NSA Playset: Bluetooth Smart

A presentation in five acts

Mike Ryan

iSEC Partners

Hack In The Box Malaysia

October 16, 2014

Standard Note

Bluetooth Smart == BLE

Act I

THE NSA PLAYSET

The NSA Playset

- NSA ANT catalog
- Sweet codenames!
  - DEITYBOUNCE
  - NIGHTSTAND
- Playset codenames
  - TWILIGHTVEGETABLE
  - SLOTSCREAMER
  - DUCHESSRIDE

Bluetooth Capabilities

- Ubertooth
- Crackle
- BlueZ

Capabilities: Ubertooth

- Discovering undiscoverable devices
- Rudimentary classic BT sniffing
- Robust BLE sniffing
- (WIP) BLE transmit

Capabilities: Crackle

- Cracking BLE key exchange
- Decrypting data

Needs to be paired with a sniffer, like Ubertooth

Capabilities: BlueZ

- Fully functional BR and BLE stack
- Able to act as master **AND** slave (BLE)
- Multipurpose utilities: recon

Existing Capabilities

→ Not bad for zero effort!

ANT Catalog: Bluetooth

ABSENT?!

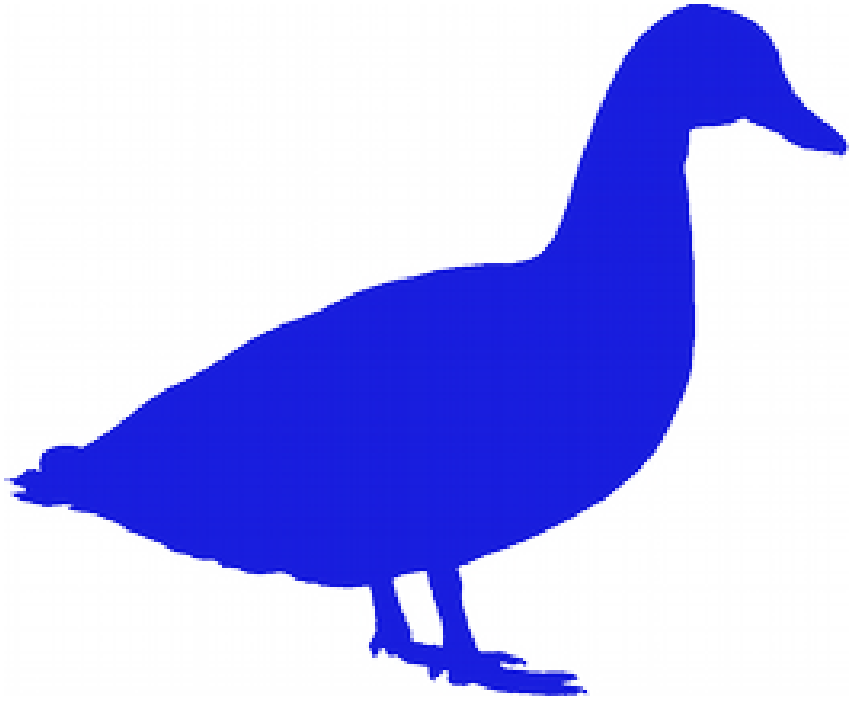
- Even the NSA doesn't care about Bluetooth
  - Yeah, right
- We don't have the full catalog
- Ellisys
  - more like elli$y$ amirite

Implied Bluetooth Capabilities

- Headset interception
- Surveillance of movement (tracking)
- Stack exploitation / RCE
- Keyboards → Now we're talking!

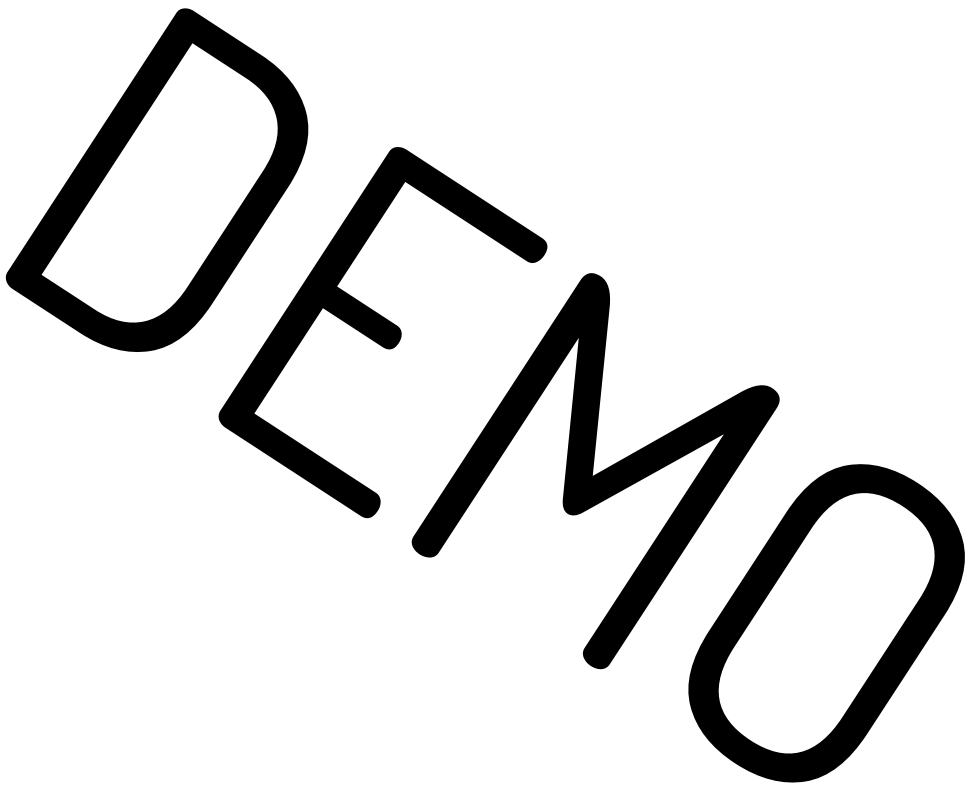
Keyboards and Mice

- Keystroke surveillance
- Keystroke injection (BlueDucky?)



TINYALAMO

- Keystroke injection!
- Duckyscript!



What just happened?

Act II

BLE PRIMER

BLE in a nutshell

- Master and slave
- Advertising

Advertising Data

- BD ADDR: 48-bits
- 16-bit Service UUIDs: 0x1812 (HID service)
- Appearance: 962 (Mouse)
- Complete Local Name: Bad Mouse

BLE in a nutshell

- Master and slave
- Advertising
- Link layer connections
- Pairing and encryption
- HID

Act III

TARGETS

Targets

OS must support HID over GATT

- Android 4.4 (Bluedroid)
- iOS 7/8 (iPad only)
- Mac OS 10.9+
- Linux / BlueZ 5 (Fedora 20, Arch, **NOT** Ubuntu)
- Windows 8 and Windows Phone 8

Not supported: BB 10, Android < 4.4

Do BLE Keyboards Exist?

- Yes, finally! (I think)
- BLE mice totally exist

HID is HID is HID

HID Encryption

- HID spec requires encryption
- Link layer **authenticated** encryption
- Pairing != Encryption

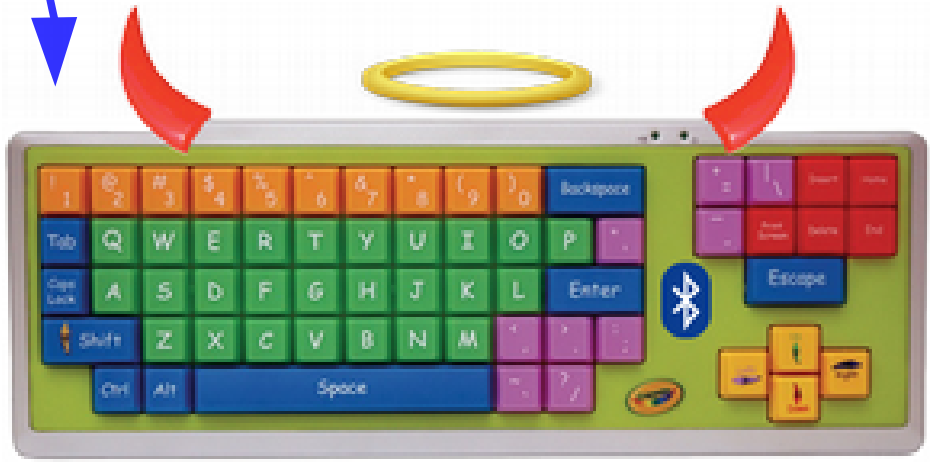
Act IV

BLE ATTACKS

Encryption Attacks

- Crack encryption, recover key
- Exploit implementation flaw
  - Encryption downgrade
  - Opportunistic downgrade
  - Forced re-pairing

Crack Encryption



Crack Encryption

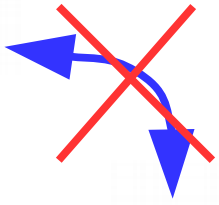
- Least invasive
- Downside: Must observe pairing
- Bonus: keystroke surveillance

Opportunistic Downgrade

Opportunistic Downgrade

- More invasive
- Distinguishable from legit keyboard
- Relies on OS bugs

Forced Unpairing



Forced Unpairing

- Most invasive
- Least broadly applicable

Naming Names

- Not yet!
- Two out of five OSes are vulnerable

BUILDING THE ATTACKS

Ingredients

- Regular Bluetooth dongle
- HCI_USER_SOCKET
- Scapy
- PyBT
- Wee bit of Python glue
- Ubertooth (optional)

HCI and HCI_USER_SOCKET

- All Bluetooth dongles speak HCI
- You can sniff this with recent Wireshark
- You can send totally illegal stuff this way
- HCI_USER_SOCKET – speak HCI from userspace
  - Linux 3.13 – all recent distros!

Scapy

- Framework for constructing packets
- Added extensive Bluetooth support
  - HCI, L2CAP, SM, ATT, GATT
  - `s = BluetoothUserSocket()`

PyBT

- I accidentally a Bluetooth stack
- BLE, SM (pairing), key management
- GATT client and server
- Roles: Central and Peripheral
- `central = LE_Central()`
- `periph = LE_Peripheral()`

Ubertooth

- Pull host and HID device addresses out of the air
- Nobody uses BLE private addresses correctly
- Nobody
- Not always necessary (hcitool lescan)

Putting it all together

- Ubertooth grabs BD ADDR
- Launch PyBT
  - Talks to dongle via HCI_USER_SOCKET
  - Builds packets with Scapy
- Advertise with address of mouse
- Wait for connection, send “no key” message
- Profit!

URL ME BRO

- https://github.com/mikeryan/PyBT
- https://bitbucket.org/mikeryan1/scapy

- Older stuff
  - https://github.com/mikeryan/crackle
  - https://github.com/greatscottgadgets/ubertooth

Thanks

- Marcel Holtmann
- Mike Ossmann / Dominic Spill / Ubertooth team
- Marcel Holtmann

- Hack In The Box!

Thank You

Mike Ryan

@mpeg4codec

mikeryan@isecpartners.com

https://lacklustre.net/