Secrets of Submitting to Black Hat:

A Guide to Writing a Successful Submission

Agenda

- Why and what to submit to Black Hat
- Makeup of a submission with examples

Who we are?

Lidia Giuliano
@pink_tangent

Marina Krotofil
@marmusha

Stefano Zanero
@raistolo

Why submit?

- Genuine desire to share your knowledge / findings
- You have something important to share with the security community and make this world a better place
- You want to showcase your competences to get your next / dream job

What to submit

- Ideally something which has never been presented
- Innovative: research-based / lessons learned
- Proven results

- Focus on:
  - Discovering a new technique, new ways to attack or defend
  - Novel methodologies, frameworks or approaches that have been tested/implemented in realistic environments
  - Fresh/unusual angle of looking at things

Makeup of a submission

- Title
- Abstract
- Presentation Outline
- New/never been presented?
- What new research, concept, technique, or approach is included?
- Takeaways
- What problem does your research solve?
- Releasing a tool?
- New vulnerability?
- Will your presentation include a demo?
- Notes to the review board

Title

- This is what people pay most attention to when deciding whether to attend the talk
- The title must be attractive, but be careful! Be sure to *properly manage expectations*
- If you say "Hacking satellites (...)", people will expect that to actually happen...

Title

Example Titles

Past Accepted Talks - Titles

- The Pool Party You Will Never Forget: New Process Injection Techniques Using Windows Thread Pools
- A Pain in the NAS: Exploiting Cloud Connectivity to PWN Your NAS
- Grand Theft House: RF Lock Pick Tool to Unlock Smart Door Lock
- Houston, We Have a Problem: Analyzing the Security of Low Earth Orbit Satellites

Abstracts

- Published on the conference website (if session is selected)
- Book Blurb but of your talk
- Should make people think, “Yeah, Wow!, I also have faced this, or I relate to this, I want to watch this”
- Would you attend this talk based on what you wrote?
- Have the “end user” (the audience) in mind when writing your abstract

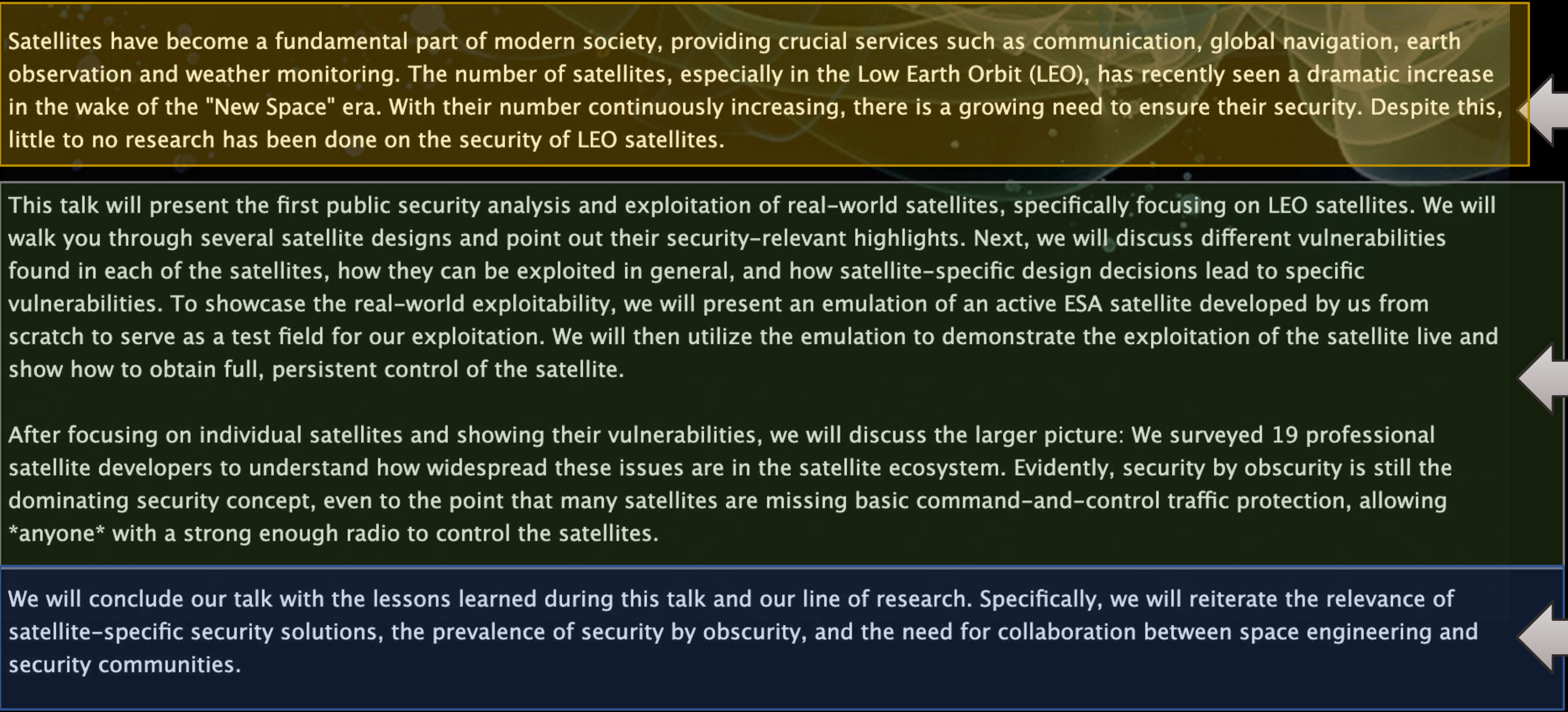
Abstracts - the breakdown

Three Parts:

- Problem statement / Hook
- What your talk is about
  - What you will discuss
  - Create intrigue
  - Include selling points
- What will they learn?
  - Takeaways

** Approx 2-4 paragraphs / 300 words **

Examples - Past Accepted Abstracts



Satellites have become a fundamental part of modern society, providing crucial services such as communication, global navigation, earth observation and weather monitoring. The number of satellites, especially in the Low Earth Orbit (LEO), has recently seen a dramatic increase in the wake of the "New Space" era. With their number continuously increasing, there is a growing need to ensure their security. Despite this, little to no research has been done on the security of LEO satellites.

Hook

This talk will present the first public security analysis and exploitation of real-world satellites, specifically focusing on LEO satellites. We will walk you through several satellite designs and point out their security-relevant highlights. Next, we will discuss different vulnerabilities found in each of the satellites, how they can be exploited in general, and how satellite-specific design decisions lead to specific vulnerabilities. To showcase the real-world exploitability, we will present an emulation of an active ESA satellite developed by us from scratch to serve as a test field for our exploitation. We will then utilize the emulation to demonstrate the exploitation of the satellite live and show how to obtain full, persistent control of the satellite.

After focusing on individual satellites and showing their vulnerabilities, we will discuss the larger picture: We surveyed 19 professional satellite developers to understand how widespread these issues are in the satellite ecosystem. Evidently, security by obscurity is still the dominating security concept, even to the point that many satellites are missing basic command-and-control traffic protection, allowing *anyone* with a strong enough radio to control the satellites.

Selling points

We will conclude our talk with the lessons learned during this talk and our line of research. Specifically, we will reiterate the relevance of satellite-specific security solutions, the prevalence of security by obscurity, and the need for collaboration between space engineering and security communities.

Lessons

Trusted Platform Module (TPM) is a tamper-resistant device and designed to provide hardware-based security functions. A TPM chip has a random number generator, non-volatile storage, encryption/decryption modules, and Platform Configuration Registers (PCRs), which can be utilized for various security applications such as BitLocker, DM-Crypt, Trusted Boot (tboot), and Open Cloud Integrity Technology (Open CIT).

TPM has been widely deployed in commodity devices to provide a strong foundation for building trusted platforms, especially in devices used in enterprise and government systems. Because TPM is the critical point in the trusted platform, many researchers have tried to find vulnerabilities in the TPM and concluded that it is hard to break it without physical access. However, this is not true anymore.

Hook

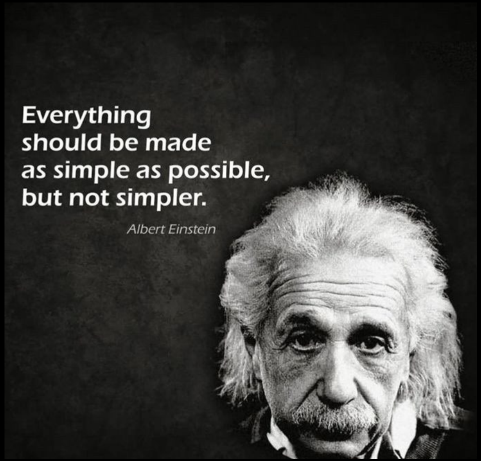
In this talk, we present two vulnerabilities, CVE-2017-16837 and CVE-2018-6622. The vulnerabilities we found can subvert the TPM with Advanced Configuration and Power Interface (ACPI). ACPI in PCs, laptops, and servers provide six sleeping states (S0-S5) for reducing power consumption. When the system enters the sleeping state, CPU, device, and RAM are powered off. Since the system powers the components off including security devices, the system should reinitialize them while waking up and this could be the attack surface. We found vulnerabilities on this attack surface without physical access.

Selling points

To mitigate the vulnerabilities, we also present countermeasures and a new tool, "Napper," to check the vulnerabilities of the TPM. Napper is a bootable USB device based-on Linux, and it has a custom kernel and a vulnerability checking software. When you boot a system with the Napper, it makes your system to take a nap to check the vulnerabilities and to report the result to you.

Lessons

Presentation Outline


Everything should be made as simple as possible, but not simpler.
Albert Einstein

- Deep dive into your topic of research
- Helps reviewers to understand your talk in detail
- Supports your submission with specifics, demos, screenshots, white/academic papers, slides draft, etc.
- This should be the most detailed part of your submission

"Not published and for the RB only"

Example Presentation Outlines

Introduction (2 minutes):

- Microcontrollers vs Microprocessors.
- Why targeting microcontrollers is worth it?

Microcontrollers programming (4 minutes):

- IDE, ASM instructions set and the assemble process.
- Writing the .hex file to program memory.

Dump the program memory (4 minutes):

- Hardware connections to the target device.
- Program memory dump into an .hex file.

Payload injection: at the entry point (12 minutes):

- Understanding a program standard structure.
- Cooking the payload.
- Injecting the payload.
- Checksum recalculation.

Advanced payload injection: at the interrupt vector (10 minutes):

- GIE, PEIE and polling inspection to identify enabled interrupts.
- Let's backdoor the EUSART (SCI) communication peripheral.

Stack Payload Injection: Controlling program flow (8 minutes):

- Understanding the Stack SFR: STKPTR, TOSU, TOSH & TOSL.
- Injecting to write the TOS (Top of Stack) data and take control of the program flow.

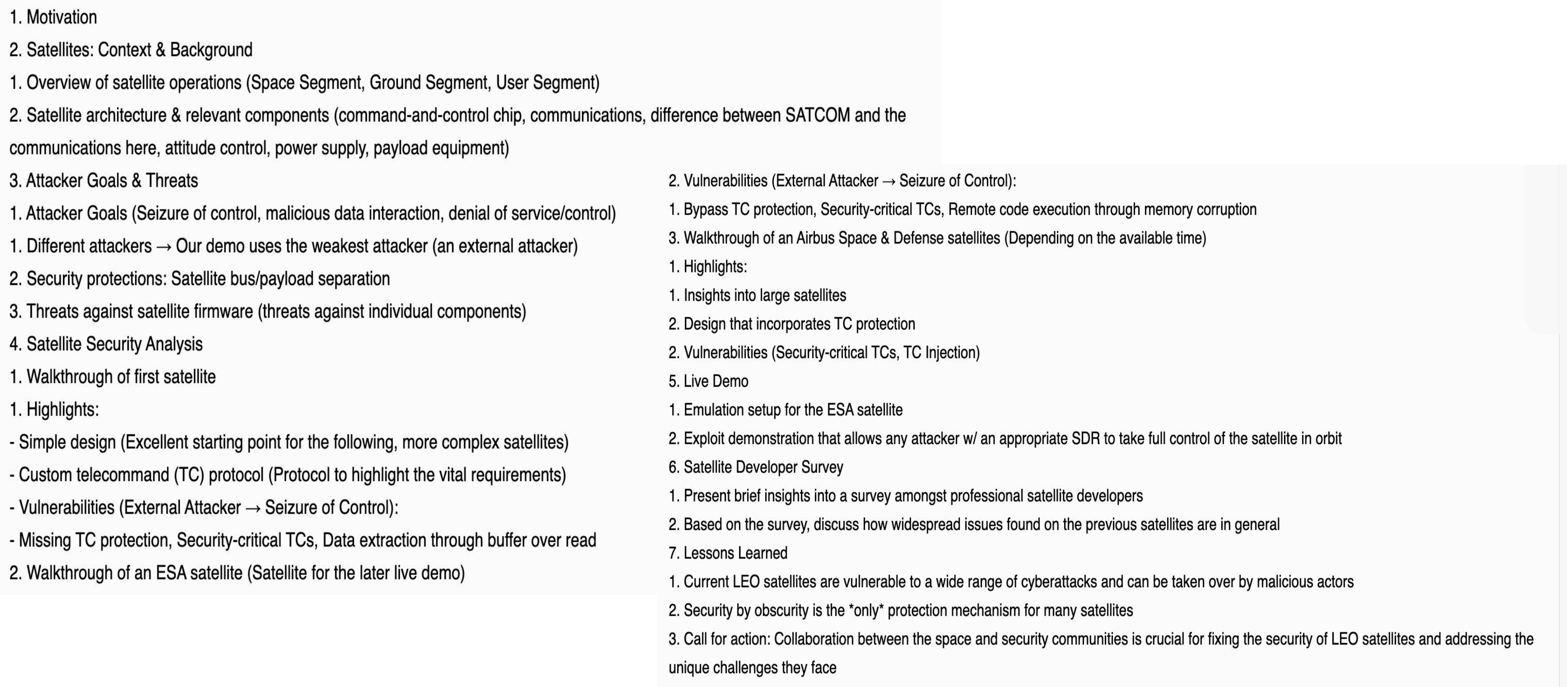
Automatizing payload injection (2 minutes):

- Presentation of a new open source tool named UCPI (UC Payload Injector) to automatizing payload injection on microcontrollers.

Program memory protections (3 minutes):

- Code Protection Bit will not protect you against backdoors.
- Boot protection.

Note: Presentation outline matches with paper contents, please check it out (it is attached) to get a sense of what will be shown at every outline point.



1. Motivation

2. Satellites: Context & Background

1. Overview of satellite operations (Space Segment, Ground Segment, User Segment)

2. Satellite architecture & relevant components (command-and-control chip, communications, difference between SATCOM and the communications here, attitude control, power supply, payload equipment)

3. Attacker Goals & Threats

1. Attacker Goals (Seizure of control, malicious data interaction, denial of service/control)

1. Different attackers → Our demo uses the weakest attacker (an external attacker)

2. Security protections: Satellite bus/payload separation

3. Threats against satellite firmware (threats against individual components)

4. Satellite Security Analysis

1. Walkthrough of first satellite

1. Highlights:

- Simple design (Excellent starting point for the following, more complex satellites)

- Custom telecommand (TC) protocol (Protocol to highlight the vital requirements)

- Vulnerabilities (External Attacker → Seizure of Control):

- Missing TC protection, Security-critical TCs, Data extraction through buffer over read

2. Walkthrough of an ESA satellite (Satellite for the later live demo)

2. Vulnerabilities (External Attacker → Seizure of Control):

1. Bypass TC protection, Security-critical TCs, Remote code execution through memory corruption

3. Walkthrough of an Airbus Space & Defense satellites (Depending on the available time)

1. Highlights:

1. Insights into large satellites

2. Design that incorporates TC protection

2. Vulnerabilities (Security-critical TCs, TC Injection)

5. Live Demo

1. Emulation setup for the ESA satellite

2. Exploit demonstration that allows any attacker w/ an appropriate SDR to take full control of the satellite in orbit

6. Satellite Developer Survey

1. Present brief insights into a survey amongst professional satellite developers

2. Based on the survey, discuss how widespread issues found on the previous satellites are in general

7. Lessons Learned

1. Current LEO satellites are vulnerable to a wide range of cyberattacks and can be taken over by malicious actors

2. Security by obscurity is the *only* protection mechanism for many satellites

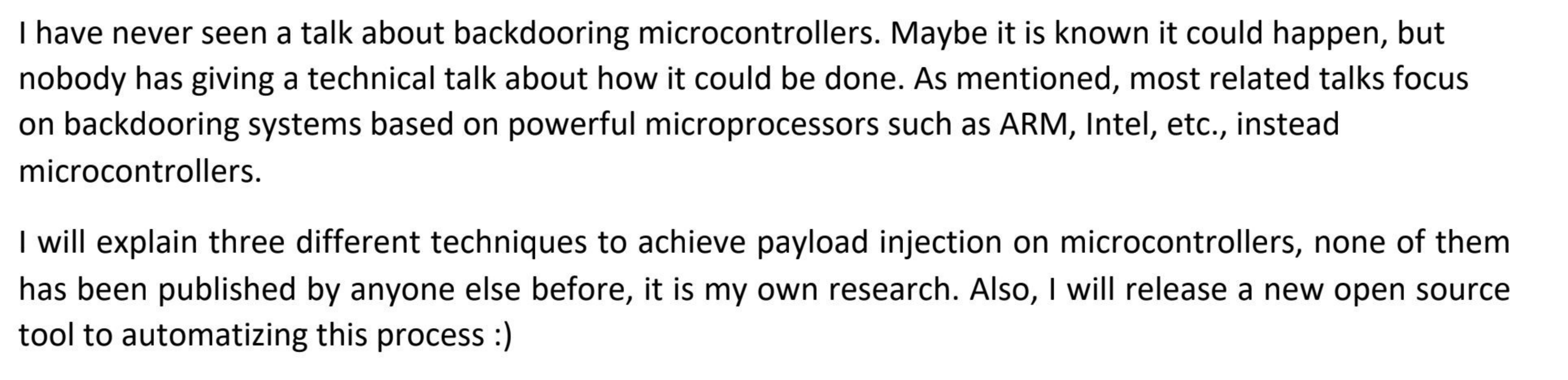
3. Call for action: Collaboration between the space and security communities is crucial for fixing the security of LEO satellites and addressing the unique challenges they face

New research, concepts and techniques

- Explain your research
- If it has been done before/built on previous work, explain what is different
- Go into detail and be specific
- Provide evidence such as a white paper (even if unpublished), CVE's, references to your work, etc.
- If you publicize your work before BH's date, your proposal "could" be of a lower priority (T&C apply)
- Don't use generative AI tools to derive your research

Examples - New Research, Concepts and Techniques

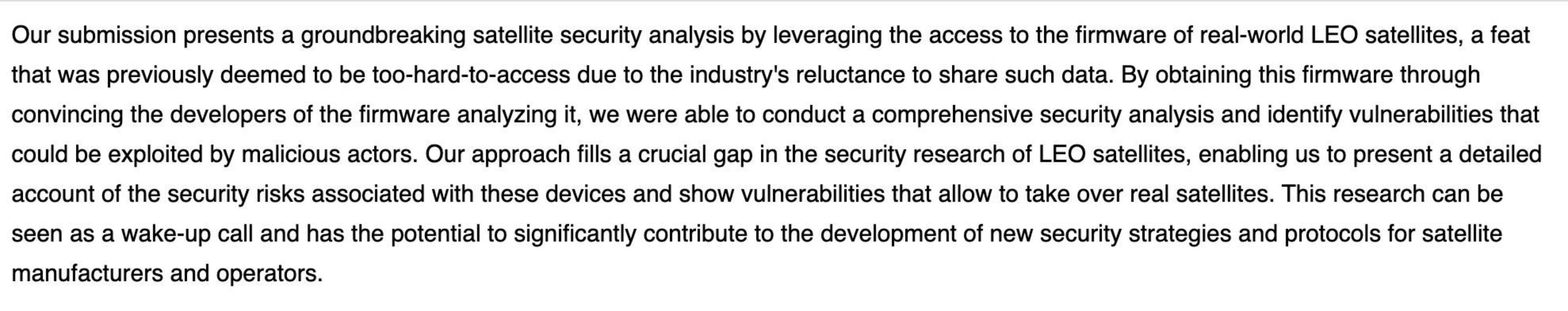
New research, concepts



I have never seen a talk about backdooring microcontrollers. Maybe it is known it could happen, but nobody has giving a technical talk about how it could be done. As mentioned, most related talks focus on backdooring systems based on powerful microprocessors such as ARM, Intel, etc., instead microcontrollers.

I will explain three different techniques to achieve payload injection on microcontrollers, none of them has been published by anyone else before, it is my own research. Also, I will release a new open source tool to automatizing this process :)

New research, concepts



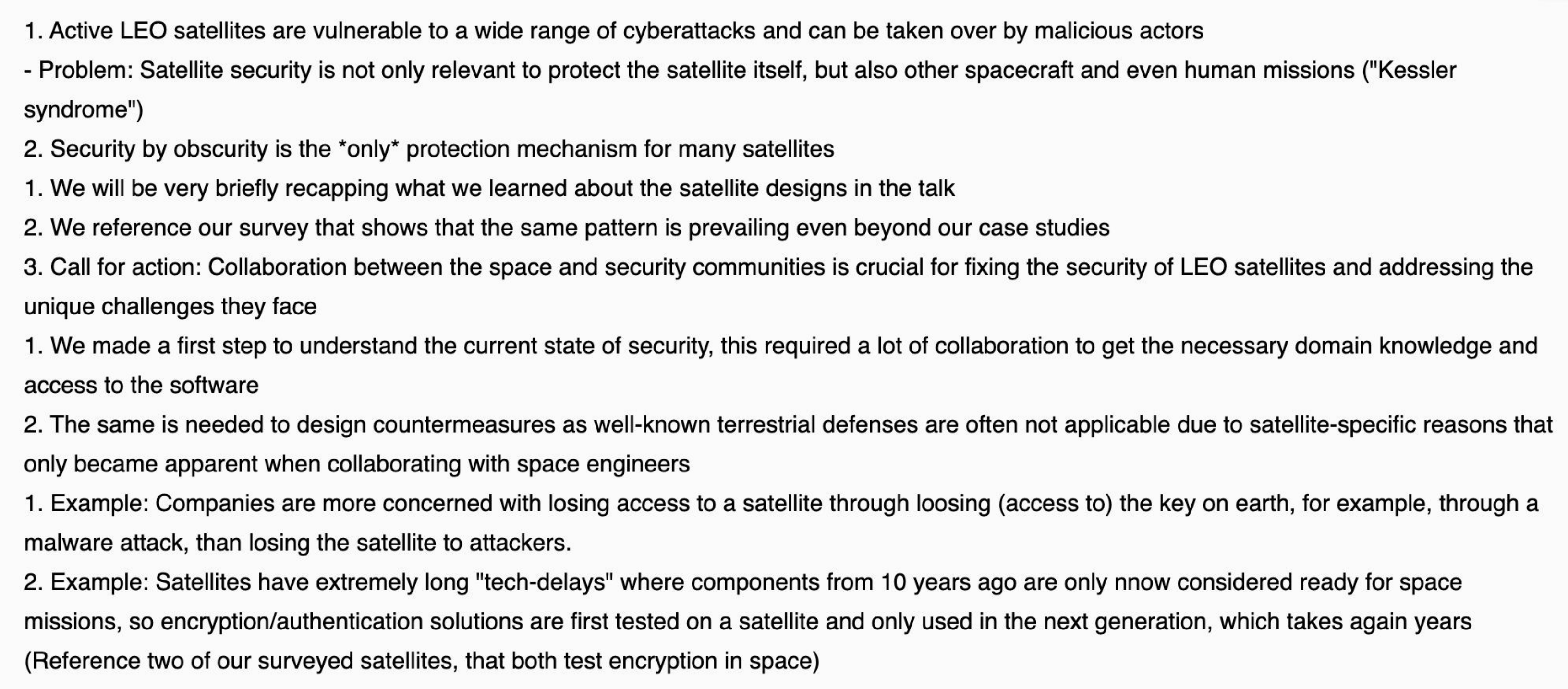
Our submission presents a groundbreaking satellite security analysis by leveraging the access to the firmware of real-world LEO satellites, a feat that was previously deemed to be too-hard-to-access due to the industry's reluctance to share such data. By obtaining this firmware through convincing the developers of the firmware analyzing it, we were able to conduct a comprehensive security analysis and identify vulnerabilities that could be exploited by malicious actors. Our approach fills a crucial gap in the security research of LEO satellites, enabling us to present a detailed account of the security risks associated with these devices and show vulnerabilities that allow to take over real satellites. This research can be seen as a wake-up call and has the potential to significantly contribute to the development of new security strategies and protocols for satellite manufacturers and operators.

Takeaways

- What are you leaving the audience with?

Example Takeaways

1. Learn to enable and configure the advanced security features on the docker daemon and its core components.
2. Learn to implement and/or improve the Docker security features for containers at run-time.
3. Learn to secure Swarm and Kubernetes environments by implementing their advanced security features.
4. Learn the attacks that might be carried out if these protections are not implemented.



1. Active LEO satellites are vulnerable to a wide range of cyberattacks and can be taken over by malicious actors

- Problem: Satellite security is not only relevant to protect the satellite itself, but also other spacecraft and even human missions ("Kessler syndrome")

2. Security by obscurity is the *only* protection mechanism for many satellites

1. We will be very briefly recapping what we learned about the satellite designs in the talk

2. We reference our survey that shows that the same pattern is prevailing even beyond our case studies

3. Call for action: Collaboration between the space and security communities is crucial for fixing the security of LEO satellites and addressing the unique challenges they face

1. We made a first step to understand the current state of security, this required a lot of collaboration to get the necessary domain knowledge and access to the software

2. The same is needed to design countermeasures as well-known terrestrial defenses are often not applicable due to satellite-specific reasons that only became apparent when collaborating with space engineers

1. Example: Companies are more concerned with losing access to a satellite through loosing (access to) the key on earth, for example, through a malware attack, than losing the satellite to attackers.

2. Example: Satellites have extremely long "tech-delays" where components from 10 years ago are only nnow considered ready for space missions, so encryption/authentication solutions are first tested on a satellite and only used in the next generation, which takes again years (Reference two of our surveyed satellites, that both test encryption in space)

Other Sections

- What problem does your research solve?
- Releasing a tool?
- New vulnerability?
- Will your presentation include a demo?

Submission Don’ts

DON’T

- Plagiarism/chatGPT/generative AI tooling
- If your submission took you 1 minute to write, it probably looks like a 1-minute submission
- A bio that is longer than your submission
- Marketing material
- Don’t sell stuff
- Presentation outline shorter than abstract
- Cut and paste of one form answer into another
- Proposals submitted by marketing/PR people on your behalf

Submission Do's

DO

- Follow the submission (CFP) guidelines
- Be your authentic-self
- Get it peer reviewed for feedback

Selection Process

- Selection process
  1. Scoring by the Review Board Members.
  2. Discussing/debating on Selection Calls.
  3. Final Selections.

Black Hat resources

https://www.blackhat.com/call-for-papers.html

- CFP submission links for each region
- Other useful information

Thank you

For further information or any questions, please email us: cfp@blackhat.com