Cheating in eSports

How to cheat at virtual cycling using USB hacks

Brad Dixon, Carve Systems





CVRWorldCup

Sweaty eSports

Virtual Cycling is Part of Cycling



Integrate E-Bike as a new discipline

The ongoing digital revolution has totally transformed our lifestyles, including the way in which we play sport. eSports have made their appearance, and the IOC has chosen to take an interest in this new form of competition, which represents a whole new concept in sport. Cycling has a significant advantage over other sports in that connected cyclists can engage in both real and virtual sport at the same time. In response to this social phenomenon, the UCI must be at the forefront of the International Federation movement and create a structure for E-Bike.

- Draw conclusions from the process of reflection engaged in by the IOC on the subject of eSports (2019-2022).
- Bring together, structure and organise E-Bike activities (2018-2022).
- Organise the UCI E-Bike World Championships and national and continental championships (2019).
- Promote the inclusion of E-Bike on the Olympic programme (2024).

 $\bigcirc \bigcirc \bigcirc \bigcirc$



DATE

21 JUL 2018

NEWS, IOC

TAGS OLYMPIC





OLYMPIC MOVEMENT, ESPORTS AND GAMING COMMUNITIES MEET AT THE ESPORTS FORUM



CHEATERS



Cycling: Over 100 Years of Cheating Innovation





1903, 1904: Hippolyte Aucouturier

1904: Maurice Garin



1947: Jean Robic

Cyclist banned for six years after racing with a hidden motor

Femke Van den Driessche has been fined \$20,569 too.

CBS News / CBS Evening News / CBS This Morning /

44 🕨 🕨

60 MINUTES EPISODES

60 MINUTES INVESTIGATES HIDE MOTORS AND PRO CYCLING

Bill Whitaker investigates whether pro cyclists have used motors to win races -- like the Tour de France -- in a spor culture of cheating

2017 JAN 29

CORRESPONDENT **BILL WHITAKER**

9.5 °℃

FACEBOOK

TWITTER y

REDDIT 5

FLIPBOARD



New York Times

Lance Armstrong Stripped of 7 Tour de France Titles

Lance Armstrong Is Stripped of His 7 Tour de France Titles ... McQuaid said that Armstrong's teams had a "win at all costs" attitude fueled by ... Armstrong's case and will probably strip him of the bronze medal he wor In-Depth · Oct 22, 2012





Will people cheat at virtual cycling, too?



"This is a sport with literally hundreds of dollars on the line, and dozens of fans...the stakes are medium!"

Marty Hass -- Tour de Pharmacy, 2017, HBO

No... Marty Hass is not a real person. Don't you recognize Jeff Goldblum? It is a silly mockumentary. Have a laugh.

Virtual Cycling: How does this work?

- Just like any MMPOG plus
- Sensors to measure real world performance
- App-controlled resistance





Bike Radar: Best Cycling Smart Trainers - 10-Way Mega-Test



Speed Estimation

- Course terrain model
- Power
- Rider mass
- Drafting model?



The Easy Way to Cheat at Virtual Cycling

given the same power...

- Lighter riders go faster
- Shorter riders draft better
- ...there are limits!

ZWIFT e-Racing Performance Limits (men, watts/kg)



Vulnerable Sensor Network





Cheat the Hard Way with USBQ





Hack'in USB ain't new

- Facedancer: *excellent*!
 - Travis Goodspeed (@travisgoodspeed)
 - Sergey Bratus (@sergeybratus)
 - Kate Temkin (@ktemkin)
 - Dominic Spill (@dominicgs)
 - Michael Ossmann (@michaelossmann)
- Hardware Village USB Links: Andrey Konovalov
- Glenn

USB Reverse Engineering: Down the Rabbit Hole: Grant "devalias"

Just want to observe USB?

tcpdump + Wireshark

- Requires Linux and the usbmon module. ullet
- **Capture with** tcpdump -i usbmon0 ... \bullet
- Wireshark is great! ullet

	X				📕 usbip.pcap		
	2	🔘 💼 🚺	N 🖸 🤇 🔶 🖷	> 🔮 🖌 👱 其 📕 🤇	Ð. Q. Q. 🎹		
📕 usb o	r usbip						
No.		Time	Source	Destination	Protocol	Length	Info
	4	0.004066s	192.168.2.42	192.168.2.45	USBIP	74	Device List Request
	12	0.015196s	192.168.2.45	192.168.2.42	USBIP	70	Device List Response
	21	3.303745s	192.168.2.42	192.168.2.45	USBIP	98	Import Request
	26	3.315538s	192.168.2.45	192.168.2.42	USBIP	378	Import Response
1 10	28	3.551844s	host	1.5.0	USB	114	GET DESCRIPTOR Request DEVICE
	29	3.555704s	1.5.0	host	USB	132	GET DESCRIPTOR Response DEVICE
	31	3.703811s	host	1.5.0	USB	114	GET DESCRIPTOR Request DEVICE
	32	3.707993s	1.5.0	host	USB	132	GET DESCRIPTOR Response DEVICE
	34	3.708255s	host	1.5.0	USB	114	GET DESCRIPTOR Request DEVICE QUALIFIER
1	35	3.711786s	192.168.2.45	192.168.2.42	USBIP	114	URB Response
	36	3.711976s	host	1.5.0	USB	114	GET DESCRIPTOR Request DEVICE QUALIFIER
i i	37	3.717327s	192.168.2.45	192.168.2.42	USBIP	114	URB Response
1	38	3.717466s	host	1.5.0	USB	114	GET DESCRIPTOR Request DEVICE QUALIFIER
1	39	3.720857s	192.168.2.45	192.168.2.42	USBIP	114	URB Response

usbip + Wireshark

- Linux usbip module can export USB devices over TCP.
- Capture TCP, observe in Wireshark. ightarrow

Stuff Brad Knows

USB Device Drivers and Kernel Code

- Emulate USB host or device functions at the ightarrowlowest level.
- Behave badly and deviate from the expectations of USB drivers.
- Use GoodFET-based board and Facedancer! ullet





USBiguitous by Benoît Camredon

- USB 2.0 MITM using loadable kernel module
- Beaglebone Black
- Python 2 userspace
- usbq core
- usbg userland



USBiquitous: USB intrusion toolkit

Benoît Camredon

benoit.camredon@airbus.com

Airbus Group

Abstract. The USBiquitous project is a set of open source tools to AUSTRACE. The USDIQUIOUS project is a set of open source woos we interact with USB communications. It is composed of a hardware part embedding a Linux system with a bespoke kernel module, and a set of userland scripts and libraries, each designed to tackle a specific problem user and scripts and noraries, each designed to tackle a specific problem linked to USB communications. Emulating a USB host, device, or simply performing a map in the middle attack between a host and a device coninked to USB communications. Emulating a USB nost, device, or simply performing a man in the middle attack between a host and a device can performing a man in the middle attack between a nost and then be done with a few lines of code in a userland script.

1 Introduction

USB devices are everywhere: keyboards, mice, USB keys, webcams, WiFi and USB interfaces are appearing on every equipment The situation is not going to change with the

USB interfaces are already widespread. Every ne or more USB plugs, that can be used either to recharge our phones, iPods... or simply to play used to retrieve logs or to update firmware. USB i face and even the most change-averse individuals a

terface is a security attack target and more and more s the robustness of systems using this interface [2,5], latory to improve our USB tools to protect systems

ew, but nevertheless growing attack vector. udits on systems having USB interfaces, we needed to lerstanding of this protocol, and the low level layers of that implement it. On this journey to understanding the we have developed the USBiquitous framework, as a means

as accumulated enough useful features to become a uring audits of systems that include a USB interface.







USBQ Architecture



Stuff Brad Knows

USB Device Drivers and Kernel Code

- Emulate USB host or device functions at the lowest level.
- Behave badly and deviate from the expectations of USB drivers.
- Consider: GreatFET One and Facedancer!



Applications Using USB Peripherals

- Inspect and mangle application-specific payloads transported across a USB bus.
- Use commodity hardware for USB hacking.
- Consider: USBQ





S

usbg version 0.1.0

Usage: usbq [OPTIONS] COMMAND [ARGS]...

USBQ: Python programming framework for monitoring and modifying USB communications.

Options: Enable usbq debug logging. --debug --logfile FILE Logfile for --debug output --trace Trace plugins. Dump USBQ packets to console. --dump --disable-plugin TEXT Disable plugin --enable-plugin TEXT Enable plugin --config FILE Read configuration from FILE. Show this message and exit. --help

Commands:

mitm Man-in-the-Middle USB device to host communications.

Available plugins:

- decode: Decode raw USBQ driver packets to Scapy representation. - encode: Encode raw USBQ driver packets to Scapy representation. hexdump: Display USBQ packet and hexdump of USB payload. - ipython: Start an IPython session so that USBQ can be updated on the fly. - lookfor: look for a specific USB device to appear - pcap: Write a PCAP file containing USB communications. - proxy: Send and receive USB packets from a USBQ proxy device using the usbq_core module. - reload: Monitor usbq_hooks.py file and reload if changed. - usbq_hooks: Optional user-provided hook implementations automatically loaded from from ./usbq_hooks.py

Default config file: /Users/rbdixon/Library/Application Support/usbq/usbq.cfg

USBQ Main Loop



DO Host/Device Packet

- 1. Wait for a packet
- 2. Get the packet
- 3. Decode the packet
- 4. Log the packet
- 5. Modify the packet
- 6. Encode the packet
- 7. Send the packet out

USBQ Plugins – Built with Pluggy



- Defined extension points for plugins to use.
- Plugins can stack and modify the results of plugins lowerdown the stack. LIFO-call order.
- Plugins can be distributed as independent Python packages.

included:

- Get and Send USB packets using the proxy kernel module Decode/Encode packets to a more useful representation Implement convenience features for development

Get Hack'in

- Inspect PCAP
- Modify plugins on-the-fly
- IPython console



Image: Second state st

2124	001	
filter	<\#/>	ē

00

Apply a display

11

USB URF

No.

Time	Source	Destination	Protocol	Length	Info	
0.000000s	host	1.1.0	USB	64	GET DESCRIPTOR Request DEVICE	
0.014017s	1.1.0	host	USB	82	GET DESCRIPTOR Response DEVICE	
0.025664s	host	1.1.0	USB	64	GET DESCRIPTOR Request STRING	
0.036267s	1.1.0	host	USB	66	GET DESCRIPTOR Response STRING	
0.046175s	host	1.1.0	USB	64	GET DESCRIPTOR Request STRING	
0.057984s	1.1.0	host	USB	98	GET DESCRIPTOR Response STRING	
0.068888s	host	1.1.0	USB	64	GET DESCRIPTOR Request STRING	
0.078970s	1.1.0	host	USB	66	GET DESCRIPTOR Response STRING	
0.089533s	host	1.1.0	USB	64	GET DESCRIPTOR Request STRING	
0.099923s	1.1.0	host	USB	112	GET DESCRIPTOR Response STRING	
0.109883s	host	1.1.0	USB	64	GET DESCRIPTOR Request STRING	
0.123019s	1.1.0	host	USB	66	GET DESCRIPTOR Response STRING	
0.135695s	host	1.1.0	USB	64	GET DESCRIPTOR Request STRING	
0.146982s	1.1.0	host	USB	106	GET DESCRIPTOR Response STRING	
5.165322s	host	1.1.0	USB	64	GET DESCRIPTOR Request CONFIGURAT	TION
5.169545s	host	1.1.0	USB	64	GET DESCRIPTOR Request CONFIGURAT	TION
10.185430s	1.1.0	host	USB	73	GET DESCRIPTOR Response CONFIGURA	ATION
10.189054s	1.1.0	host	USB	73	GET DESCRIPTOR Response CONFIGURA	ATION
10.192974s	host	1.1.0	USB	64	GET DESCRIPTOR Request CONFIGURAT	TION
10.196074s	host	1.1.0	USB	64	GET DESCRIPTOR Request CONFIGURAT	TION
10.198678s	host	1.1.0	USB	64	GET DESCRIPTOR Request CONFIGURAT	TION

Frame 6: 98 bytes on wire (784 bits), 98 bytes captured (784 bits)

STRING DESCRIPTOR bLength: 34 bDescriptorType: 0x03 (STRING) bString: ANT USB-m Stick

0|mac:edope \$ usbq --enable-plugin ipython mitm

bq.pm]:	Loading plugins: reload, proxy, pcap, decode, encode, lookfor, ipython, usbq
eload]:	Monitoring usbq_hooks.py for changes.
proxy]:	Device listen to 0.0.0.0:64240
proxy]:	Host send to 10.0.10.90:64241
.pcap]:	Logging packets to PCAP file usb.pcap
okfor]:	Searching for USB device 0fcf:1009
ngine]:	Starting USB processing engine with IPython UI.
ckend]:	Loading KWallet
ckend]:	Loading SecretService
ckend]:	Loading Windows
ckend]:	Loading chainer
ckend]:	Loading macOS
2018, ⁻	15:22:33)
or 'licer	nse' for more information
Interact	tive Python. Type '?' for help.





What is next for USBQ?

- Release: Visit usbq.org
- Need help with / working on:
 - **USBIP support**: Native Linux kernel system for remote USB
 - Device emulation with Function FS
 - Replace USBiguitous kernel module? Need Linux kernel USBIP + Multipoint USB Highspeed Dual-Role Controller (MUSB)
 - GreatFET One: Looks awesome... need to fiddle with it!
 - More plugins and tools





eSports Leet Automatic

Network Cheating Enhancement



EPO Mode



Sustain performance with less effort and more guilt!

Tour de Pharmacy

 Boost your power with a multiplier Make the world flat



Slacker Mode



Why even risk sweating a little?

 Automatic pedal POWER Cruise control with random jitter Terrain-sensitive heart rate and cadence data generation

ELANCE Plugins for USBQ

- Decode ANT+ USB Payload.
- Decode three different ANT+ Payload types: fitness, HRM, and cadence.

8.6.7 Page 25 (0x19) – Specific Trainer/Stationary Bike Data

Data page 25 shall [MD_0006] be transmitted by stationary bikes and trainers. Any optional field that is not used shall [self-verify] be set to the invalid value as stated in Table 8-25. All fields in this message shall [MD_0010] be set as described in Table 8-25.

Table 8-25. Specific Trainer	Data	Page	Forma
------------------------------	------	------	-------

Byte	Description	Length	Value	Units	Range or Rollover
0	Data Page Number	1 Byte	0x19 - Page 25	N/A	N/A
1	Update Event Count	1 Byte	Event counter increments with each information update	N/A	256
2	Instantaneous Cadence 1 Byte Crank cadence – if available Otherwise: 0xFF indicates invalid		RPM	0-254rpm	
3	Accumulated Power LSB	2 Puter	Accumulated power	1 Watt	65536W
4	Accumulated Power MSB	2 Bytes	1-watt resolution		
5	Instantaneous Power LSB		Instantaneous power		
6 (bits 0-3)	Instantaneous Power MSN	1.5 Bytes	0xFFF indicates BOTH the instantaneous and accumulated power fields are invalid	1 Watts	0 - 4094W
6 (bits 4-7)	Trainer Status Bit Field	4 Bits (4:7)	Refer to Table 8-27	N/A	N/A
-	Flags Bit Field	4 Bits (0:3)	Refer to bit field description (8.6.7.6)	N/A	N/A
/	FE State Bit Field	4 Bits (4:7)	Refer to bit field description (8.5.2.7)	N/A	N/A

at

ANT+ Profile Pages

ANT+

USB Host or Device

USBQ Host, Device, or Management

Cheat the Hard Way with USBQ





1	Terminal	Shell	Edit	View	Window	Help
	Ter IIIII ai	Shen	Edit	VIEW	WIIIGOW	. Huip

(python-3.7.0) 0|mac:edope \$ edope slacker

Terminal — -bash — 137×8









Could it work?

1.Workouts

Photo by Simon Connellan on Unsplash

2. Online racing 3. Live event racing

Workouts

- Yeah, go ahead and cheat yourself.
- You'll need to use sensible limits.



UH OH!

Either you've missed your calling as a pro cyclist or your equipment is not set up properly.

Ensure that your power meter or smart trainer is properly calibrated. Your superhuman times will not show up in the leaderboards this ride.

of the World St. or a part of the second state



Online Racing*

- Plausible to *stretch* a mediocre rider into a competitor.
- Use multiple accounts to establish the actual performance limits for verification.
- Build an IRL riding record and a public Strava profile.
- Verification cheats:
 - 2nd power monitor / IRL power mo
 - Either real height + weight or fake
 - Bribe / dodge / fake 3rd party verif

* Never actually tried to cheat in an online race nor applied the techniques listed above.



ZWIFT

Rules and Regulations

Version 1.0.1 - 6/20/2019

onitor		
e videos		
fication	la	b

Power Adjuster 4 Change your power meter data.	Section Remover © Remove selected sections of data from your file.	Device Changer	
« Close	Launch »	Launch »	
Corrup Fix files with	t Time Fixer Peak Ro Corrupt timestamps. Remove high power Launch » Laur	emover or heart rate records.	
Step 1	Step 2	Step 3	Version 1.0.0 Page 1
Drop FIT file here or select files	Select power adjustment percentage	Go!	

Live Event Racing*

- This is harder but live events are rare.
- High-stakes events use equipment provided by race.
- Probably can't fake weigh-in.
- Infiltrate a NSA COTTONMOUTH-I style hacked cable?
- Working on some other techniques, too.

* Never tried this, either. There is no way anyone would believe I'm an elite cyclist. Not even for a second.

CVRWorldCup

(TS//SI//REL) COTTONMOUTH-I (CM-I) is a Universal Serial Bus (USB) hardware implant which will provide a wireless bridge into a target network as well as the ability to load exploit software onto target PCs.

Wrap up

- Overall system not designed for high-integrity competition.
- 2. Insecure sensor networks and untrusted hardware are not a good foundation for security.
- 3. Electronics and software are part of cycling. New domains for cheaters to exploit.

Winners never cheat. Cheaters never win. Hackers sometimes cheat for fun.

edope.bike