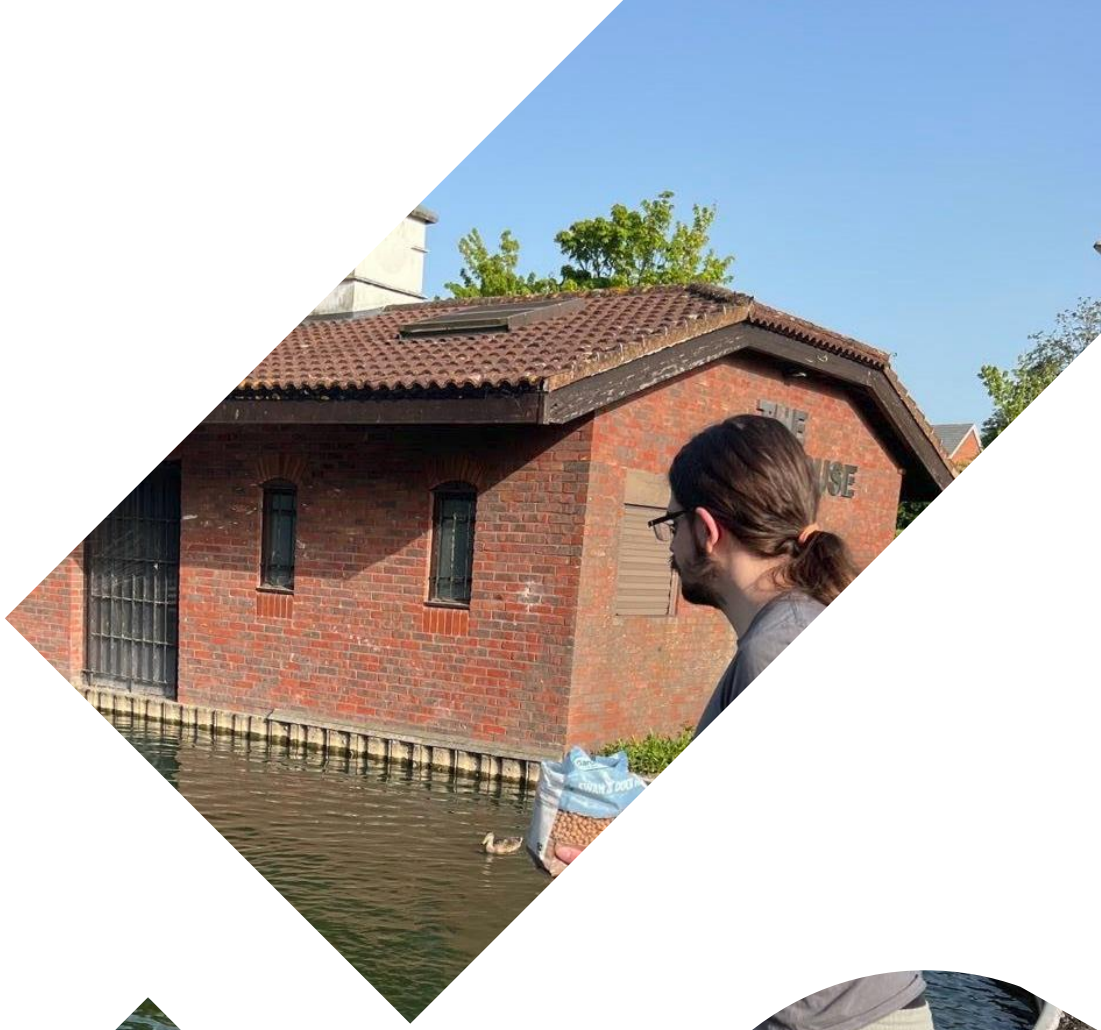
Sniffing Keyboards
(turns out some of them stink)

Miłosz Gaczkowski

WITH secure

Who am I?

- Miłosz Gaczkowski
  - /ˈmi.wɔʂ/
- Past life: University teaching
  - Computer science
  - Cybersecurity
- Current life: Mobile Security Lead at WithSecure
  - Android/iOS apps
  - Android devices
  - BYOD Mobile Application Management setups
- Knows the ONE WEIRD TRICK to get the attention of local ducks
  - Bring actual duck food, not bread



Today's talk

Not mobile security

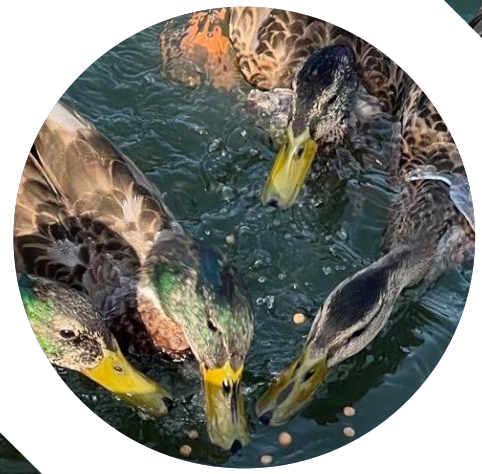
Results of a few days of investigating a suspected incident

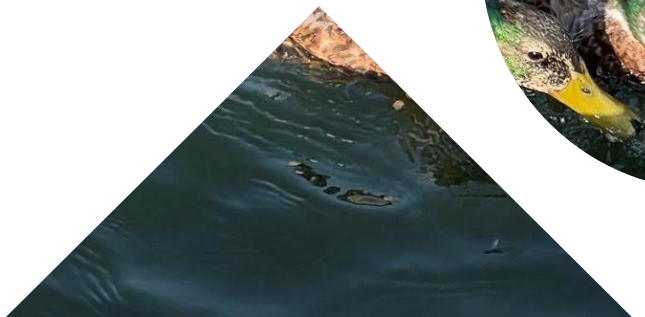
Goals for today:

- Entertainment
- Basic technical understanding of how keyboards work and how things can go wrong
- The process – especially the failures along the way!

When I started this research I was like:

And now I'm like:

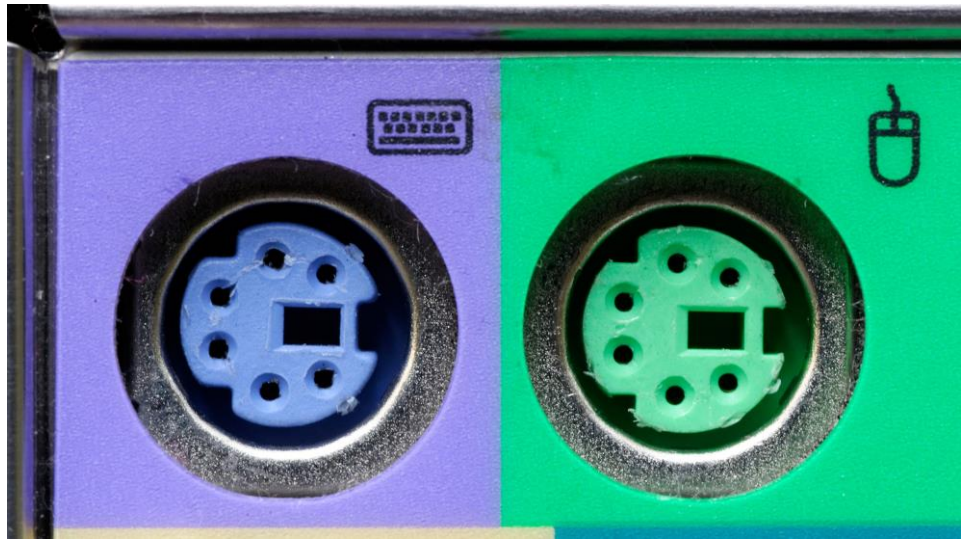




W / T H
secure

Keyboards!

- People want to be able to type
- Keyboards let them do that
- The way things used to be: parallel/serial, keyboard sends interrupts to host
  - Religious following among some gamers

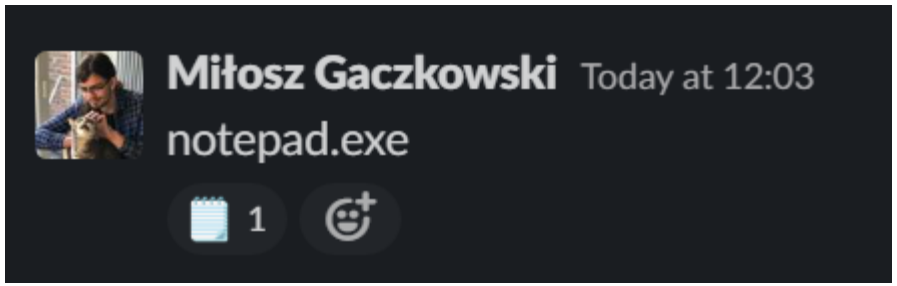


USB keyboards!

- Common approach today – USB HID
  - USB human interface device (class)
- Predefined set of standardised functions
- Covers any mouse, keyboard, game controller, etc.
- Good news: plug in any keyboard into any device, and it just works
- Bad news: plug anything that *says* it's a keyboard, and it… just works…
- Also: HID is super simple, so a device in the middle could intercept keypresses
- Simplicity at odds with security?

BadUSB keyboards!

- Keyboards generally not subject to many security measures
- Plug one in, and your OS is like “sure thing, buddy, that’s a keyboard, I will now accept keystrokes from this thing” :)
- How does the OS know we’re a keyboard?
  - Well, it’s ‘cause the device told it so
  - It wouldn’t just lie, right?
- BadUSB: USB Rubber Ducky and similar devices
  - Tell the computer you’re a keyboard, enter a bunch of predefined characters
  - Definitely didn’t end up posting “notepad.exe” on Slack by playing with it



WITH secure

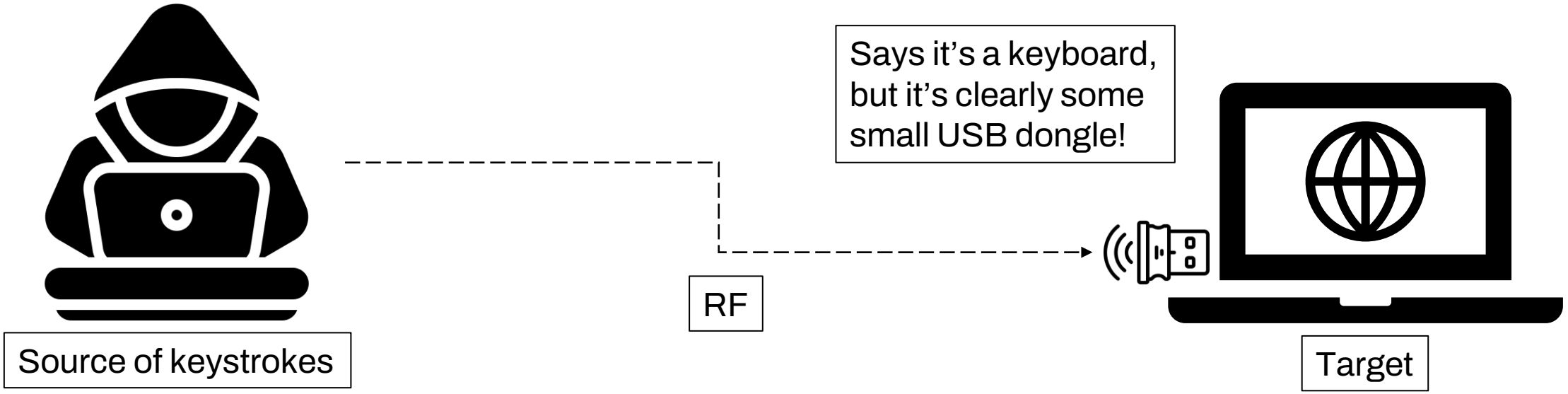
Things are bad, but not terrible

- Physical access required
- Moreover: access to a USB port required
- Still, potential for trickery and mischief is there:
  - Trick someone into plugging in an innocuous-looking USB device
  - Gets you anything a keyboard would, and it does it *fast*
  - Also handy if your target doesn't *have* a keyboard

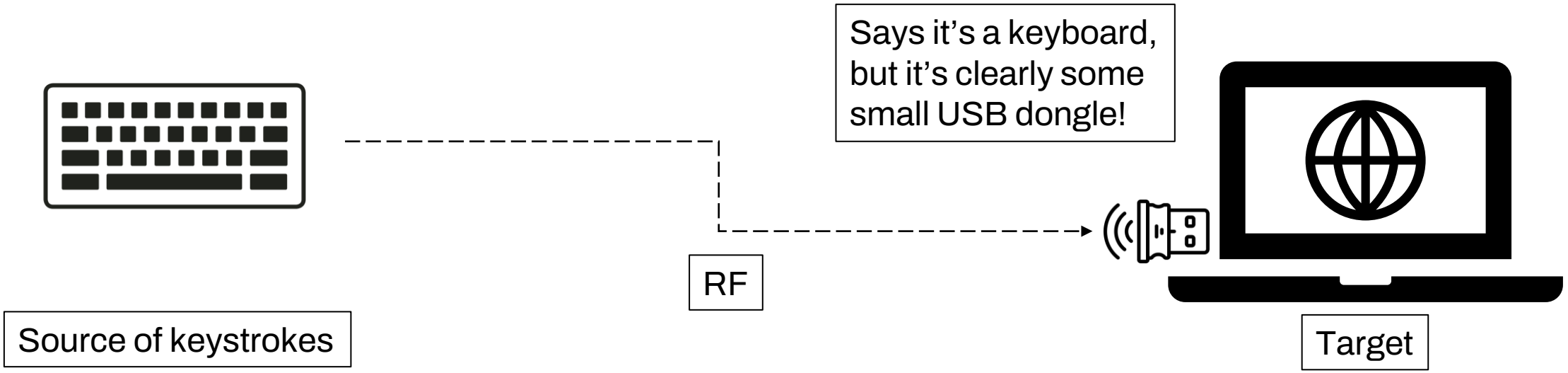
Could we take this further?

Maybe it doesn't have to rely on a pre-written script – **you could feasibly accept inputs remotely and have your device forward those on**

Hypothetical idea



Introducing... the wireless keyboard!



Our story!

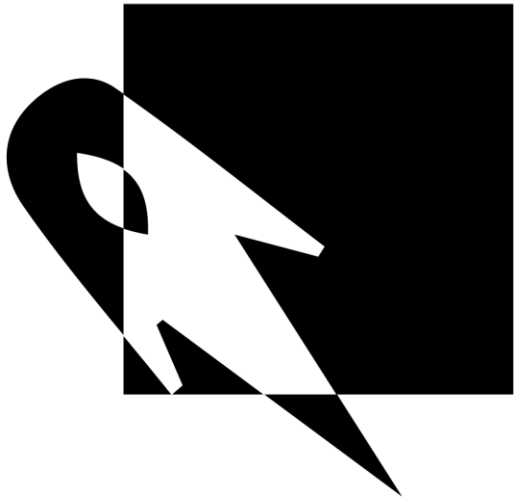
Things that happened

Story time!

- The following story is loosely based on facts
- We’ll have to blur some details and tell a few white lies
- But the bits that matter are true

Introducing: the University of Something

- Located somewhere in the Caribbean
- Pretty standard educational institution
  - Large lecture theatres
  - Most staff use desktops installed in the podiums
- Pretty standard IT practices
  - Don’t ask questions you don’t want to hear the answers to
- Probably gonna call them UoS/Something from this point on
- For the sake of the story:
  I’m there, helping with some minor IT stuff



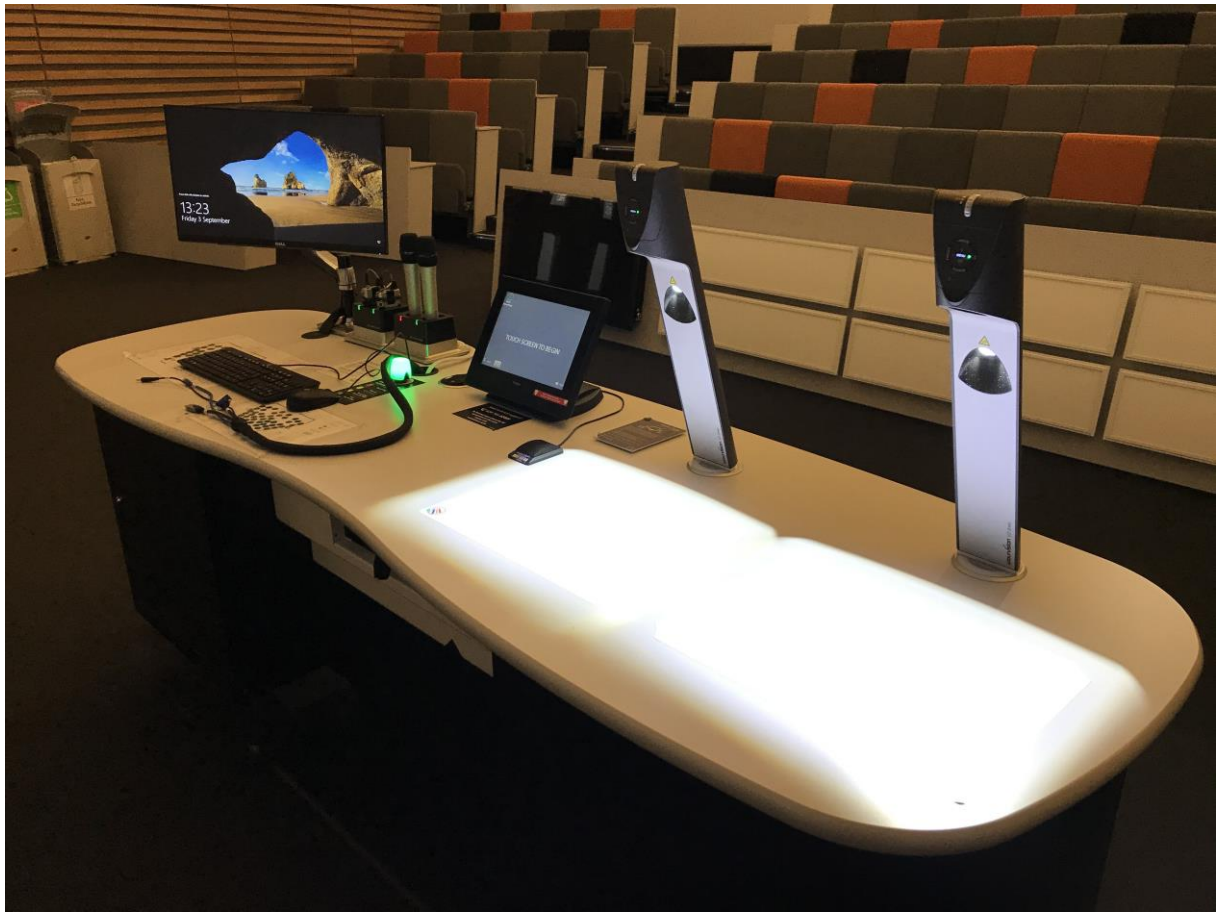
UNIVERSIDAD DE SÓMETHING

Ayo pizza nunc hic sita est

Introducing: the University of Something

Your standard lecture room contains:

- Audience seats and voice reinforcement (duh)
- A podium or lectern
- A desktop PC, maybe other display sources, AV equipment to switch between those
- BYOD laptops supported, but the desktops are very popular
- Relatively tamper-proof
  - Not secure enough to withstand theft/destruction
  - Enough to stop well-intentioned users from getting too hands-on



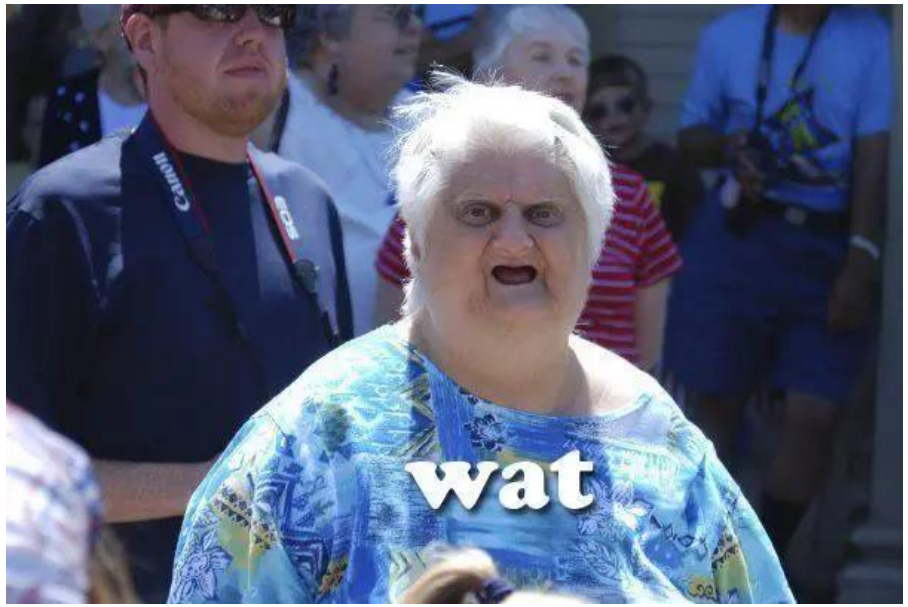
Strange IT tickets

- UoS started getting occasional IT tickets
- Wireless keyboard in <room> ran out of battery and needs replacing
- Gosh, how amateurish, why aren't they replacing batteries periodically?

...because there aren't any wireless keyboards in those rooms, **because** changing batteries would be a faff.

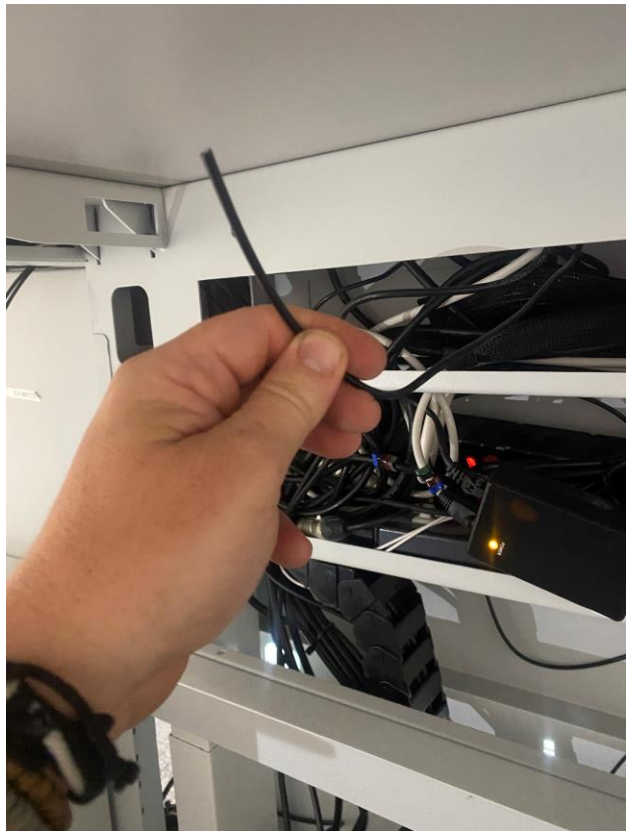
Wired keyboard are used in all rooms.

Wait... what?



Strange IT tickets

- OK, let's go to the room and find out what's going on
- The wired keyboard is no longer there, and, sure enough, there's a wireless keyboard just chilling on the podium
- The original USB cable for the keyboard has not been neatly removed – it's been **cut** and tucked away out of sight
- USB dongle plugged into the front of the machine
- Implies whoever did this couldn't access the back
- …what?



Initial response

- This is weird, but ultimately harmless, right?
- Just replace the keyboard back and forget about it.
- nope.png
- The keyboards just keep appearing in more and more rooms
- At this point, we'd really like to know **why**

The Keyboard →

OK... so why is this happening?

Hypotheses:

- Maybe someone in IT is doing this and this is all a misunderstanding
- They wanted to steal the wired keyboards
- Random user **really** hates wired keyboards so they replace them
- Just a prank?
- Some sort of security issue?
  - (ok look the context of this talk kinda spoils it, but we didn't want to jump to conclusions at the time)

OK... so why is this happening?

Hypotheses:

- Maybe someone in IT is doing this and this is all a misunderstanding
  - We checked, and we checked again - nope
- They wanted to steal the wired keyboards
  - We're talking about **very cheap** wired keyboards – highly unlikely someone would steal them
- Random user **really** hates wired keyboards so they replace them
  - I mean, I guess, but why wouldn't you just carry one keyboard with you? Why lose it to the room? Why remove the original?
- Just a prank?
  - Maaaybe? But after like 10 keyboards it's starting to look a little expensive.
- Some sort of security issue?
  - (ok look the context of this talk kinda spoils it, but we didn't want to jump to conclusions at the time)
  - **...maybe?**
  - **It *is* a university – stolen credentials could be valuable**
  - **It does sound contrived, though**
  - **Oh well, let's investigate anyway**

Initial investigation

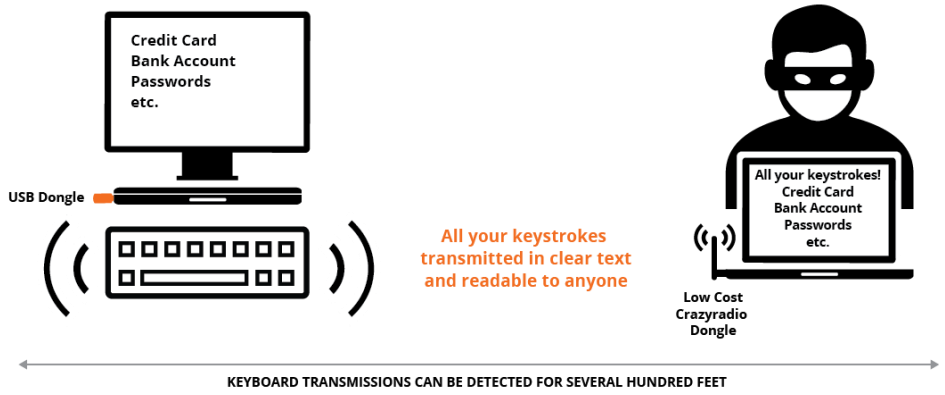
- Take the keyboards apart and find the haxx0r board that does all this
  - nope.bmp
- Take the dongle apart!
  - By which I mean smash it open with a hammer
  - Nothing there, and it seemed unlikely that an attacker would get anywhere with modding these
- Send the entire PC to IT security for deeper investigation!
  - No indicators of badness – just a normal PC with a normal keyboard dongle
- Frantically Google things until something makes sense!!!!!!!
  - But wait…
  - …what's this?
  - …a potential lead!

Introducing KeySniffer!

- https://keysniffer.net/ - originally reported on by Bastille
- tl;dr: a few keyboards have been identified in 2016 as trivial to intercept
  - RF communications trivial to intercept
  - Not encrypted
  - Easy to infer the specification
- Get yourself a receiver, tune in to the right channel, listen for keystrokes
  - Keystrokes not encrypted and protocol easy enough to reverse
  - You could also inject keystrokes (by just crafting packets)

Introducing KeySniffer!

- Previous issues of similar types were already known, but this one is a little easier
- Normally (e.g. Mousejack) you wouldn't be able to find an affected keyboard unless someone was actively typing on it
- KeySniffer-vulnerable dongles **really** like to announce they're on
  - Constant sync packets as long as the dongle is plugged in
  - So, theoretically, you can find find your targets more easily, prepare yourself, and sniff away



https://keysniffer.net/

Introducing KeySniffer!

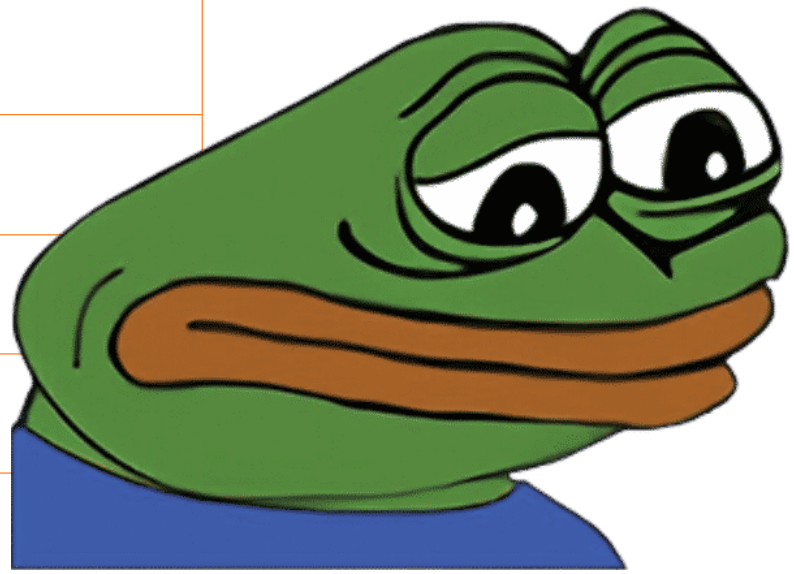
So, what do we need to play with this?

- Crazyradio PA dongle
  - Open project radio dongle
  - Based on nRF24LU1+ chip
  - €30 - €40
- Alternatively: Logitech unifying receiver
  - Same chip inside
  - Surprisingly flashable!
  - €15 - €20
- A vulnerable keyboard

...first things first, *is* our keyboard vulnerable?

Introducing KeySniffer!

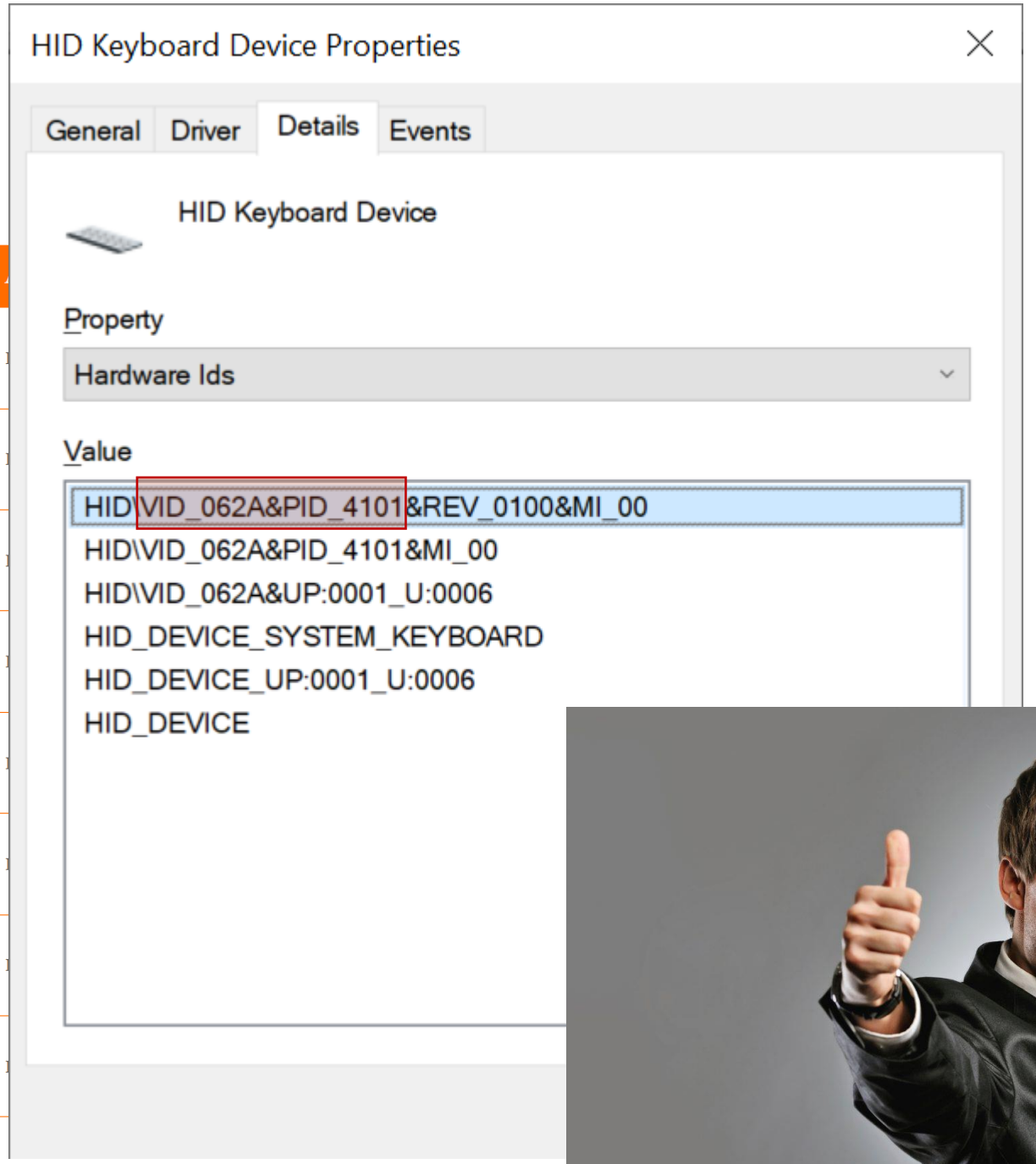
Vendor	Affected Devices	Advisory	Vendor Response
Anker	Anker Ultra Slim 2.4GHz Wireless Compact Keyboard Anker USB dongle (USB ID 062a:4101)	Link	Link
EagleTec	EagleTec K104 / KS04 2.4 GHz Wireless Combo keyboard EagleTec USB dongle (USB ID 062a:4101)	Link	
General Electric	GE 98614 wireless keyboard GE 98614 USB dongle (USB ID 05b8:3245)	Link	Link
Hewlett-Packard	HP Wireless Classic Desktop wireless keyboard HP Wireless Classic USB dongle (USB ID 3938:1032)	Link	
Insignia	Wireless Keyboard NS-PNC5011 USB dongle (USB ID 3938:1032)	Link	
Kensington	Kensington ProFit Wireless Keyboard Kensington USB dongle (USB ID 062a:4101)	Link	Link
Radio Shack	RadioShack Slim 2.4GHz Wireless Keyboard RadioShack USB dongle (USB ID 062a:4101)	Link	
Toshiba	Toshiba PA3871U-1ETB wireless keyboard Toshiba PA3844D USB dongle (USB ID 0458:00ce)	Link	



WITH secure

But wait!

Vendor	Affected Devices
Anker	Anker Ultra Slim 2.4GHz Wireless Compact Keyboard Anker USB dongle (USB ID 062a:4101)
EagleTec	EagleTec K104 / KS04 2.4 GHz Wireless Combo keyboard EagleTec USB dongle (USB ID 062a:4101)
General Electric	GE 98614 wireless keyboard GE 98614 USB dongle (USB ID 05b8:3245)
Hewlett-Packard	HP Wireless Classic Desktop wireless keyboard HP Wireless Classic USB dongle (USB ID 3938:1032)
Insignia	Wireless Keyboard NS-PNC5011 USB dongle (USB ID 3938:1032)
Kensington	Kensington ProFit Wireless Keyboard Kensington USB dongle (USB ID 062a:4101)
Radio Shack	RadioShack Slim 2.4GHz Wireless Keyboard RadioShack USB dongle (USB ID 062a:4101)
Toshiba	Toshiba PA3871U-1ETB wireless keyboard Toshiba PA3844D USB dongle (USB ID 0458:00ce)



OK, well, let's test it

- KeySniffer's GitHub repo gives us a handy tool to detect affected devices
- Sure enough, it detects our keyboard

```
milosz@cruz-missile:~/keysniffer/tools$ sudo ./mosart-device-discovery.py
[2022-11-30 16:17:43.393]  MOSART dongle found on channel 30 with address 4B:78:38:8C
```

- So, we've cracked the case!
- And now we know it's a MosArt dongle
- We're finally getting somewhere!
- ...right?

OK, well, let's test it

- Bastille didn't just give us a working sniffer script
- But Marc Newlin did run a DEFCON talk in which they explained the entire protocol
  - https://media.defcon.org/DEF%20CON%2024/DEF%20CON%2024%20presentations/DEF%20CON%2024%20-%20Marc-Newlin-MouseJack-Injecting-Keystrokes-Into-Wireless-Mice-WP.pdf
- Good enough!

MOSART Keypress Packet		
Field	**Length**	**Description**
Preamble	2 bytes	AA:AA
Address	4 bytes	
Frame Type	4 bits	0x07
Sequence Number	4 bits	
Key State	1 byte	0x81 (down) or 0x01 (up)
Key Code	1 byte	
CRC	2 bytes	CRC-16 XMODEM
Postamble	1 byte	FF

Table 23: MOSART Keypress Packet

An 'a' keystroke is transmitted over the air in the following format:

```
AA:AA:AE:DD:D4:E8:23:DB:48:19:06:FF // 'a' key down
AA:AA:AE:DD:D4:E8:20:5B:48:D1:44:FF // 'a' key up
```

OK, well, let's test it

- So now we just need to edit the detection script to pick out a single known packet. Let's say an "a". Easy enough, since the script gives us enough to view raw packets.
- This will be done in no time!
- ...huh, I'm not seeing anything
- Ok, fine, let's just view raw packets and see if any of them make sense!
- ...huh, none of them *do* make sense
- ...and this keyboard is communicating on multiple channels
  - It's not supposed to do that!
- The packets aren't even the right length!
- Ok, this isn't working.

```
50  35  B4:B2:A0:73:11:22:70:A5:F0:82:0E:BC:BC:0D:2C:F9:3F:8F:BF:34:AC:13:AD:30:A7:6F:7F
50  35  B4:B2:80:11:11:23:70:A5:0E:77:FF:B0:81:73:0E:AD:A2:33:4C:EF:06:F0:E0:0C:80:00:7F:38:F3
50  35  B4:B2:80:F3:11:22:F0:A5:74:B6:F7:01:08:E9:00:B0:11:6D:35:7E:F3:EF:00:F3:F1:07:0F:2F
50  35  B4:B2:A0:73:11:20:70:A7:0A:F4:B7:77:8F:F3:B7:0C:E0:CC:97:70:0C:CD:6C:E0:F6:FB:12
50  35  B4:B2:80:73:10:32:70:A5:0C:F7:08:D4:8C:70:CC:27:80:A4:E0:8E:F6:68:8B:06:A0:89:30:CF
50  35  B4:B2:80:33:10:22:F0:A1:00:EF:2F:3F:75:0D:7E:F7:70:F4:E4:F5:3F:E0:85:F4:7F:8D:0A
50  35  B4:B2:80:F3:91:22:70:E5:12:0C:33:1E:F8:90:F5:4F:70:D0:A7:0E:4C:F8:16:39:70:0B:46
50  35  B4:B2:80:73:11:02:70:A5:87:90:0F:CC:D0:CE:0F:8B:15:0C:8B:F8:40:31:F0:FB:6F:33:3F
50  35  B4:B2:80:73:21:02:78:A5:A7:0F:E8:B5:E4:27:0C:81:0C:FF:33:E1:43:2E:ED:33:E0:F1:30:D1:F7
50  35  B4:B2:80:73:91:22:70:A4:8B:EC:87:18:F3:CE:E9:01:F7:B3:0C:CD:2E:ED:EB:A2:F7:14:89
50  35  B4:B2:80:F3:11:32:30:A7:C1:76:30:F3:67:14:0C:70:AC:B0:F0:9C:CC:CF:11:B1:9C:94:F0
50  35  B4:B2:80:73:31:22:70:A5:B9:30:F4:17:21:48:C7:00:CF:31:CC:AF:0B:DE:3E:D0:CC:F2:F6:6E
50  35  B4:B2:80:73:01:02:70:A5:2D:2F:0F:C3:F6:F3:1C:F3:43:37:33:0F:2F:0B:F7:C3:7F:F2:F7:57
50  35  B4:B2:80:73:11:22:70:A5:27:B3:83:FF:0C:D0:5B:63:3E:2B:00:71:39:0F:F5:F0:B7:F3:CD
50  35  B4:B2:88:73:11:32:70:A5:2B:31:33:B0:F0:AE:C0:C9:C9:EE:EC:8E:A8:B0:F1:FF:4B:5F:AB
50  35  B4:B2:80:73:01:32:70:A5:C4:F0:10:2C:EF:03:18:93:F2:C9:03:3B:FF:90:16:12:F0:48:0F
50  35  B4:B2:82:73:11:22:70:EC:FF:A1:3F:00:C5:4A:48:4A:5E:72:5E:DE:8B:7A:70:DE:D8:5A:08
```

Hmm...

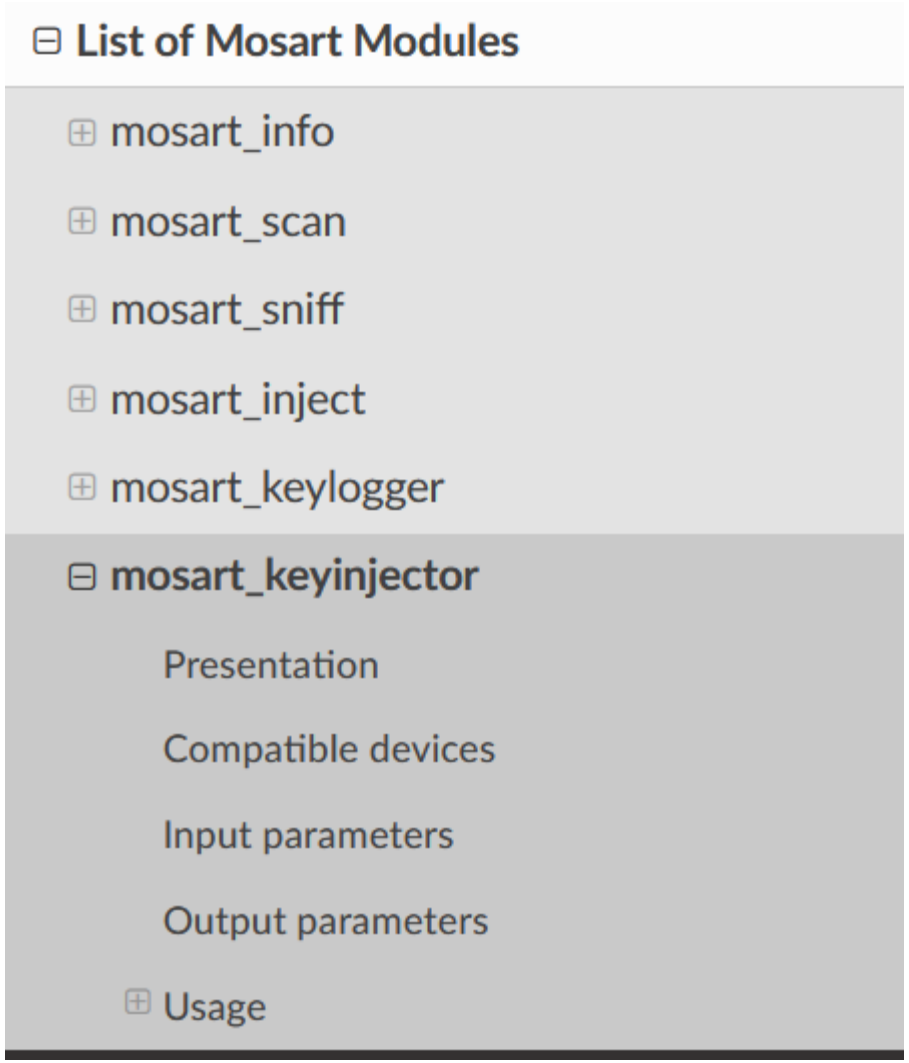
- OK, maybe I'm doing something horribly wrong
- But now I at least know some keywords to search for, has someone written a tool that will do this for me?
- **Yes!** Enter mirage by Romain Cayre
- https://github.com/RCayre/mirage

- **Ok, now we're definitely hackin'**

No luck!

- I'm still not seeing any keystrokes from my keyboard.
- Mirage implementation very similar to my modified script
- And people say it works…
- Clearly there's something we're missing here
- Let's look at other MosArt modules
- `mosart_keyinjector` sounds like fun, I hadn't thought about that before!

https://homepages.laas.fr/rcayre/mirage-documentation/mosartmodules.html



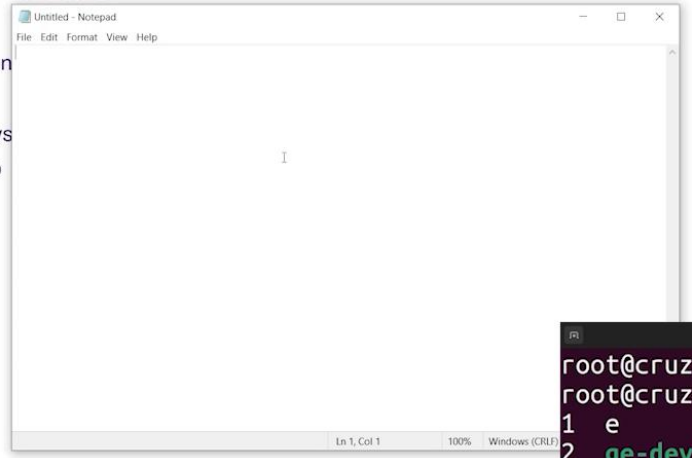
```
milosz@cruz-missile:~/keysniffer/tools$ sudo ./mosart-device-discovery.py
[2022-11-30 16:32:05.828]  MOSART dongle found on channel 30 with address 4B:78:38:8C
milosz@cruz-missile:~/keysniffer/tools$ sudo mirage
```

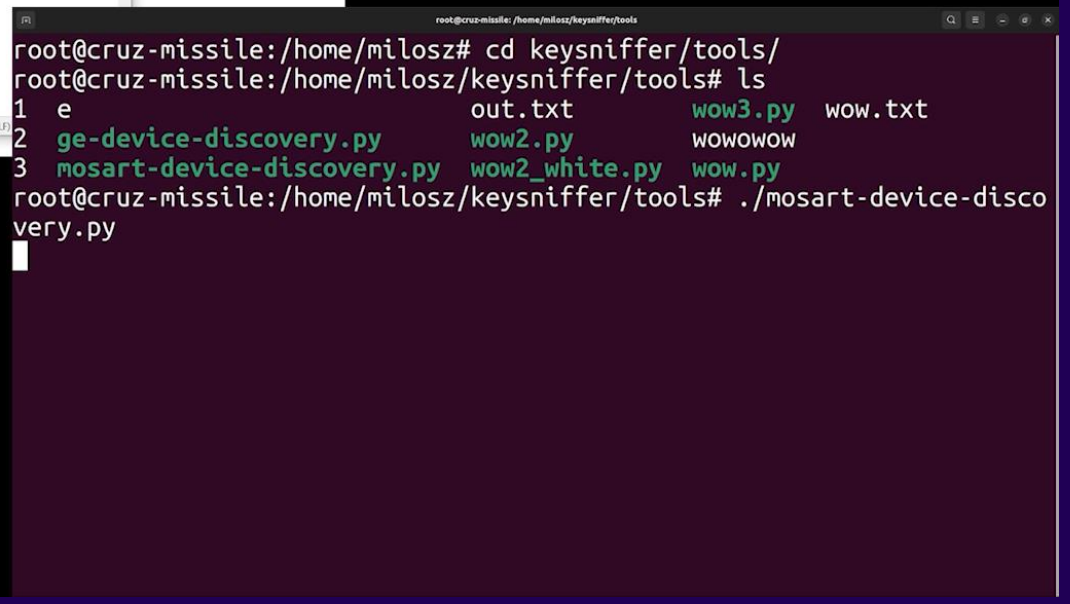

```
 ~~> load mosart_keyinjector
[INFO] Module mosart_keyinjector loaded !
 << mosart_keyinjector >>~~> set TARGET 4B:78:38:8C
 << mosart_keyinjector >>~~> set CHANNEL 30
 << mosart_keyinjector >>~~> run
i[INFO] Injecting:i
 [INFO] Injecting:
l[INFO] Injecting:l
o[INFO] Injecting:o
v[INFO] Injecting:v
e[INFO] Injecting:e
 [INFO] Injecting:
b[INFO] Injecting:b
a[INFO] Injecting:a
n[INFO] Injecting:n
a[INFO] Injecting:a
```

Demo!

nope.java

- mirage is clearly doing what I want it to do
- And people say it works
- Doesn't always detect the don
- No keystrokes though
- A little bit of fun, though – keys
- Let's play with that, it's fun :D





Conclusion

- The dongle can clearly **receive** the old MOSART packets
- But the keyboard is not sending them anymore
- Educated guesses follow:
  - This specific keyboard changed its firmware from the default Chinesium
  - Other vendors have done so, and we never had proper confirmation that the ADVENT keyboard would be vulnerable
  - Maybe it's easy to reverse – I haven't put in the time yet
  - It's possible the attacker (if they are one at all) knows something we don't
  - Injection is fun, and potentially malicious
    - Essentially a remote Rubber Ducky
- We told UoS what we knew, and they agreed that blocking the USB dongle across the estate is a good idea
- A victory, I guess, but it's a little bittersweet

And then I lost the dongle

So I’ll never talk about this to anyone – the end!

WITH secure

Okay, *fine...*

- I do want to talk about this
- Sure, it's incomplete, and the lack of a sniffable keyboard is frustrating, but it's a neat idea
- And not many people are thinking about it
- Guess I'll need a new dongle. So, basically, guess I'll need a new keyboard.
- To Curry's!

ADVENT AKBWL15 Wireless Keyboard

(235) • Ask an owner

£15.99

Flexible credit on orders over £99

Delivery unavailable | Collection

Collection available: RG21 7TZ

Change your location

Your local store:

Basingstoke
Units 5–6 St Michael's Retail Park Winchester Road
Basingstoke RG22 4AZ

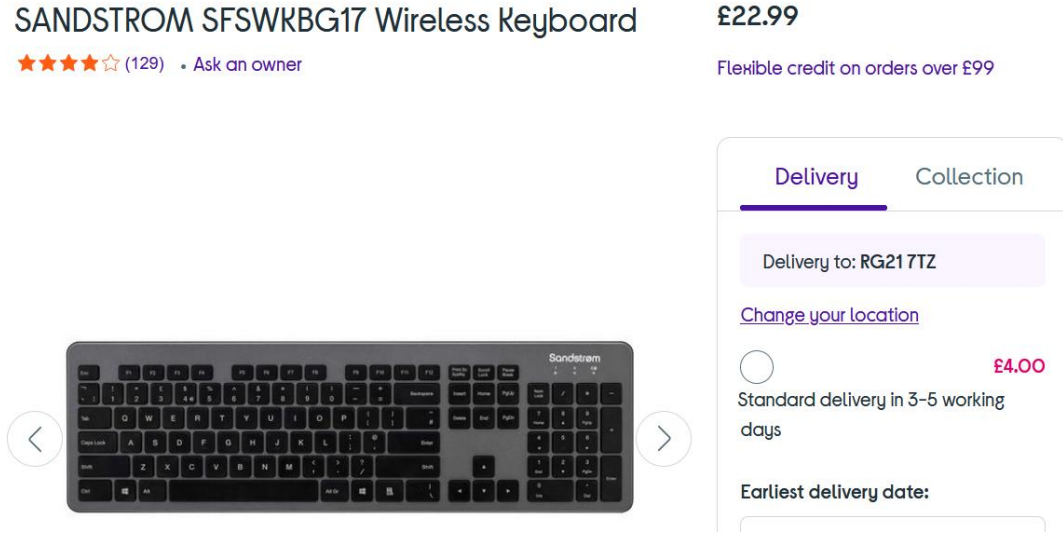
1.4 miles away Store info

Free collection today from store

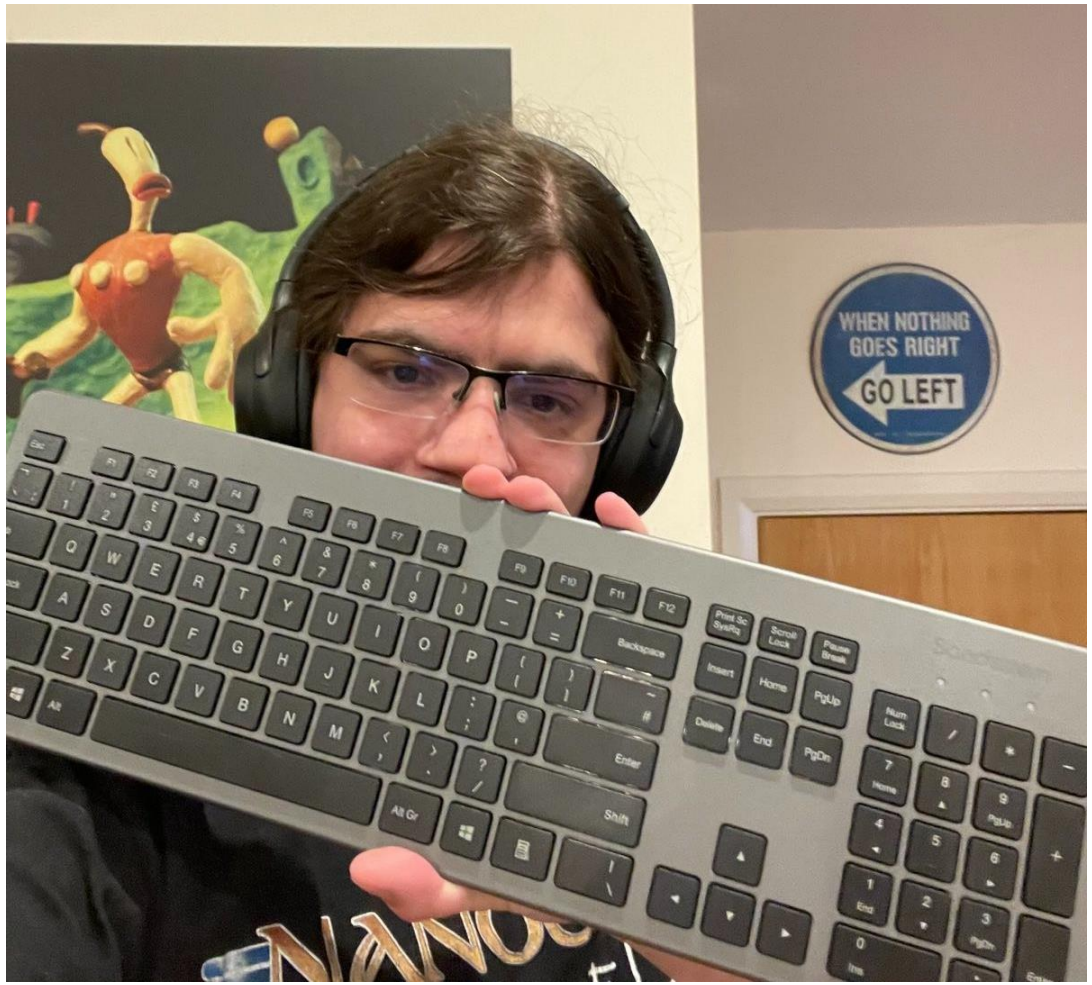
WITH secure

Ah, but if I'm going anyway...

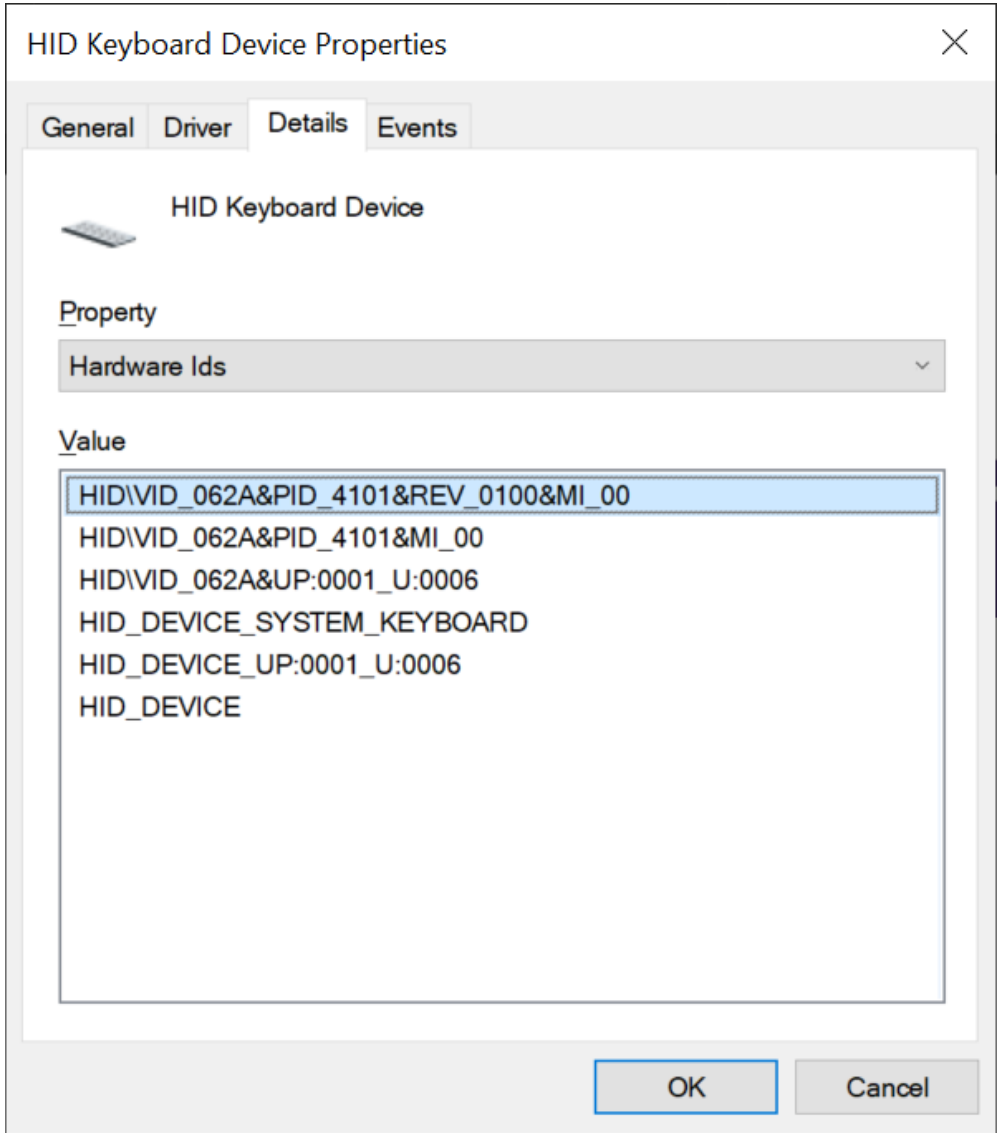
- Can I get more questionable quality keyboards and play with them?
- Kind of. Two candidates meet my criteria:
  - Looks *questionable*
  - I can get it from Basingstoke Curry's
- One gets ruled out because it explicitly says it's encrypted to provide peace of mind
  - Would be fun to look at, but not now
- One… looks strangely perfect
- Whatever, let's drop £23 on a keyboard because I think it ***looks bad***
  - I know nothing about this keyboard
  - Is it even the right protocol?
  - Will it be relevant at all?
  - This all sounds like good financial decision-making



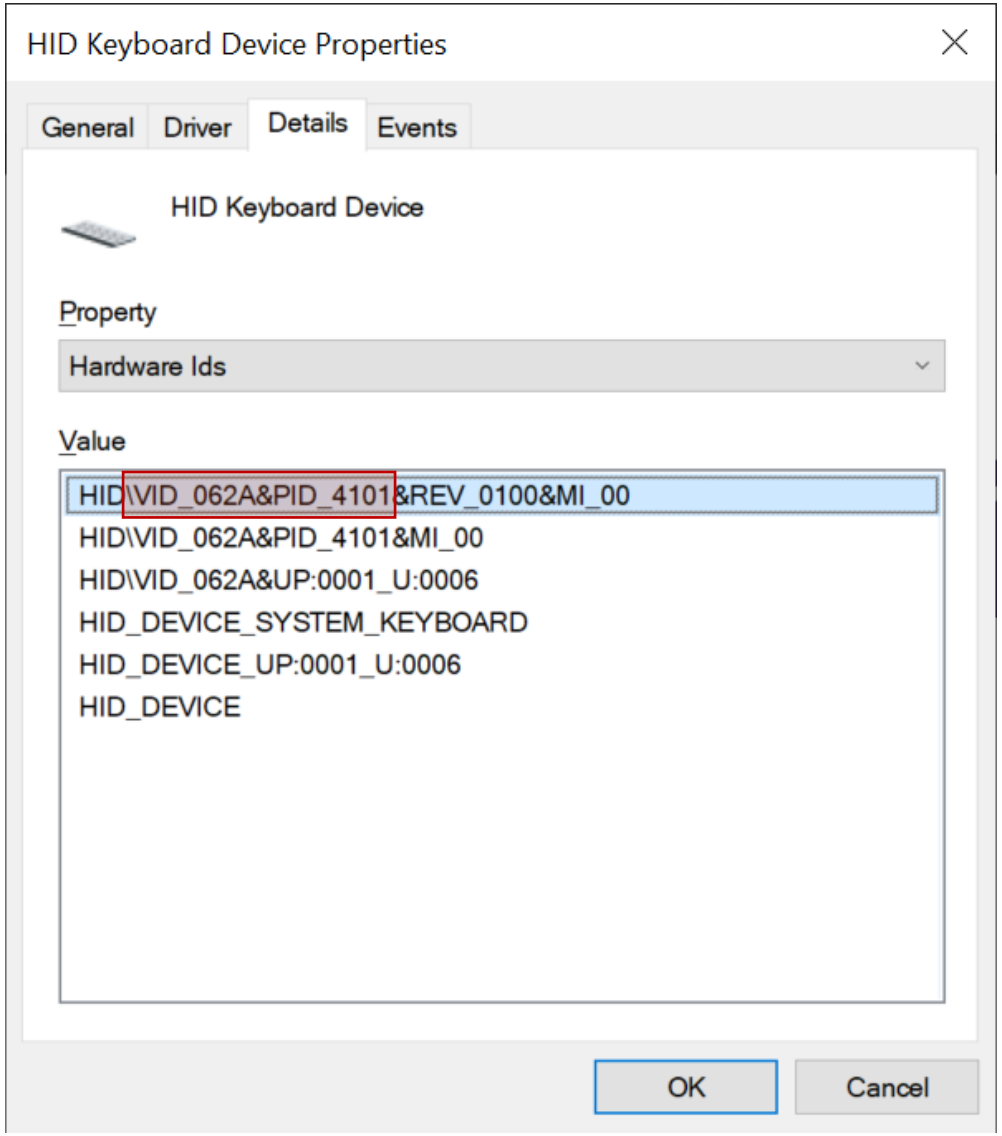
OK I did it!!!!!



Let's plug it in!



oh

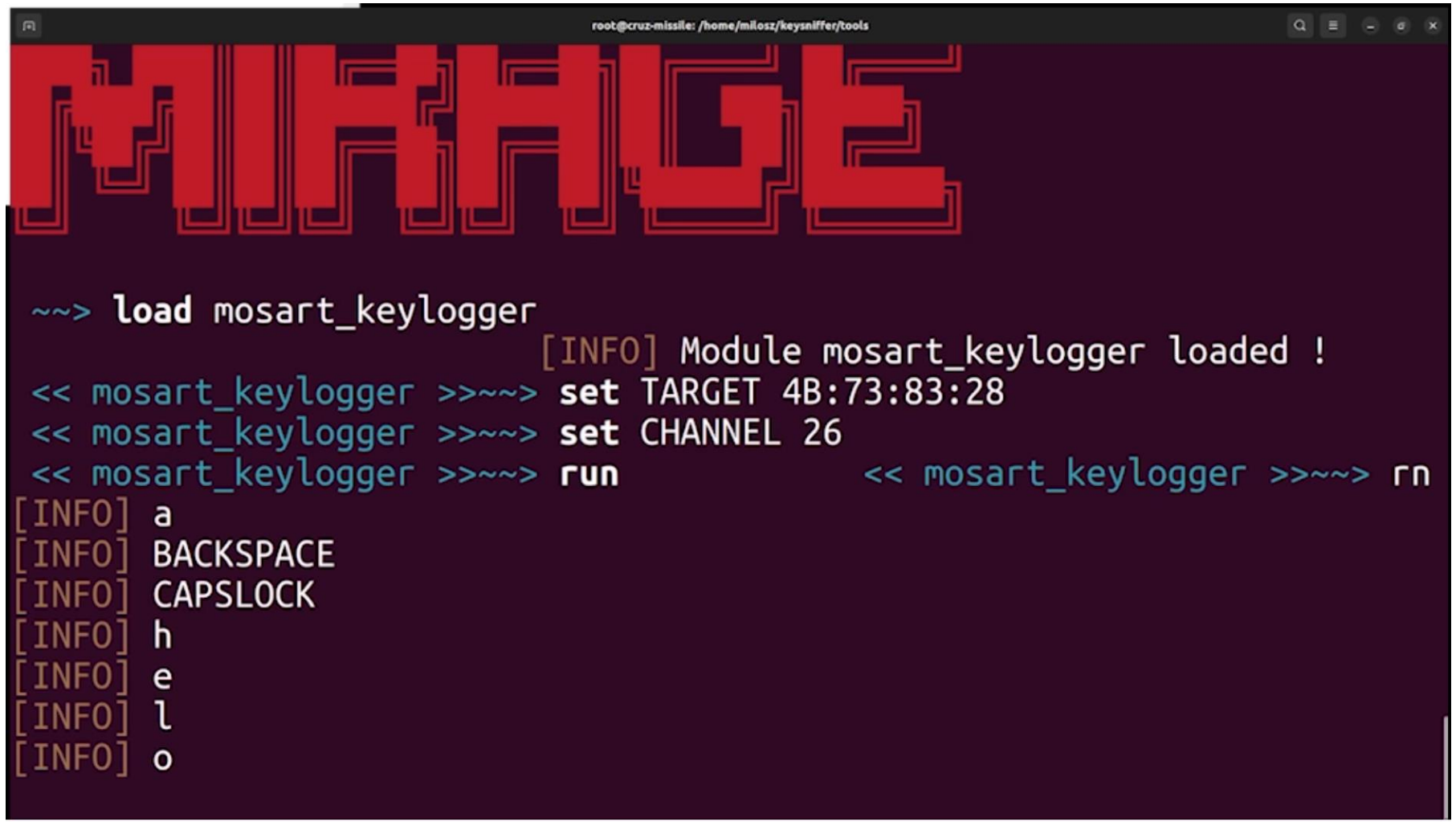


WITH secure

ohhhh

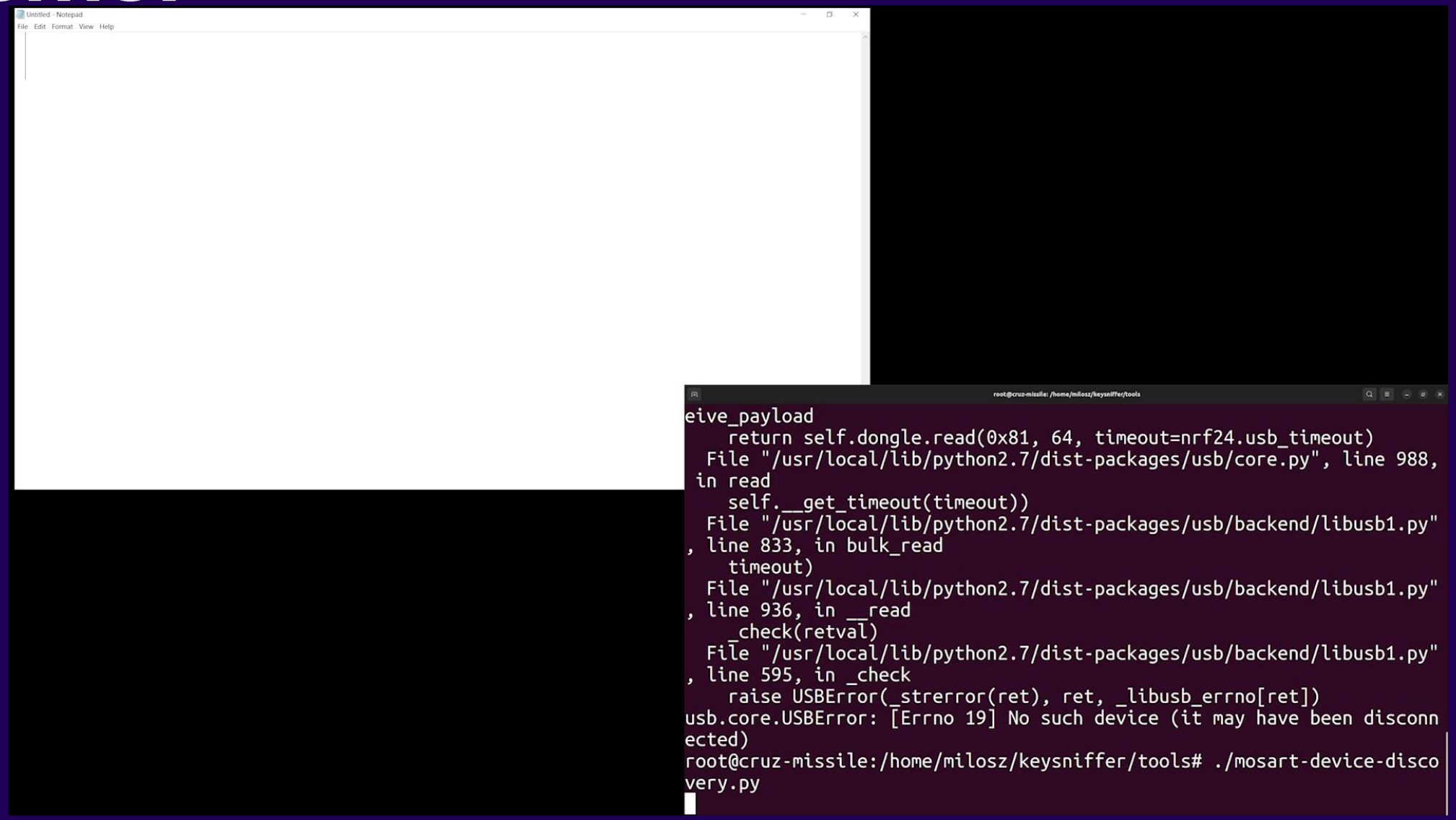
```
milosz@cruz-missile:~/keysniffer/tools$ sudo ./mosart-device-discovery.py
[2022-11-30 16:51:16.463]  MOSART dongle found on channel 26 with address 4B:73:83:28
milosz@cruz-missile:~/keysniffer/tools$
```

OH!


root@cruz-missile: /home/milosz/keysniffer/tools
MIRAGE
~~> load mosart_keylogger
[INFO] Module mosart_keylogger loaded !
<< mosart_keylogger >>~~> set TARGET 4B:73:83:28
<< mosart_keylogger >>~~> set CHANNEL 26
<< mosart_keylogger >>~~> run
<< mosart_keylogger >>~~> rn
[INFO] a
[INFO] BACKSPACE
[INFO] CAPSLOCK
[INFO] h
[INFO] e
[INFO] l
[INFO] o

OH!

Demo!

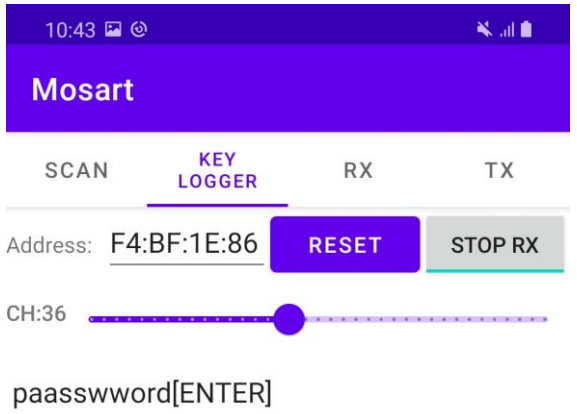

Untitled - Notepad
File Edit Format View Help
root@cruz-missile: /home/milosz/keysniffer/tools
eive_payload
return self.dongle.read(0x81, 64, timeout=nrf24.usb_timeout)
File "/usr/local/lib/python2.7/dist-packages/usb/core.py", line 988,
in read
self.__get_timeout(timeout))
File "/usr/local/lib/python2.7/dist-packages/usb/backend/libusb1.py"
, line 833, in bulk_read
timeout)
File "/usr/local/lib/python2.7/dist-packages/usb/backend/libusb1.py"
, line 936, in __read
_check(retval)
File "/usr/local/lib/python2.7/dist-packages/usb/backend/libusb1.py"
, line 595, in _check
raise USBError(_strerror(ret), ret, _libusb_errno[ret])
usb.core.USBError: [Errno 19] No such device (it may have been disconn
ected)
root@cruz-missile:/home/milosz/keysniffer/tools# ./mosart-device-disco
very.py

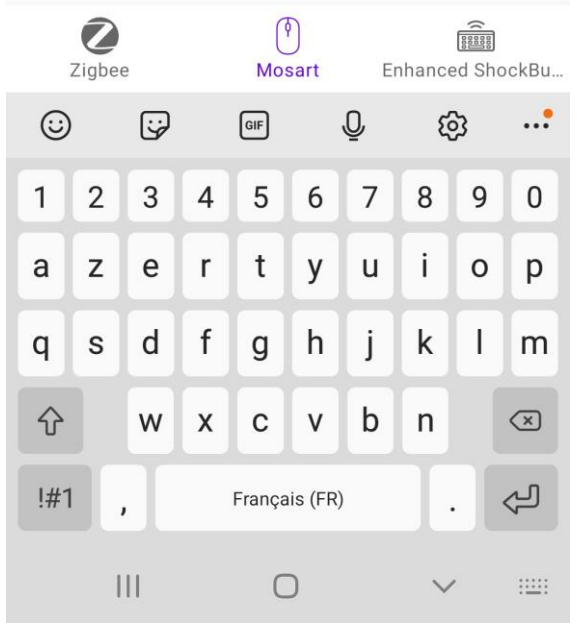
New conclusions!

- OK, so the original keyboard was definitely not doing the thing, it's not just me
- Needs more work to figure out what's up with that
  - Likely encrypted or obfuscated in some way
- BUT you can apparently eyeball a bad keyboard and find one that will work
- That's Not Great™
- Bastille claim 250ft (76m) range on a Crazyradio
- I tried it in large lecture rooms, and I could get it to work with several walls between me and the target, probably about 50m
- Remember: these dongles advertise themselves promiscuously
  - You *could* just start walking around and trying to opportunistically find keyboards to attack

Future work

- Debug mirage – it hates duplicate keystrokes
- Figure out what's up with that other keyboard
- The creator of mirage also made https://github.com/RCayre/radiosploit - a series of patches for the Samsung Galaxy S20's Bluetooth controller that allows sniffing/injection of multiple protocols





WITH secure

That’s it from me!

Twitter: @cyberMilosz

E-mail: milosz.gaczkowski@withsecure.com

WITH secure