Hack-A-Sat 2020 CTF

Cyrus Malekpour

Captain, PFS Capture the Flag Team

Who am I?

- Full-time security engineer, focused on red-teaming software
- Primary focus is on "low-level" binary security and reverse engineering
- Motivating professional interest:

  "How can we improve software by thinking like an attacker?"

What is Capture the Flag (CTF)?

- Capture the Flag is a form of computer security competition

- Teams compete to solve cybersecurity challenges, demonstrating the application of real-world expertise

- Freely run and organized by companies, universities, clubs, and groups worldwide
  - In 2020, the top 10 teams hailed from 9 different countries

CTF then and now

Credit: CNET, Aug. 2006

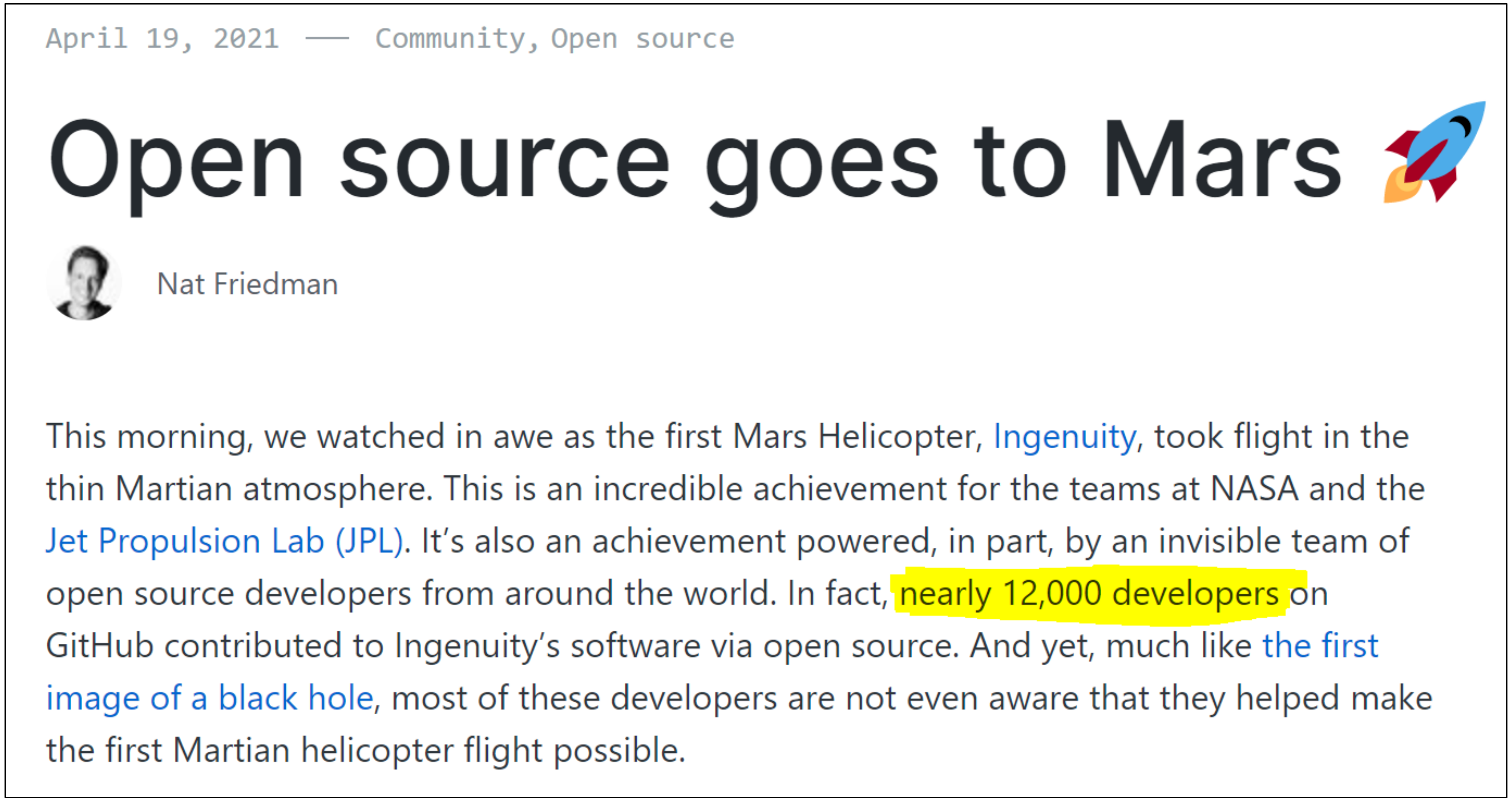
Credit: @RealWorldCTF twitter, May 2020

CTF then and now

Credit: CNET, Aug. 2006

Credit: @RealWorldCTF twitter, May 2020

Space and Software



April 19, 2021 —— Community, Open source

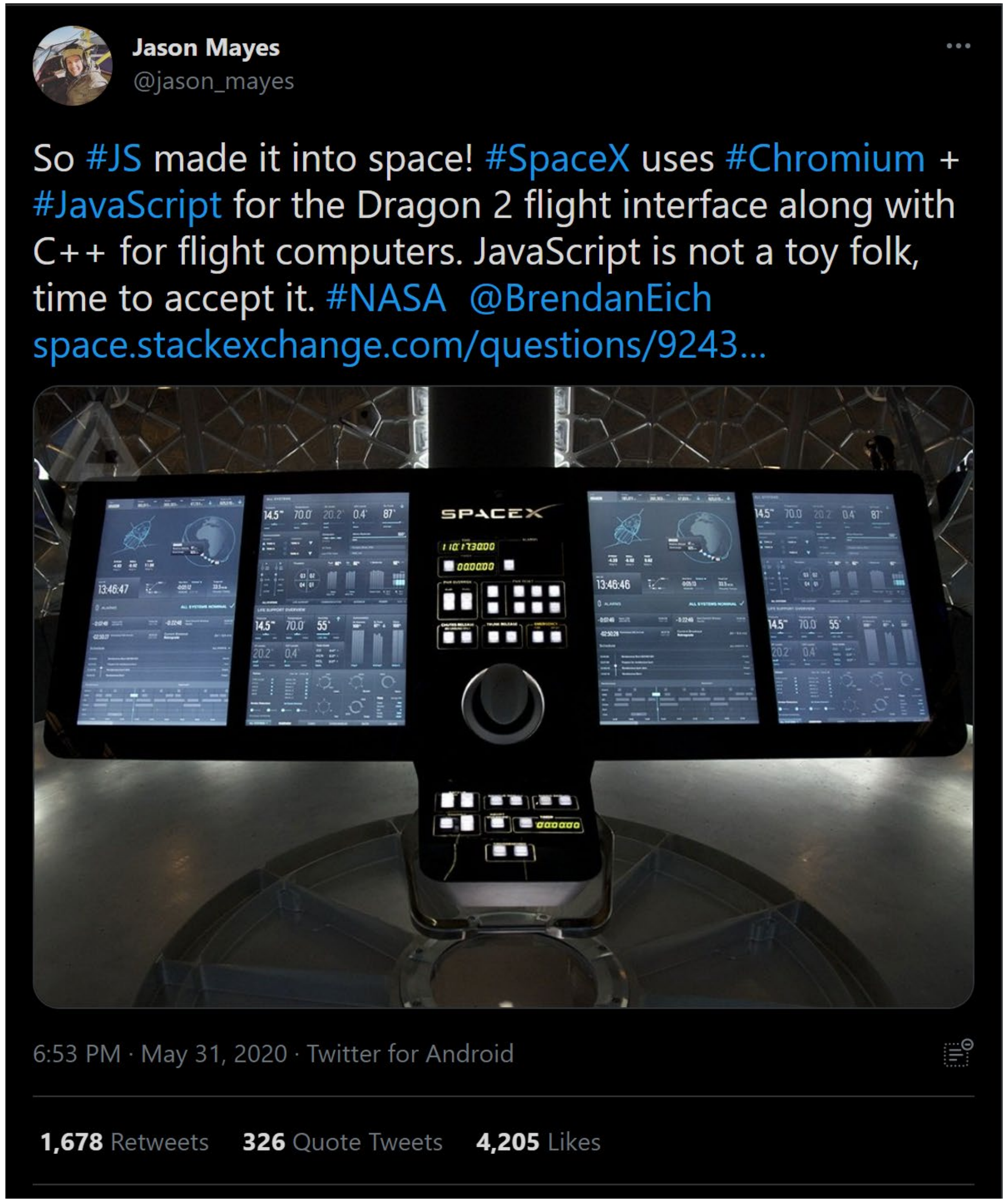
Open source goes to Mars 🚀

Nat Friedman

This morning, we watched in awe as the first Mars Helicopter, Ingenuity, took flight in the thin Martian atmosphere. This is an incredible achievement for the teams at NASA and the Jet Propulsion Lab (JPL). It's also an achievement powered, in part, by an invisible team of open source developers from around the world. In fact, nearly 12,000 developers on GitHub contributed to Ingenuity's software via open source. And yet, much like the first image of a black hole, most of these developers are not even aware that they helped make the first Martian helicopter flight possible.

> "We use **C++** for all vehicle control systems, **Python** for tools, testing and automation, and **Javascript/HTML/CSS** for our displays [...] Our flight systems use a custom **Linux kernel** with the PREEMPT_RT patch."
>
> Jeff Dexter, Dir. Cybersecurity at SpaceX

Next generation of Space ventures can benefit from open source software...

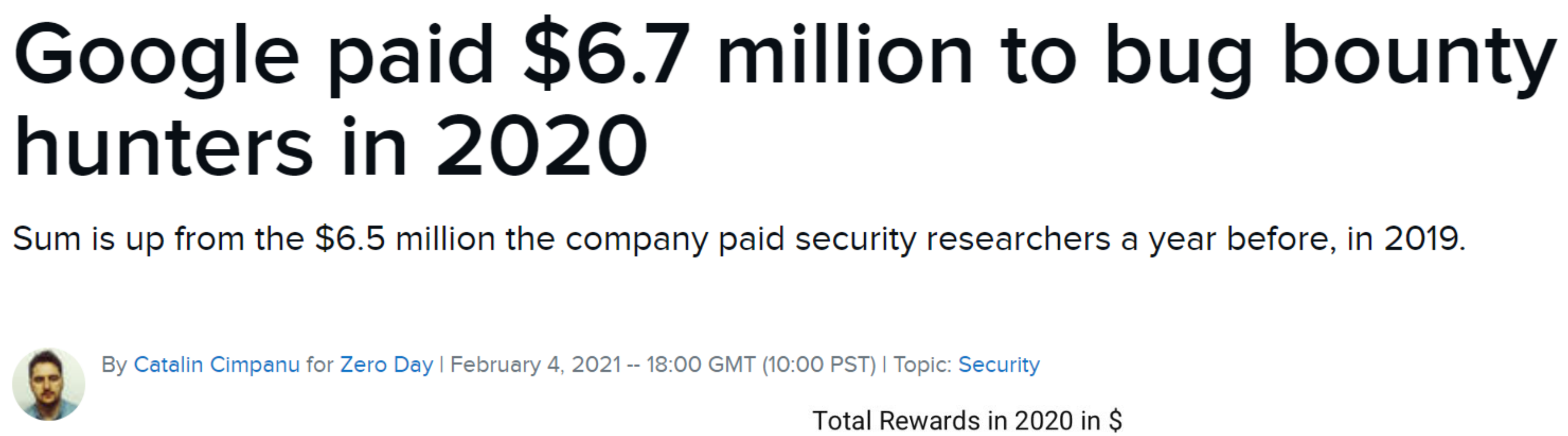


Jason Mayes
@jason_mayes

So #JS made it into space! #SpaceX uses #Chromium + #JavaScript for the Dragon 2 flight interface along with C++ for flight computers. JavaScript is not a toy folk, time to accept it. #NASA @BrendanEich space.stackexchange.com/questions/9243...

6:53 PM · May 31, 2020 · Twitter for Android

1,678 Retweets **326** Quote Tweets **4,205** Likes



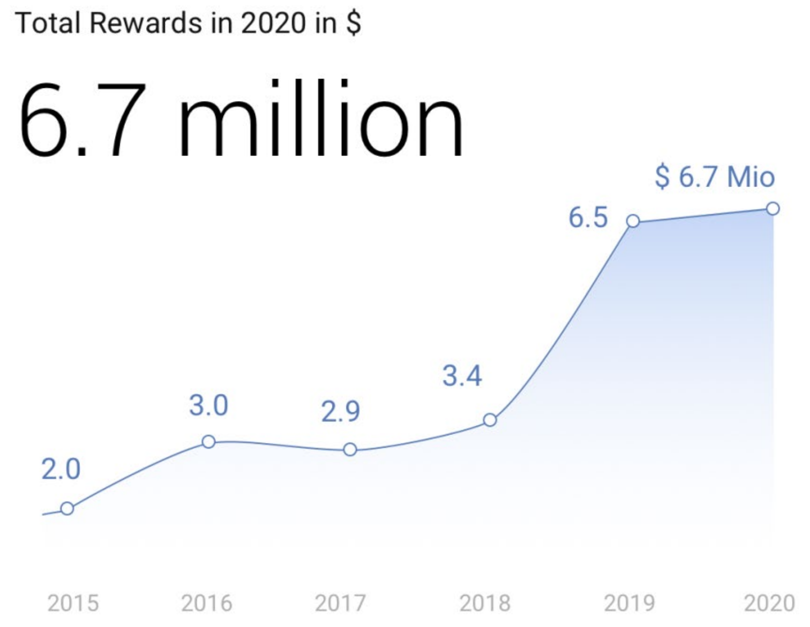
Google paid $6.7 million to bug bounty hunters in 2020

Sum is up from the $6.5 million the company paid security researchers a year before, in 2019.

By Catalin Cimpanu for Zero Day | February 4, 2021 -- 18:00 GMT (10:00 PST) | Topic: Security

> **Most of last year's bug prizes were awarded in the Chrome VRP** [...] more than $2.1 million to security researchers for **300 bugs** identified in Google's flagship browser.

Credit: ZDNet



Credit: Google

Past Events

DARPA CGC (2015-2016)

- DARPA Grand Challenge to create first "automatic defense system" to discover/prove/correct flaws
- Significant research impact and big advance in publicly available security tooling
- Very well publicized in the press; raised a lot of awareness

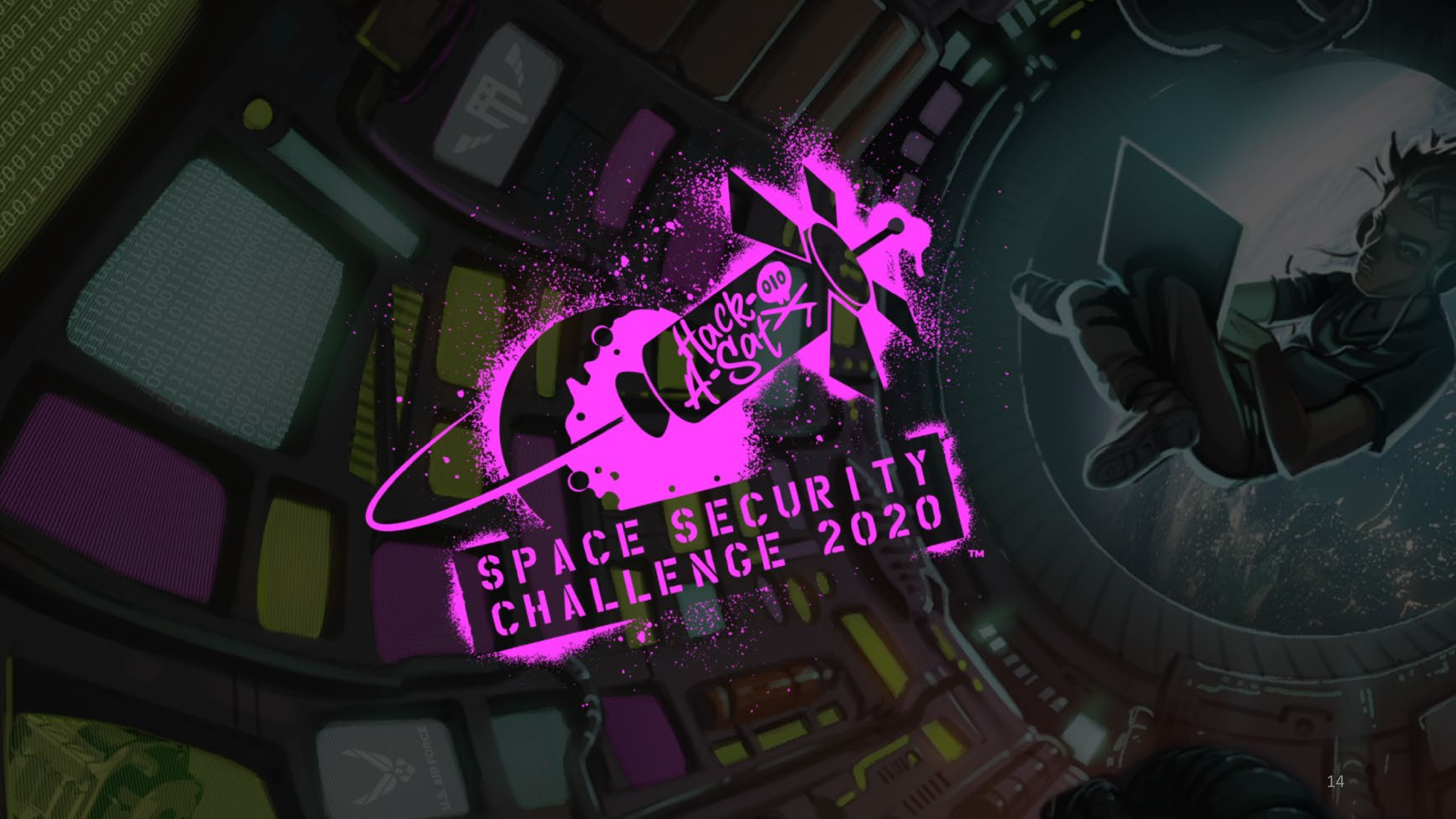
Hack the Air Force (2016-)

- Bug bounty program to audit public-facing Air Force websites
- 5+ years running; 1000+ vulnerabilities identified and fixed
- Important milestone in allowing for community security review of AF systems

(Potential) motivating questions

1. What barriers prevent the security community from engaging with space security research?

2. How can we raise awareness of cybersecurity in space, both in industry and in government?


Hack-A-Sat
SPACE SECURITY CHALLENGE 2020™

Hack-A-Sat CTF

- Joint event by Defense Digital Service + Air Force Research Lab
- Give researchers access to satellite systems they couldn't otherwise play around with
- Get hackers together in one place, give them access to hardware, and help raise a lot of awareness

BRIAN BARRETT SECURITY 09.17.2019 07:00 AM

The Air Force Will Let Hackers Try to Hijack an Orbiting Satellite

At the Defcon hacking conference next year, the Air Force will bring a satellite for fun and glory.

Competition structure

1. **Online Qualifiers:** Open to any team, "jeopardy style", occurring over a 48 hour period

2. **Preparation time:** Teams train + develop tooling in order to attack and defend satellite hardware

3. **Finals Competition:** On-site finals with flatsat devices, intended as a spectator showcase as teams compete to gain control. Includes an in-orbit component on an active satellite

Our qualifier experience

- 48 hour online qualifiers, Friday night to Sunday night, across a wide range of challenge categories

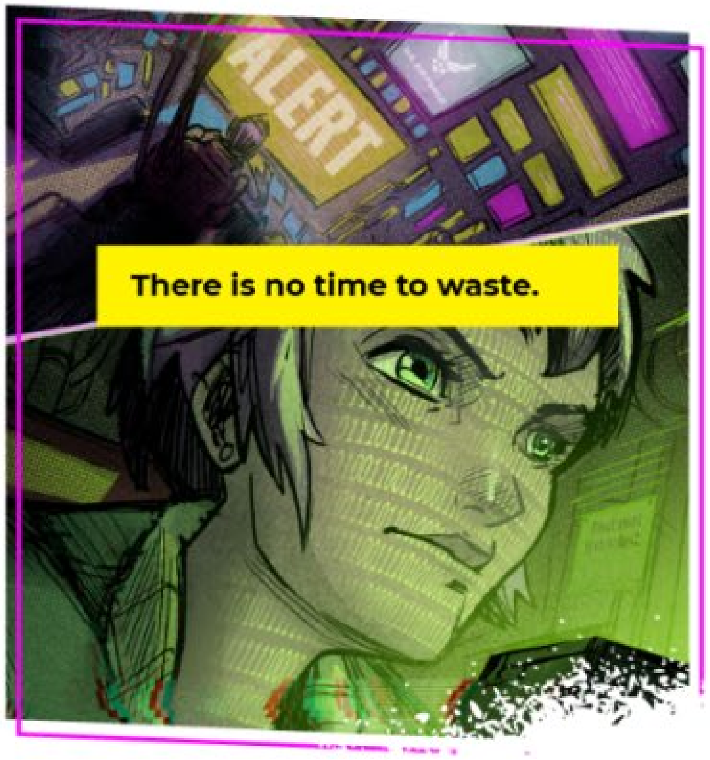
- Harder challenges are worth more points than easier challenges, but points decay over time as more teams solve
  - Teams must demonstrate ability to solve hardest challenges to retain points
  - Let's be strategic! Keep an eye on how many points each challenge is worth

- ~1300 teams solved at least one challenge, 8 slots in the finals

Breadth of knowledge required

- Hack-a-Sat Qualifier challenges covered a lot of ground with 30+ challenges released
- Topics included RF, low-level debugging, systems familiarity (KubOS/COSMOS), reverse engineering, and orbital mechanics
- Bottom line: the most important skill was the ability to **learn quickly**

We need your help to gain control of a rogue satellite.


ALERT
There is no time to waste.

We need your skill, creativity and goodwill to hack into its system and turn the camera gimbals to take a moon shot.

The plan for finals

- 8 teams qualified to compete on-site at Caesar's Palace in Las Vegas
  - Planned to occur during the annual DEF CON Conference
- Each team is tasked with gaining remote access to the satellite (1), regaining control of the sat (2), removing all traces of the attacker (3), and reorienting it for a "Moonshot" (4)
- "Moonshot" plans would be sent to an active satellite and include specific flight plans to capture a real picture of the Moon

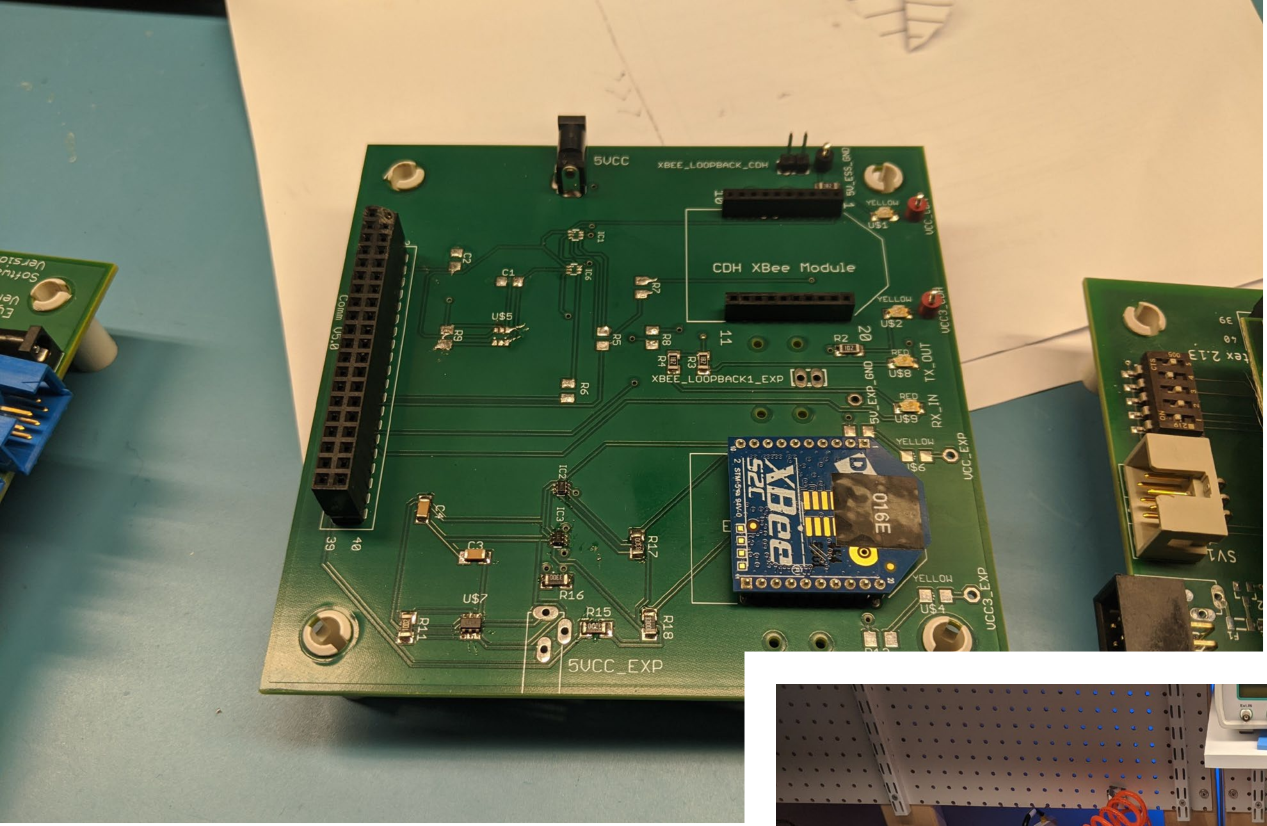
COVID update: Satellite@Home

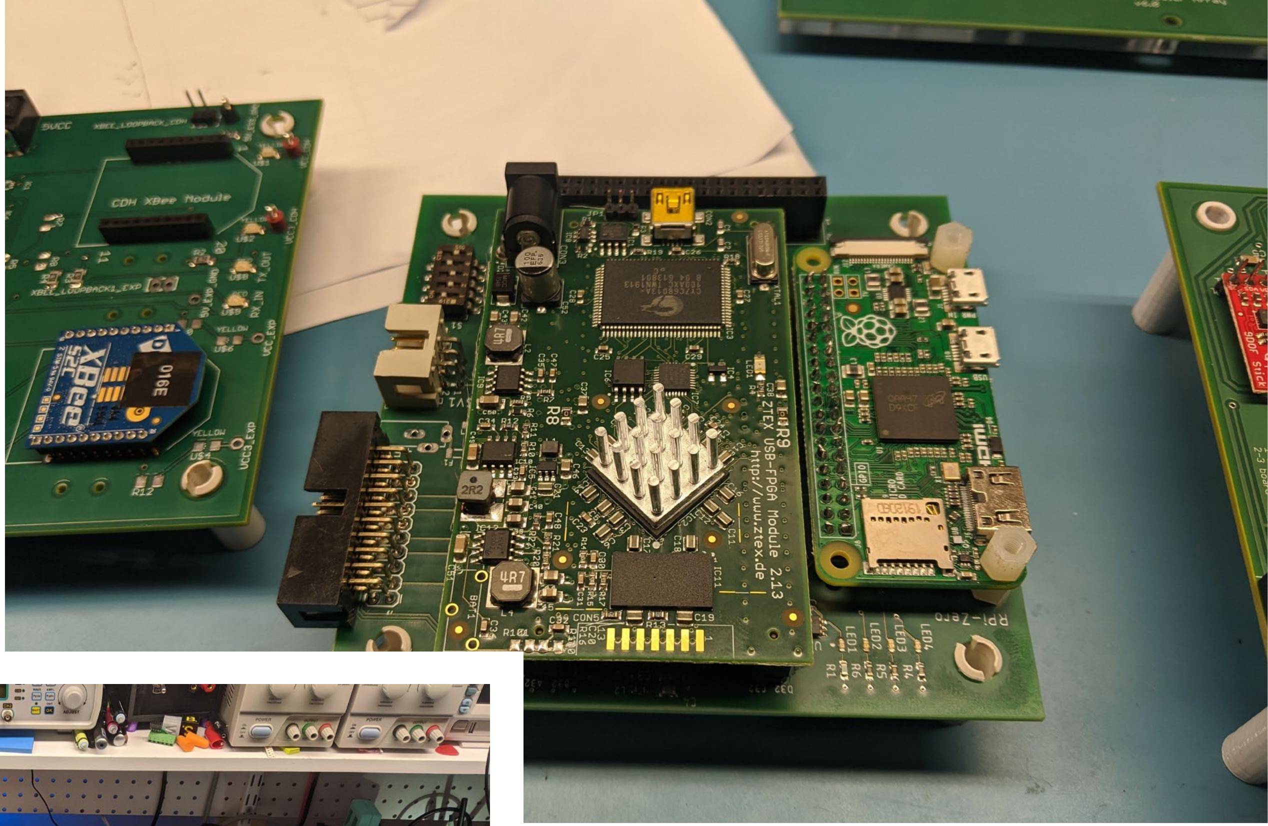
- Because of COVID, the competition had to be retooled to work remotely
- Decision was made to send each team their own Eyessat flatsat device, which the competition would be based on
- In parallel, teams could test on their local device while organizers ran the "live" hardware

Our preparation

- We immediately fully disassembled it and studied everything
  - Dumped all software from the device
  - Carefully went over all documentation for hardware/software
  - Developed some in-house tooling to aid software reverse engineering

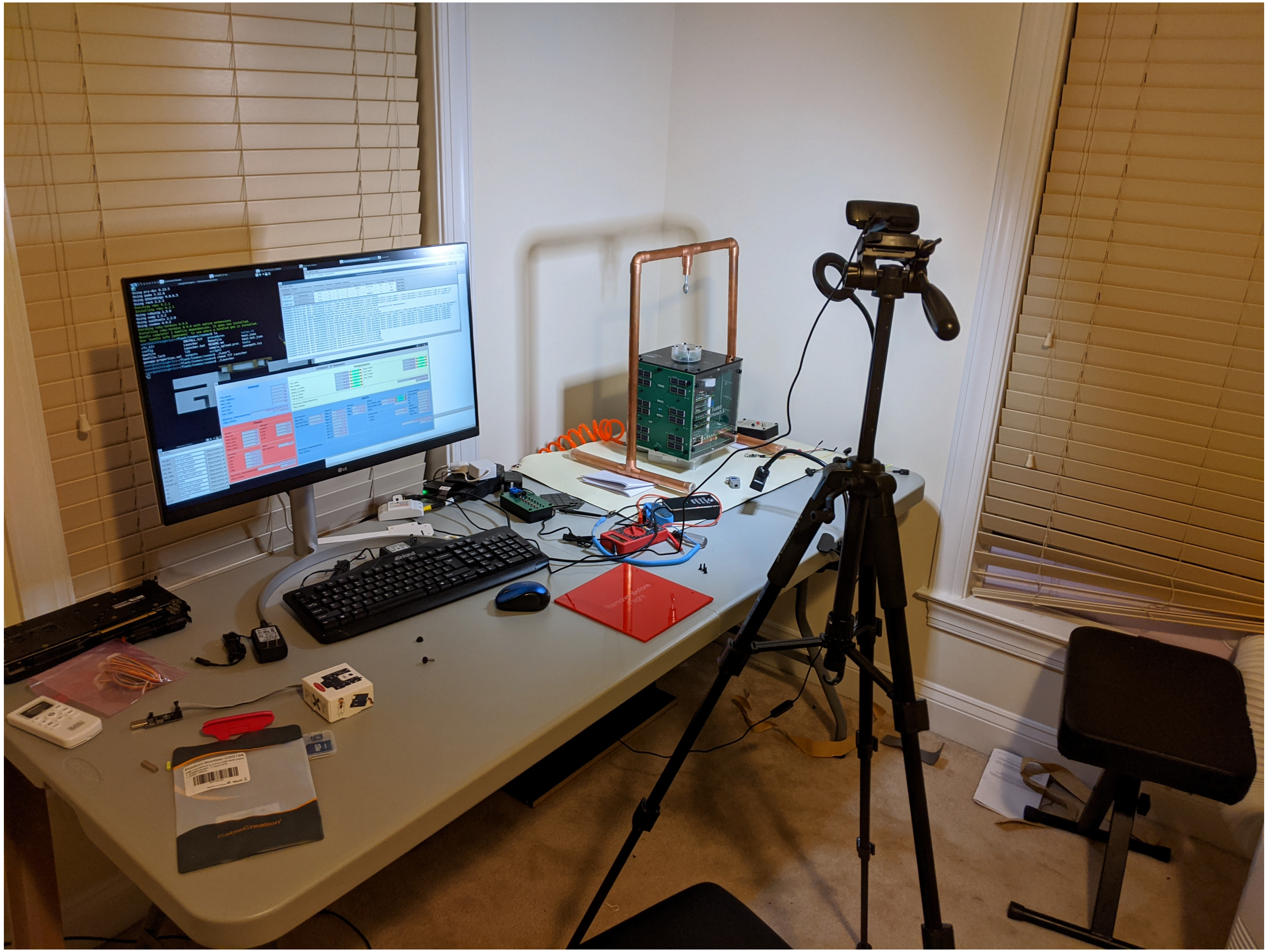
- 1:1 copy of satellites run by CTF organizers on-site
  - So we can enable all the debug features we want on our own device :)


5VCC
XBEE_LOOPBACK_CDH
CDH XBee Module
XBEE_LOOPBACK1_EXP
5VCC_EXP
XBee
016E


CDH XBee Module
ZTEX USB-FPGA Module 2.13
http://www.ztex.de
RPi-Zero

ATMEL
ATMEGA12
16U-TH
35454U

2005D
The Classroom Satellite
CableCreation
Employees must wash hands before returning to libc.
cromulence


Remove Before Flight

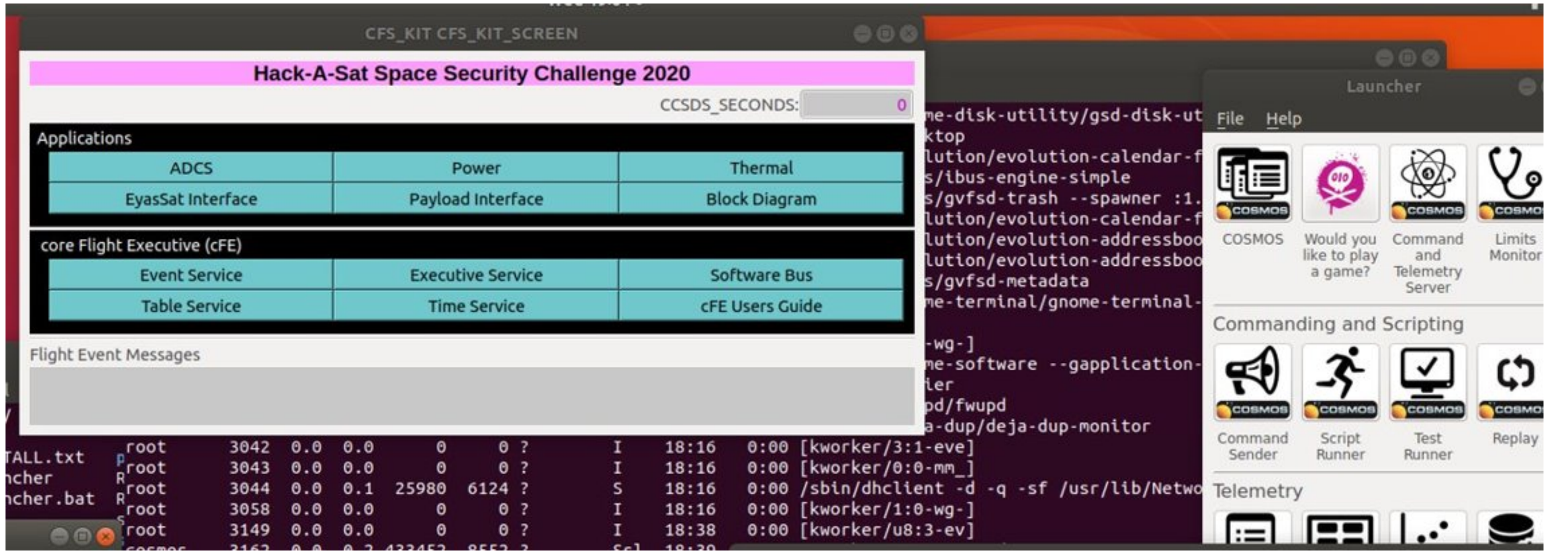
Playing finals

- 2 days of competition (Friday-Saturday) with 7 active hours each day
- On Friday night, teams could continue working with their local satellite but had to wait until the morning to resume play
- Challenges were sequential; have to solve each stage to proceed
- First solver gets max points, afterwards decay with time

Finals, Day 1

Regain Control

1. Gain access to the "ground station" via web exploitation
2. Use ground controls to stablilize satellite with its torque rods
3. Develop our moonshot plan!



```
me-disk-utility/gsd-disk-ut
ktop
lution/evolution-calendar-f
s/ibus-engine-simple
s/gvfsd-trash --spawner :1.
lution/evolution-calendar-f
lution/evolution-addressboo
lution/evolution-addressboo
s/gvfsd-metadata
me-terminal/gnome-terminal-

-wg-]
me-software --gapplication-
ier
pd/fwupd
a-dup/deja-dup-monitor
```

```
          root      3042  0.0  0.0      0     0 ?      I    18:16   0:00 [kworker/3:1-eve]
TALL.txt  root      3043  0.0  0.0      0     0 ?      I    18:16   0:00 [kworker/0:0-mm_]
ncher     root      3044  0.0  0.1  25980  6124 ?      S    18:16   0:00 /sbin/dhclient -d -q -sf /usr/lib/Netwo
ncher.bat root      3058  0.0  0.0      0     0 ?      I    18:16   0:00 [kworker/1:0-wg-]
          root      3149  0.0  0.0      0     0 ?      I    18:38   0:00 [kworker/u8:3-ev]
```

SPACE SECURITY
CHALLENGE 2020
Sat 08/08/2020 14:31:30 ET

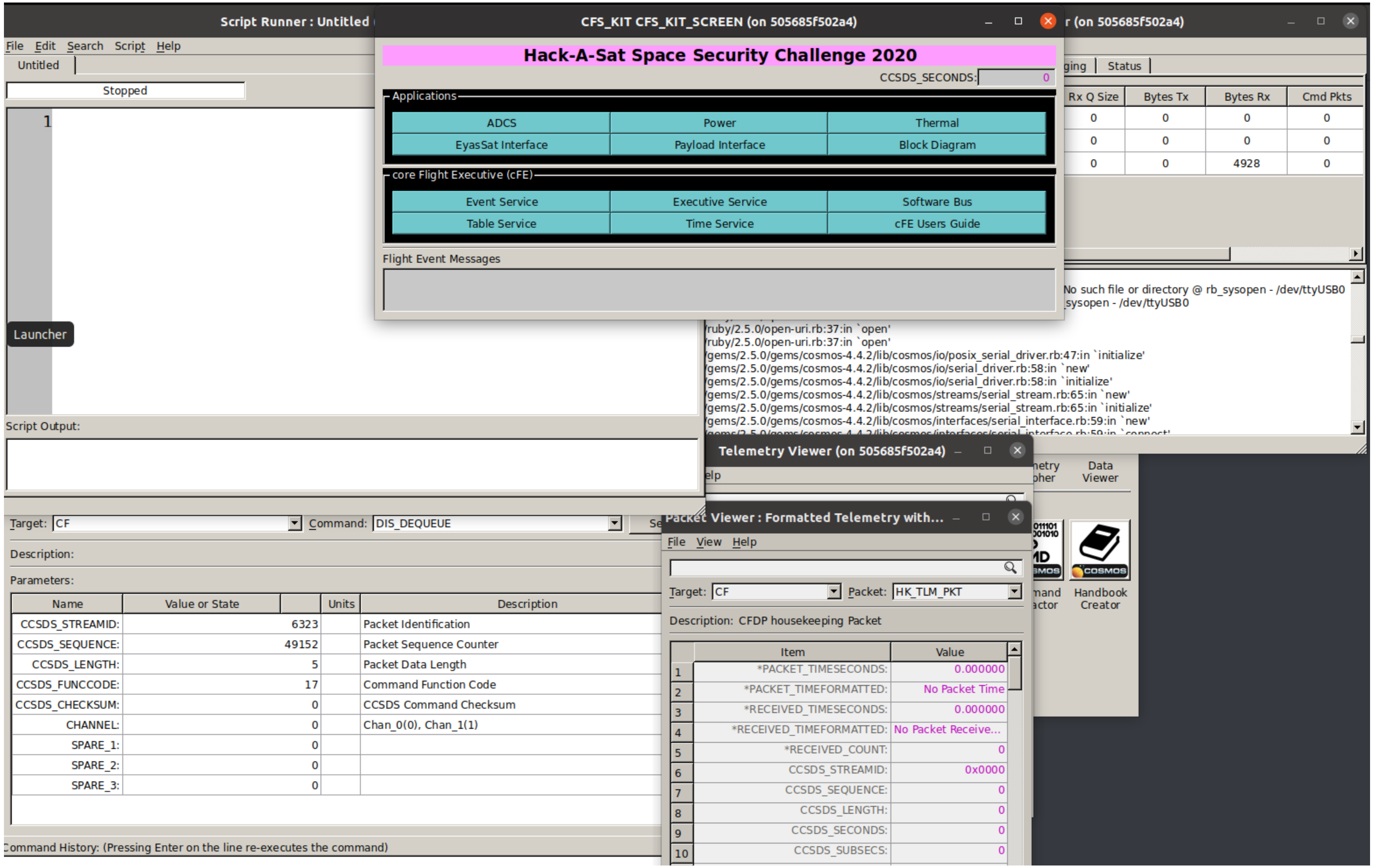
Finals, Day 2

Remove the Attacker

1. Analyze binary code of malicious attacker module to identify the backdoor
2. Use the attacker's backdoor to purge them from the system

Complete our mission

1. Write software for unknown peripheral to operate on-board camera
2. Direct the satellite to capture a picture of an object placed by the organizers in the room


Script Runner : Untitled
File Edit Search Script Help
Untitled
Stopped
Launcher
CFS_KIT CFS_KIT_SCREEN (on 505685f502a4)
r (on 505685f502a4)
ging
Status
Telemetry Viewer (on 505685f502a4)
elp
netry
pher
Data Viewer
mand
actor
Handbook Creator
Packet Viewer : Formatted Telemetry with...
File View Help

Hack-A-Sat Space Security Challenge 2020

CCSDS_SECONDS: 0

Applications

ADCS	Power	Thermal
EyasSat Interface	Payload Interface	Block Diagram

core Flight Executive (cFE)

Event Service	Executive Service	Software Bus
Table Service	Time Service	cFE Users Guide

Flight Event Messages

Rx Q Size	Bytes Tx	Bytes Rx	Cmd Pkts
0	0	0	0
0	0	0	0
0	0	4928	0

```
No such file or directory @ rb_sysopen - /dev/ttyUSB0
sysopen - /dev/ttyUSB0
/ruby/2.5.0/open-uri.rb:37:in `open'
/ruby/2.5.0/open-uri.rb:37:in `open'
/gems/2.5.0/gems/cosmos-4.4.2/lib/cosmos/io/posix_serial_driver.rb:47:in `initialize'
/gems/2.5.0/gems/cosmos-4.4.2/lib/cosmos/io/serial_driver.rb:58:in `new'
/gems/2.5.0/gems/cosmos-4.4.2/lib/cosmos/io/serial_driver.rb:58:in `initialize'
/gems/2.5.0/gems/cosmos-4.4.2/lib/cosmos/streams/serial_stream.rb:65:in `new'
/gems/2.5.0/gems/cosmos-4.4.2/lib/cosmos/streams/serial_stream.rb:65:in `initialize'
/gems/2.5.0/gems/cosmos-4.4.2/lib/cosmos/interfaces/serial_interface.rb:59:in `new'
/gems/2.5.0/gems/cosmos-4.4.2/lib/cosmos/interfaces/serial_interface.rb:59:in `connect'
```

Script Output:

Target: CF Command: DIS_DEQUEUE

Description:

Parameters:

Name	Value or State	Units	Description
CCSDS_STREAMID:	6323		Packet Identification
CCSDS_SEQUENCE:	49152		Packet Sequence Counter
CCSDS_LENGTH:	5		Packet Data Length
CCSDS_FUNCCODE:	17		Command Function Code
CCSDS_CHECKSUM:	0		CCSDS Command Checksum
CHANNEL:	0		Chan_0(0), Chan_1(1)
SPARE_1:	0		
SPARE_2:	0		
SPARE_3:	0		

Command History: (Pressing Enter on the line re-executes the command)

Target: CF Packet: HK_TLM_PKT

Description: CFDP housekeeping Packet

	Item	Value
1	*PACKET_TIMESECONDS:	0.000000
2	*PACKET_TIMEFORMATTED:	No Packet Time
3	*RECEIVED_TIMESECONDS:	0.000000
4	*RECEIVED_TIMEFORMATTED:	No Packet Receive...
5	*RECEIVED_COUNT:	0
6	CCSDS_STREAMID:	0x0000
7	CCSDS_SEQUENCE:	0
8	CCSDS_LENGTH:	0
9	CCSDS_SECONDS:	0
10	CCSDS_SUBSECS:	0

Finals wrap-up

- We captured a decisive lead during the first day
- Barely made it within acceptable bounds for our "Moonshot" plan
- Second day was tighter, with another team edging us out on points
- Ultimately, we ended up winning since they had not submitted a satisfactory "Moonshot" plan!

Hack-A-Sat

SPACE SECURITY CHALLENGE 2020™

Takeaways from Hack-A-Sat

- Space software can be understood by security researchers like anything else
  - **Existing approaches** to security analysis can be applied – both for good and bad!

- The unique factor is the interface (radio) and hard-to-access hardware
  - Underlying software systems are often familiar! Linux, VxWorks, seL4, …
  - But can't buy a new satellite online, broadcast on certain frequencies, etc

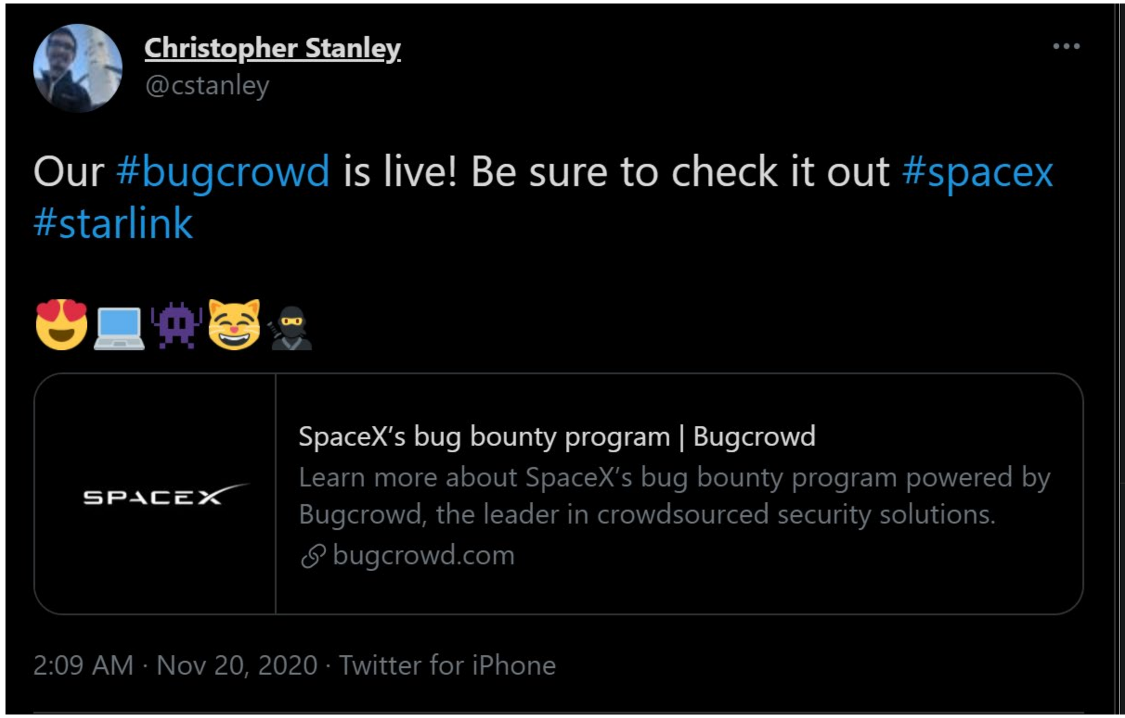
Takeaways from Hack-A-Sat

- Hack-A-Sat helped us peek through some of the security-through-obscurity of satellites and space software

- There's value in helping researchers perform security audits of your code, in a controlled way – would love to see more opportunities

- Software industry has moved to bug bounty model – can it work for satellites and space software?

SpaceX launched new Starlink bug bounty in Nov. 2020!



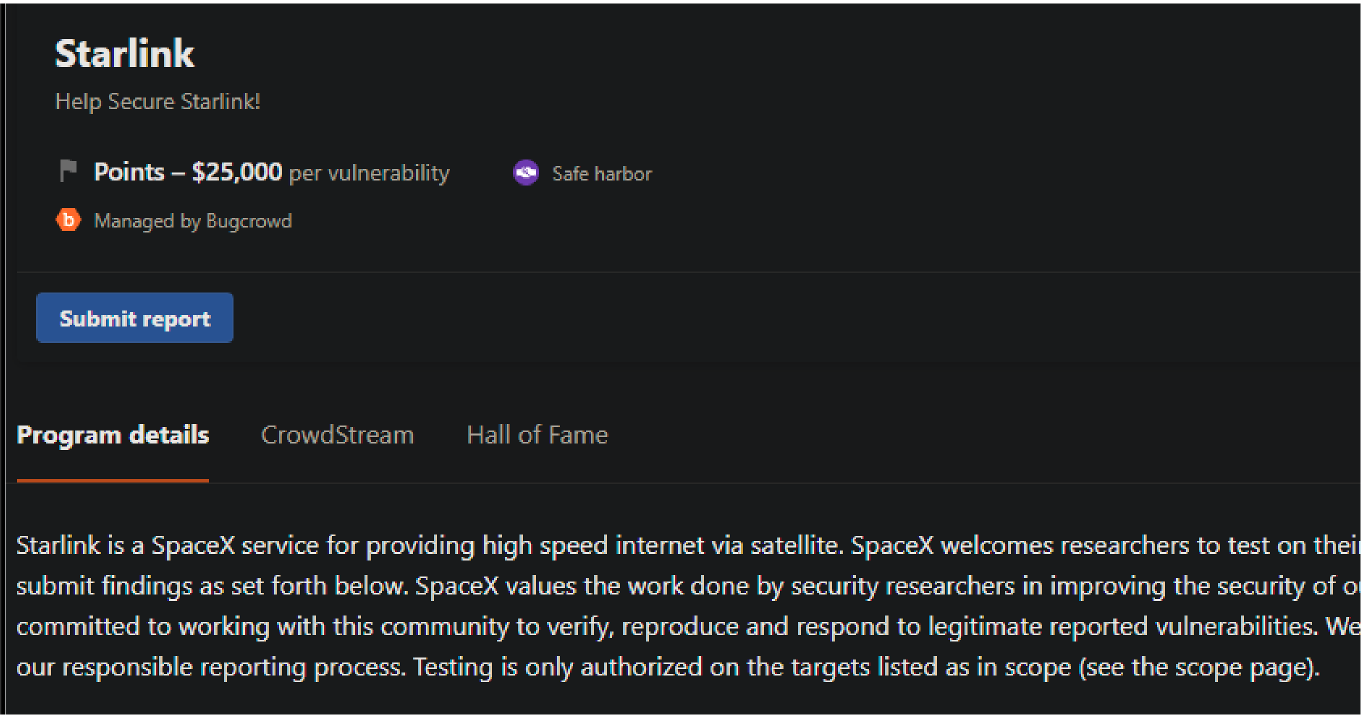
Christopher Stanley
@cstanley

Our #bugcrowd is live! Be sure to check it out #spacex #starlink

😍💻👾😸🥷

SpaceX's bug bounty program | Bugcrowd
Learn more about SpaceX's bug bounty program powered by Bugcrowd, the leader in crowdsourced security solutions.
bugcrowd.com

2:09 AM · Nov 20, 2020 · Twitter for iPhone



Starlink
Help Secure Starlink!

Points – $25,000 per vulnerability
Safe harbor
Managed by Bugcrowd

Starlink is a SpaceX service for providing high speed internet via satellite. SpaceX welcomes researchers to test on thei
submit findings as set forth below. SpaceX values the work done by security researchers in improving the security of o
committed to working with this community to verify, reproduce and respond to legitimate reported vulnerabilities. We
our responsible reporting process. Testing is only authorized on the targets listed as in scope (see the scope page).

Thanks!