Secure Micropatching on the ISS

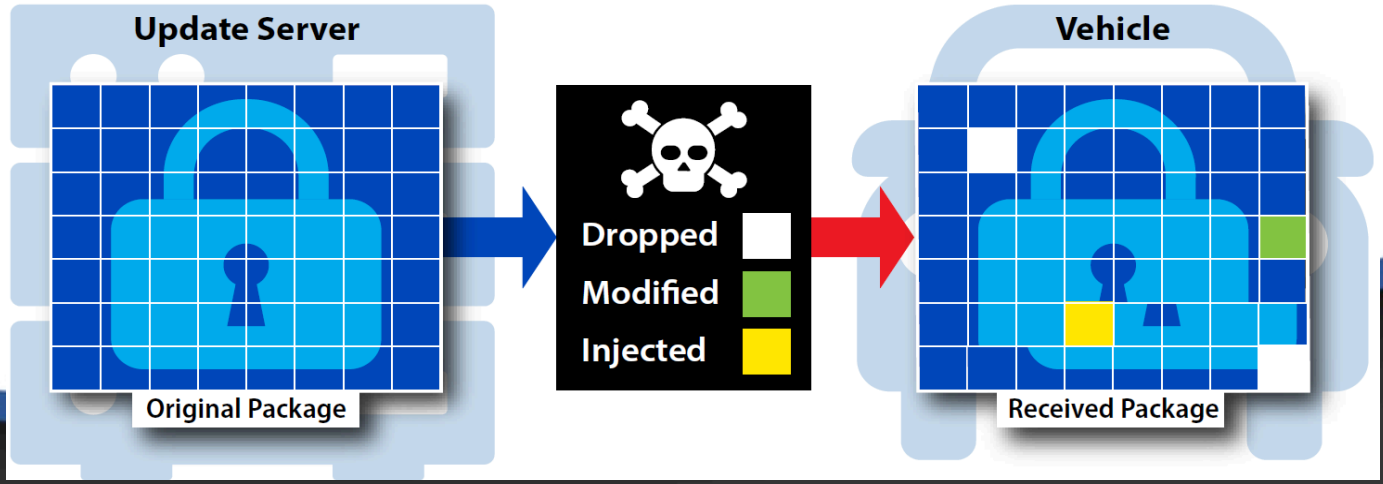
Henry Haswell
August 11 2023

Background: Current State of OTA Updates in Space

- OTA updates have small window to deploy
- At best, failed updates need to resend
- At worst, failed updates cause system errors

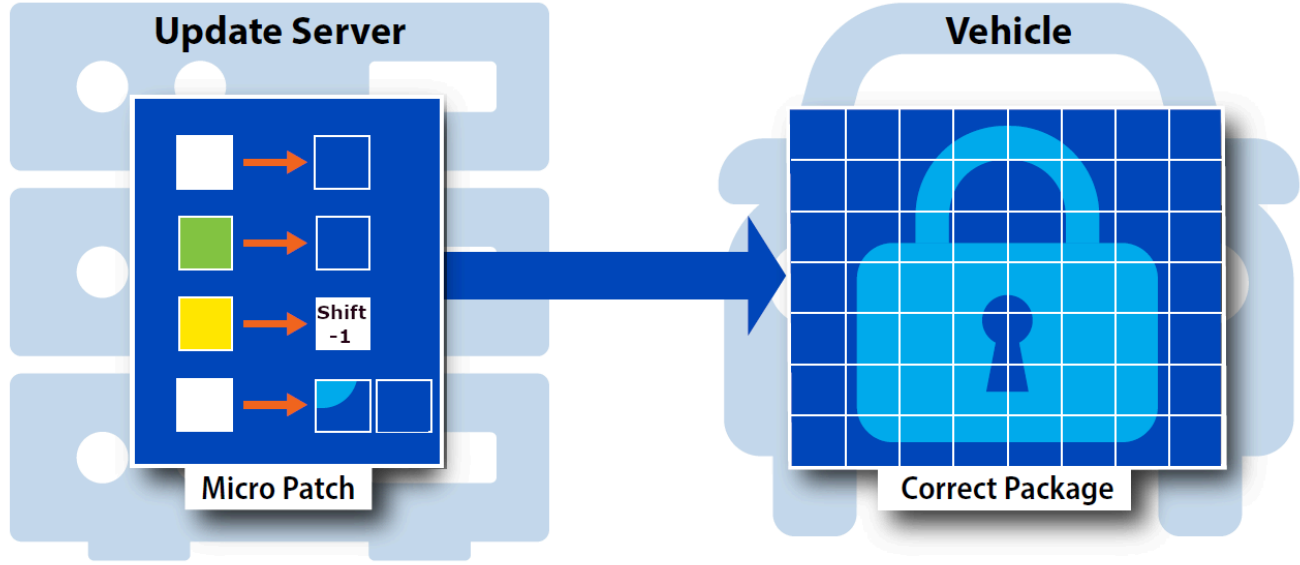
Common Mitigations for Data Errors

- CRC is standard for message authentication
- Common EDAC methods:
  - BCH
  - Reed-Solomon
- Correcting larger bit errors needs exponentially more memory



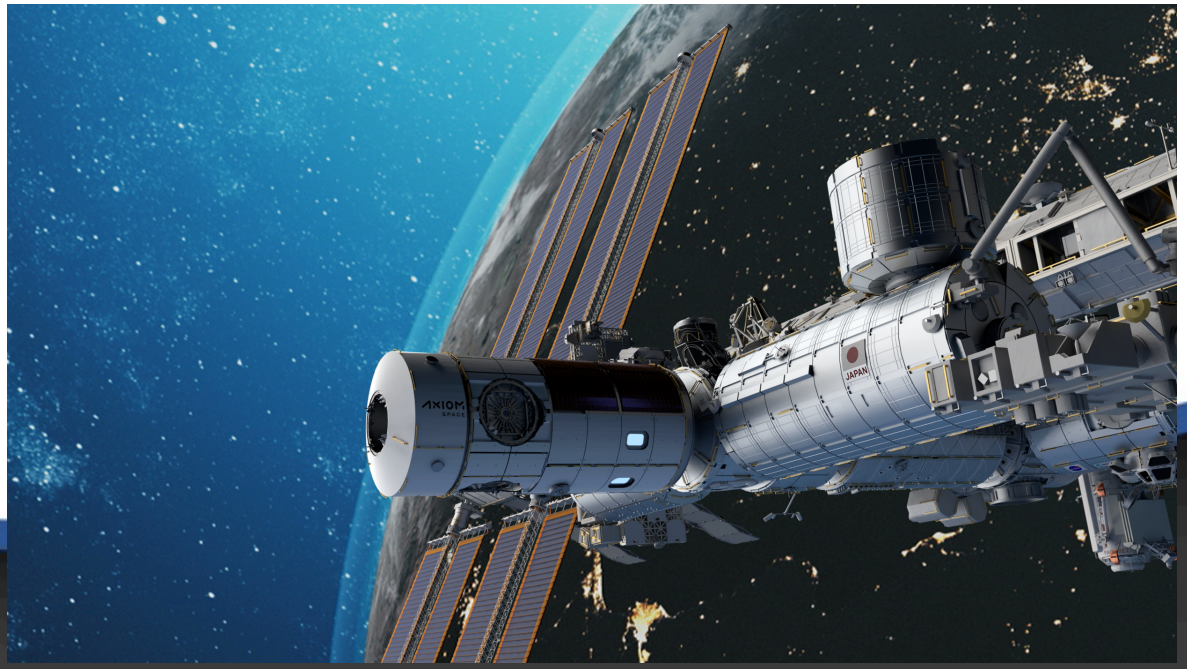
Previous Work In Secure Micropatching

- Allows for correction of insertion, deletion, or modification of data
- Reduced transmit time compared to resending a full update
- Deployed on ground vehicle environment

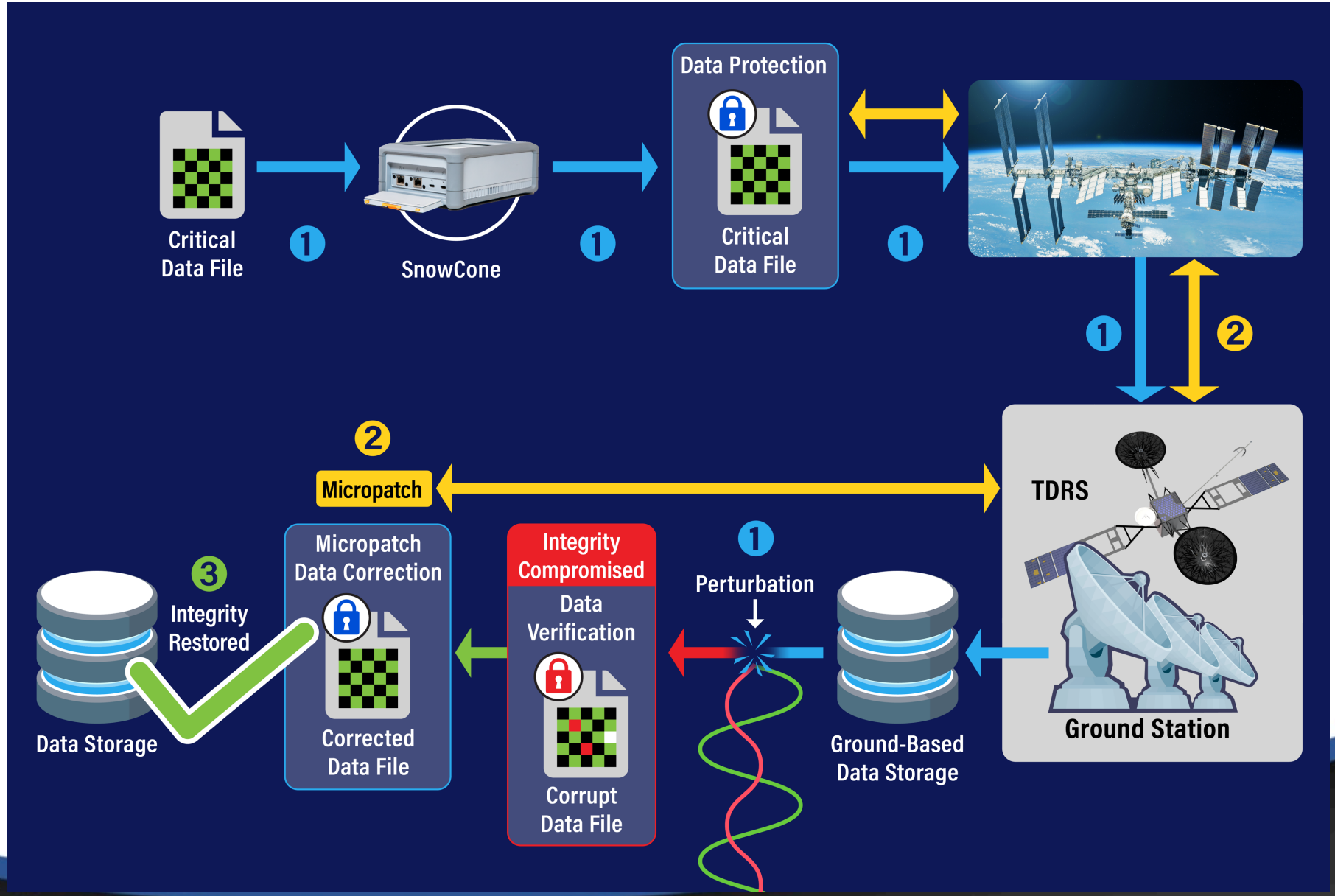


Why Team to Demo on ISS?

- US side of ISS has AWS Snowcone onboard
- Running our micropatching tool could help demonstrate cyber use case for space servers
- First step to improve micropatching for space environments

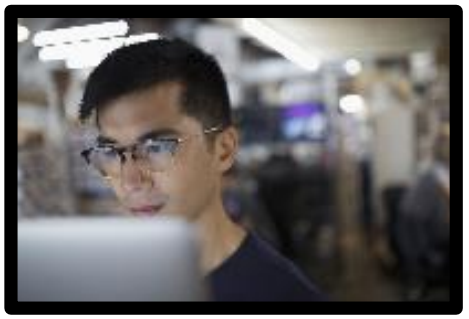


Micropatching at a High Level



Phases of Internal Research

- **Phase #1:**
  - **Ground HIL Testing**
- **Phase #2:**
  - **Optimize Micropatching for Space**
- **Phase #3:**
  - **Demonstration of Micropatching on ISS Snowcone**

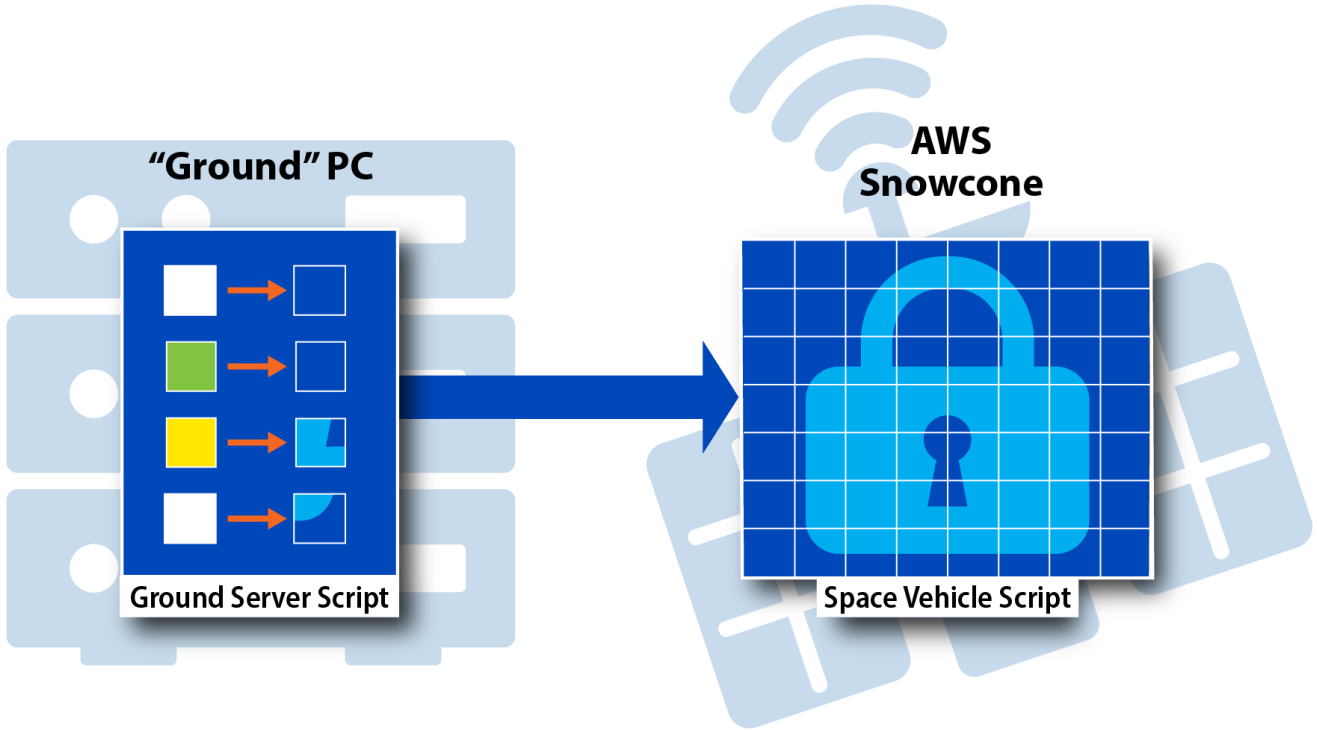


AWS Snowcone

- **Compact Server for Edge Computing**
- **Utilize Amazon EC2 Linux Instance**
- **14TB of SSD storage in ~4.5 pounds**

HIL Ground Simulation

- Snowcone acting as space asset
- Lab laptop acting as ground station
- Implemented client-server setup



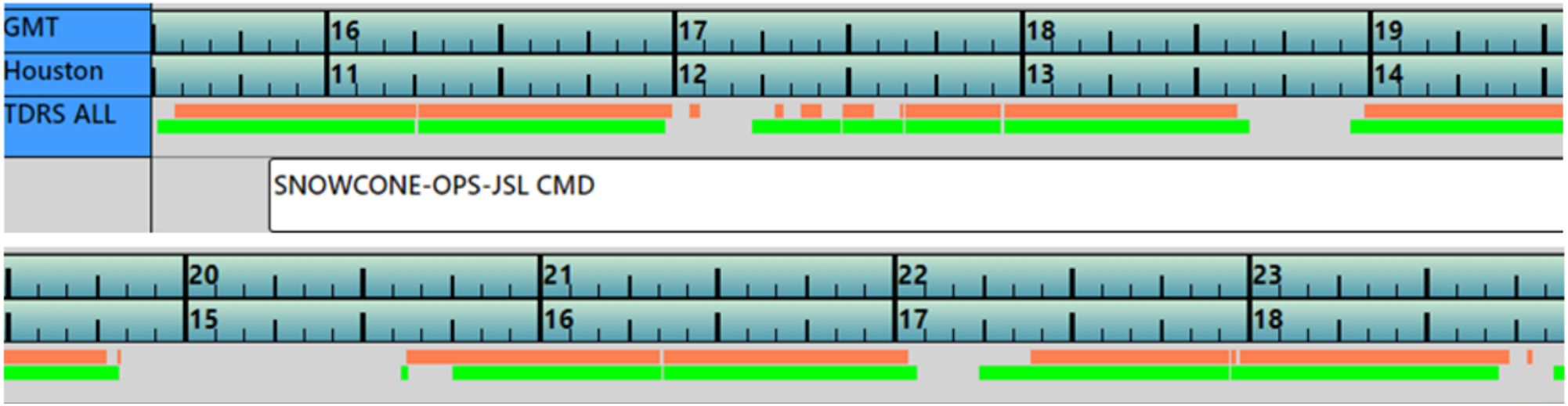
Ground Simulation Results

- **Simulated expected latency of space network**
- **Compared performance to full resend and EDAC**

	Open Network					Simulated Latency				
File	**40 KB**	**400 KB**	**1 MB**	**5 MB**	**100 MB**	**40 KB**	**400 KB**	**1 MB**	**5 MB**	**100 MB**
DBS	0.35	0.43	0.51	0.92	12.91	68.92	77.20	82.58	98.76	127.71
Recursive	0.02	0.14	0.15	0.59	12.56	4.11	10.60	13.19	43.71	59.33
Full	0.32	5.23	12.75	59.83	1186.24	3.45	56.36	143.64	718.92	1565.92
Full/DBS	**0.91**	**12.16**	**25.00**	**65.03**	**91.89**	**0.05**	**0.73**	**1.74**	**7.28**	**12.26**
Full/R	**16.00**	**37.36**	**85.00**	**101.41**	**94.45**	**0.84**	**5.32**	**10.89**	**16.45**	**26.39**

Changes Needed For ISS Demo

- **Reducing Latency of Micropatches**
- **Accounting for potential LOS**



Running the Demo

- Worked with Axiom Space staff on ground to start ground script
- Astronaut on ISS startup Snowcone
- Variety of file size and error rates
- ~5Mb/s transfer speed
- ~800ms latency
- Expected LOS about every 40 minutes

ISS Demo Results

- **Micropatching faster than python implementation of EDAC when >1mb file**
- **Multiple LOS did not impact micropatches**

File	40 KB	400 KB	1 MB	5 MB	100 MB
DBS	212.95	251.77	225.89	264.58	312.79
Recursive	7.31	22.81	21.51	27.56	46.29
Full	2.26	8.24	16.31	73.70	1400.63
Full/DBS	**0.01**	**0.03**	**0.07**	**0.28**	**4.48**
Full/R	**0.31**	**0.36**	**0.76**	**2.67**	**30.26**

Future Work and Concluding Remarks

- Space assets face a need for reduced downtime and increased file integrity
- Micropatching corrects insertion, deletion, and modification of bits
- Reduced repair time compared to a full resend or python implementation of EDAC
- This marks the first micropatching deployment to an asset on the ISS