

Cyber Range for Space Systems: Training Scenarios for Satellite Cybersecurity Preparedness

Matteo Ciccaglione^{1,*†}, Lorenzo Bracciale^{2,*‡}, Pierpaolo Loreti^{3‡} and Arianna Miraval Zanon^{4‡}

¹National Laboratory of Network Assessment, Assurance and Monitoring, CNIT, Rome, Italy

²Department of Electronic Engineering, University of Rome Tor Vergata, Rome, Italy

³DICII, Tor Vergata University of Rome, Rome, Italy.

⁴ASI - Agenzia Spaziale Italiana

Abstract

The increasing exposure of satellite systems to cyber threats requires new approaches to operational training and testing of countermeasures. Traditional assumptions about the security of space infrastructures—based on limited access and high technical barriers—are no longer valid in the face of accessible tools like Software Defined Radio (SDR) and cloud-based services such as Ground Station as a Service (GSaaS). In this paper, we present four realistic cyber attack scenarios specifically designed for integration into OpenSatRange (OSR), an open-source cyber range developed for training and experimentation in the satellite domain. The proposed scenarios cover key vulnerabilities in satellite ecosystems, including ground segment compromise, broadcast channel cryptographic flaws, inter-satellite link saturation (DDoS), and firmware-level hijacking via memory corruption. Each scenario is modeled with an emphasis on hands-on learning, narrative realism, and technical reproducibility. This work contributes to building a structured, operational framework for cybersecurity training in space systems, enabling simulation-based evaluation of both detection and response strategies in an environment that accurately reflects satellite communication constraints.

Keywords

Satellite cybersecurity, Cyber range, Training scenarios,

1. Introduction


Historically, the satellite domain has been considered relatively secure compared to other areas of telecommunications, partly because of a security-by-obscurity¹ approach and partly because of the high infrastructure costs required to carry out attacks. The required equipment, such as ground stations, high-powered transceivers, and specialized radio systems, presented a technological and economic barrier that made it impractical for non-state actors or otherwise

Joint Proceedings of IS-EUD 2025: 10th International Symposium on End-User Development, 16-18 June 2025, Munich, Germany.

*Corresponding author.

✉ matteo.ciccaglione@cnit.it (M. Ciccaglione); lorenzo.bracciale@uniroma2.it (L. Bracciale); pierpaolo.loreti@uniroma2.it (P. Loreti); arianna.miraval@asi.it (A. M. Zanon)

ORCID 0009-0005-7348-2674 (M. Ciccaglione); 0000-0002-6673-3157 (L. Bracciale); 0000-0002-2348-5077 (P. Loreti)

 © 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

¹Security by obscurity refers to the practice of attempting to protect a system by keeping its design, implementation, or internal mechanisms hidden. While it can provide a superficial layer of security, it is generally considered ineffective when relied upon alone, as determined attackers can eventually uncover or bypass the obscured elements.

not significantly resourced to compromise. As an illustration, the developers of the Iridium network claimed that the system was “too complex to attack” [1].

This perception, however, is no longer valid in the current context. The satellite communications ecosystem has evolved rapidly, with markedly lowered barriers to entry both economically and technically. Services such as Ground Station as a Service (GSaaS), offered by cloud providers such as Amazon Web Services and Microsoft Azure, allow any user with minimal skills to send and receive data from satellites, making the ground infrastructure a potential entry point for cyber attacks. In parallel, open source and open hardware projects such as SatNOGS [2] enable the construction of affordable amateur ground stations, further democratizing access to space. In addition, the advent of technologies such as Software Defined Radio (SDR) has revolutionized the threat landscape, allowing individuals to monitor, analyze, and interfere with satellite radio signals using commercial hardware and open source software [1, 3]. This lowering of technical complexity has also made sophisticated attacks, such as spoofing, jamming, and injection of malicious payloads into satellite firmware, more accessible [4, 3, 5].

In the face of a constantly evolving threat landscape, it is clear that the security of space systems can no longer rely on passive measures or technological complexity alone. In fact, today’s cyber threats surpass traditional defensive approaches based on isolation or obscurity of technical specifications [6]. Instead, proactive and systematic approaches are needed, capable of enabling a dynamic and resilient defense based on situational awareness, controlled experimentation, and operational training. In particular, there is an emerging need for specific tools for training, experimentation, and evaluation of countermeasures in realistic operational contexts, such as cyber ranges, that enable the reproduction of concrete scenarios and evaluation of defensive strategies in simulated environments [7].

This is especially true in the satellite domain, where the coexistence of physical constraints (latency, orbital mobility, limited visibility) and the complexity of on-board protocols impose specialized skills and technologically advanced tools [4]. However, while cyber ranges focused on ground-based infrastructure are now widespread and established in academia and industry [8], a flexible and accessible infrastructure for simulating attacks and defenses in the space domain is still lacking.

To fill this gap, the development of OpenSatRange (OSR) [9], a cyber range specifically designed to simulate satellite communication networks and enable hands-on exercises in realistic scenarios, has been initiated. OSR aims to train operators, technicians and analysts in the management of cyber incidents involving both terrestrial (ground stations, terminals) and space (satellite constellations, edge protocols) segments through immersive and configurable simulations. With this in mind, this paper proposes a set of four attack scenarios modeled to be implemented in the OSR context. Each addresses a specific aspect of satellite system vulnerabilities and represents a possible real-world threat. These scenarios are designed to assess detection, response, and recovery capabilities, as well as provide an operational context for the development of new countermeasures and defense policies.

2. Related Work

2.1. Real-World Satellite Attacks

The security of satellite systems has historically been underestimated, mainly due to high infrastructure costs and a security-by-obscurity approach. However, recent events show that such systems are increasingly exposed to sophisticated and accessible threats. For example, during the invasion of Ukraine in February 2022, a cyber attack compromised Viasat's KA-SAT satellite network, disrupting communication services for tens of thousands of users in Europe and Ukraine [10, 11].

Recent studies have highlighted the vulnerability of the firmware and software protocols used aboard the satellites. In particular, the analysis conducted by Willbold et al. [4] showed that real systems suffer from serious weaknesses in terms of authentication, firmware integrity and access protection. In parallel, targeted attacks on satellite services such as GNSS—for example, through spoofing and jamming—are becoming increasingly common, partly due to the availability of low-cost tools such as SDR (Software Defined Radio) [3, 5].

Specifically, Giuliani et al. [3] demonstrated that it is possible to conduct DDoS attacks against low-orbit (LEO) constellations by saturating inter-satellite links, with the potential to compromise communications across entire geographic areas. In addition, the public availability of orbital data and technical parameters of satellites accentuates the attack surface and encourages targeted malicious operations. In this regard, Manulis et al. [12] present a comprehensive analysis of threats in the New Space context, illustrating the evolution of attacks and entry vectors in the space and ground segments.

The security of satellite location services is also increasingly the focus of attention. GPS spoofing and meaconing attacks have been documented in real-world scenarios, as in the case of the RQ-170 drone captured in Iran [11], and recent studies confirm their ease of execution in the presence of unauthenticated signals [5].

2.2. Cybersecurity Training in Sector-Specific Domain

Training in cybersecurity has long moved beyond a generalist approach, evolving toward specialized programs for vertical domains. This trend is particularly evident in sectors with distinctive technical or operational characteristics, where threats take different forms than in traditional IT contexts.

In the industrial sector, for example, courses such as those offered by CISA for ICS systems [13] focus on operational aspects, minimal latencies, and legacy environments, while academic initiatives such as KYPO4Industry provide hands-on environments for testing exploits in real-world control networks [14]. Similarly, in the healthcare domain, institutes such as SANS offer pathways focused on ransomware attacks and patient data protection, with a focus on specific regulations such as HIPAA [15].

In aerospace and defense, the DoD Cyber Crime Center (DC3) offers a suite of advanced courses on critical infrastructure protection and forensic investigation in operations [16]. There are also centers dedicated to law enforcement training, such as UCD's Center for Cybersecurity & Cybercrime Investigation, which adapts OSINT and digital forensics techniques to real-world investigative scenarios [17].

These examples show how the declination of cyber training to different operational contexts is a must. In this landscape, the spatial domain-with physical constraints, proprietary protocols

2.3. Cyberranges for Satellite Security

Unlike traditional cyber ranges focused on IT or ICS infrastructure, satellite cyberrange platforms replicate the unique features of space operations, such as physical constraints, custom protocols, and the interaction between ground and orbital segments.

At the European level, the European Space Agency (ESA) has opened in Tallinn, Estonia, the first Space Cyber Range fully dedicated to simulating attack and defense scenarios against space infrastructure. This initiative is designed to support exercises and resilience testing by public and private entities in the European aerospace sector [18].

In the U.S., NASA has built specific tools for the simulation of small satellites such as CubeSats. In particular, the NOS3 (NASA Operational Simulator for Small Satellites) framework, developed for the Simulation-to-Flight-1 (STF-1) mission, allows the entire software life cycle of the satellite to be simulated in a virtual environment. This open-source tool is designed to support early verification of on-board functionality and pre-flight risk reduction, providing a valuable resource for pre-launch operational testing.

Also in academia, initiatives such as the Unified Cybersecurity Testing Lab for Satellite, Aerospace, Avionics, Maritime and Drone (SAAMD) have demonstrated the importance of multi-domain environments for security assessment in the space domain [19].

This landscape includes OpenSatRange (OSR), an open-source cyber range promoted by the Italian Space Agency (ASI), developed to enable operational exercises, attack simulations, and countermeasure testing in realistic satellite scenarios. OSR is distinguished by its modularity, ability to integrate SDR segments, and focus on supporting LEO/GNSS and ground-segment compromise scenarios [9].

3. Training Scenarios

In this section, we present four scenarios that can be implemented within a satellite-focused cyber range such as the one described in [9]. Each scenario is designed to replicate realistic cyber attack situations in the space domain, providing an operational context for training and evaluation activities.

Each scenario is described through the following three dimensions:

Story: We outline the environment, the narrative behind the attack, and the attacker's objective. This serves as the script guiding the actions that the attacker must perform to successfully complete the training.

Network: Given the satellite context, special attention is devoted to describing the network topology, including both terrestrial and space segments.

Relevance: For each scenario, we highlight its relevance by drawing parallels with real-world cyber attacks that have occurred in the satellite sector.

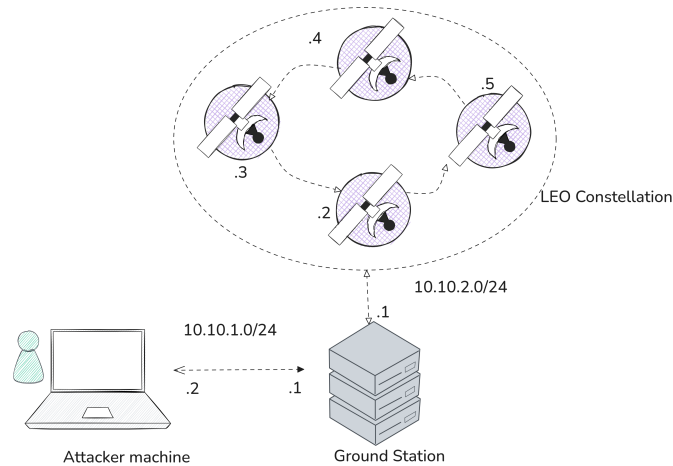


Figure 1: Network Infrastructure for Scenario 1

3.1. Scenario 1: Ground Segment Attack

This scenario describes an hypothetical attack that involves the ground segment of a satellite environment.

Story: A big company is hosting an infrastructure of Ground Station as a Service (GaaS) which is accessible through a web site always hosted by the company. Due to an intricate security flaw, an external attacker is able to gain partial access to the machine hosting the service. Exploiting his privileges, he can dump and patch the satellite firmware, that is later uploaded and installed on a satellite through a periodical update procedure. In this way the ground station is used as an attack vector to reach satellites in orbit.

Network: The network infrastructure is summarized in figure 1 The space segment consists of a LEO constellation with a (sufficient) number of satellites in such a way that at least one of them is in the ground station visibility range.

Relevance: Numerous cyber attacks have targeted user devices or ground segments in satellite communication systems. One of the most recent and widely known incidents is the attack on Viasat's KA-SAT network in February 2022 [10]. The attackers deployed a wiper malware known as *AcidRain*, which was specifically designed to erase the flash memory of satellite modems. Although the satellite itself and the core ground infrastructure were not compromised, the attack rendered thousands of modems inoperable, significantly disrupting satellite internet services across Europe and Ukraine. Another example is about the Russian satellite telecom provider Dozor which was compromised in a cyberattack that resulted in damage to user terminals and a temporary network outage. The attackers successfully infiltrated systems responsible for managing customer equipment, highlighting critical vulnerabilities in satellite ground infrastructures [20].

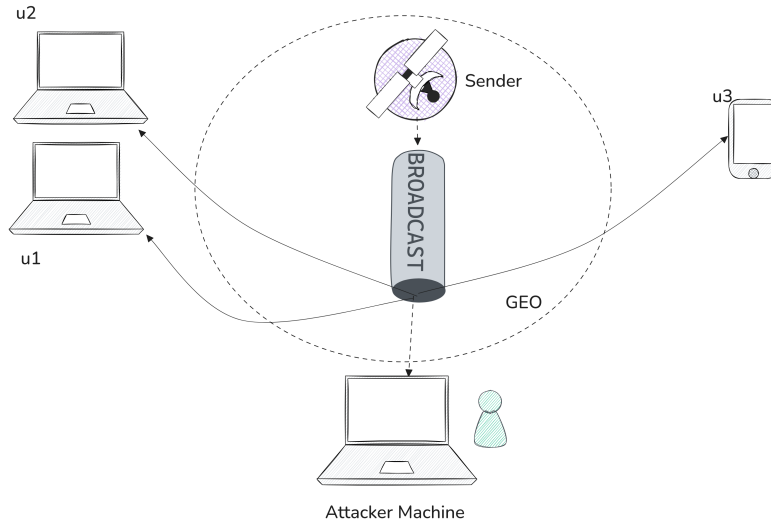


Figure 2: Network Infrastructure for Scenario 2

3.2. Scenario 2: Decrypting a broadcast channel

This scenario focuses on the role of cryptography in securing satellite communications.

Story: This scenario aims to exploit a misconfiguration in a custom satellite key exchange protocol to demonstrate how cryptographic vulnerabilities can be leveraged to compromise the confidentiality of message exchanges. The attacker's objective is to decrypt communications transmitted over a satellite broadcast channel, revealing sensitive data that was assumed to be securely encrypted. A central satellite periodically sends a message containing an encrypted symmetric key in a broadcast channel. The message is formatted as follows:

- The RSA public key (in PEM format) used to cipher the symmetric key;
- The encrypted symmetric key.

Each node that receives the message checks the public key to see if it matches its private key, and if so, decrypts the symmetric key that will be used to encrypt the rest of the communication. An external attacker is sniffing the traffic over the broadcast channel and due to incorrect configuration of pre-distributed RSA keys, he can perform an RSA common factor attack to decrypt the message and thus obtain the symmetric encryption key.

Network: The network infrastructure, illustrated in Figure 2, consists of a GEO satellite that communicates over a broadcast channel with multiple user terminals.

Relevance: Crypto attacks are also a real threat in the satellite environment as shown real-time inversion attacks on the GMR-2 cipher used in satellite phones were reported [21].

3.3. Scenario 3: Distributed Denial of Service in a LEO Constellation

A Distributed Denial of Service (DDoS) is a cyber attack, widely known in the telecommunications sector, which uses a huge amount of dummy data traffic to saturate network links,

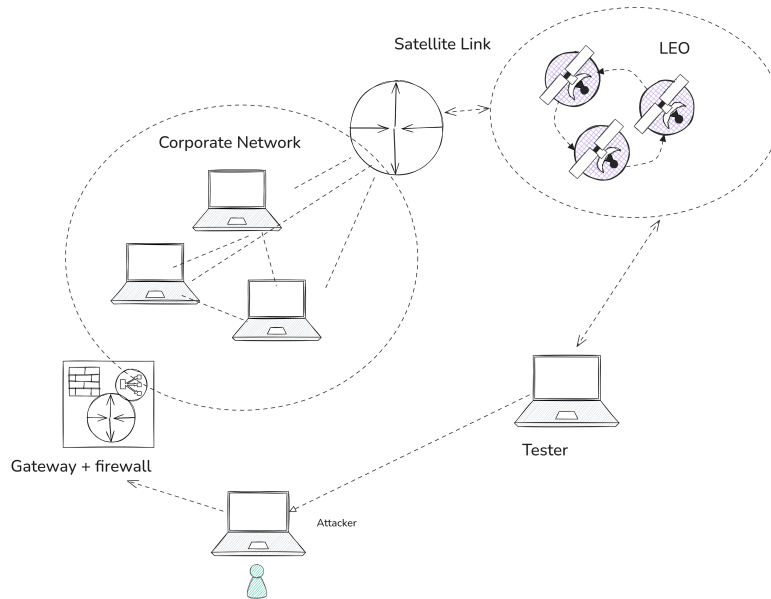


Figure 3: Network Topology for Scenario 3

exploiting the limited bandwidth capacity. Typically, botnets are used to generate the aforementioned traffic, that is, an attempt is made to infect a set of machines within a local network (for example a corporate network) that has an access point to the target network, taking control of them and transforming them into a set of bots at the service of an external user, through software that can be traced back to the Command & Control (C2) family [22].

Story: This scenario tried to reproduce the ICARUS attack [3] providing a vulnerable corporate network isolated by a wrong-configured firewall and a broken gateway. Gaining access to one machine on the network allows the attacker to quickly take control of others and set up a botnet, which can be used to flood a huge amount of traffic on a satellite link with limited bandwidth. The goal is to saturate the inter-satellite links thus leading to unavailability of the service.

Network: The network topology provided is summarized in figure 3. For technical reasons, we need a test machine that pings two satellites and provides the attacker with the code to proceed with the exercise when one of the two is no longer reachable.

Relevance: Despite being largely known in standard telecommunications systems, DDOS attacks have recently studied also in a satellite system (ICARUS Attack [3]).

3.4. Scenario 4: Satellite Hijacking

This attack targets a vulnerable implementation of a library used to parse telecommands.

Story: This scenario aims to show how simple it is to break a satellite system with an obscured simple vulnerability. The provided network topology is shown in figure 4. The idea behind the attack is the following: a satellite is piloted by firmware that makes use of vulnerable library functions like `strcpy` that can be exploited by an attacker to perform a buffer overflow and rewrite part of the satellite's memory. The goal is to corrupt the satellite's memory, causing it

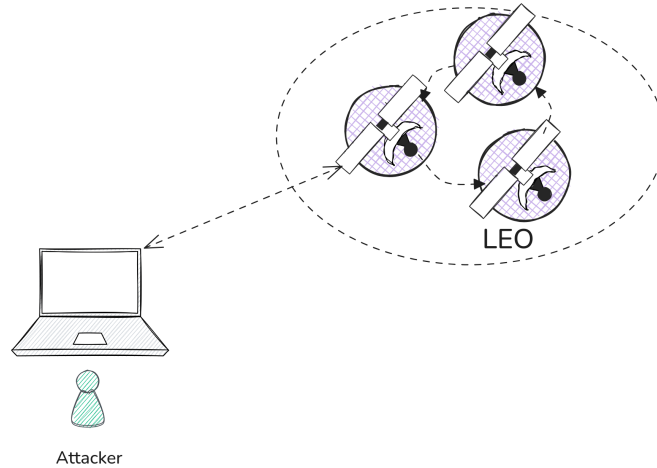


Figure 4: Network Topology for Scenario 4

to operate in a manner controlled by the attacker.

In this scenario, we have a weather forecast satellite, which receives input requests on the collected data, and an external attacker that carries out the attack by exploiting the above-mentioned vulnerability. To make everything more realistic, communication with the satellite uses the CCSDS format [23].

Network: Although we only need one satellite to set up the attack, the provided topology allows for a sufficient number of satellites to allow, with the parameters provided, that the intervals of non-visibility (in which the attacker does not see any satellites) are very short, so as to allow an effective exercise.

Relevance: The concept of security through obscurity is prevalent in satellite systems, as the firmware used is often proprietary and specifically developed for custom hardware architectures. Although this is true, the Space Odyssey study [4] shows that attacks based on information gathered through reverse engineering are feasible.

4. Implementation

This section offers a more in-depth technical perspective on the training scenarios outlined in Section 3. All software implementation details are designed with the interactive nature of the cyber range in mind. This means that each successfully exploited vulnerability must yield a flag (namely unique code) that is provided to the user as proof of exploitation.

4.1. Technical details about satellite software vulnerabilities: A hands-on approach

4.1.1. Scenario 1

To emulate this attack, we need to:


```

1  while(1){
2
3      if ((new_socket = accept(server_fd,
4          (struct sockaddr *)&address, &addr_len)) < 0) {
5          close(server_fd);
6          exit(EXIT_FAILURE);
7      }
8      if(handle_conn(new_socket,address) == 231){ // This is the conditional branch to patch
9          break;
10     }
11     else{
12         close(new_socket);
13     }
14 }
15 // Secret part
16 int fd = open("secret.txt",O_RDONLY,0);
17 struct stat st;
18 stat("secret.txt", &st);
19 int size = st.st_size;
20 char *buf = malloc(size+1);
21 read(fd,buf,size);
22 write(new_socket,buf,size);

```

Listing 1: C code snippet example for hackable firmware

1. Develop a vulnerable server hosting a Ground Station as a Service web page. In particular, we want to hide a git repository used for platform deployment and maintenance behind a hidden virtual host;
2. Set up the git repository with explanatory logs and a credentials file removed with the last commit;
3. Set up a server on the Ground Station machine with an automated firmware update routine and an easy-to-crack firmware executable.

To emulate a firmware tampering in an emulated environment, while still delivering a valid flag to the user, we follow this approach: the provided firmware includes a "secret" code segment, that is, a theoretically unreachable portion of code, which becomes executable through a firmware patch (e.g., modifying an immediate value in a loop condition). This hidden code is responsible for triggering the flag. Consequently, the satellite environment can be kept simple, requiring only a `flag.txt` file and a lightweight server that reads the input bytes, builds an executable, and runs it. As an illustrative example, refer to the code snippet shown in 1, which contains a hidden code segment accessible only under a specific condition. This condition can be easily bypassed through binary tampering, allowing the execution flow to be redirected to unauthorized code.

Attacker walkthrough: From the attacker's perspective, the intended approach for this exercise is the following. Using enumeration tools like *Gobuster*, the attacker first identifies the presence of the virtual host, then iterates the process to locate the git repository. Once found, the attacker can clone the repository to their machine using tools like *GitDumper*. By leveraging Git's features and analyzing commit messages in the log, they can locate the file containing the credentials and restore it. With the credentials, the attacker can gain SSH access to the machine. After enumerating the files and active cron jobs, they can identify the firmware to patch. Using

```

1  class SimpleProxy(http.server.BaseHTTPRequestHandler):
2
3      def do_GET(self):
4          self._proxy_request('GET')
5
6      def do_HEAD(self):
7          self._proxy_request('HEAD')
8
9      def do_POST(self):
10         content_type = self.headers.get('Content-Type')
11
12         if content_type != 'image/jpeg':
13             self.send_response(415)
14             self.send_header('Content-type', 'text/plain')
15             self.end_headers()
16             self.wfile.write(b'""""Unsupported Media Type:
17             Only JPEG images are allowed.""""')
18             return
19
20         self._proxy_request('POST')

```

Listing 2: A possible way to implement a gateway for MIME type inspection

reverse engineering tools like *Ghidra* and binary patching tools like *Okteta*, they can apply the necessary changes to compromise the satellite.

4.1.2. Scenario 2

The goal of this Scenario is to violate the confidentiality of satellite transmissions by obtaining symmetric encryption keys by intercepting key-distribution messages sent from a satellite acting as a master-key server.

To implement this, we must pre-distribute the private keys across the various nodes, ensuring that two nodes share a common prime factor in the RSA key, denoted as N . This setup allows the attacker to intercept network traffic using tools like *Wireshark* and launch a common factor attack, leveraging the public key present in the messages as outlined in 3.2.

Attacker walkthrough: Using sniffing tools, the attacker must gather a large set of public key-symmetric key pairs. Then, a cryptographic attack can be performed, for example, leveraging the capabilities of the Python library *PyCryptoDome*. If successful, the private keys obtained can be used to decrypt the corresponding symmetric keys, which, for the purpose of the exercise, will contain the flag.

4.1.3. Scenario 3

In this scenario we want to emulate the well-known ICARUS attack. To do that we have to:

1. Model a corporate network hosting a web service with the capability to upload images;
2. Put a firewall and a gateway to protect the network. The firewall checks TCP connection requests, denying incoming connections on port 4242, which is used by machines on the network to host a custom remote shell service. This can be implemented more simply by leveraging iptables rules such as `iptables -A INPUT -p tcp -dport 4242`

```
! -s 192.168.1.0/24 -j DROP, assuming that 192.168.1.0/24 is the network mask of the LAN. The gateway on the other hand performs checks on post requests to the website, verifying that the MIME type actually corresponds to an image, applying checks like the one reported in listing 2;
```

3. Implement a custom remote shell service that includes a basic and commonly encountered vulnerability—such as an SQL injection in the access control system or a similar low-level flaw. The specific nature of the vulnerability is left to the discretion of the reader, as it does not influence the overall success of the attack scenario.

The success of the attack hinges on accurately modeling an intersatellite link with sufficiently low network bandwidth.

Attacker walkthrough: The attacker's first step is to detect the presence of the web service using *Nmap*. The firewall will mask port 4242 as filtered. Upon accessing the web service, the attacker discovers the file upload feature, which can be exploited by uploading a PHP file and subsequently viewing it via the "view image" button. After some testing, the attacker identifies the presence of a gateway and bypasses it by using a *Burp* proxy server to intercept requests and modify the MIME type, circumventing the gateway's soft checks. At this point, the attacker can set up a local file inclusion (LFI) attack, causing the web server to execute the uploaded PHP page, which can contain code for a remote shell. Once inside the network, the attacker exploits the vulnerable custom service to move between machines and install the botnet. To create the botnet, the attacker develops both a server for remote control and a client that sends packets when triggered via a designated link. To make the attack feasible, the user will be provided with information about the vulnerable satellite link using the facilities provided by the cyber range.

4.1.4. Scenario 4

In this scenario we chose to implement the vulnerable server in C to gain greater control over memory layout and object placement within the address space. This also required developing a custom C library to handle the CCSDS packet protocol. The core idea is to define a packet data structure without any padding between its members, making the exploit more straightforward to execute. By leveraging a vulnerable function such as *strcpy*, an attacker can overwrite a structure member located immediately after the buffer used to store the message. A software routine then inspects this overwritten field and, if it has been correctly manipulated, the program ceases normal operation and begins responding with the flag.

Attacker walkthrough: The exercise includes the executable file that implements the vulnerable server-side service. The attacker must use reverse engineering tools like *Ghidra* to analyze the binary and identify the described vulnerability. Once understood, the attacker can self-host the service to perform tests and determine the appropriate payload to exploit the vulnerability. To carry out the attack, the attacker needs to implement a client that, using the provided C library for generating CCSDS packets, sends the correct payload to the server, triggering a memory corruption that ultimately results in the flag being distributed, as previously explained.

5. Conclusions

The increasing exposure of satellite systems to cyber threats is a priority challenge for space infrastructure security, both civil and military. In a context where accessibility to space is expanding through the use of COTS technologies, cloud-based services such as GSaaS, and tools such as SDRs, the space domain can no longer be considered an inaccessible environment.

This paper presents four didactic attack scenarios designed to be integrated within a cyber range such as OpenSatRange. The scenarios are inspired by real cases documented in the literature and incident reports, and cover a wide range of attack vectors: firmware compromise, cryptographic exploits, inter-satellite DDoS, and advanced spoofing on broadcast services. The goal is to provide operational and narrative tools that allow operators to practice in controlled but reality-adherent environments.

Defining these scenarios is a first step toward building a framework for practical training and evaluation of countermeasures in the space environment. The narrative-technical approach adopted allows enhancing both the educational aspect and reproducibility in simulated environments. In addition, their future integration into OSR will allow not only training operators, but also testing defensive solutions and monitoring tools under realistic conditions.

Declaration on Generative AI

During the preparation of this work, the author(s) used X-GPT-4 in order to: Grammar and spelling check. After using these tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

Acknowledgments

This research is supported by the project “OpenSatRange: Un cyber range aperto per la formazione in cyber security di operatori di sistemi e reti satellitari” supervised and financed by the Italian Space Agency (Agenzia Spaziale Italiana, ASI) in the framework of the Research Day “Giornate della Ricerca Spaziale” initiative through the contract no. ASI-2023-2-U.0, and by the CYBER4SPACE project, funded by the 2024 Scientific Research Grant of the University of Rome Tor Vergata.

References

- [1] G. Falco, The Vacuum of Space Cyber Security, in: AIAA SPACE and Astronautics Forum and Exposition, American Institute of Aeronautics and Astronautics, 2018.
- [2] Satnogs: Satellite networked open ground station, <https://satnogs.org/>, ????. Accessed: 2025-04-11.
- [3] G. Giuliani, T. Ciussani, A. Perrig, A. Singla, ICARUS: Attacking Low Earth Orbit Satellite Networks, in: 2021 USENIX Annual Technical Conference (USENIX ATC 21), USENIX Association, 2021, pp. 317–331.

- [4] J. Willbold, M. Schloegel, M. Vögele, M. Gerhardt, T. Holz, A. Abbasi, Space Odyssey: An Experimental Software Security Analysis of Satellites, in: 2023 IEEE Symposium on Security and Privacy (SP), IEEE, San Francisco, CA, USA, 2023, pp. 1–19. doi:10.1109/SP46215.2023.10351029.
- [5] Z. Wu, Y. Zhang, Y. Yang, C. Liang, R. Liu, Spoofing and anti-spoofing technologies of global navigation satellite system: A survey, IEEE Access 8 (2020) 165444–165496. doi:10.1109/ACCESS.2020.3022149.
- [6] G. Falco, N. Boschetti, A Security Risk Taxonomy for Commercial Space Missions, in: ASCEND, American Institute of Aeronautics and Astronautics, 2021, p. 4241.
- [7] G. Bernardinetti, S. Iafrate, G. Bianchi, Nautilus: A Tool for Automated Deployment and Sharing of Cyber Range Scenarios, in: Proceedings of the 16th International Conference on Availability, Reliability and Security (ARES), ACM, 2021, pp. 1–7. doi:10.1145/3465481.3469206.
- [8] J. Vykopal, R. Ošlejšek, P. Čeleda, M. Vizváry, D. Tovarňák, KYPO Cyber Range: Design and Use Cases, IEEE Transactions on Education (preprint) (2017). Available from Masaryk University technical report or conference proceedings; see also <https://crp.kypo.muni.cz>.
- [9] F. Patrone, P. Loreti, L. Fiscariello, L. Bracciale, A. Amici, A. Detti, C. Roseti, F. Zampognaro, M. Luglio, G. Bianchi, et al., Opensatrange: An open cyber range for operators and users of satellite communication networks, in: CEUR WORKSHOP PROCEEDINGS, volume 3731, CEUR-WS, 2024.
- [10] CyberPeace Institute, Viasat cyberattack case study, <https://cyberconflicts.cyberpeaceinstitute.org/law-and-policy/cases/viasat>, 2022. Accessed: 2025-04-11.
- [11] M. Kang, S. Park, Y. Lee, A survey on satellite communication system security, Sensors 24 (2024) 2897. URL: <https://www.mdpi.com/1424-8220/24/9/2897>. doi:10.3390/s24092897.
- [12] M. Manulis, C. P. Bridges, R. Harrison, V. Sekar, A. Davis, Cyber security in new space: Analysis of threats, key enabling technologies and challenges, International Journal of Information Security 20 (2021) 287–311. URL: <https://doi.org/10.1007/s10207-020-00503-w>. doi:10.1007/s10207-020-00503-w.
- [13] Cybersecurity and Infrastructure Security Agency (CISA), Ics training available through cisa, 2025. URL: <https://www.cisa.gov/resources-tools/programs/ics-training-available-through-cisa>, accessed April 11, 2025.
- [14] P. Čeleda, J. Vykopal, V. Švábenský, K. Slaviček, Kypo4industry: A testbed for teaching cybersecurity of industrial control systems, in: Proceedings of the 51st ACM Technical Symposium on Computer Science Education (SIGCSE), 2020, pp. 1234–1240. URL: <https://doi.org/10.1145/3328778.3366908>. doi:10.1145/3328778.3366908.
- [15] SANS Institute, Cybersecurity courses & certifications, 2025. URL: <https://www.sans.org/cyber-security-courses/>, accessed April 11, 2025.
- [16] Department of Defense Cyber Crime Center (DC3), Cyber training academy, 2025. URL: <https://www.dc3.mil/TrainingAcademy>, accessed April 11, 2025.
- [17] University College Dublin, Centre for cybersecurity & cybercrime investigation (cci), 2025. URL: <https://www.ucd.ie/cci/>, accessed April 11, 2025.
- [18] European Space Agency (ESA), Estonia to host europe’s new space cybersecurity testing ground, 2025. URL: https://www.esa.int/Applications/Connectivity_and_Secure_Communications/Estonia_to_host_Europe_s_new_space_cybersecurity_testing_ground,

accessed April 11, 2025.

- [19] A. Costin, H. Turtiainen, S. Khandker, T. Hämäläinen, Towards a unified cybersecurity testing lab for satellite, aerospace, avionics, maritime, drone (saamd) technologies and communications, arXiv preprint arXiv:2302.08359 (2023). URL: <https://arxiv.org/abs/2302.08359>.
- [20] V. Petkauskas, Russian satellite telecom dozor hit by hackers, 2023. URL: <https://cybernews.com/cyber-war/dozor-russian-satellite-telecom-hacked/>, accessed: 2025-04-11.
- [21] J. Hu, R. Li, C. Tang, A real-time inversion attack on the gmr-2 cipher used in the satellite phones, Science China Information Sciences 61 (2018) 1–18.
- [22] H. R. Zeidanloo, A. A. Manaf, Botnet command and control mechanisms, in: 2009 Second International Conference on Computer and Electrical Engineering, volume 1, IEEE, 2009, pp. 564–568.
- [23] The Consultative Committee for Space Data Systems, CCSDS publications manual, 2014.