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
CTF then and now
Space and Software
Open source goes to Mars

April 19, 2021 — Community, Open source

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...
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.

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.
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 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
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.

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 :)}
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 stabilize satellite with its torque rods
3. Develop our moonshot plan!
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
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!
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!
Thanks!