WhatTheToool

Bluetooth Security Workshop

July 30th 2005, Liempde, Netherlands

Adam Laurie, Marcel Holtmann, Martin Herfurt

Who we are

- Adam Laurie
  - CSO of The Bunker Secure Hosting Ltd.
  - Co-Maintainer of Apache-SSL
  - DEFCON Staff/Organiser
- Marcel Holtmann
  - Maintainer and core developer of the Linux Bluetooth Stack BlueZ
- Martin Herfurt
  - Security Researcher & Java Programmer
  - Founder of trifinite.org

Bluetooth Technology Overview

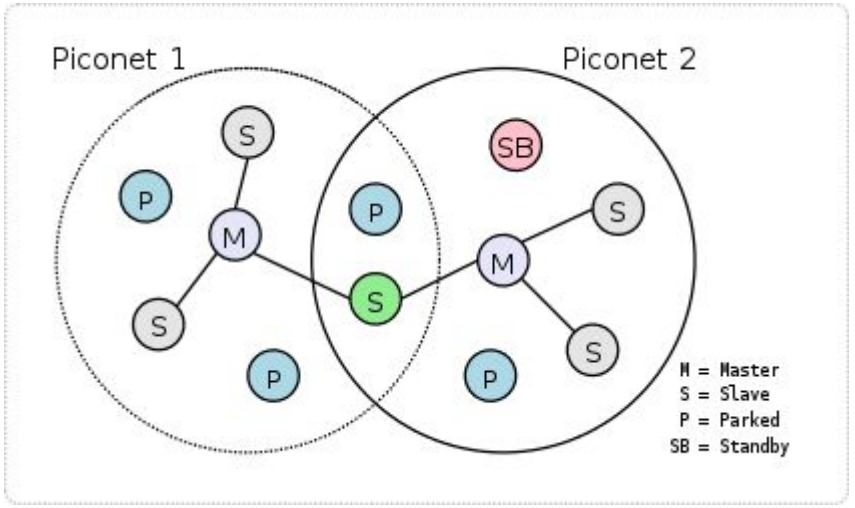
- Bluetooth SIG
  - Trade Association
  - Founded 1998
  - Owns & Licenses IP
  - Individual membership free
  - Promoter members: Agere, Ericsson, IBM, Intel, Microsoft, Motorola, Nokia and Toshiba
  - Consumer http://www.bluetooth.com
  - Technical http://www.bluetooth.org

Bluetooth Piconet

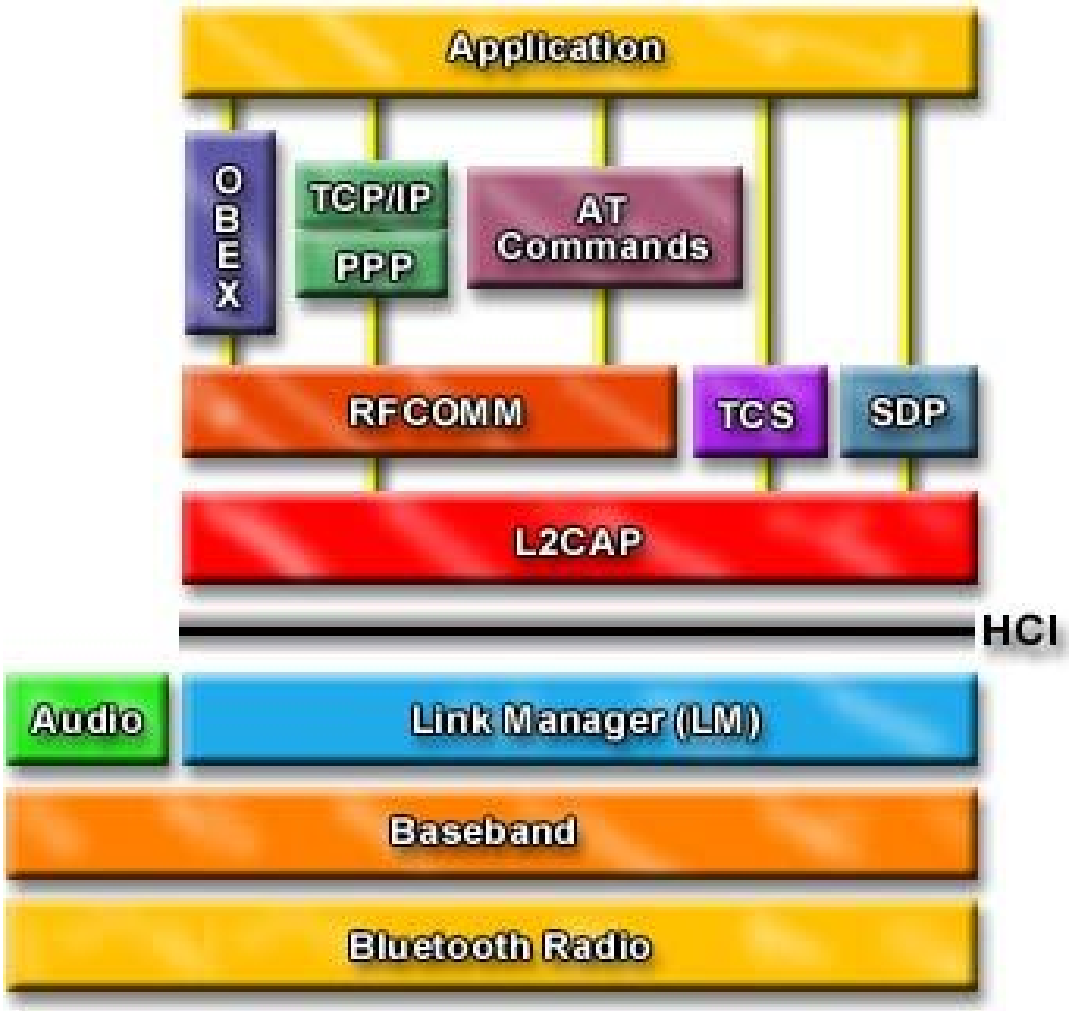
- Bluetooth devices create a piconet
  - One master per piconet
  - Up to seven active slaves
  - Over 200 passive members are possible
  - Master sets the hopping sequence
  - Transfer rates of 721 Kbit/sec
- Bluetooth 1.2 and EDR (aka 2.0)
  - Adaptive Frequency Hopping
  - Faster connection times
  - Transfer rates up to 2.1 Mbit/sec

Bluetooth Scatternet

- Connected piconets create a scatternet
  - Master in one and slave in another piconet
  - Slave in two different piconets
  - Only master in one piconet
  - Scatternet support is optional



Bluetooth Stack



Application specific security mechanisms

Bluetooth host security mechanisms

Security mechanisms on the Bluetooth chip

Security Mode

- Security mode 1
  - No active security enforcement
- Security mode 2
  - Service level security
  - On device level no difference to mode 1
- Security mode 3
  - Device level security
  - Enforce security for every low-level connection

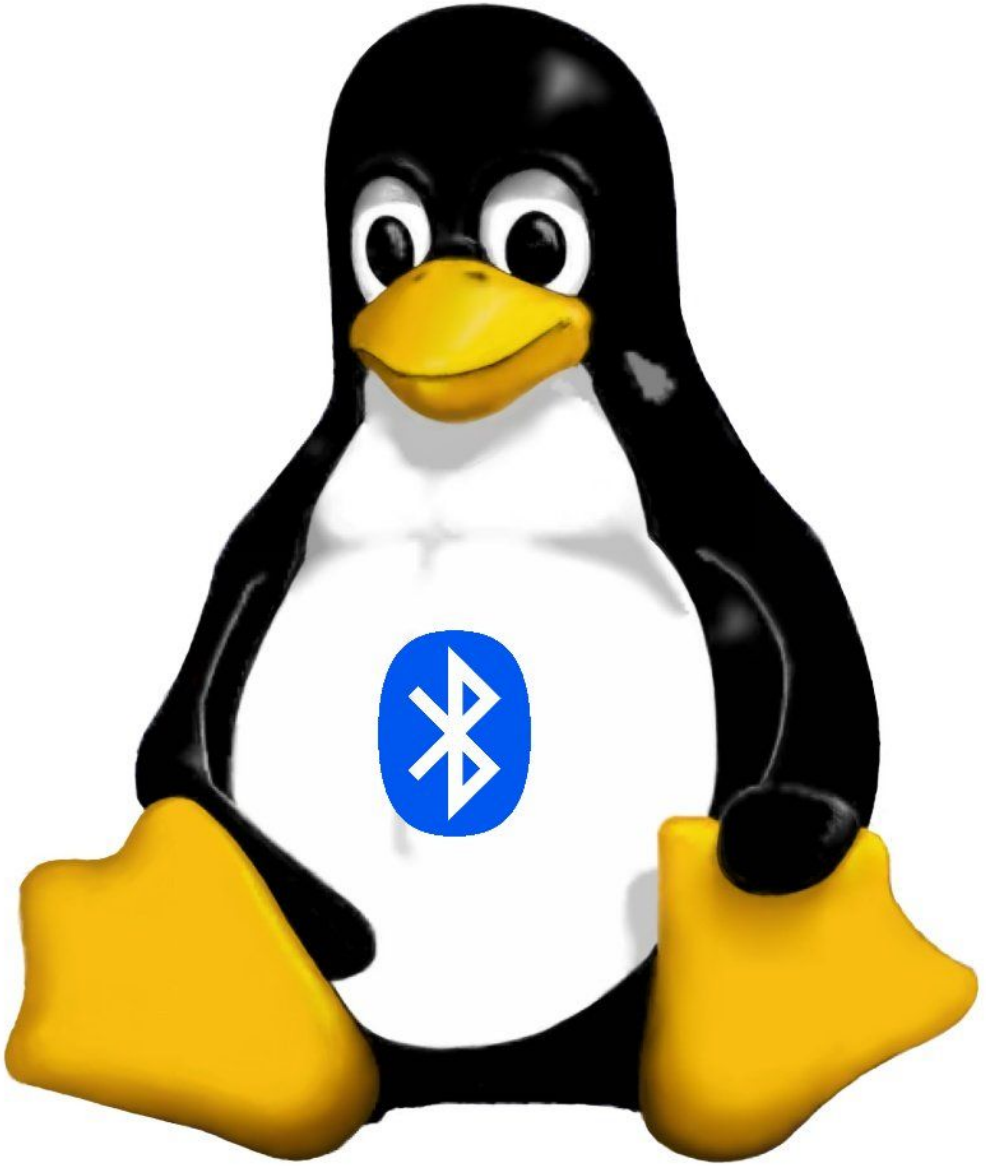
Sniffing with hcidump

- Recording of HCI packets
  - Commands, events, ACL and SCO data packets
- Only for local connections
- Decoding of higher layer protocols
  - HCI and L2CAP
  - SDP, RFCOMM, BNEP, CMTP, HIDP, HCRP and AVDTP
  - OBEX and CAPI
- No sniffing of baseband or radio traffic

How pairing works

- First connection
  (1) HCI_Pin_Code_Request
  (2) HCI_Pin_Code_Request_Reply
  (3) HCI_Link_Key_Notification
- Further connections
  (1) HCI_Link_Key_Request
  (2) HCI_Link_Key_Request_Reply
  (3) HCI_Link_Key_Notification (optional)

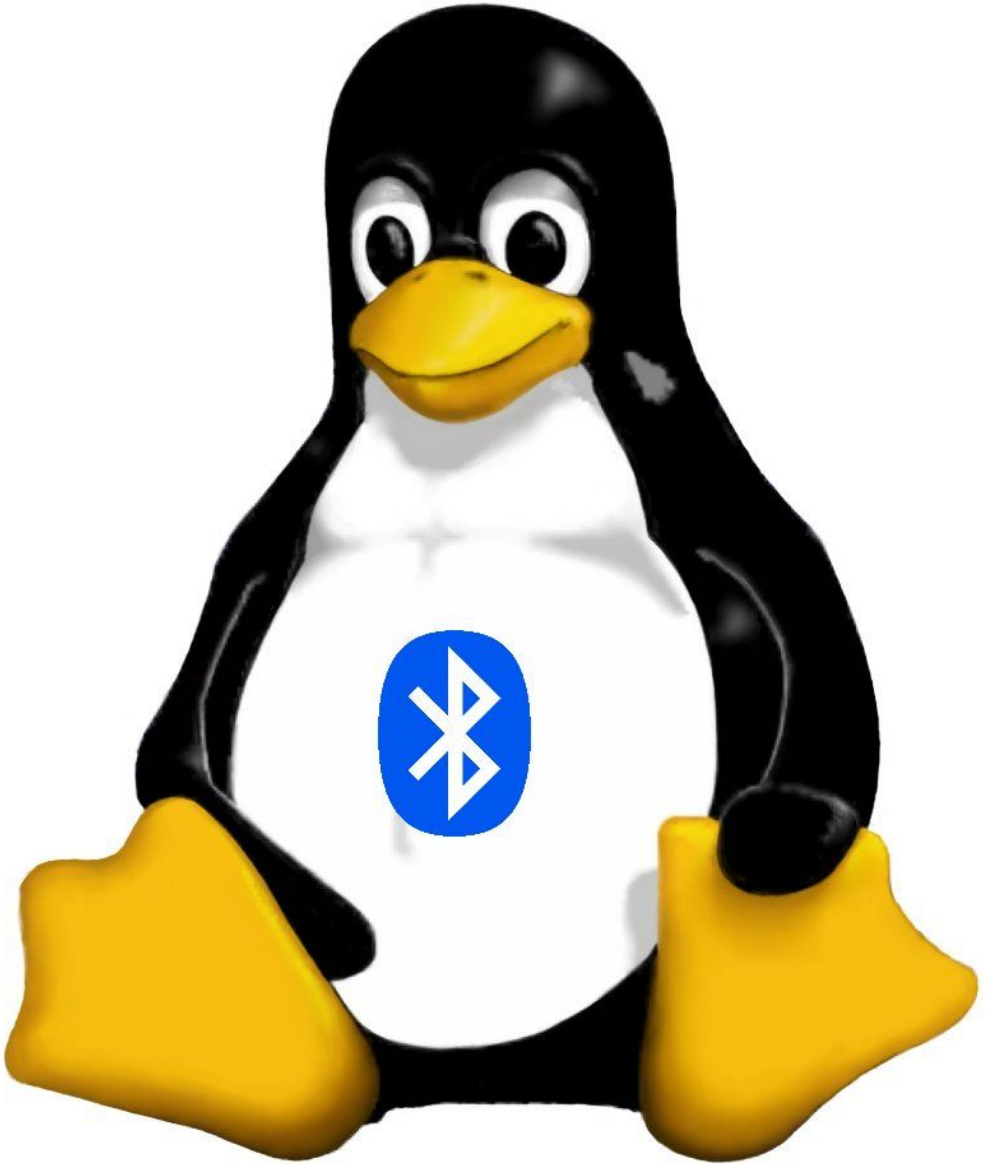
Demonstration: Exploring Phones



BlueSnarf

- Trivial OBEX PUSH channel attack
  - PULL known objects instead of PUSH
  - No authentication
- Infrared Data Association
  - IrMC (Specifications for Ir Mobile Communications)
    - e.g. telecom/pb.vcf
- Sony Ericsson T68, T68i, R520m, T610, Z1010
- Nokia 6310, 6310i, 8910, 8910i
- Devicelist on bluestumbler.org

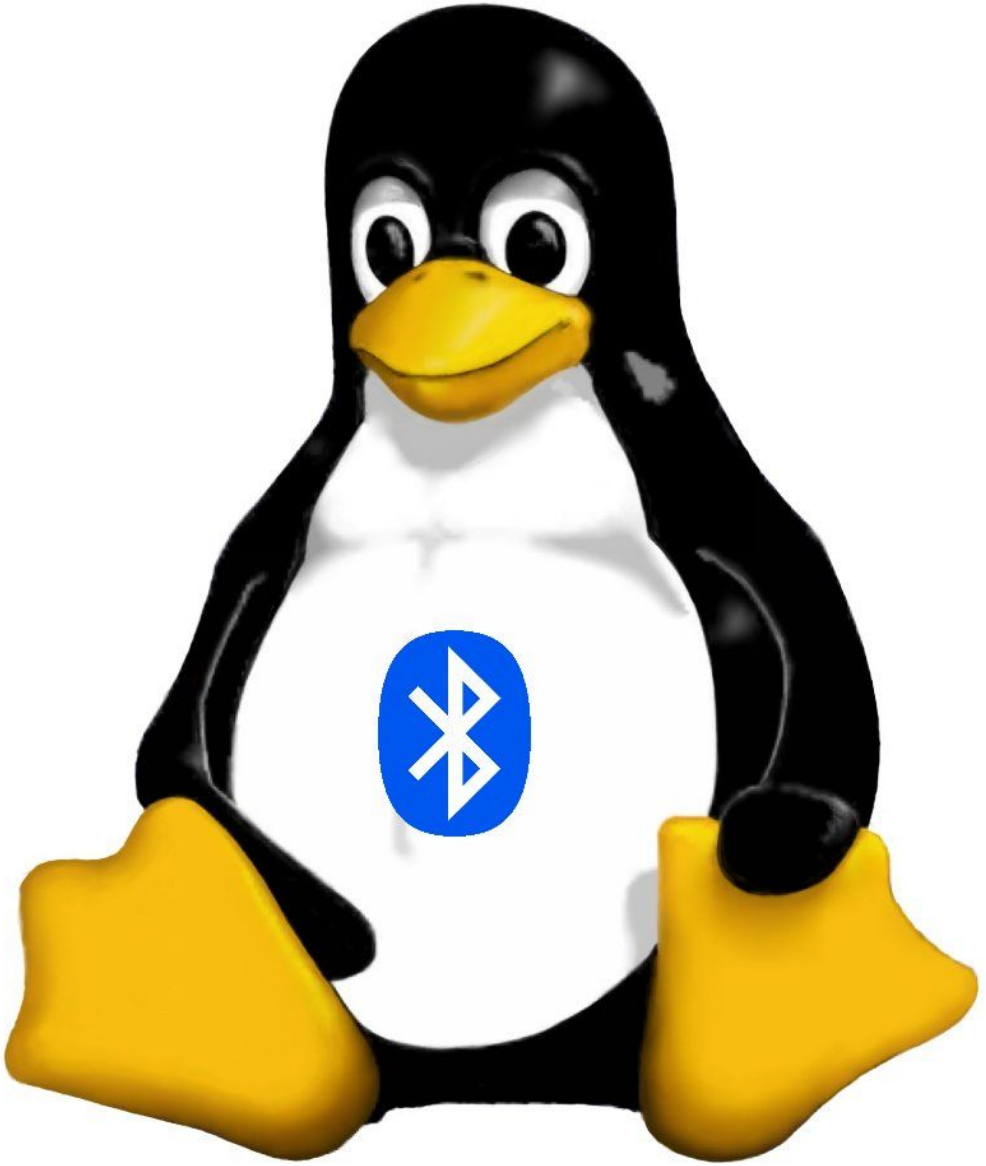
Demonstration: BlueSnarf



BlueSnarf++

- Trivial OBEX PUSH channel attack
  - Connect to Sync, OPP or BIP UUID/target
  - No authentication
  - Contents Browseable
  - Full read/write access
  - External Media Storage

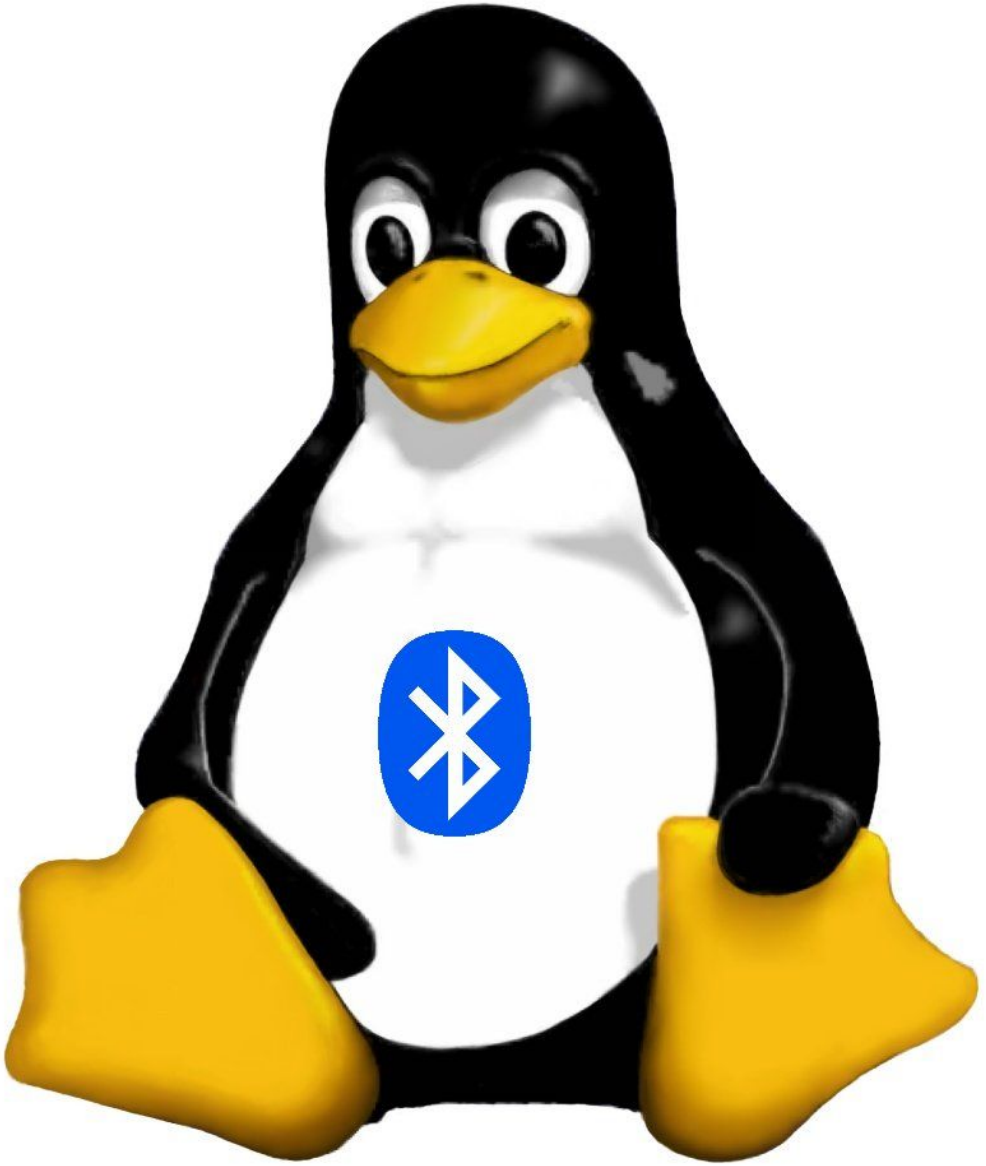
Demonstration: BlueSnarf++



BlueBug

- Issuing AT Commands to covert service
  - BlueBug is based on AT Commands (ASCII Terminal)
    - Very common for the configuration and control of telecommunications devices
  - High level of control...
    - Call control (turning phone into a bug)
    - Sending/Reading/Deleting SMS
    - Reading/Writing Phonebook Entries
    - Setting Forwards
    - -> causing costs on the vulnerable phones!

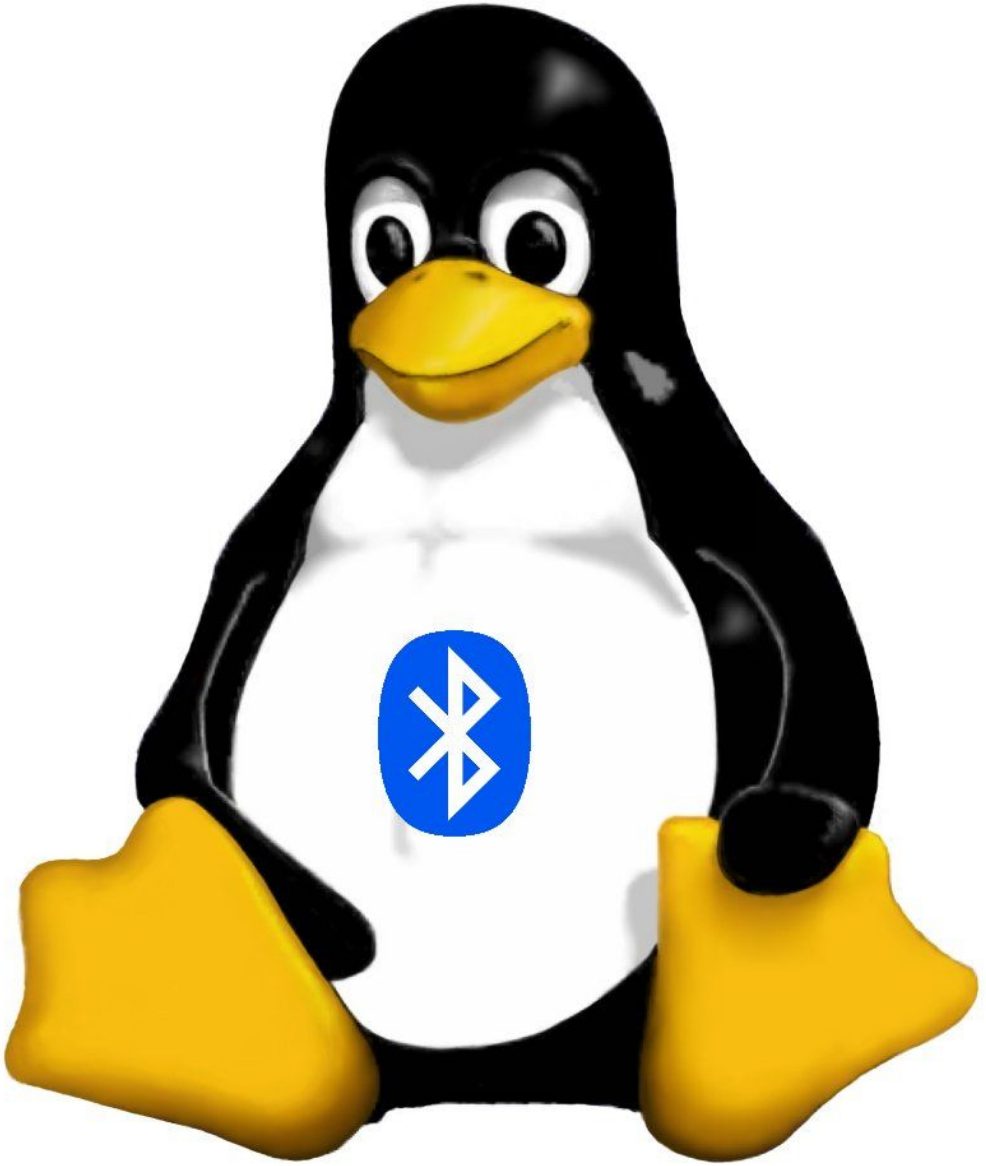
Demonstration: BlueBug



HeloMoto

- Requires entry in 'My Devices'
- OBEX PUSH to create entry
- Connect RFCOMM to Handsfree or Headset
  - No Key required
  - Full AT command set access
- Motorola V80, V5xx, V6xx and E398

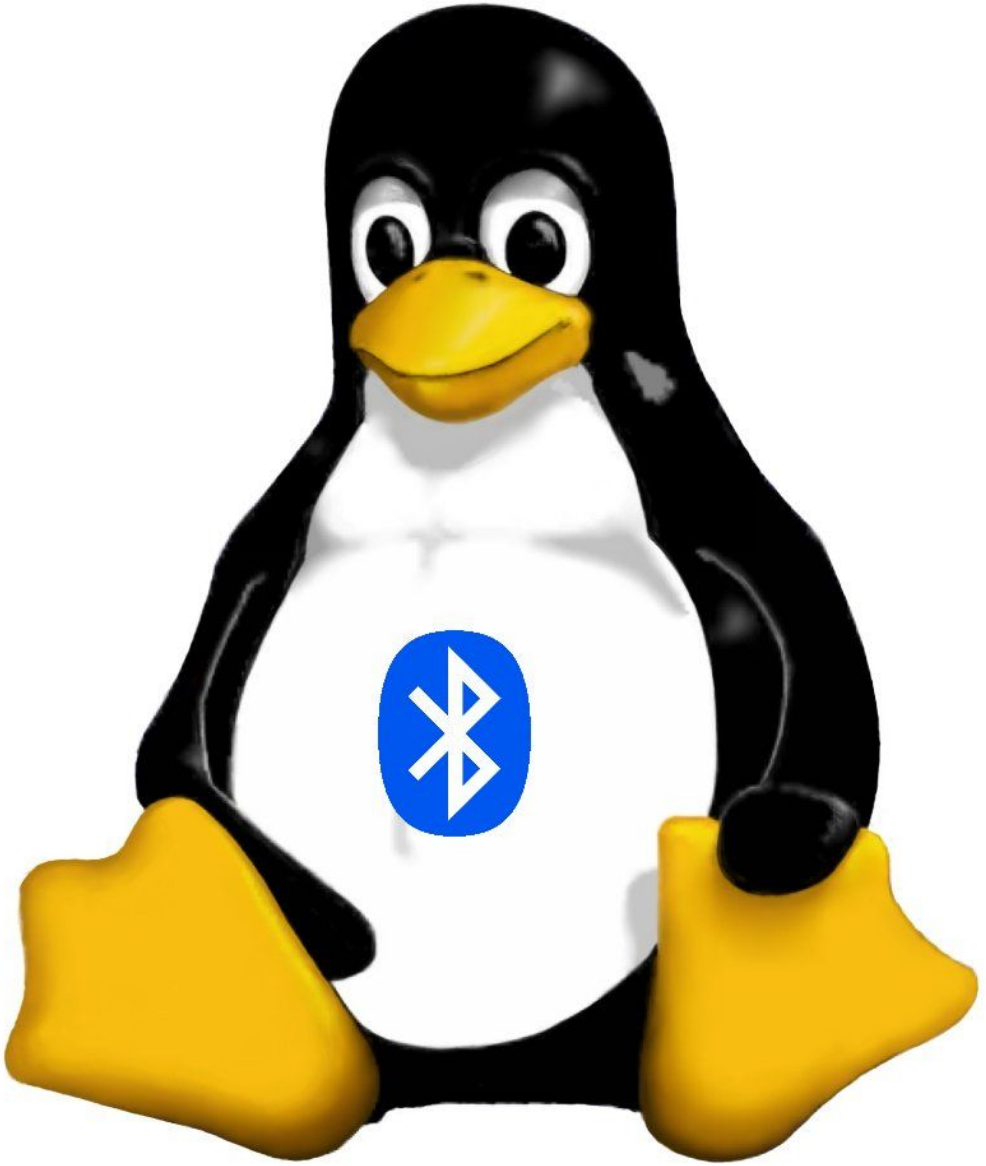
Demonstration: HeloMoto



Mode3 Abuse

- Create Pairing
  - Authenticate for benign task (e.g. vCard exchange)
  - Force authentication if required (set Mode 3)
- Connect to unauthorised Channels
  - Serial Profile, OBEX FTP, etc.

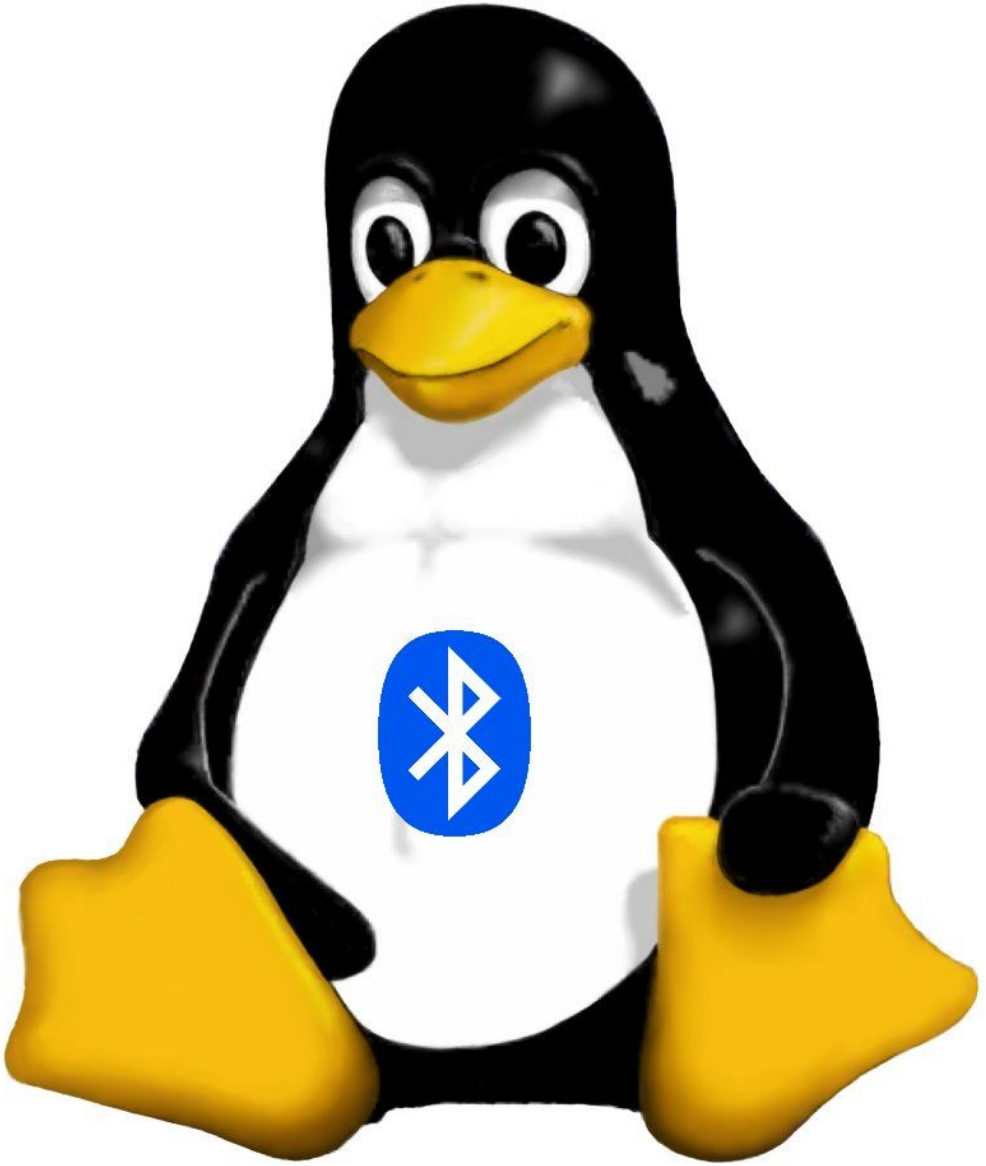
Demonstration: Mode-3 Abuse



BlueSmack

- Using L2CAP echo feature
  - Signal channel request/response
  - L2CAP signal MTU is unknown
  - No open L2CAP channel needed
- Buffer overflow
- Denial of service attack

Demonstration: BlueSmack

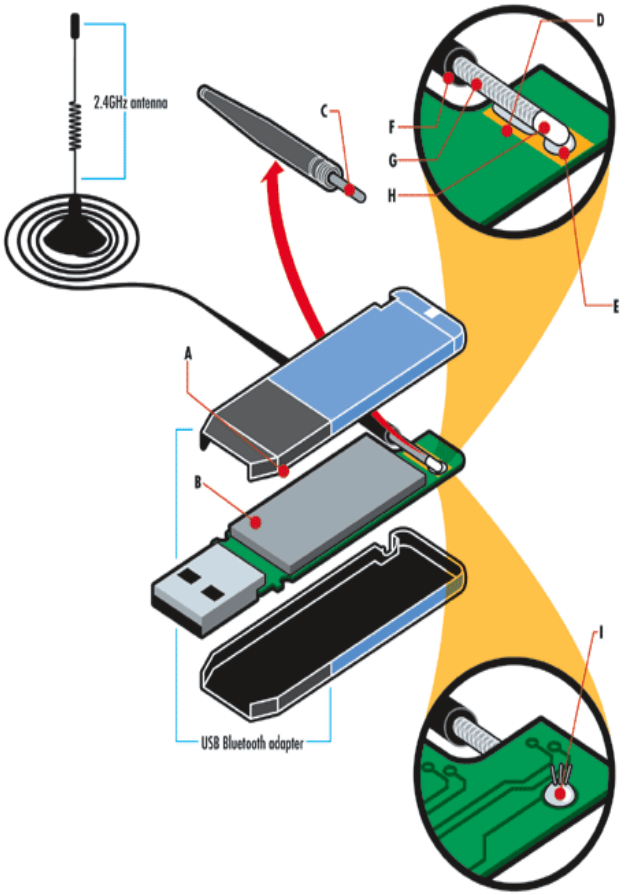


BlueBump

- Forced Re-keying
  - Authenticate for benign task (e.g. vCard exchange)
  - Force authentication if required (Mode 3)
- Partner deletes pairing
  - Hold connection open
  - Request Link Key Exchange
- Connect to unauthorised Channels
  - Serial Profile, OBEX FTP, etc.

Bluetooone

- Enhancing the range of a Bluetooth dongle by connecting a directional antenna -> as done in the Long Distance Attack
- Original idea from Mike Outmesguine (Author of Book: “Wi-Fi Toys”
- Step by Step instruction on trifinite.org



Long-Distance Attacking

- Beginning of August 2004 (right after DEFCON 12)
- Experiment in Santa Monica California with Flexilis
- Modified Class-1 Dongle Snarfing/Bugging Class-2 device (Nokia 6310i) from a distance of 1,78 km (1.01 miles)

Blueprinting – What is it?

Blueprinting™

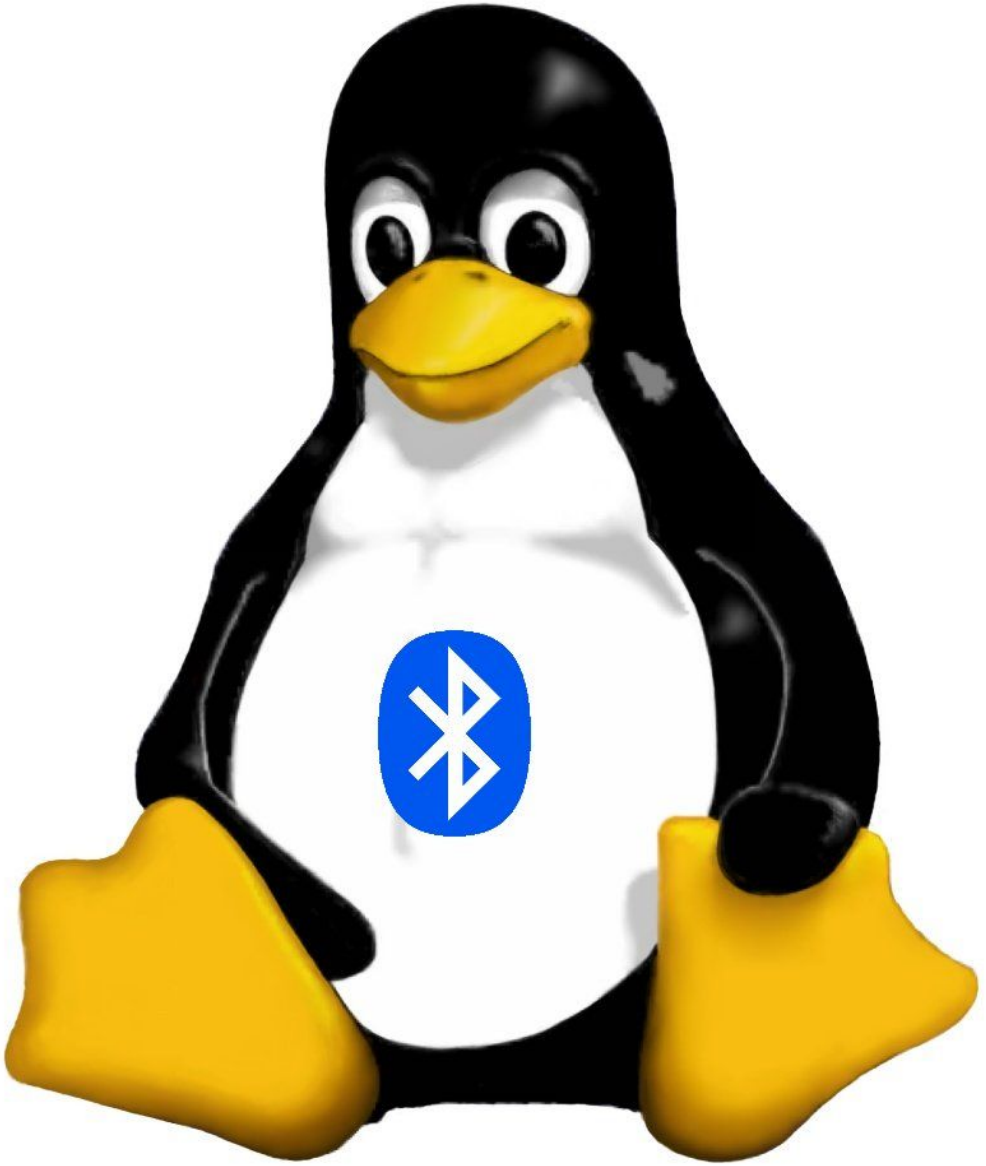
- Blueprinting is fingerprinting *Bluetooth* Wireless Technology interfaces of devices
- This work has been started by Collin R. Mulliner and Martin Herfurt
- Relevant to all kinds of applications
  - Security auditing
  - Device Statistics
  - Automated Application Distribution
- Released paper and tool at 21C3 in December 2004 in Berlin

Blueprinting - How

- Hashing Information from Profile Entries
  - RecordHandle
  - RFCOMM channel number
  - Adding it all up ($RecHandle_1$*$Channel_1$)+ ($RecHandle_2$*$Channel_2$)+...+($RecHandle_n$*$Channel_n$)
- Bluetooth Device Address
  - First three bytes refer to manufacturer
- Example of Blueprint

00:60:57@2621543

Demonstration: Blueprinting



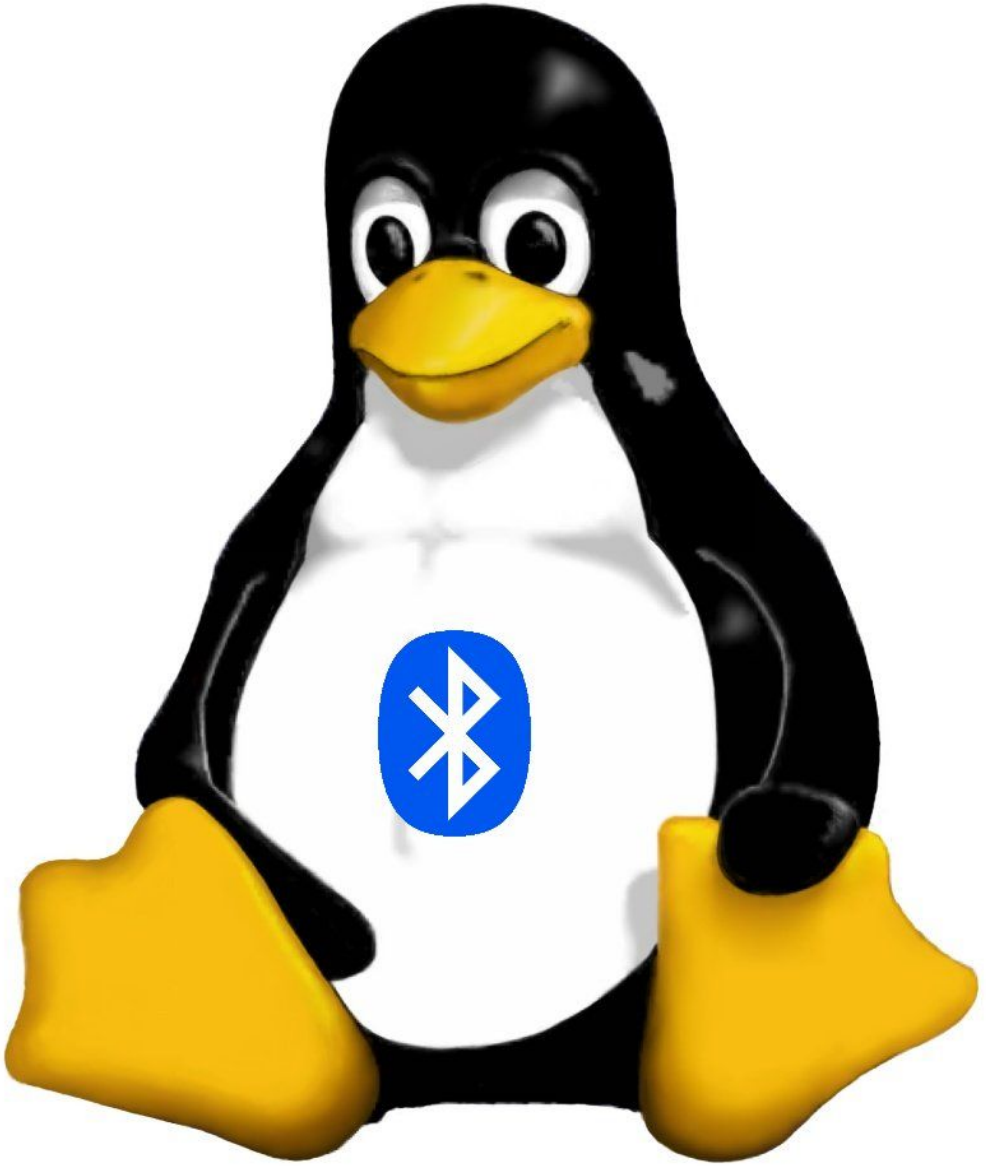
Blooover -What is it?

- Blooover - *Bluetooth* Wireless Technology Hoover
- Proof-of-Concept Application
- Educational Purposes only
- Phone Auditing Tool
- Running on Java
  - J2ME MIDP 2.0
  - Implemented JSR-82 (Bluetooth API)
  - Nokia 6600, Nokia 7610,
    Nokia 6670, ... Series 60
    Siemens S65
    SonyEricsson P900 ...

Blooover- What does it do?

- Blooover performs the BlueBug attack
  - Reading phonebooks
  - Writing phonebook entries
  - Reading/decoding SMS stored on the device (buggy..)
  - Setting Call forward (predef. Number) +49 1337 7001
  - Initiating phone call (predef. Number) 0800 2848283
- Please use this application responsibly!
  - Not with phones of strangers...
- Release at 21C3 in Berlin 29th December 2004
  - Over 60000 Downloads so far (avg. 500/day)

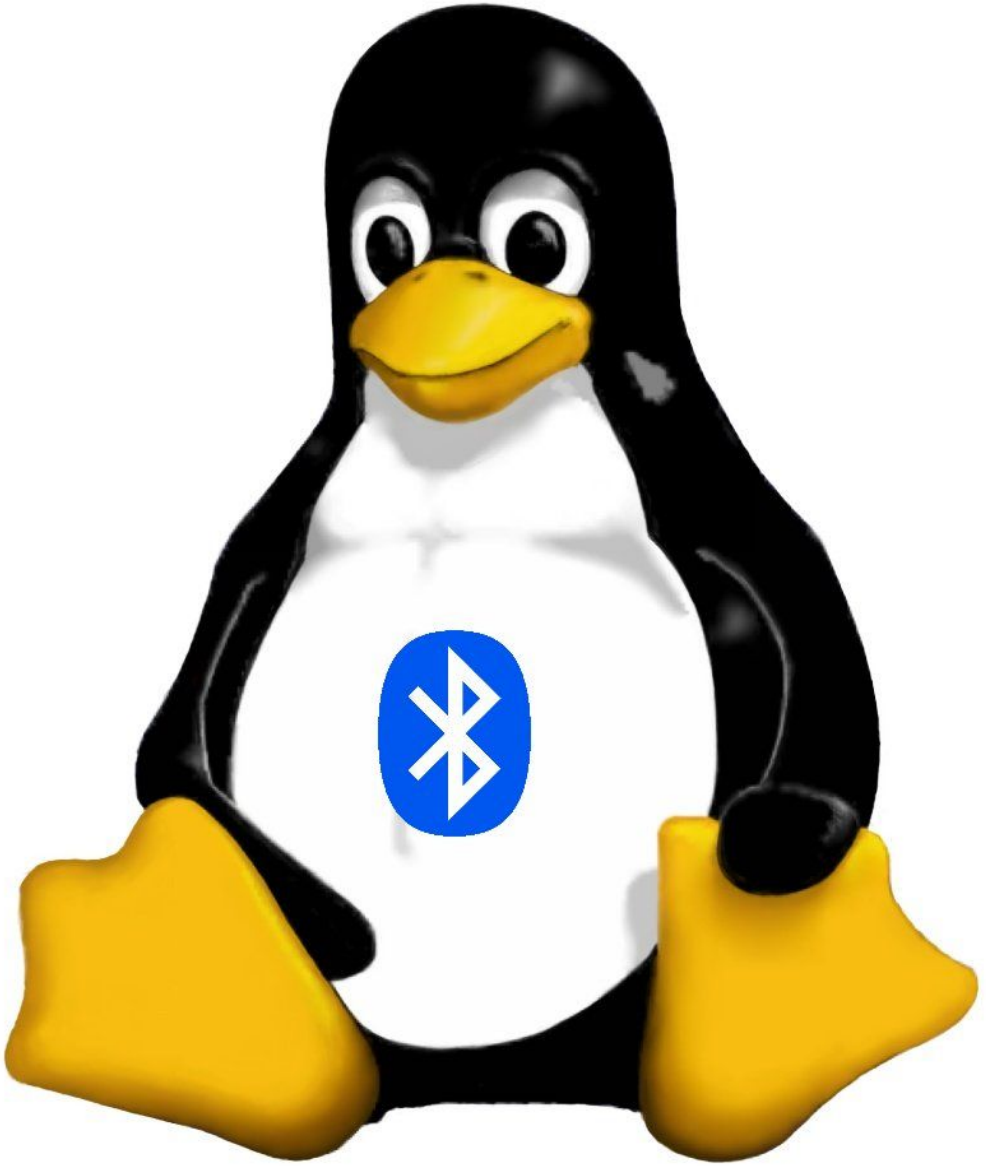
Demonstration: Blooover



BluePot

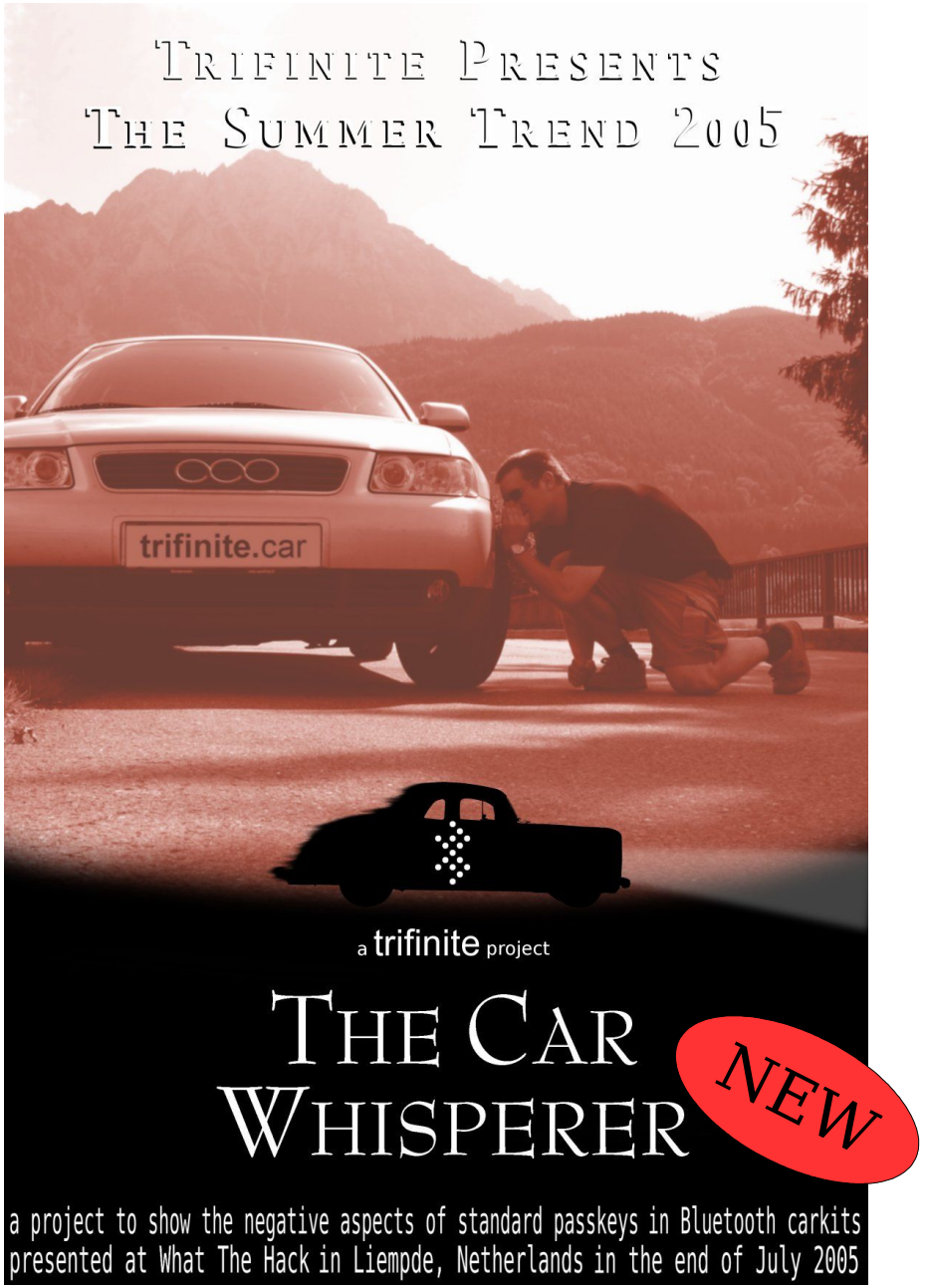
- Bluetooth HoneyPot
  - Runs on J2ME phones
  - Imitates vulnerable phone
  - Logs incoming attacks & device info
  - Strikeback capable

Demonstration: BluePot

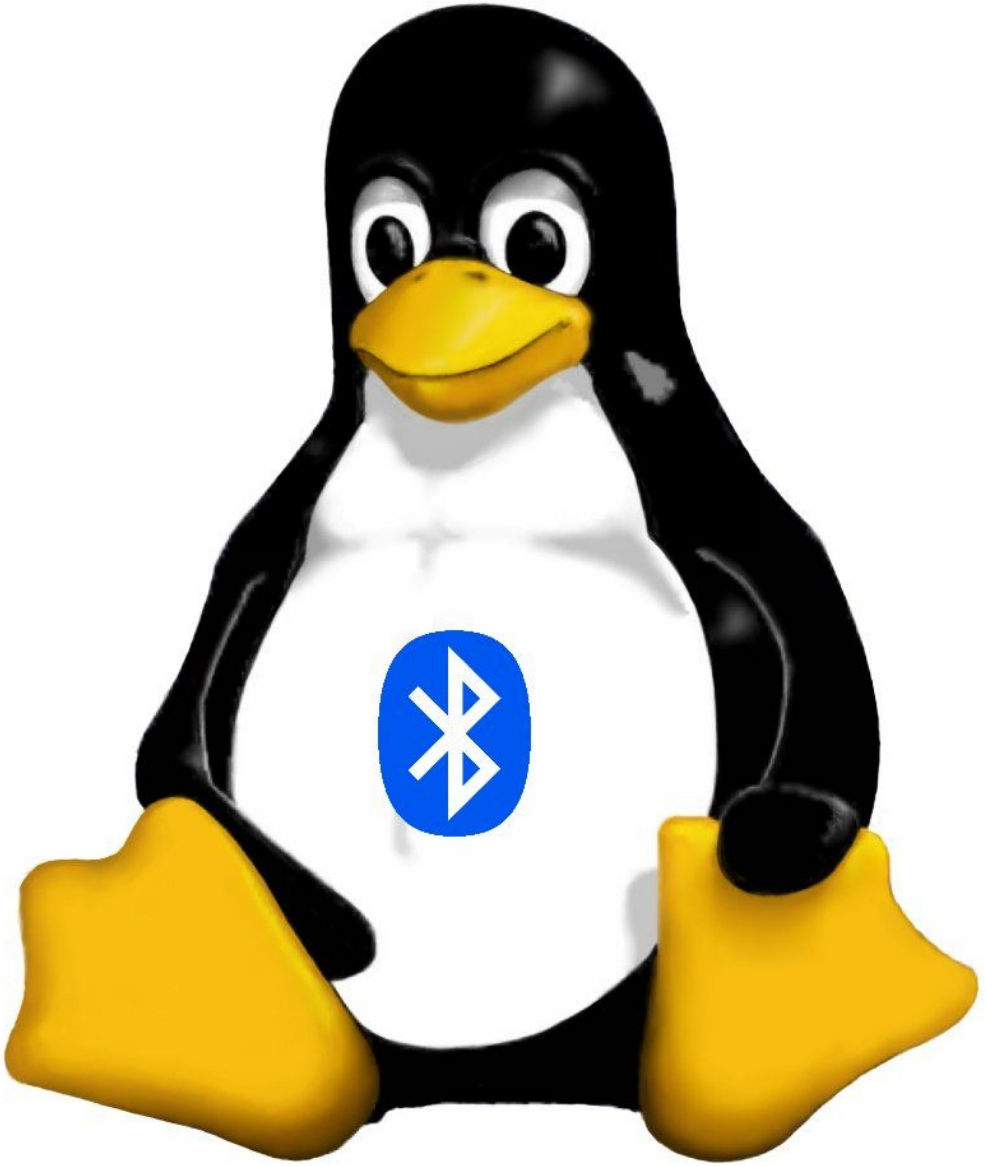


The Car Whisperer

- Uses standard passkey to connect to carkits
- Injects audio
- Records audio
- Use it
  - for talking to other drivers (be nice)
  - for eavesdropping to conversations in other cars
  - responsible!



Demonstration: Car Whisperer



Conclusions

- Bluetooth is a secure standard (per se)
  - Problems at application level
- Cooperation with Bluetooth SIG
  - Pre-release testing at UPF (UnplugFests)
  - Better communication channels for external testers
    - Security Expert Group mailing list
    - bluetooth.org more open areas
  - Mandatory security at application level

trifinite.org

- http://trifinite.org/
- Loose association of BT security experts
- Features
  - **trifinite.**blog
  - **trifinite.**stuff
  - **trifinite.**album
  - **trifinite.**group

trifinite.group

- Adam Laurie (the Bunker Secure Hosting)
- Marcel Holtmann (BlueZ)
- Collin Mulliner (mulliner.org)
- Tim Hurman (Pentest)
- Mark Rowe (Pentest)
- Martin Herfurt (trifinite.org)
- Spot (Sony)

trifinite.3

- Three things we provide to help you secure your products
  - **trifinite.**testing
  - **trifinite.**trust
    - Blooover
    - Blooonix
  - **trifinite.**training
- Contact us for details

Questions / Feedback / Answers

- Contact us via mailto:whatthehack@trifinite.org (group alias for Adam, Marcel and Martin)