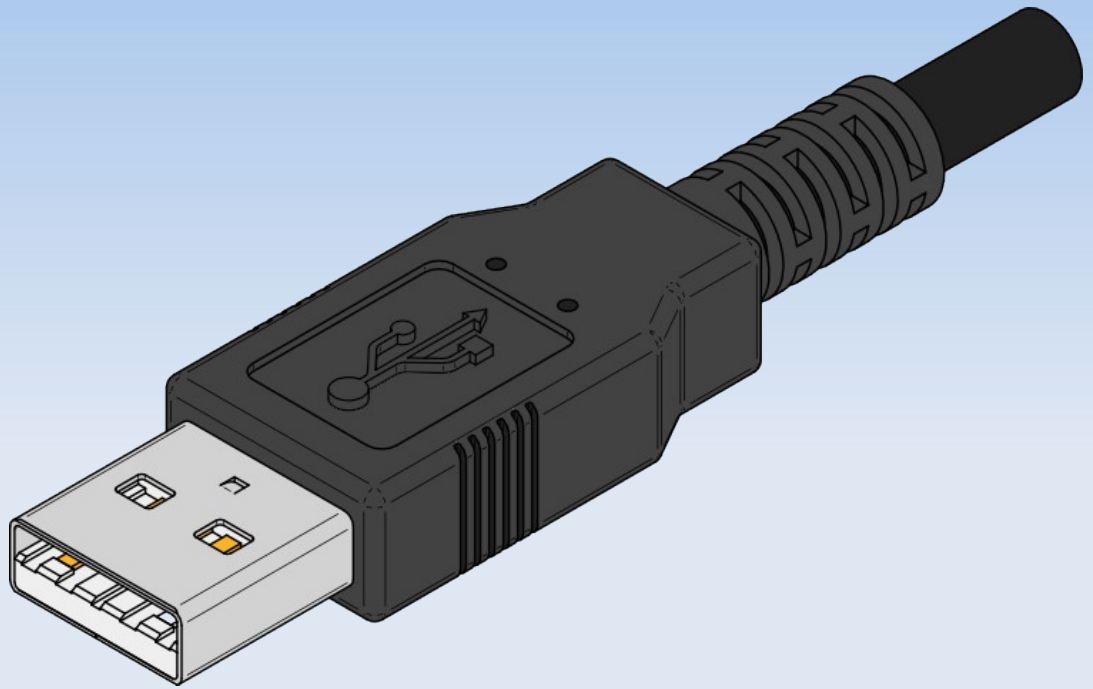
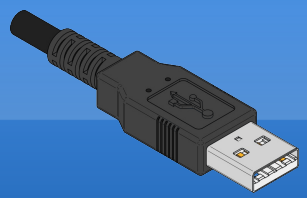
USB Reverse Engineering



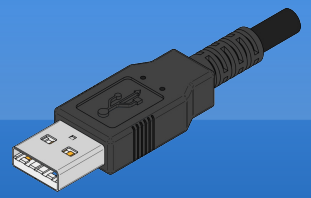
Alexander „alech“ Klink
darmstadt.ccc.de!alech

Chaostreff Darmstadt
17.03.2009

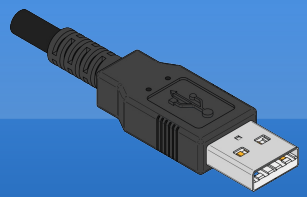


USB Reverse Engineering

agenda

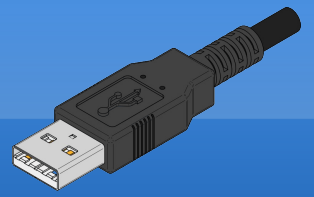


- Warum?
- Wie?
- Alternativen
- Q & A
- (evtl.) Ausprobieren

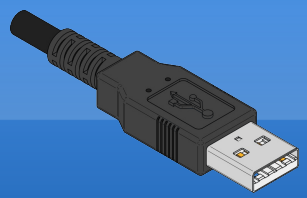


USB Reverse Engineering

Warum?

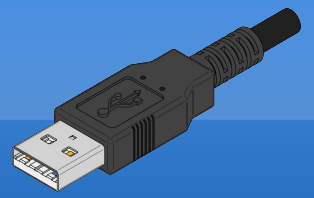


- Freie Software für USB-Devices
  - Konkret: Sony *α*700 DSLR
- (Security-)Analyse des Traffics
  - z.B. PINs bei Smartcardlesern
  - Software-Protection-Dongles
- „Because we can!“



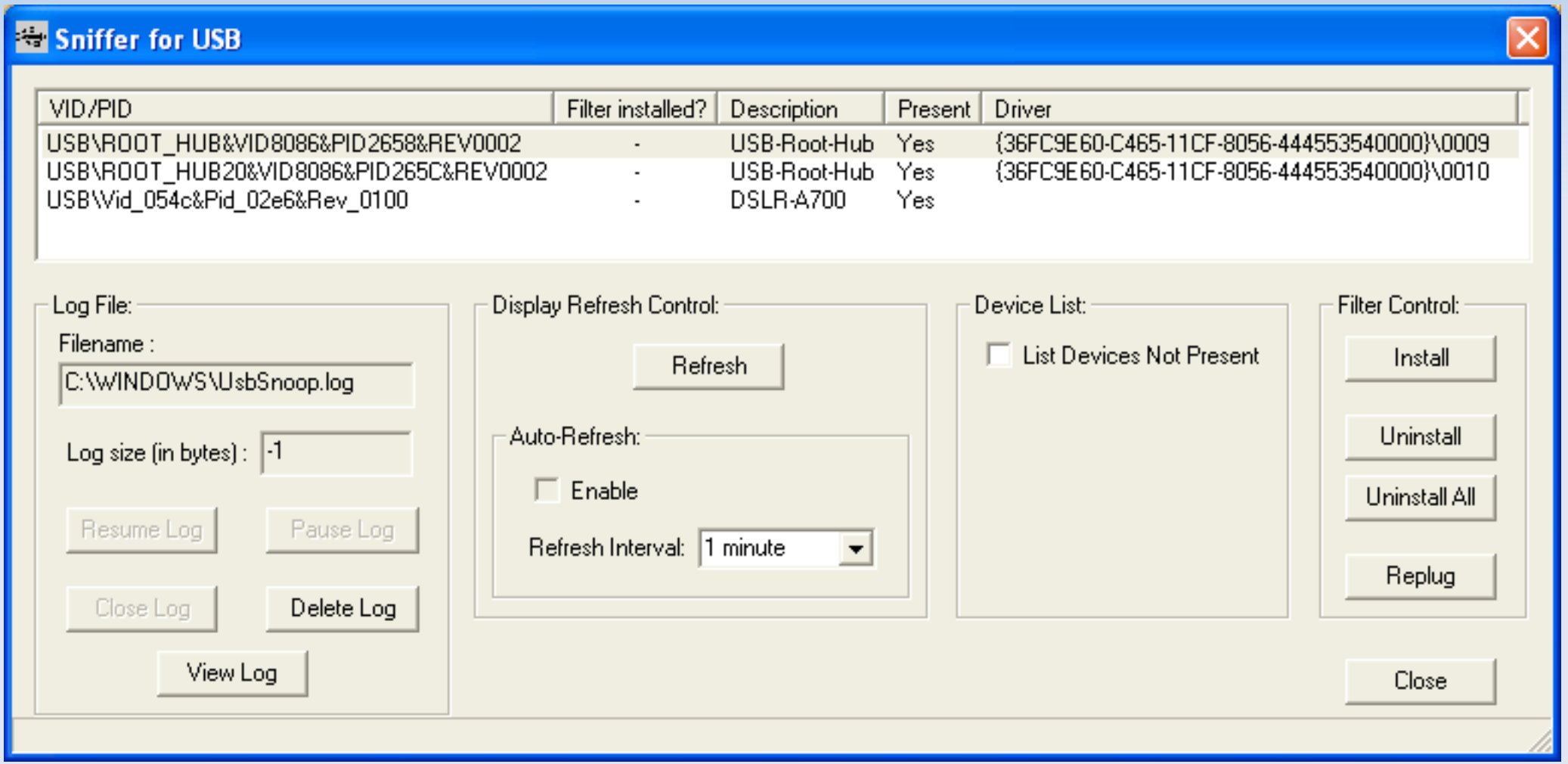
USB Reverse Engineering

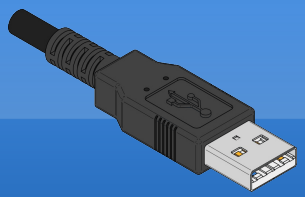
Wie?

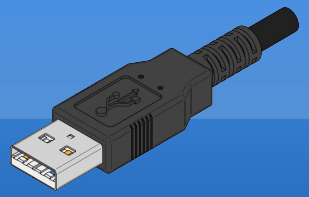


- Annahme: (Windows-)Software vorhanden
- Sniff
- Replay
- (Educated) Guessing

¶ Usbsnoop von Benôit Papillaut



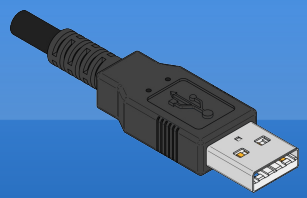




¶ Produziert (große) Logfiles:

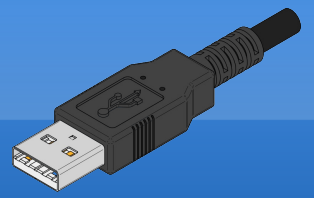
Terminal - usbsnoop.log (/tmp) - VIM

```
[300895 ms]  >>>  URB 7 going down  >>>
-- URB_FUNCTION_BULK_OR_INTERRUPT_TRANSFER:
  PipeHandle            = 82fc93f4 [endpoint 0x00000001]
  TransferFlags         = 00000002 (USBD_TRANSFER_DIRECTION_OUT, USBD_SHORT_TRANSFER_OK)
  TransferBufferLength  = 0000000c
  TransferBuffer        = 00000000
  TransferBufferMDL     = 829c40e8
    00000000: 0c 00 00 00 01 00 01 10 01 00 00 00
  UrbLink               = 00000000
[300900 ms] UsbSnoop - MyInternalIOCTLCompletion(f5b09126) : fido=82c986e8, Irp=829a1a8
8, Context=82f4c440, IRQL=2
[300900 ms]  <<<  URB 7 coming back  <<<
-- URB_FUNCTION_BULK_OR_INTERRUPT_TRANSFER:
  PipeHandle            = 82fc93f4 [endpoint 0x00000001]
  TransferFlags         = 00000002 (USBD_TRANSFER_DIRECTION_OUT, USBD_SHORT_TRANSFER_OK)
  TransferBufferLength  = 0000000c
  TransferBuffer        = 00000000
  TransferBufferMDL     = 829c40e8
  UrbLink               = 00000000
~
```

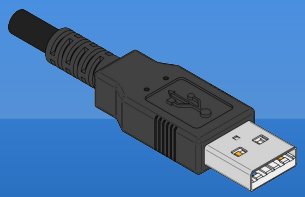


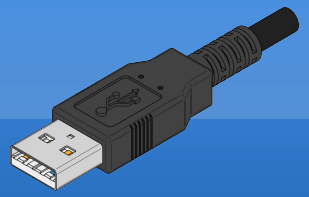
USB Reverse Engineering

Wie? – Replay



- Praktisch wenn wenig Traffic
- Sony *α*700 produziert viel „NOP“-Traffic
- Replay um rauszufinden, wo bzw. wann wirklich was passiert
- usbsnoop2libusb.pl

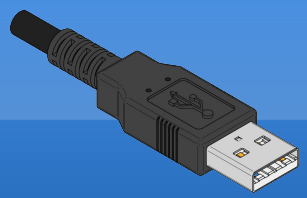




¶ Generiert C-Code (libusb)

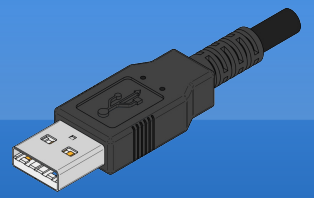
Terminal - shutter.c (~/devel/alphamote) - VIM

```
    ret = usb_bulk_write(devh, 0x00000001, buf, 0x0000010, 1000);
    printf("3982 bulk write returned %d, bytes: ", ret);
    print_bytes(buf, ret);
    printf("\n");
    usleep(7*1000);
    memcpy(buf, "\x0e\x00\x00\x00\x02\x00\x07\x92\x0e\x05\x00\x00\x02\x00", 0x000
0000e);
    ret = usb_bulk_write(devh, 0x00000001, buf, 0x000000e, 1000);
    printf("3983 bulk write returned %d, bytes: ", ret);
    print_bytes(buf, ret);
    printf("\n");
    usleep(6*1000);
    ret = usb_bulk_read(devh, 0x00000082, buf, 0x0000200, 1030);
    printf("3984 bulk read returned %d, bytes: ", ret);
    print_bytes(buf, ret);
    printf("\n");
    usleep(68*1000);
    memcpy(buf, "\x0c\x00\x00\x00\x01\x00\x08\x92\x0f\x05\x00\x00", 0x000000c);
    ret = usb_bulk_write(devh, 0x00000001, buf, 0x000000c, 1000);
    printf("3985 bulk write returned %d, bytes: ", ret);
    print_bytes(buf, ret);
    printf("\n");
    usleep(4*1000);
    ret = usb_bulk_read(devh, 0x00000082, buf, 0x0000200, 1030);
    printf("3986 bulk read returned %d, bytes: ", ret);
    print_bytes(buf, ret);
    printf("\n");
    usleep(9*1000);
                                                  48,1          43%
```

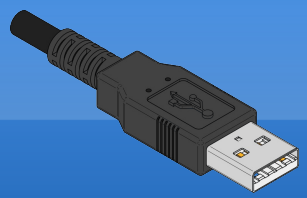


USB Reverse Engineering

Wie? – Educated Guessing

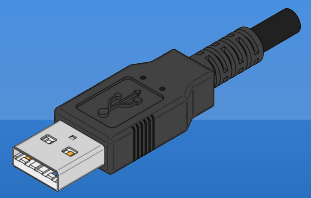


- Output der Antworten anstarren
- #ifdef 0 / #endif
- Strukturen bzw. bereits bekannte Protokolle identifizieren
  - im konkreten Fall: PTP-Erweiterung

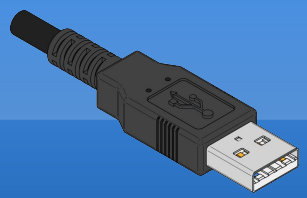


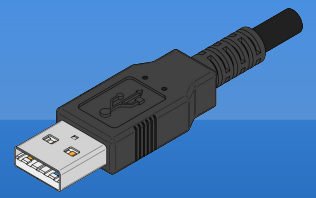
USB Reverse Engineering

Alternativen



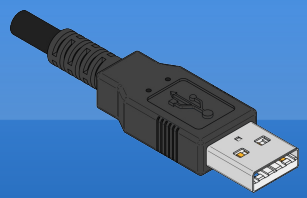
- Hardware USB-Analyzer
  - sicher schick, kostet aber $$$
- Virtual USB Analyzer von VMWare
  - wird benutzt um deren USB-Stack zu testen
  - GUI mit Timeline, Diff-Modus





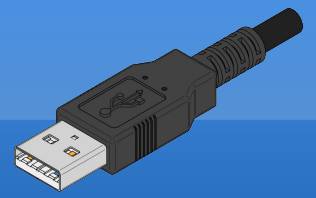
¶ Danke!

¶ Q & A



USB Reverse Engineering

Links



- Usbsnoop
  - http://benoit.papillault.free.fr/usbsnoop/
- usbsnoop2libusb.pl
  - http://iki.fi/lindi/darcs/usbsnoop2libusb/usbsnoop2libusb.pl
- Virtual USB Analyzer
  - http://vusb-analyzer.sourceforge.net/
- Alphamote
  - http://repo.or.cz/w/alphamote.git