Hacking satellites

~~From SDR to RCE.~~

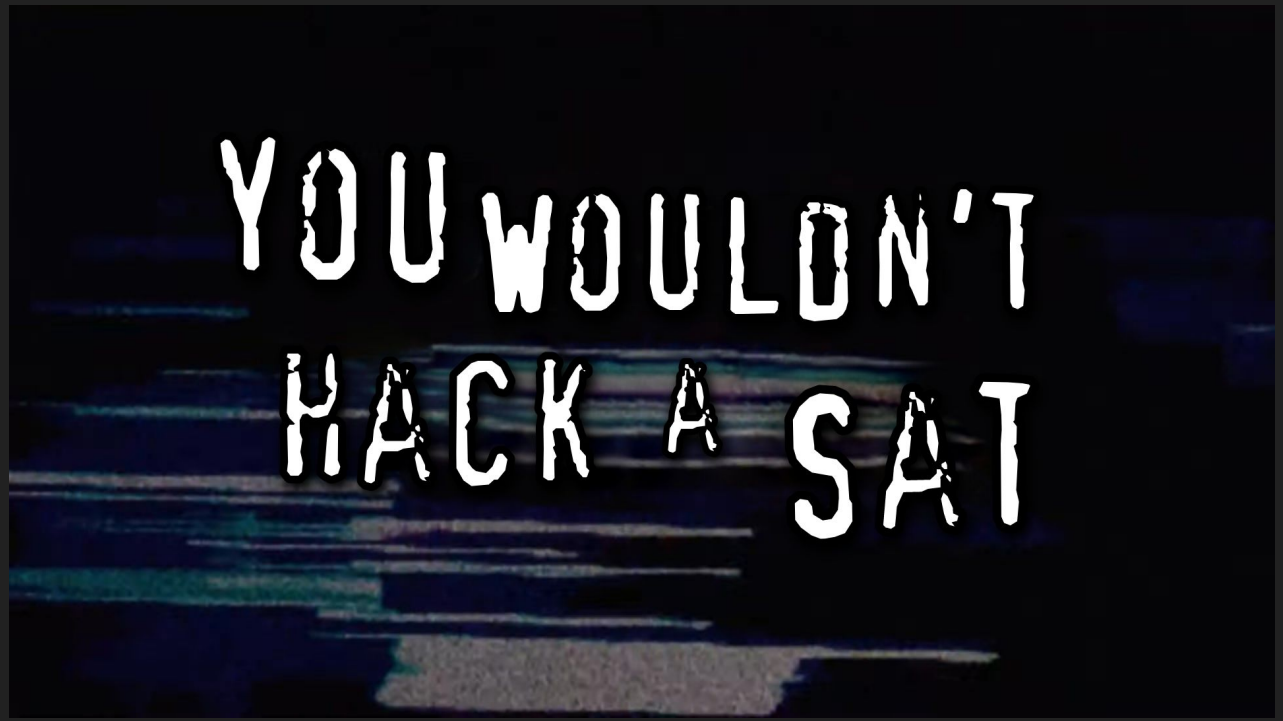
De la radio logicielle à l'exécution de commande à distance.

Présentation

whoami

- **Salim LARGO | @2ourc3**
- **Security Engineer @ Nexova - RF Pentest team**
- **Focus sur les techniques DAST/SAST**
- **Github https://github.com/20urc3 | Site https://bushido-sec.com/**

Avertissement légal



Cyberattaques.

Chinese hackers suspected of interfering with US satellites

Two US government satellites fell victim to cyber-attacks in 2007 and 2008, claims report highlighting control systems' vulnerability

Landsat 7 satellite image of the area around Los Alamos showing forest fires in New Mexico. The Landsat 7 was hacked in 2007 and 2008. Photograph: Rob Simmon/AP

Chinese hackers are suspected of having interfered with the operation of two US government satellites on four occasions via a ground station, according to a report being prepared for the US Congress.

BBC

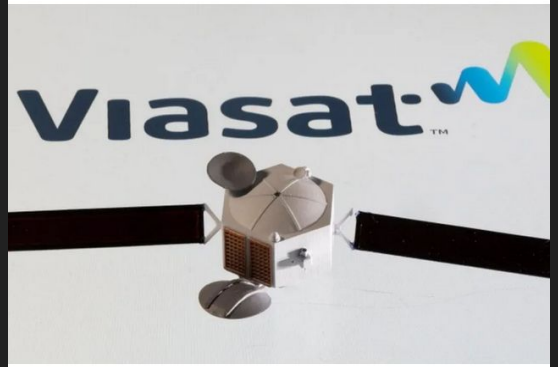
Home News Sport Business Innovation Culture Travel Earth Video Live

UK blames Russia for satellite internet hack at start of war

10 May 2022

By Chris Vallance, Technology Reporter

Share



Russia was behind a cyber-attack targeting American commercial satellite internet company Viasat, UK and US intelligence suggests.

the japan times

JAPAN

Japan's space agency hit by repeated cyberattacks since last year

During the attacks, private data on JAXA employees as well as information on external parties might have leaked, it has been reported. | BLOOMBERG

BY GABRIELE NINIVAGGI
STAFF WRITER

SHARE Jun 21, 2024

Listen to this article
0:00 / 2:39 1x

The Japan Aerospace Exploration Agency (JAXA) has experienced several cyberattacks since last year, the government said Friday.

L'importance des satellites dans l'infrastructure moderne.

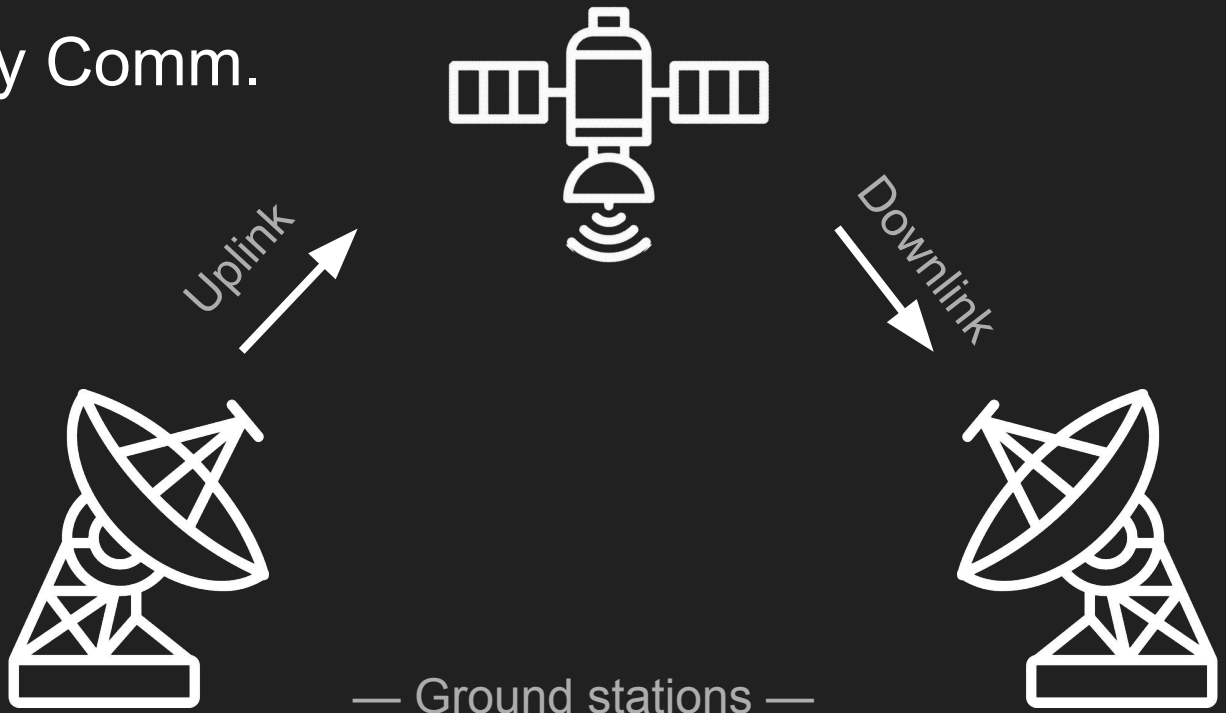
Type de satellites

Différentes orbites:

- LEO: de 160 km à 2000 km
- MEO: 2000 km à 35000 km
- GEO: 35786 km

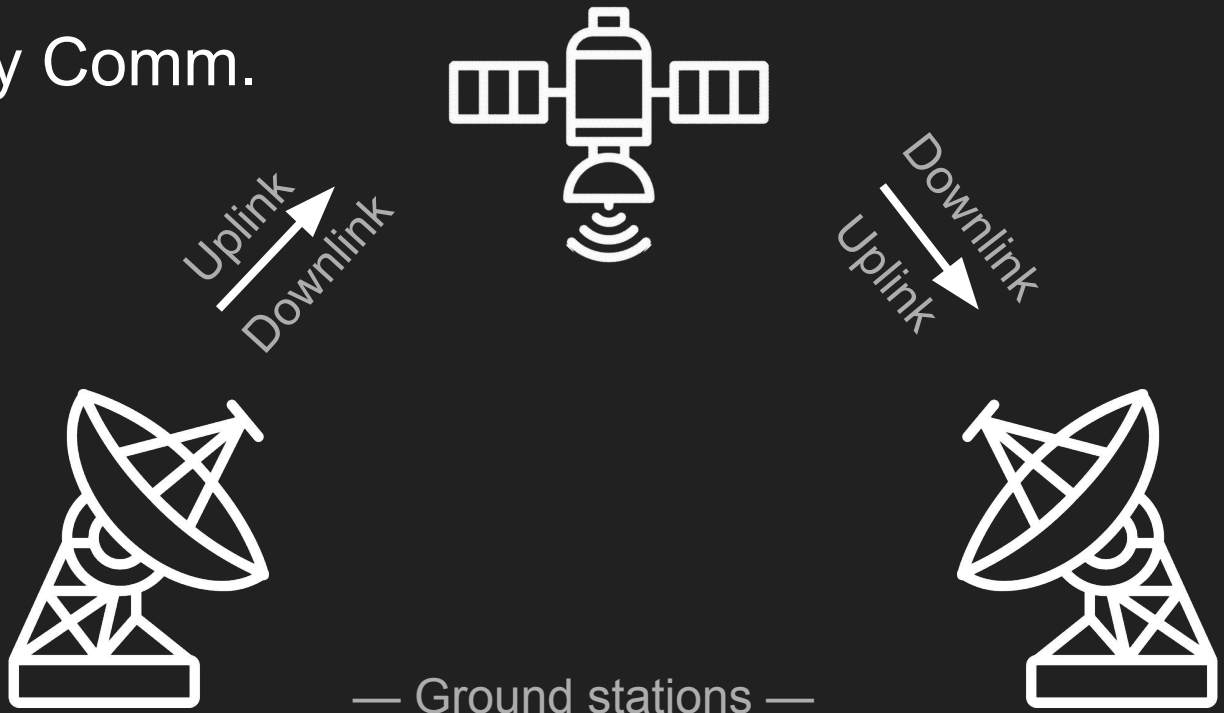
Méthode de communication.

One way Comm.



Méthode de communication.

Two way Comm.



Bande de fréquences:

- **UHF** (300 MHz à 3 GHz)
- **Bande L** (1 à 2 GHz)
- **Bande S** (2 à 4 GHz)
- **Bande Ku** (12 à 18 GHz)
- **Bande Ka** (26,5 à 40 GHz)

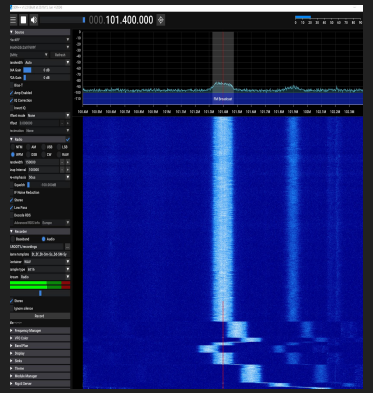
Aperçu des protocols de communications.

Introduction à la SDR

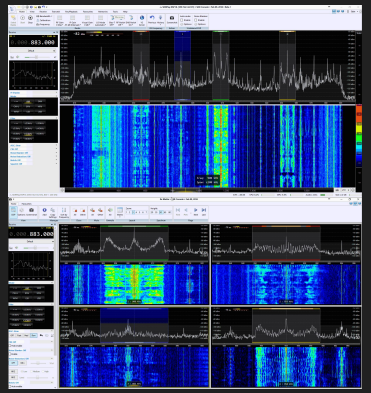
Software defined radio processing

Analog/Hardware defined radio processing

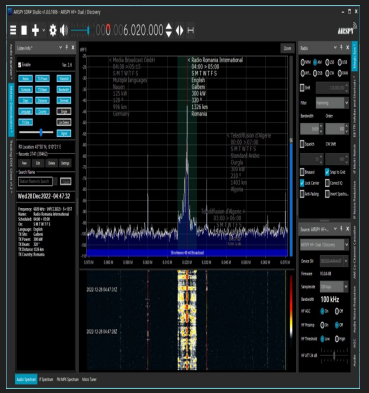
Introduction à la SDR



https://www.sdrpp.org/



https://www.sdr-radio.com/console



https://www.rtl-sdr.com/tag/sdrsharp/

https://www.sdrplay.com/sdruno/

Introduction à la SDR

https://greatscottgadgets.com/hackrf/one/

https://www.nooelec.com/store/sdr/sdr-receivers/nesdr-smartee-xtr.html

https://kb.ettus.com/B200/B210/B200mini/B205mini

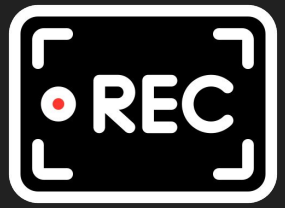
Antenne: DIY

Antenne S-band

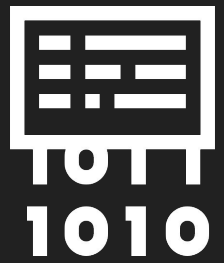
Fabriquer son antenne | Comprendre le fonctionnement des antennes

Le rôle de la SDR dans l’audit de Satellites

- SatNOGS
- OrbitalFocus
- n2yo
- SAT Passes
- AMSAT



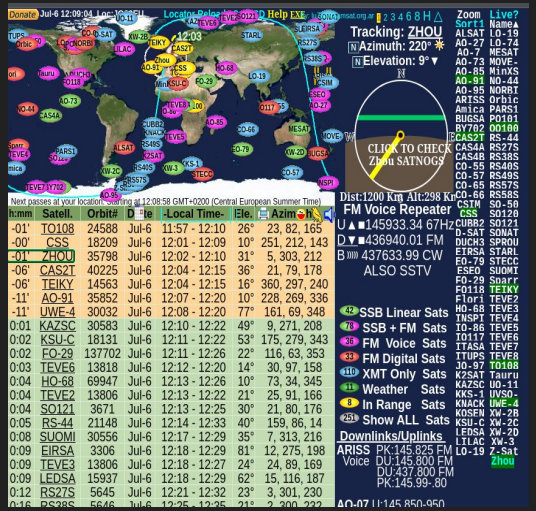
- https://www.sdrpp.org/
- https://www.sdr-radio.com/console
- https://www.rtl-sdr.com/tag/sdrsharp/
- https://www.sdrplay.com/sdruno/

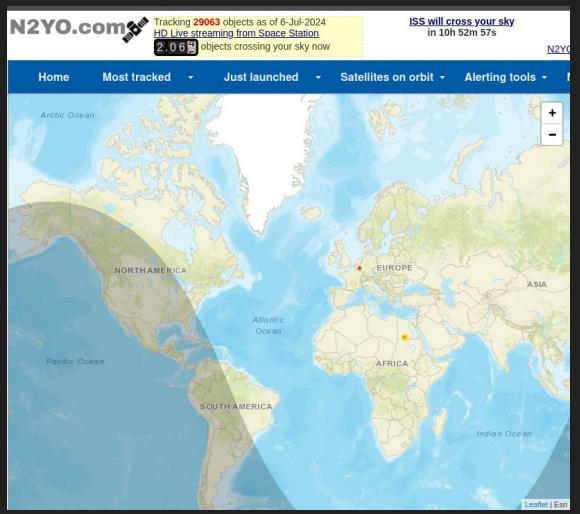


- https://github.com/ssloxford/gsextract
- https://github.com/jopohl/urh

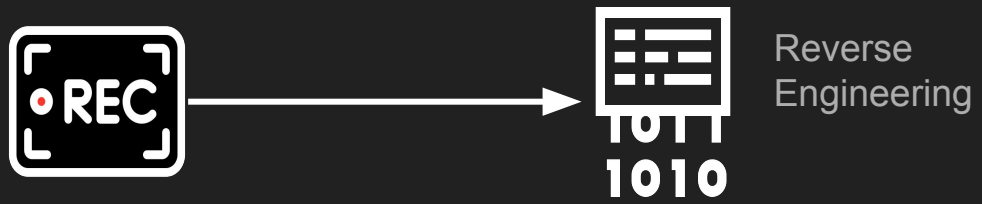
Prédiction de passage

- SatNOGS
- OrtbitalFocus
- n2yo
- SAT Passes
- AMSAT





Rétro-ingénierie de signal



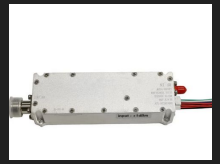
- https://www.youtube.com/watch?v=8kIxlMIGctc
- https://github.com/ssloxford/gsextract
- https://github.com/jopohl/urh

Attaquer un satellite, est-ce possible ?

Attaqueur configuration:

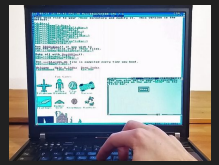
- BlackSDR B200 mini: 500€

- RF Amplifier 50w: 220€



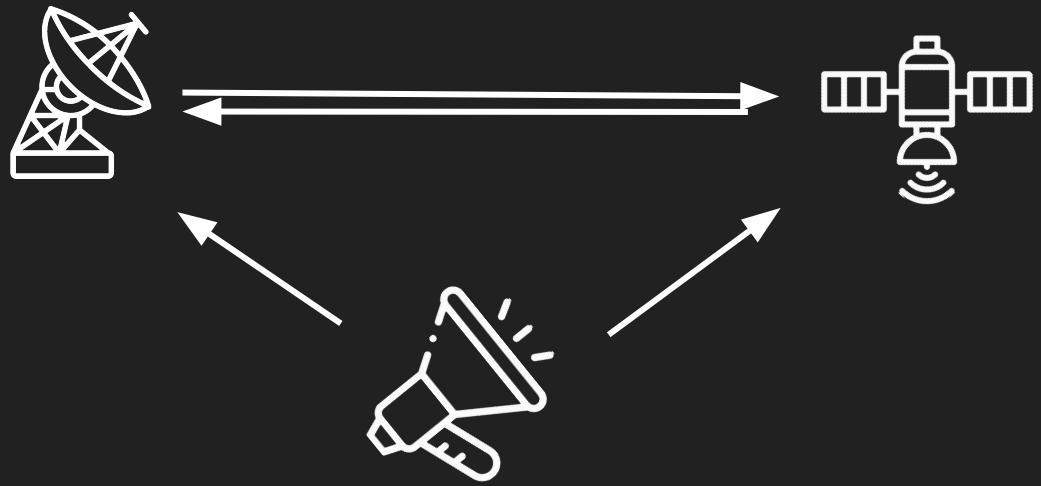
- Monture Azimuthal: 150-500€

- Laptop milieu de gamme: 750-1000€

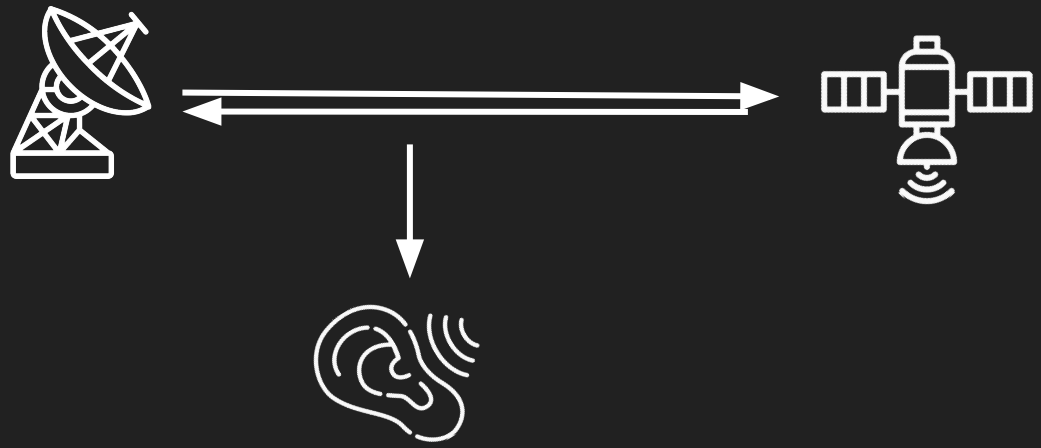


Antenne: DIY

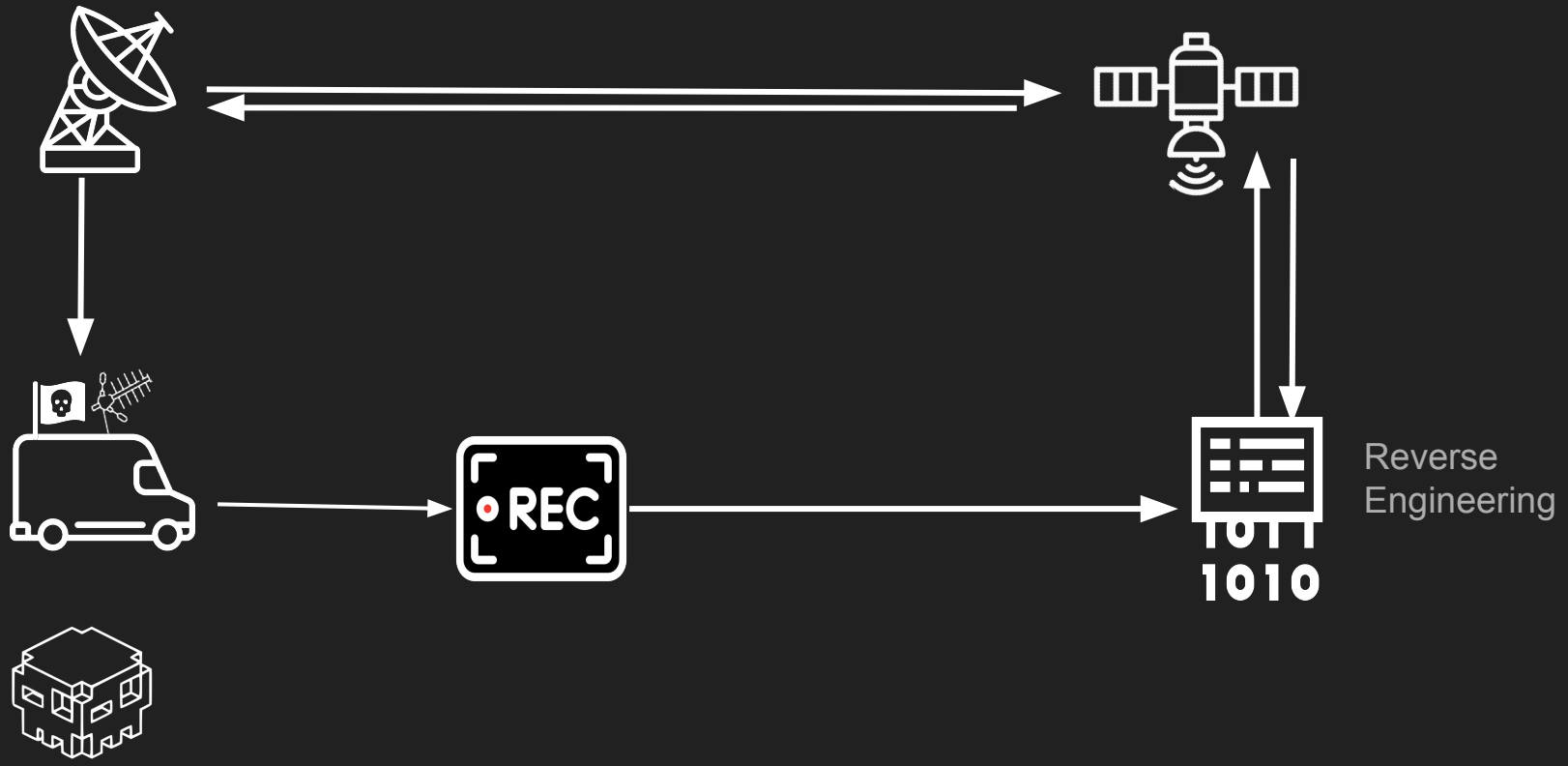
Jamming - Brouillage



Eavesdropping - écoute

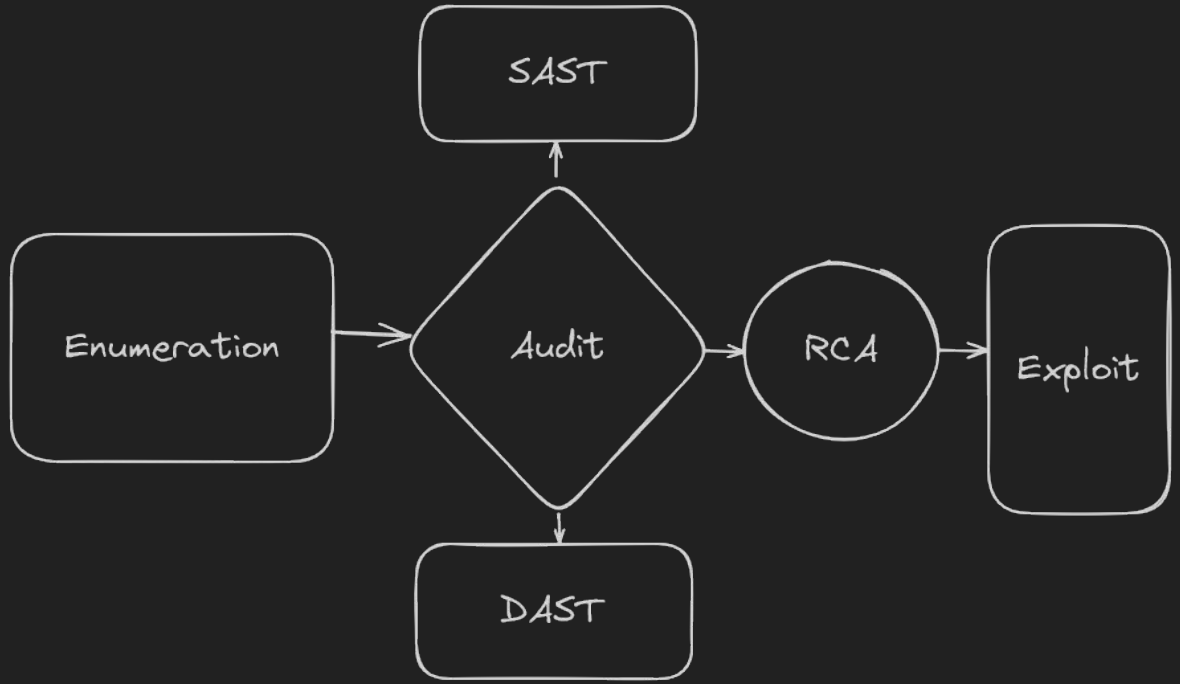


Link takeover - Prise de control



Audit des de logiciels Satellites.

Workflow:



Audit des protocols réseaux SAT.

- ESA Open Source Software: https://www.esa.int/Enabling_Support/Space_Engineering_Technology/Radio_Frequency_Systems/Open_Source_Software_Resources_for_Space_Downstream_Applications%20#Table3
- Nasa core Flight System: https://cfs.gsfc.nasa.gov/
- Cubesat: https://en.wikipedia.org/wiki/Cubesat_Space_Protocol
- LibCSP: https://github.com/libcsp/libcsp/tree/develop/doc

Introduction aux techniques de DAST/SAST.

script d'installation:
https://github.com/20urc3/Talks/blob/main/leHack/sdast.sh

Appliquer les SAST

Recherche de vulnérabilité: Résultats SAST

Bug Summary

Bug Type	Quantity	Display?
All Bugs	27	☑
Logic error		
Branch condition evaluates to a garbage value	2	☑
Dereference of null pointer	4	☑
Function call with invalid argument	1	☑
Uninitialized argument value	3	☑
Memory error		
Memory leak	1	☑
Stream handling error		
Resource leak	1	☑
Unused code		
Dead assignment	15	☑

Scan Status

Scanning 478 files (only git-tracked) with 35 Code rules:

CODE RULES

Language	Rules	Files
c	35	135
cpp	35	47

Origin	Rules
Custom	35

SUPPLY CHAIN RULES

No rules to run.

PROGRESS

100% 0:00:00

221 Code Findings

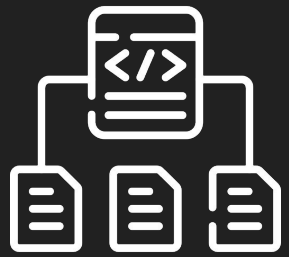
Line ↓	File	Message
cpp/tainted-format-string cpp/tainted-format-string 2		
⚠ 284	.c	The value of this argument may come from buffer read by read and
⚠ 1697	.c	The value of this argument may come from an environment variable

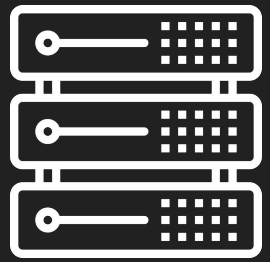
```
warning: Possible null pointer dereference: last [nullPointer]
```

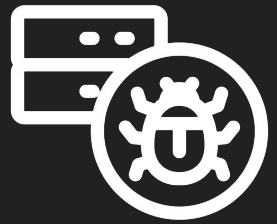
Recherche de vulnérabilité: DAST / Fuzzing

Introduction au fuzzing

Comment s’applique le fuzzing à libcsp/ZeroMQ

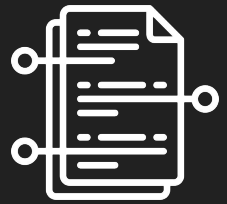




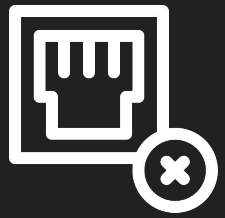


[Le fuzzing de protocol réseau]

Comment s’applique le fuzzing à libcsp/ZeroMQ



Spécifier une grammaire



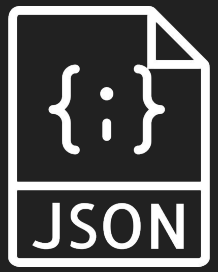
de-sock le programme cible

Configurer AFL++ avec votre grammar-mutator

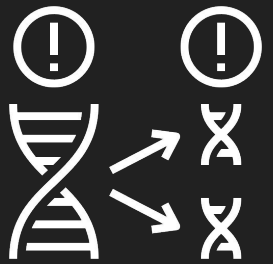
AFL-Grammar | Preeny

Créer votre custom-grammar mutator

ZeroMQ RFC



Custom grammar



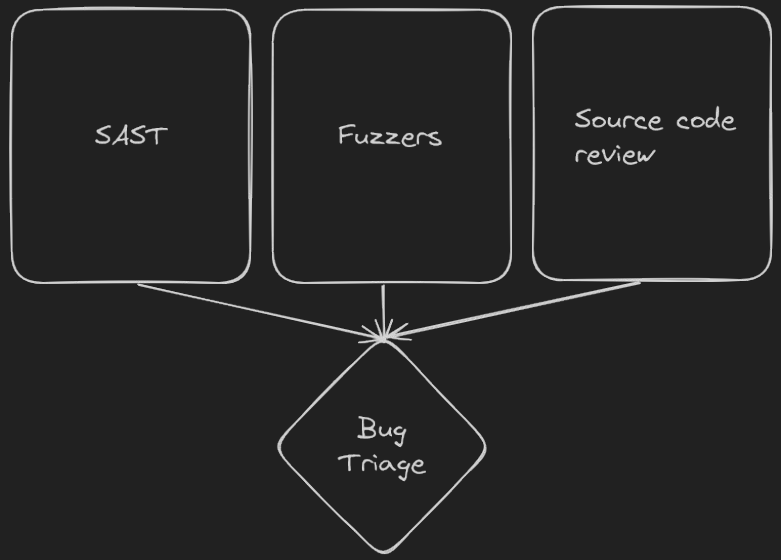
Créer votre mutator

ZMQ-Mutator | ZMQ-Grammar

Recherche de vulnérabilité: Résultats DAST

```
output/Secondary3/crashes/id:000001,sig:11,src:000008+000231,time:2453485,execs:8835,op:splice,rep:9
output/Secondary3/crashes/id:000002,sig:05,src:000299+000000,time:26398746,execs:77328,op:splice,rep:2
output/Secondary3/crashes/id:000000,sig:06,src:000233+000014,time:1635427,execs:6426,op:splice,rep:2
output/Secondary1/crashes/id:000000,sig:06,src:000232+000148,time:932389,execs:4401,op:splice,rep:1
output/Secondary25/crashes/id:000000,sig:11,src:000239+000149,time:637558,execs:3545,op:splice,rep:1
output/Secondary9/crashes/id:000000,sig:11,src:000249+000019,time:4198117,execs:13451,op:splice,rep:1
output/Secondary18/crashes/id:000000,sig:11,src:000008+000018,time:1604173,execs:6220,op:splice,rep:2
output/Master/crashes/id:000002,sig:06,src:000308+000263,time:26180146,execs:74096,op:splice,rep:2
output/Master/crashes/id:000000,sig:06,src:000250,time:4665670,execs:14991,op:havoc,rep:9
output/Master/crashes/id:000001,sig:05,src:000308+000292,time:22742297,execs:64702,op:splice,rep:8
output/Secondary28/crashes/id:000000,sig:11,src:000240+000091,time:6774476,execs:20200,op:splice,rep:2
output/Secondary4/crashes/id:000000,sig:11,src:000239+000120,time:833136,execs:4028,op:splice,rep:3
output/Secondary8/crashes/id:000000,sig:06,src:000008,time:2914454,execs:10190,op:havoc,rep:13
output/Secondary26/crashes/id:000000,sig:05,src:000272+000000,time:21597256,execs:61928,op:splice,rep:3
output/Secondary29/crashes/id:000000,sig:06,src:000298+000008,time:23009741,execs:66202,op:splice,rep:7
output/Secondary17/crashes/id:000000,sig:11,src:000239+000023,time:740805,execs:3779,op:splice,rep:7
output/Secondary6/crashes/id:000004,sig:06,src:000260+000253,time:14828781,execs:59976,op:splice,rep:2
output/Secondary6/crashes/id:000002,sig:06,src:000024+000014,time:364500,execs:3113,op:splice,rep:5
output/Secondary6/crashes/id:000001,sig:06,src:000024+000014,time:361935,execs:3106,op:splice,rep:6
output/Secondary6/crashes/id:000000,sig:06,src:000024+000103,time:344865,execs:3047,op:splice,rep:5
output/Secondary6/crashes/id:000006,sig:06,src:000304+000303,time:23574066,execs:93125,op:splice,rep:3
output/Secondary6/crashes/id:000005,sig:06,src:000306+000303,time:23017984,execs:90974,op:splice,rep:4
output/Secondary6/crashes/id:000003,sig:11,src:000239+000148,time:2379686,execs:10980,op:splice,rep:7
output/Secondary6/crashes/id:000007,sig:06,src:000295+000262,time:23710211,execs:93657,op:splice,rep:2
output/Secondary19/crashes/id:000000,sig:06,src:000238+000014,time:464721,execs:3525,op:splice,rep:12
output/Secondary19/crashes/id:000001,sig:06,src:000262+000105,time:15250442,execs:61029,op:splice,rep:3
output/Secondary19/crashes/id:000003,sig:06,src:000311+000259,time:24735673,execs:97306,op:splice,rep:1
output/Secondary19/crashes/id:000002,sig:06,src:000273+000283,time:16706086,execs:66672,op:splice,rep:1
output/Secondary10/crashes/id:000000,sig:11,src:000023+000092,time:8082990,execs:24065,op:splice,rep:3
output/Secondary10/crashes/id:000001,sig:05,src:000291+000305,time:25008157,execs:70764,op:splice,rep:2
output/Secondary7/crashes/id:000000,sig:06,src:000277+000264,time:18270034,execs:54401,op:splice,rep:1
```

Recherche de vulnérabilité



Recherche de vulnérabilité: Bug triage

?

RCA - Analyse de vulnérabilité

CVE in libcsp: https://www.cvedetails.com/cve/CVE-2016-8598/

Buffer overflow in the zmq interface in csp_if_zmqhub.c in the libcsp library v1.4 and earlier allows hostile computers connected via a zmq interface to execute arbitrary code via a long packet.

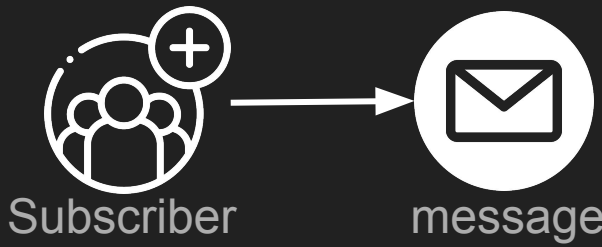
RCA - Analyse de vulnérabilité

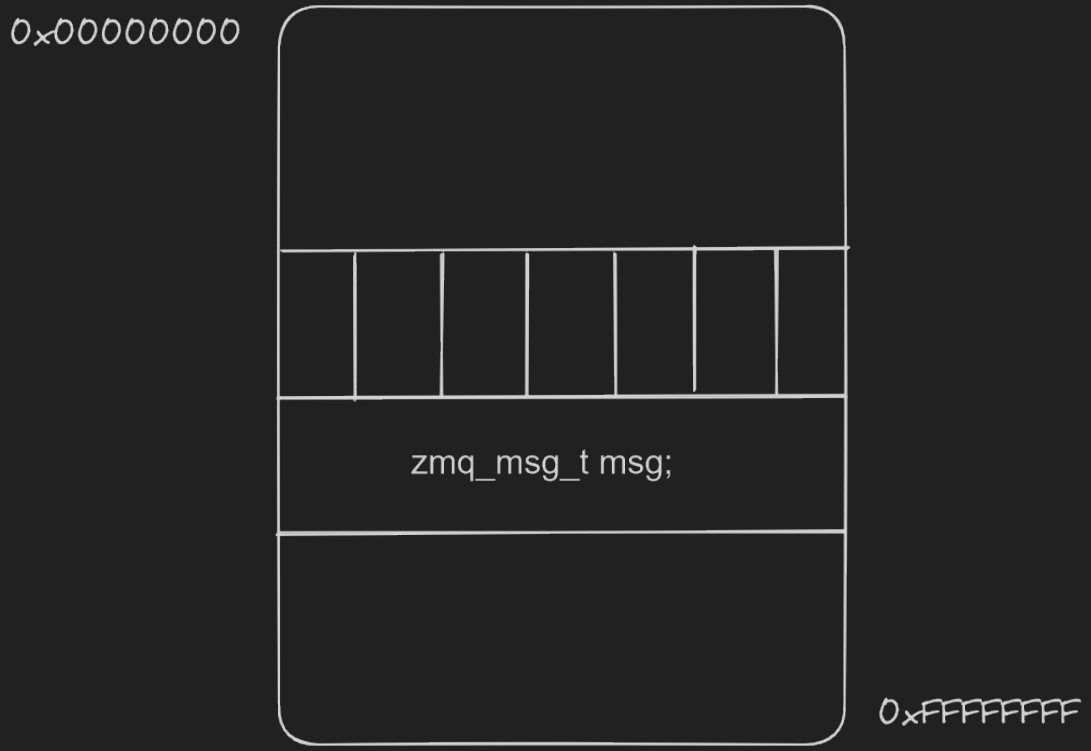
```
CSP_DEFINE_TASK(csp_zmqhub_task) {
    while(1) {
        zmq_msg_t msg;
        assert(zmq_msg_init_size(&msg, 1024) == 0);
        /* Receive data */
        if (zmq_msg_recv(&msg, subscriber, 0) < 0) {
            zmq_msg_close(&msg);
            csp_log_error("ZMQ: %s", zmq_strerror(zmq_errno()));
            continue;
        }
        int datalen = zmq_msg_size(&msg);
        if (datalen < 5) {
            csp_log_warn("ZMQ: Too short datalen: %u", datalen);
            while(zmq_msg_recv(&msg, subscriber, ZMQ_NOBLOCK) > 0)
            zmq_msg_close(&msg);
            continue;
        }
        /* Create new csp packet */
        csp_packet_t * packet = csp_buffer_get(256);
        if (packet == NULL) {
            zmq_msg_close(&msg);
            continue;
        }

        /* Copy the data from zmq to csp */
        char * satidptr = ((char *) &packet->id) - 1;
        memcpy(satidptr, zmq_msg_data(&msg), datalen);
        packet->length = datalen - 4 - 1;
        /* Queue up packet to router */
        csp_qfifo_write(packet, &csp_if_zmqhub, NULL);
        zmq_msg_close(&msg);
    }
    return CSP_TASK_RETURN;
}
```

RCA - Analyse de vulnérabilité

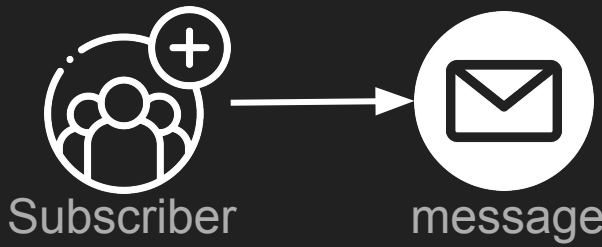
```
00125 /*****************************************************************************/
00126 /*  0MQ message definition.                                                   */
00127 /*****************************************************************************/
00128
00129 typedef struct {unsigned char _ [32];} zmq_msg_t;
00130
00131 typedef void (zmq_free_fn) (void *data, void *hint);
00132
00133 ZMQ_EXPORT int zmq_msg_init (zmq_msg_t *msg);
00134 ZMQ_EXPORT int zmq_msg_init_size (zmq_msg_t *msg, size_t size);
00135 ZMQ_EXPORT int zmq_msg_init_data (zmq_msg_t *msg, void *data,
00136     size_t size, zmq_free_fn *ffn, void *hint);
00137 ZMQ_EXPORT int zmq_msg_close (zmq_msg_t *msg);
00138 ZMQ_EXPORT int zmq_msg_move (zmq_msg_t *dest, zmq_msg_t *src);
00139 ZMQ_EXPORT int zmq_msg_copy (zmq_msg_t *dest, zmq_msg_t *src);
00140 ZMQ_EXPORT void *zmq_msg_data (zmq_msg_t *msg);
00141 ZMQ_EXPORT size_t zmq_msg_size (zmq_msg_t *msg);
00142 ZMQ_EXPORT int zmq_getmsgopt (zmq_msg_t *msg, int option, void *optval,
00143     size_t *optvallen);
```

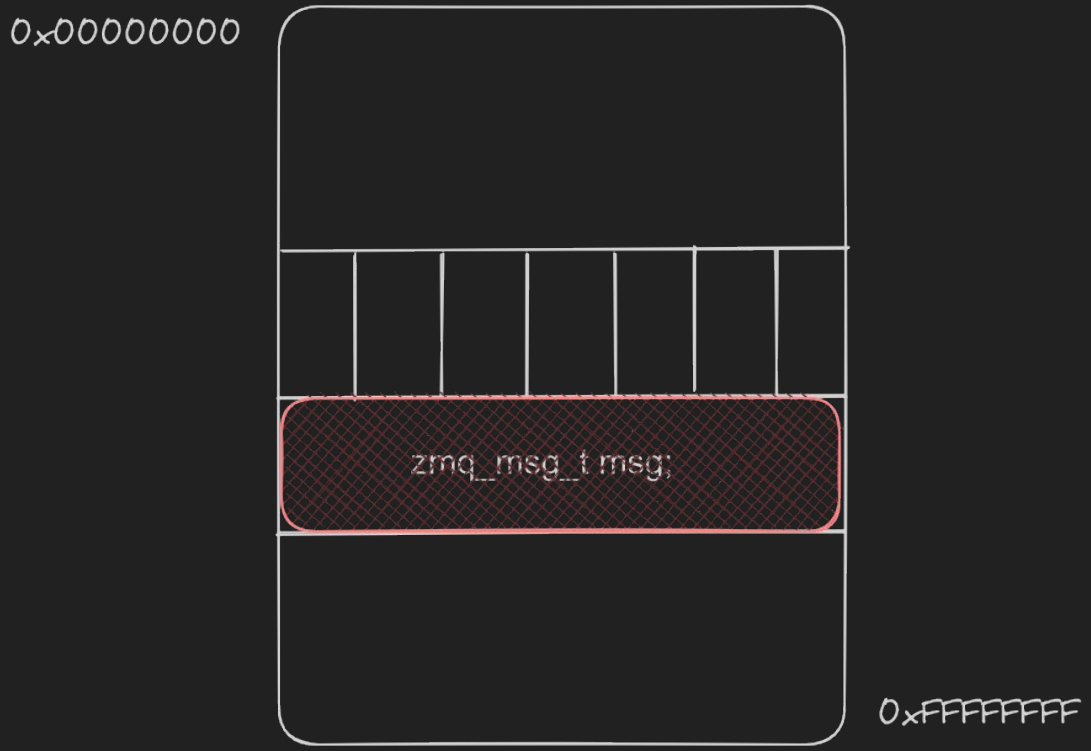
RCA - Analyse de vulnérabilité



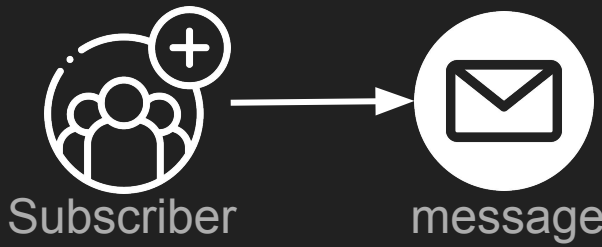


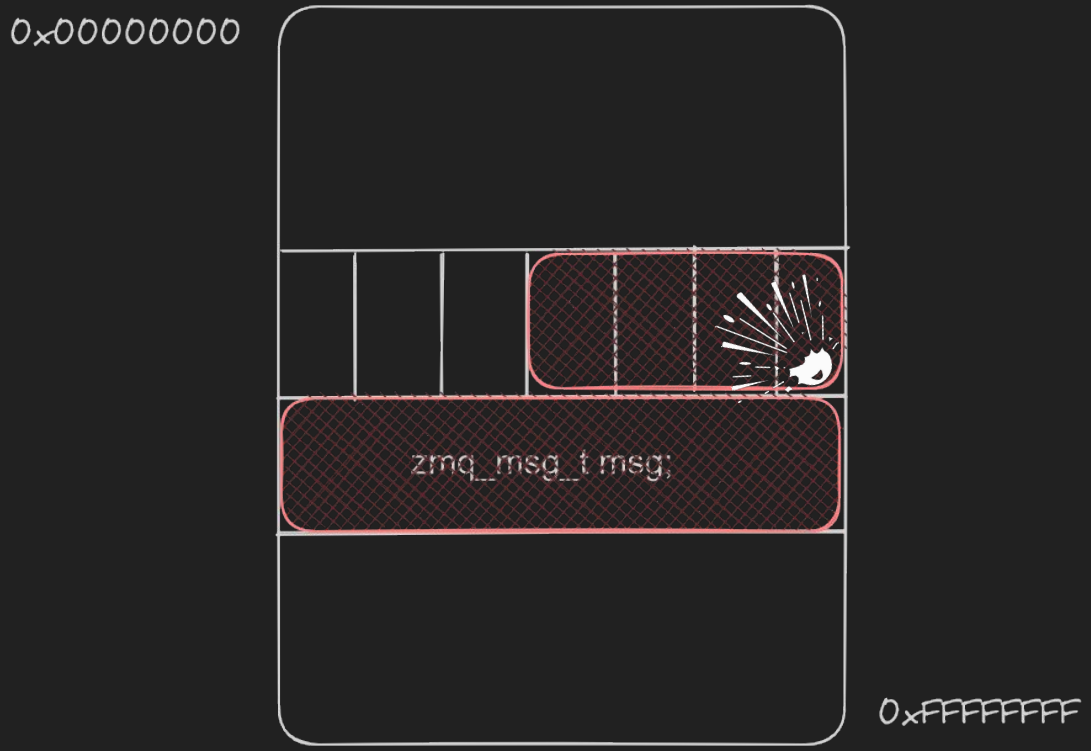
RCA - Analyse de vulnérabilité



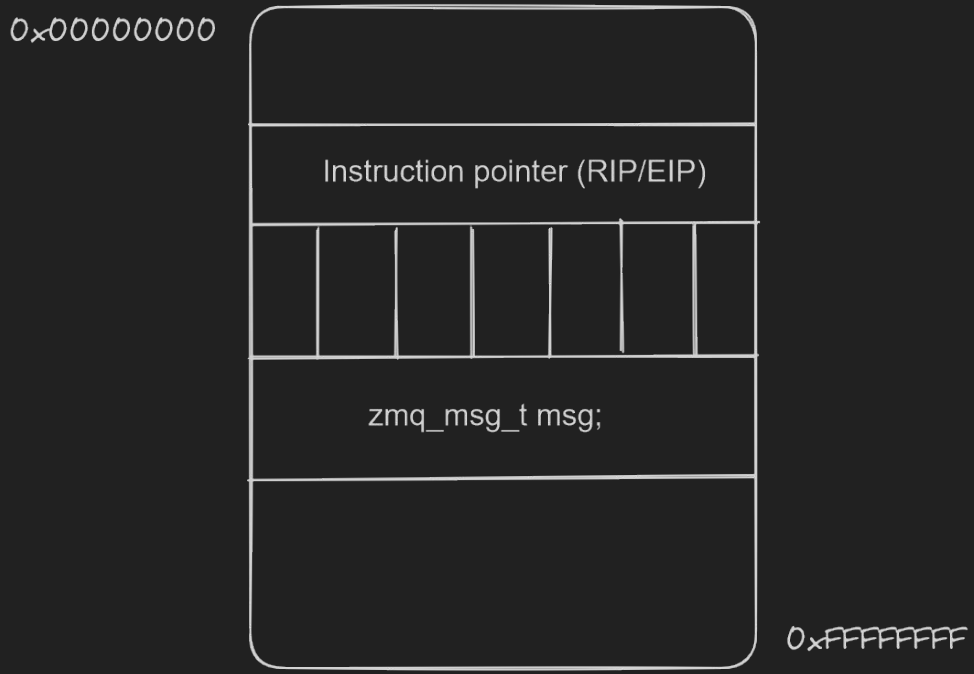


RCA - Analyse de vulnérabilité



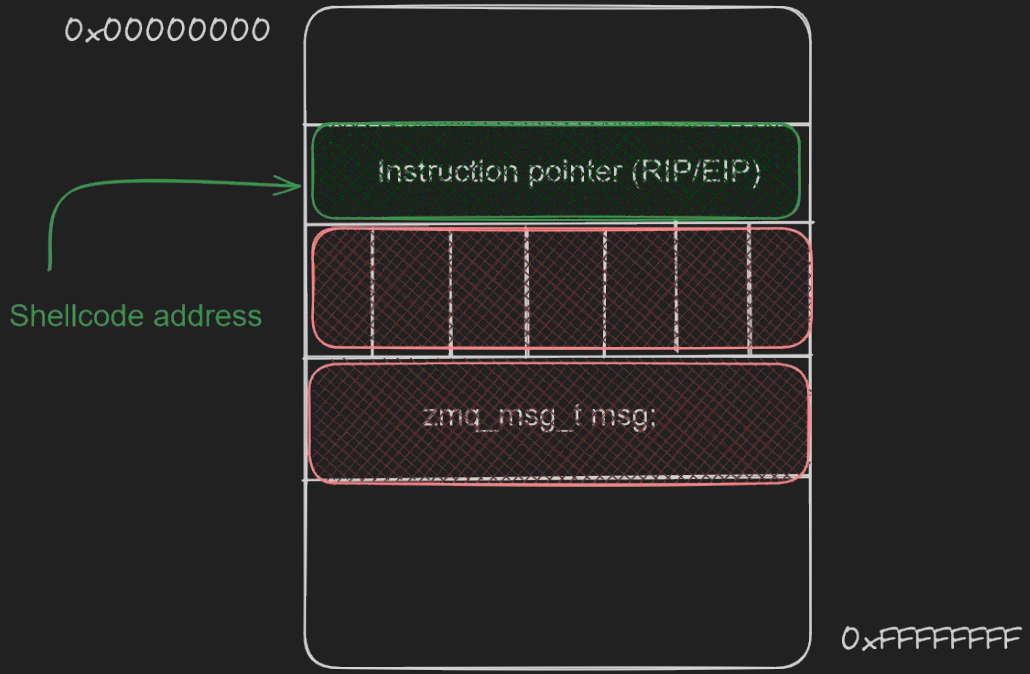


Techniques d'exploitation



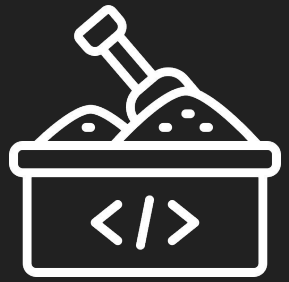
[Smashing the stack for fun and profit]

Techniques d'exploitation

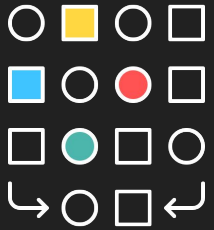


[Smashing the stack for fun and profit]

Exploit mitigations

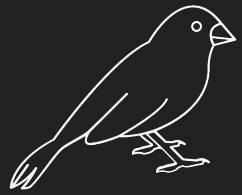


Sandboxing



ASLR

CFI

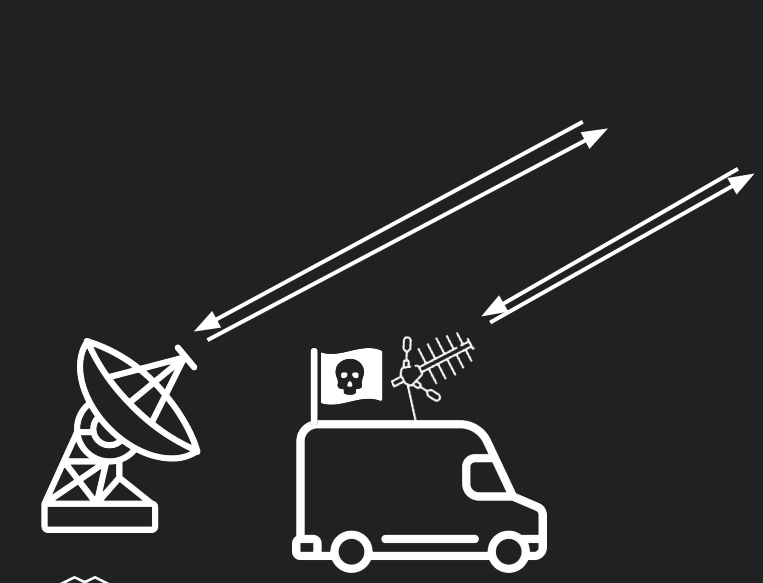


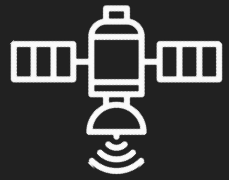
Stack canary

Comprendre les mitigations:
- NCCGroup liste

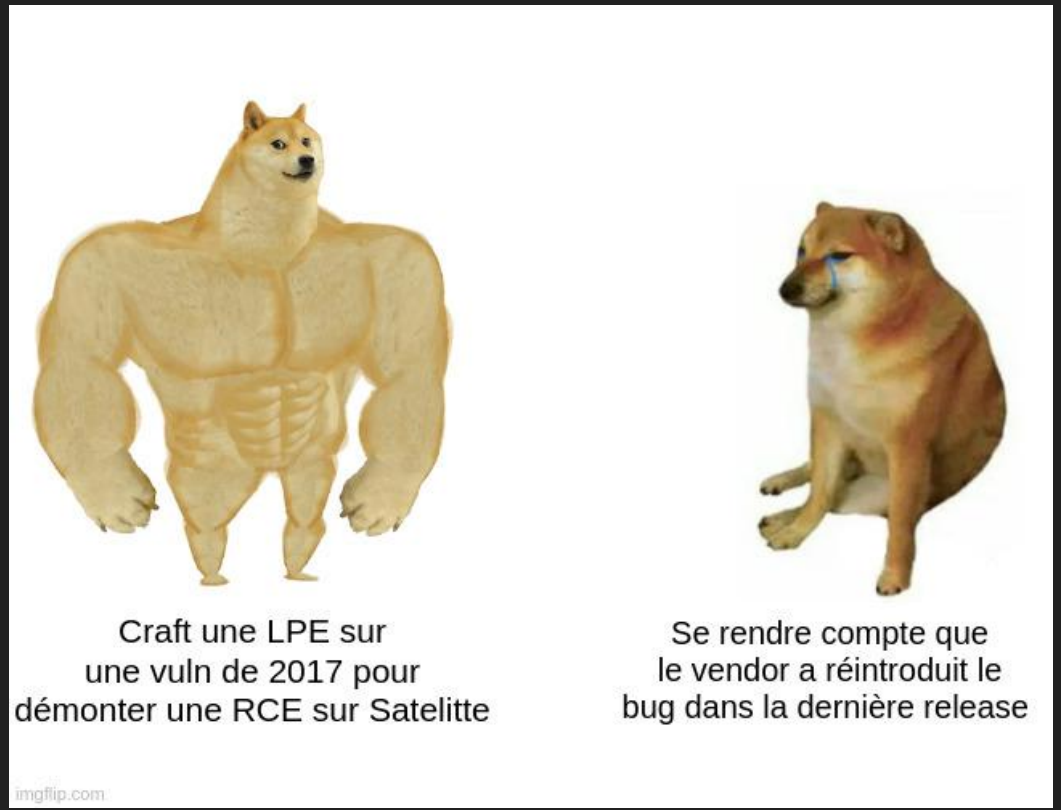
Comprendre les mitigations bypass:
- pwn.college
- OffByOne

Scénario d’attaque





Démo vidéo de la RCE



Manipulation de données

- Manipulation de données
- Perturbation de systèmes critiques
- Dommages physiques

Besoin d’audit de cybersécurité pour les Satellites.

Bonus: Créez votre lab

- HackRF, BlackSDR, RTL-SDR dongle
- Logiciel SDR
- RaspberryPI Simulator: https://github.com/alanbjohnston/CubeSatSim
- Cable RX/TX: SMA male coaxial cable
- Alternative OS: FreeRTOS
- Alternative OS: Yocto Linux

Relecteurs

- @djnn1337 @TQN @Vsim
- RF Pentest team @ Nexova

Merci :-)

Références

- https://web.archive.org/web/20111118180309/http://www.uscc.gov/annual_report/2011/annual_report_full_11.pdf
- https://securingdemocracy.gmfus.org/incident/russian-cyberattack-takes-down-satellite-communications-in-ukraine
- https://cyberconflicts.cyberpeaceinstitute.org/law-and-policy/cases/viasat
- https://english.kyodonews.net/news/2024/06/c9eed01c27a4-japan-space-agency-suffers-multiple-cyberattacks-since-last-year.html
- https://github.com/jekhokie/raspberry-noaa-v2
- https://github.com/SatDump/SatDump
- https://alicja.space/
- https://github.com/ssloxford/gsextract
- https://universemagazine.com/en/how-to-destroy-a-satellite-without-firing-a-single-shot/
- https://github.com/AlexandreRouma/SDRPlusPlus
- https://github.com/jopohl/urh
- https://docs.google.com/document/d/1yjAO3jTBa9lAFIuiteK_GLWh7-Xk-kSD2d0DUxQe_vU/edit
- https://www.youtube.com/watch?v=gTISW-5Uy6I
- https://github.com/antonio-morales/Apache-HTTP-Fuzzing
- https://github.com/TImada/raspi4_freertos
- https://insecure.org/stf/smashstack.html
- https://github.com/zardus/preeny
- https://github.com/AFLplusplus/Grammar-Mutator
- https://www.youtube.com/watch?v=FWCN_uI5ygY
- https://www.rtl-sdr.com/building-an-s-band-antenna-for-the-hackrf/
- https://www.oreilly.com/library/view/the-art-of/0321444426/
- https://www.nexovagroup.eu/en