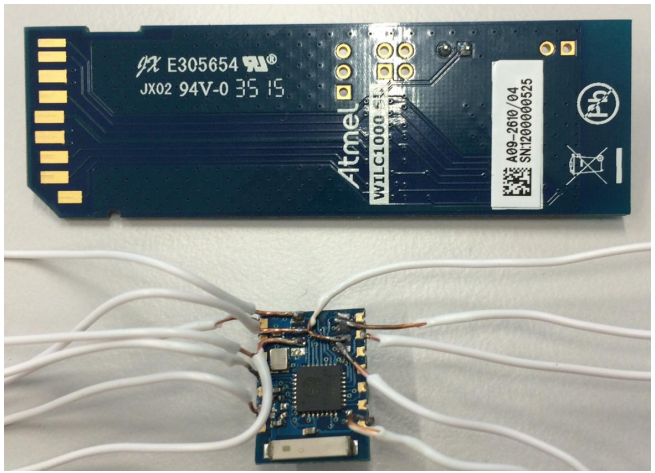
If host supports SD only

- Most microcontrollers do not natively support the protocol
- Using commodity hardware for bitbanging could be cumbersome
- No open source protocol analyzers tools
- Complex solutions like FPGA + IP core software
  - Expensive
  - Steep learning curve
  - Requires business logic programming



Modify existing firmware

- Get the firmware
- Find hooking points
- Rewrite specific functions
- Two main options:
  - Firmware embedded in SD card
  - Firmware loaded to device by the driver



SDIO-based vs. USB-based attacks

	SDIO attack	BadUSB
Hosts	Laptops, tablets, PDAs	Desktops, laptops, printers, routers
Devices	Limited vendors and applications	Many vendors and applications
Stealthiness	Embedded in port	Protruding from port
Ease of exploitation	No “off-the-shelf” products	USB Armory, Rubber Ducky, known vulnerable firmware

Discussion

- Licensing required by SD Association
- Attack feasible but:
  - Time consuming
  - Expensive
  - Not possible to create general purpose malicious firmware
- Likelihood
  - Affects SDIO aware hosts only
  - Kernel module needs to be loaded
- Impact:
  - Wide range of attacks possible
- Mitigation: vendors should sign or encrypt their firmware

Conclusions

- SDIO cards are supported by various types of hosts
  - Laptops, phones, tablets, PDAs
- SDIO is an attack vector
  - No protections found
- Firmware might be modified, or developed from scratch
  - SD is more effective than SPI
- Currently, SDIO-based attacks seem less likely than USB-based attacks
  - Ease of exploitation
  - Number of vendors / products supporting SDIO
  - Kernel module needs to be loaded

References

Research material:

- SDIO specifications
  - https://www.sdcard.org/downloads/pls/index.html
- BADUSB - On Accessories that Turn Evil by Karsten Nohl + jakob Lell
  - https://www.youtube.com/watch?v=nuruzFqMgIw
- SD card hack
  - http://hackaday.com/2013/12/29/hacking-sd-card-flash-memory-controllers/
- A Microcontroller-based HF-RFID Reader Implementation for the SD-Slot
  - http://www.thinkmind.org/download.php?articleid=icds_2011_4_30_10048

Images:

- https://www.parallella.org/create-sdcard/
- http://www.techrific.com.au/2005/06/wifi-sd-card-spectec-in-stock.html
- http://www.actel.com/ipdocs/iW-SDIO_Slave_demo_board_DS.pdf
- https://www.sdcard.org/developers/overview/sdio/index.html