Perilous Paths: The Cybersecurity Issues of a Space-Based Cloud Imagery Processing System

AIAA ASCEND 2022, TECH.SEC-06: Cybersecurity Frameworks and Architectures for Operational Systems II

October 25, 2022, Las Vegas, NV

Bryce L. Meyer

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Document Markings

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM22-0681

Outline

Threat space: Where can I find threats?

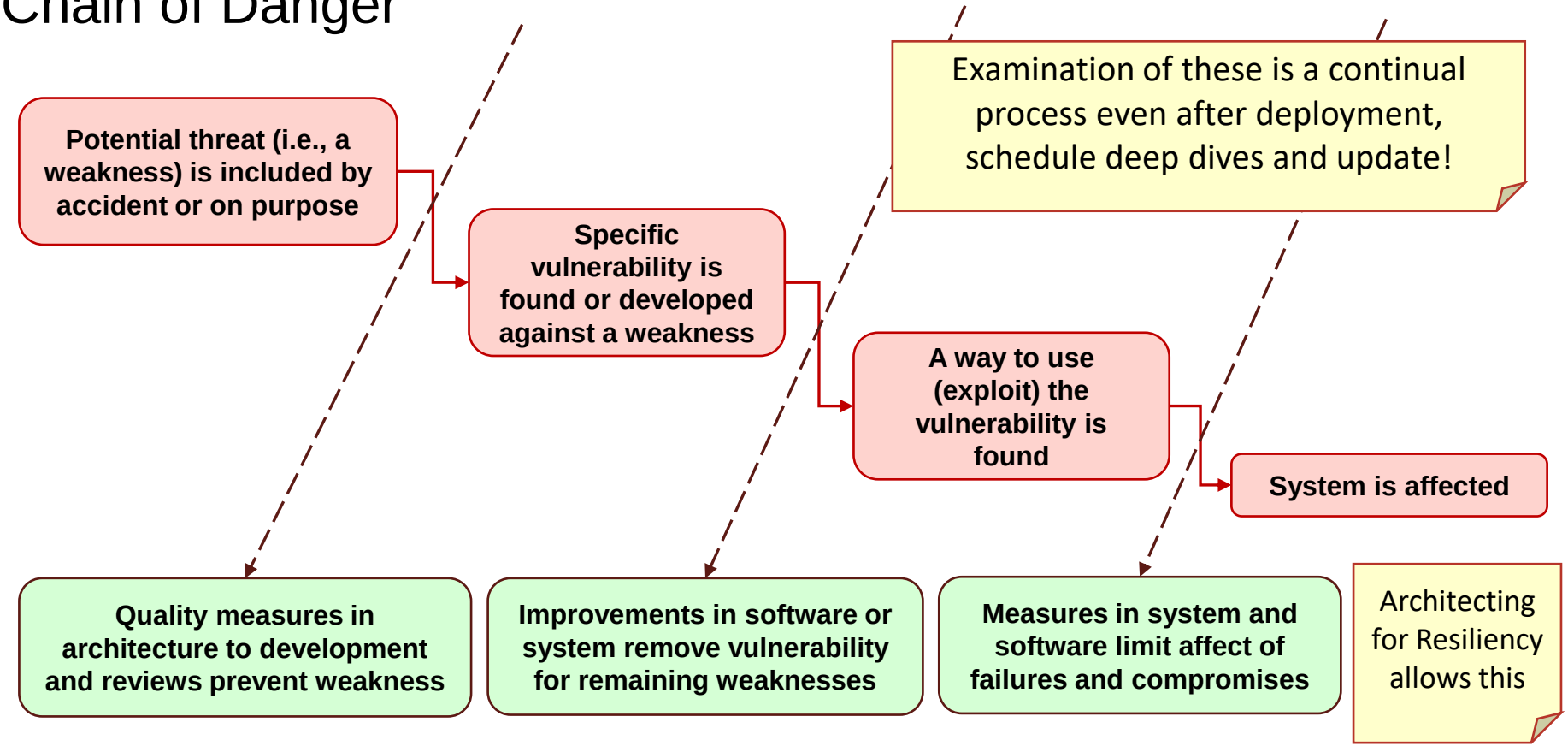
Scenario and definitions

Phases of notional scenario and threat map for each

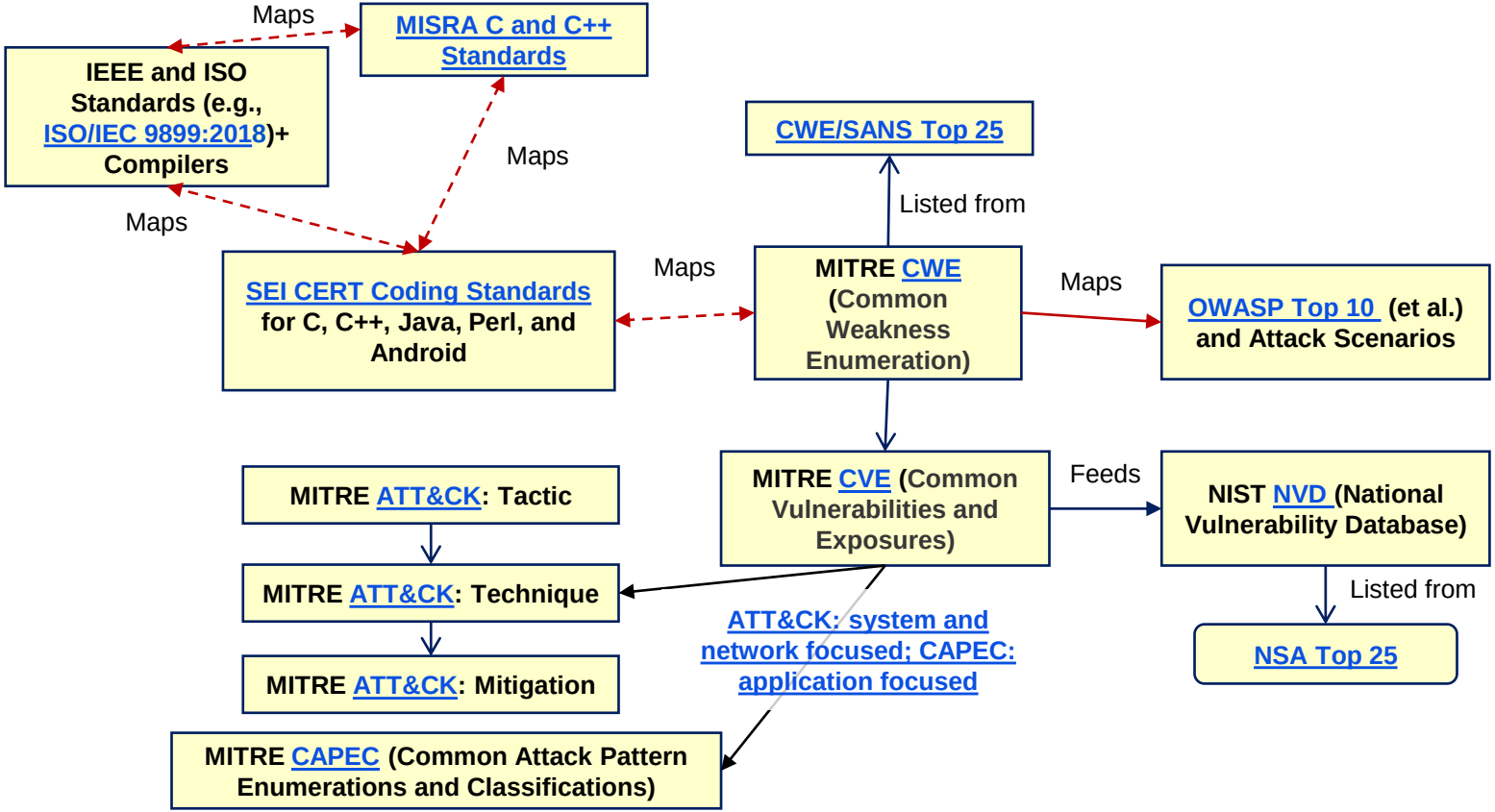
Mitigations

Overall map

Chain of Danger



Where can I find threats?



Threat Databases and Uses

Source	Time Frame	Tools	Notes
• **SEI CERT Coding Standards for C, C++, Java, Perl, and Android** • **IEEE standards (e.g., ISO/IEC 9899:2011)+**	During Architecture and BEFORE USE: • During coding in development environment • Part of DevSecOps via static analysis	• Clang, other modern development environments (Eclipse, Visual), and static analysis tools • Manual review	• Standards from IEEE/IEC/ISO informed secure coding practices and examples in the SEI CERT Coding Standards; secure coding should part of an overall code quality process • Many map to CWEs
• **MITRE CWE** • **OWASP Top 10** • **CWE/SANS Top 25**	During Architecture and BEFORE USE: • During coding in development environment • Part of DevSecOps via static analysis IN UPDATE/PATCHING	• Static analysis tools • Manual review	• Weaknesses can lead to vulnerabilities (CWEs can be mapped to CVEs if a vulnerability is found for a weakness) • Top lists are the most likely weaknesses for exploit
• **MITRE CVE, which feeds ATT&CK Techniques** • **MITRE CAPEC** • **NIST NVD** • **NSA Top 25**	During Architecture and BEFORE USE: • To patch in-use software and systems • As checks as part of DevSecOps process • When upgrades and revisions are required to fix known vulnerabilities in fielded code and systems.	• Static and dynamic analysis tools • Manual review • Custom comparison: release to vulnerability • ATT&CK focuses on system level, CAPEC on application and software level	• Vulnerabilities can be exploited by various hacking tools • NVD maps CVEs with known exploits; the worst vulnerabilities become NSA Top 25 risks • CVEs can have exploitation techniques listed in ATT&CK and in CAPEC

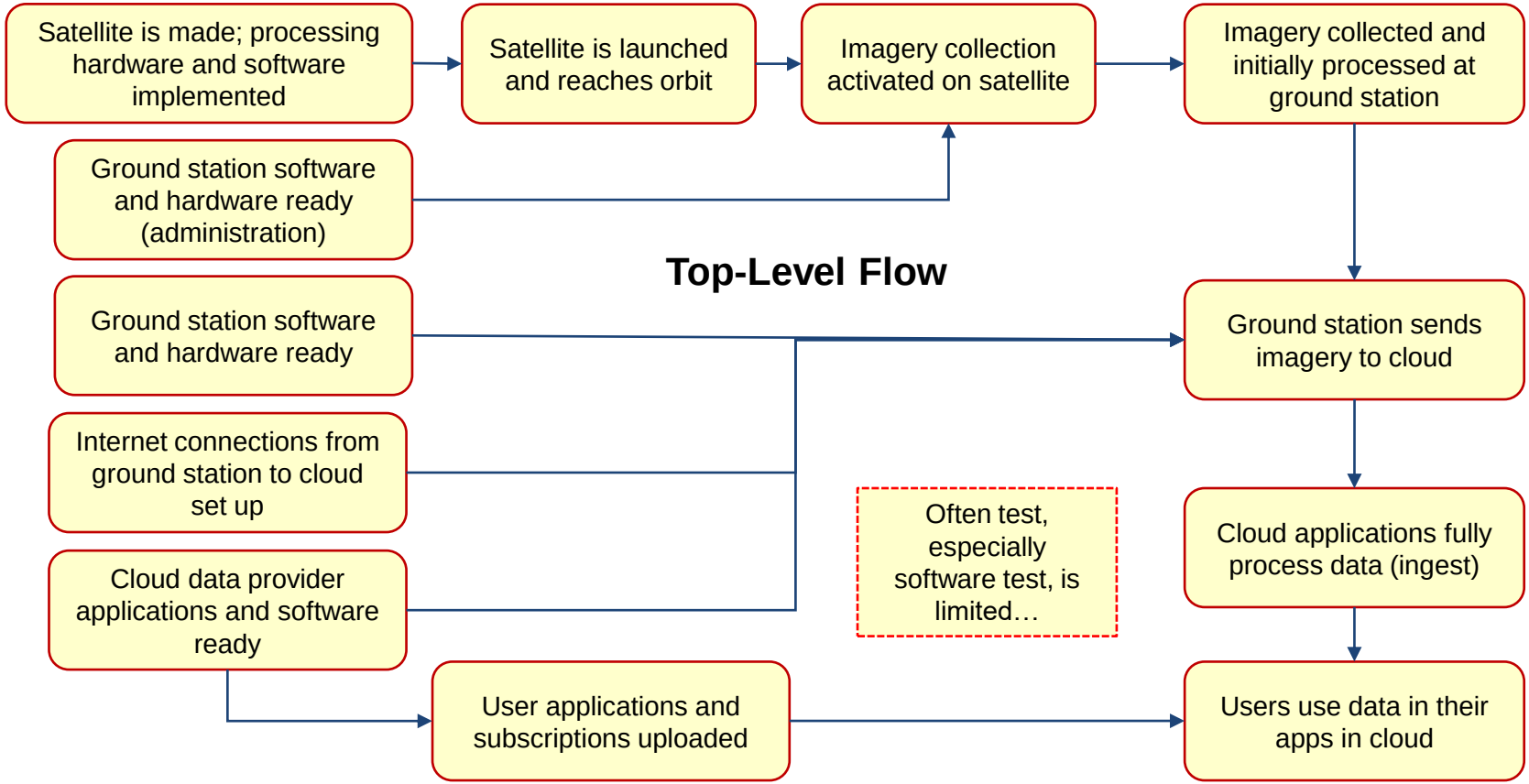
Definitions and Scenario

GEO = geostationary Earth orbit; **LEO** = low Earth orbit; **LAN** = local area network; **WAN** = wide area network; **ISP** = Internet service provider; **data provider** = owner of imagery from satellites

Scenario:

- Multiple GEO and LEO satellites send imagery to multiple ground stations.
- Satellites have a control channel to manage the satellite and data collection and a data channel to send data down to ground stations.
- Ground stations have antennas, equipment, and servers to receive and initially process imagery from satellites on a LAN. LAN connects via ISP to the Internet to push data to an instance on a commercial cloud. Some ground stations have control and operations servers.
- In commercial cloud, data provider has containers, services, and applications to check data files for security and format and then tag files for use. Data enters cloud via on-ramp.
- Users can search and retrieve data files and use their own applications to further process data.

Five Phases: Pre-Launch, Post Launch, Ground Station, WAN, Cloud

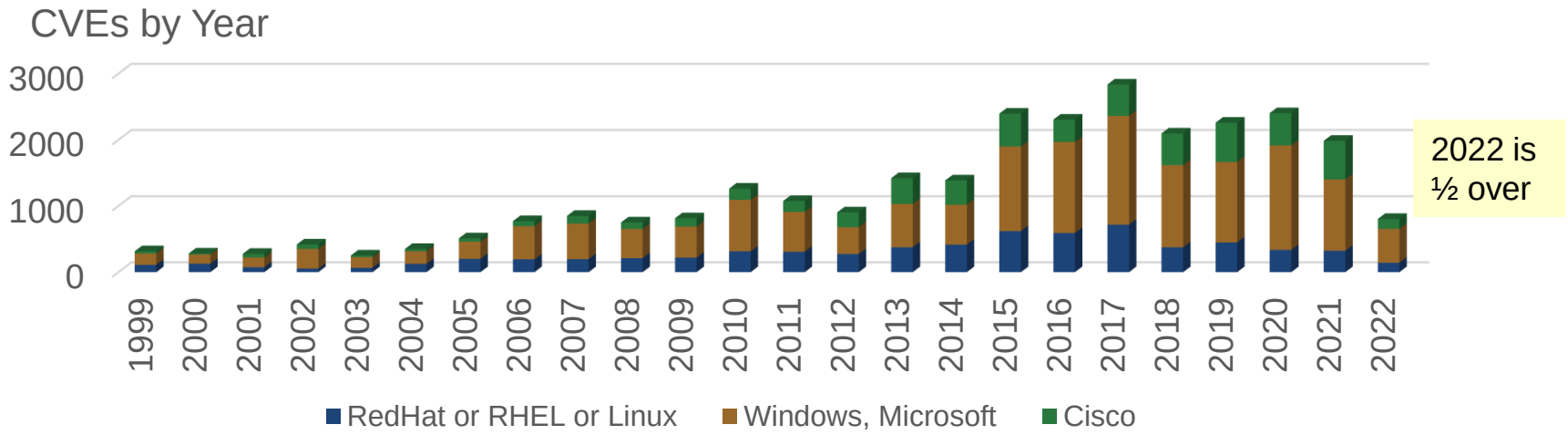


Threats in General: Looking at OSs

Satellite: for SUM(Linux, RedHat, VxWorks, RTOS) >400 possible vulnerabilities per year

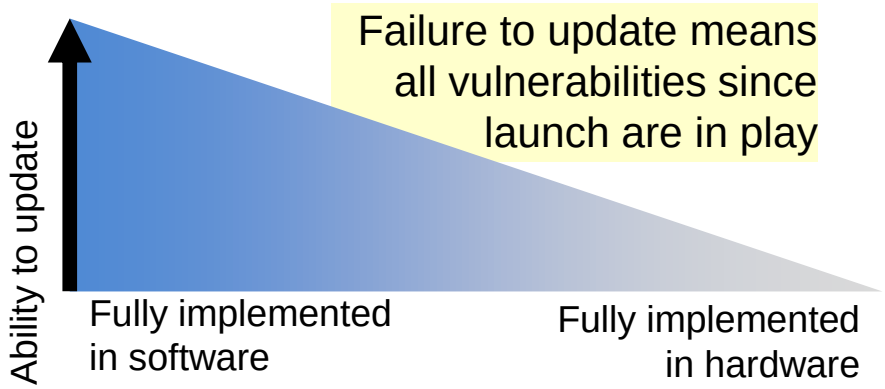
Servers: typically updated by a good IT staff, but around 100 possible vulnerabilities ***per month*** on average across all Microsoft products

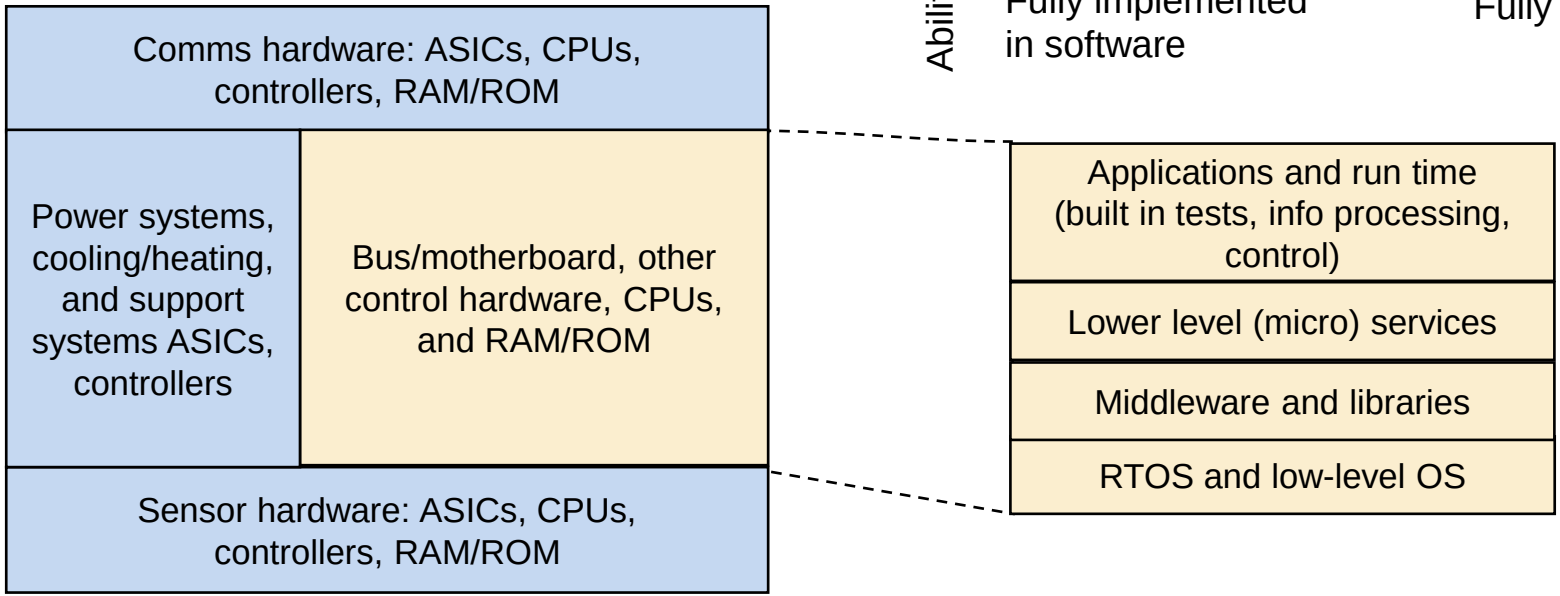
Network and related: 300–400 possible vulnerabilities per year



Pre-Launch: Satellite System

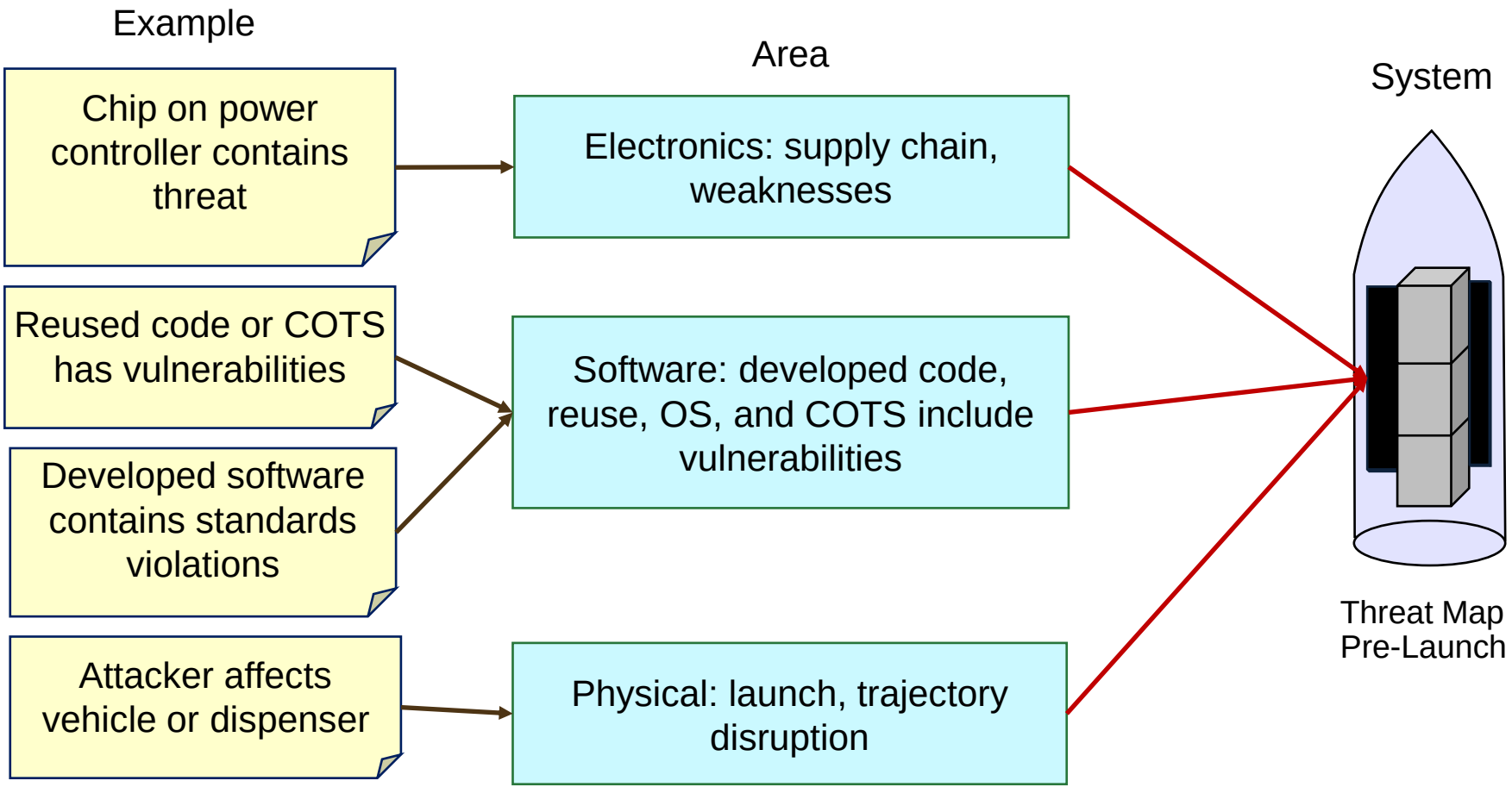
Tradeoffs in functions using hardware implementations versus software determine updatability and threat exposure





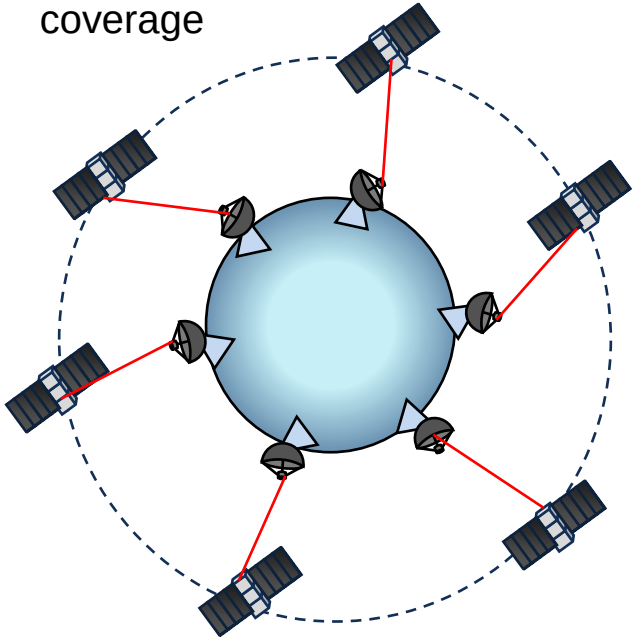
Pre-Launch Example Threats

Threat	Type	Impact
• Open source or COTS weakness, vulnerability, or exploit • Chipset exploit or vulnerability • COTS or externally sourced component vulnerability/exploit	• Supply chain	• Sleeping issues that can be triggered by conditions or communication, failure of satellite, control of satellite, or contamination of data streams
• Insider threat: exploits or errors inserted to hardware/software	• Insider threat	• Issues planted by developers • Data stream sent to unauthorized receivers • Control of systems
• Weak code (contains rules violations)	• Quality	• Undefined behaviors that can cause failures
• Attacks against launch vehicle or range control, etc.	• Direct assault	• Launch failure • Inability to make correct orbit



Post-Launch Dynamics

LEO/MEO need multiple stations for coverage



Imagery sensor collects raw data

Administration controls satellite (and updates SW) via Control Channel

Control Channel

Data Channel

Image Capture

Admin System

Ground Stations receive imagery via data channel

Geosynchronous that are not equatorial (Geostationary) need multiple stations…

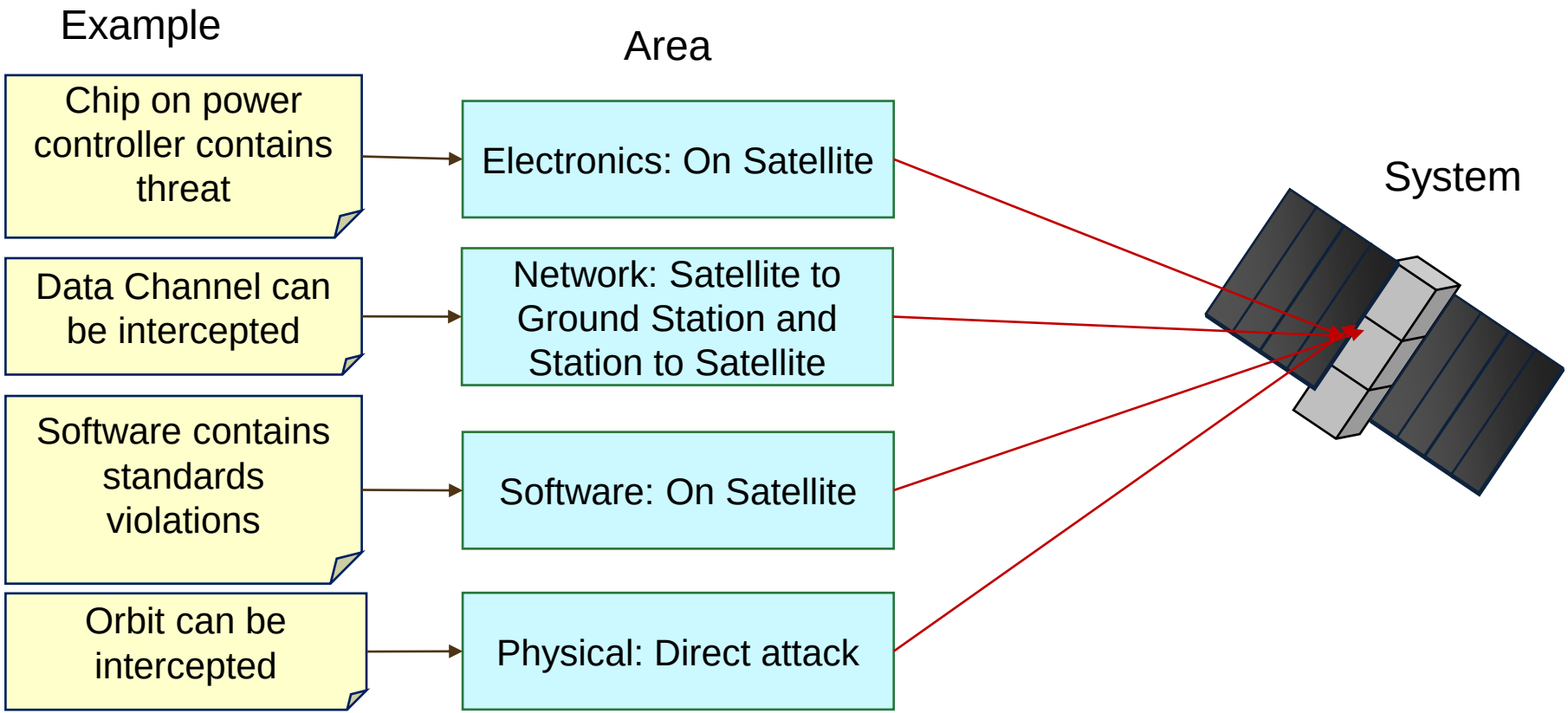
Ground Station (Receiver) LAN

Emerging Concern: Sat to Sat crosslink

Post-Launch Example Threats

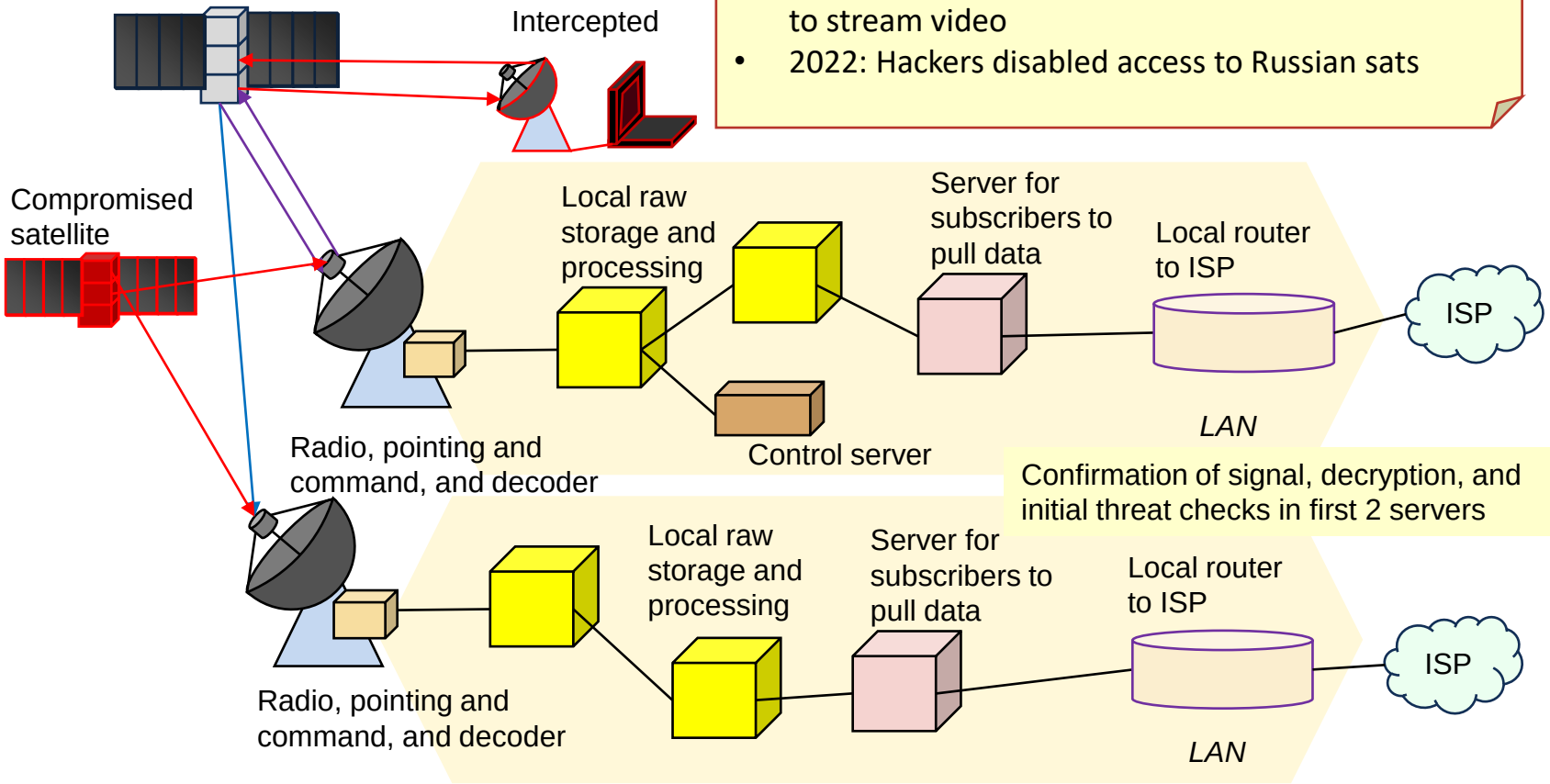
Threat	Type	Impact
• Chip on power controller or other area contains threat, via supply chain or insider threat (custom chip)	• Electronic	• Satellite compromise • Data or control loss • Data alteration
• Software contains standards violations, and they are exploited	• Software	• Satellite compromise • Data or control loss • Data alteration
• Memory errors due to natural radiation or direct electronic attack	• Environmental • Direct electronic	• Depends on fault tolerance of hardware and resiliency of software • Effects can be subtle and appear random • Some EM attacks can introduce values into variables or inject code
• Control Channel compromise/MitM • Data channel cloning/MitM	• Interception via wireless	• Control Channel compromise can cause loss of control of sat and data channel interception
• Data Channel can be intercepted	• Data compromise via wireless	• Since data is sent wirelessly over an area, any antenna in the area can catch the signal
• Data or Control Channel degradation due to DoS or E-M attack	• Direct electronic • Cyber attack	• Loss or degradation of data channel
• Orbit can be intercepted via kinetic attack • Orbital degradation due to laser, etc.	• Direct physical attack	• Loss of satellite

Top-Level Risks: Satellite Post-Launch



Ground Stations

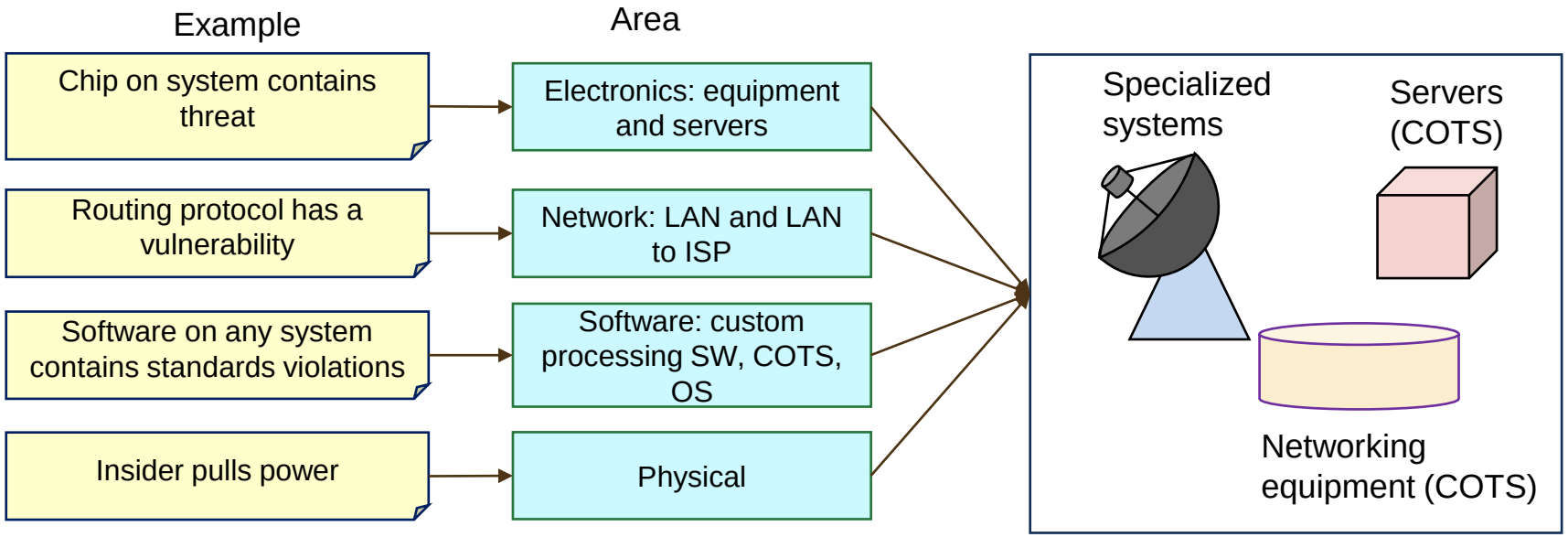
- 2022:Research group used a decommissioned sat to stream video
- 2022: Hackers disabled access to Russian sats



Ground Station Threats

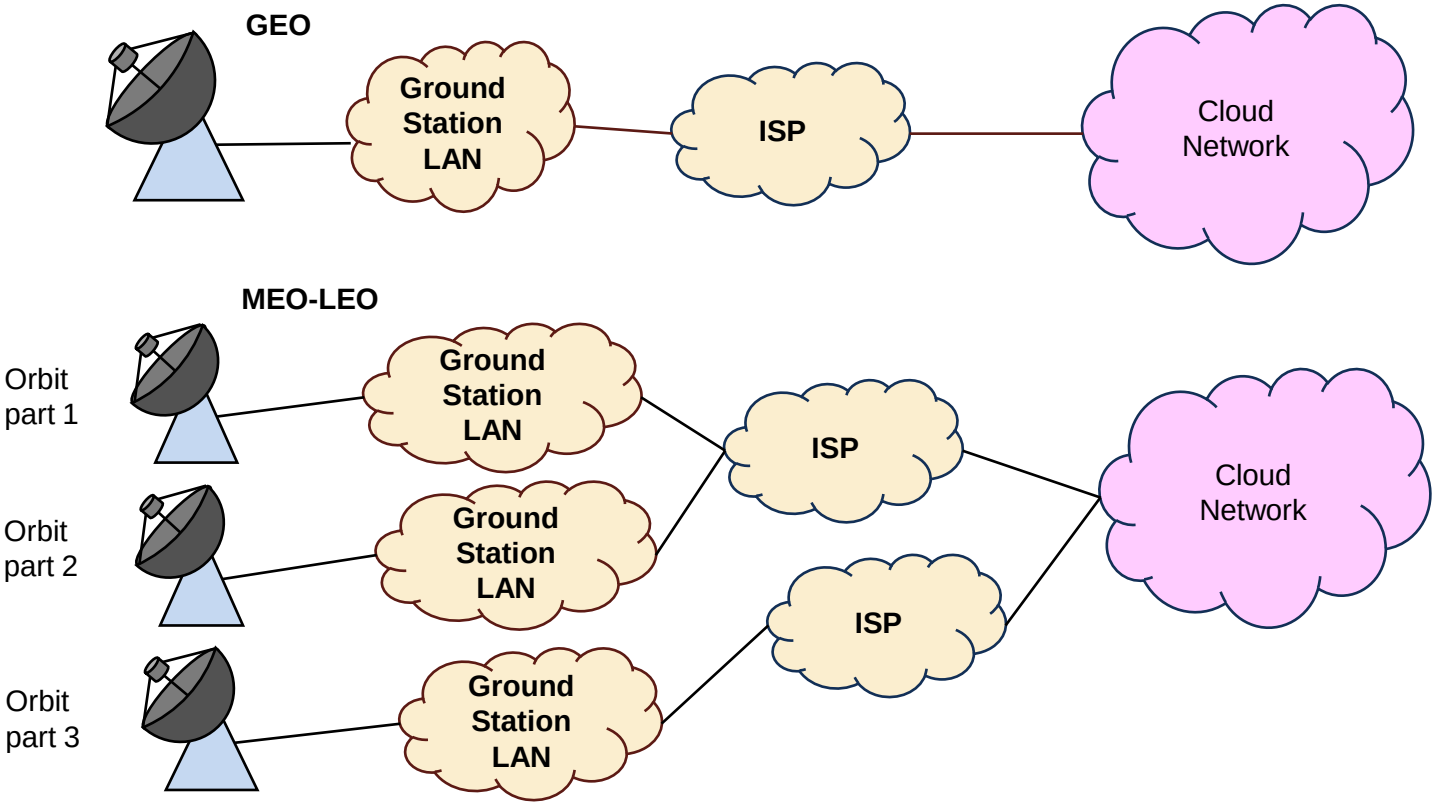
Threat	Type	Impact
• Secure Coding Standards violations for custom applications • Reused code • Downloaded applications • COTS	• Software code	• Data loss or compromise • Disruption • Loss of control • Compromise of control
• Denial of service via network, traffic flood, or protocol	• Network	• Data loss • Loss of connectivity to cloud or remote users
• Malicious code inclusions in COTS or reused code in control software, data processing applications, or off-the-shelf installed code	• Software code	• Data or control loss • Data or control compromise
• Insider threat or adversary assault: direct access to ground stations for disruption, takeover, or destruction • Ex: Insider pulls power	• Physical	• Data loss or compromise • Disruption • Loss of control • Compromise of control
• Chip on servers, satellite communications systems, control systems, or network hardware contains threat	• Electronics: supply chain	• Data loss or compromise • Disruption • Loss of control • Compromise of control

Top-Level Risks: Ground Station



Most threats for the Ground Station are software, based on the servers, due to the many applications installed on servers and common operating systems

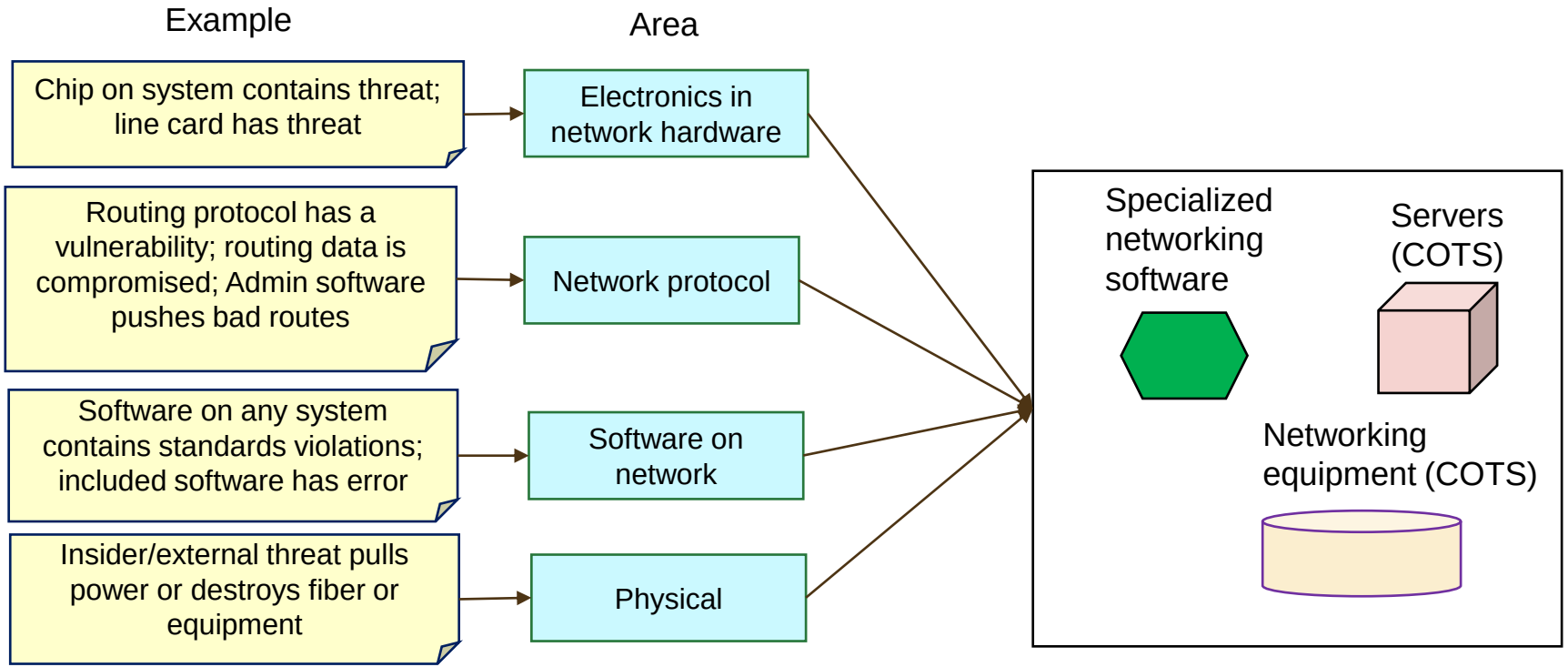
Ground Stations WAN: Ground Station via ISP to Cloud



WAN Threats

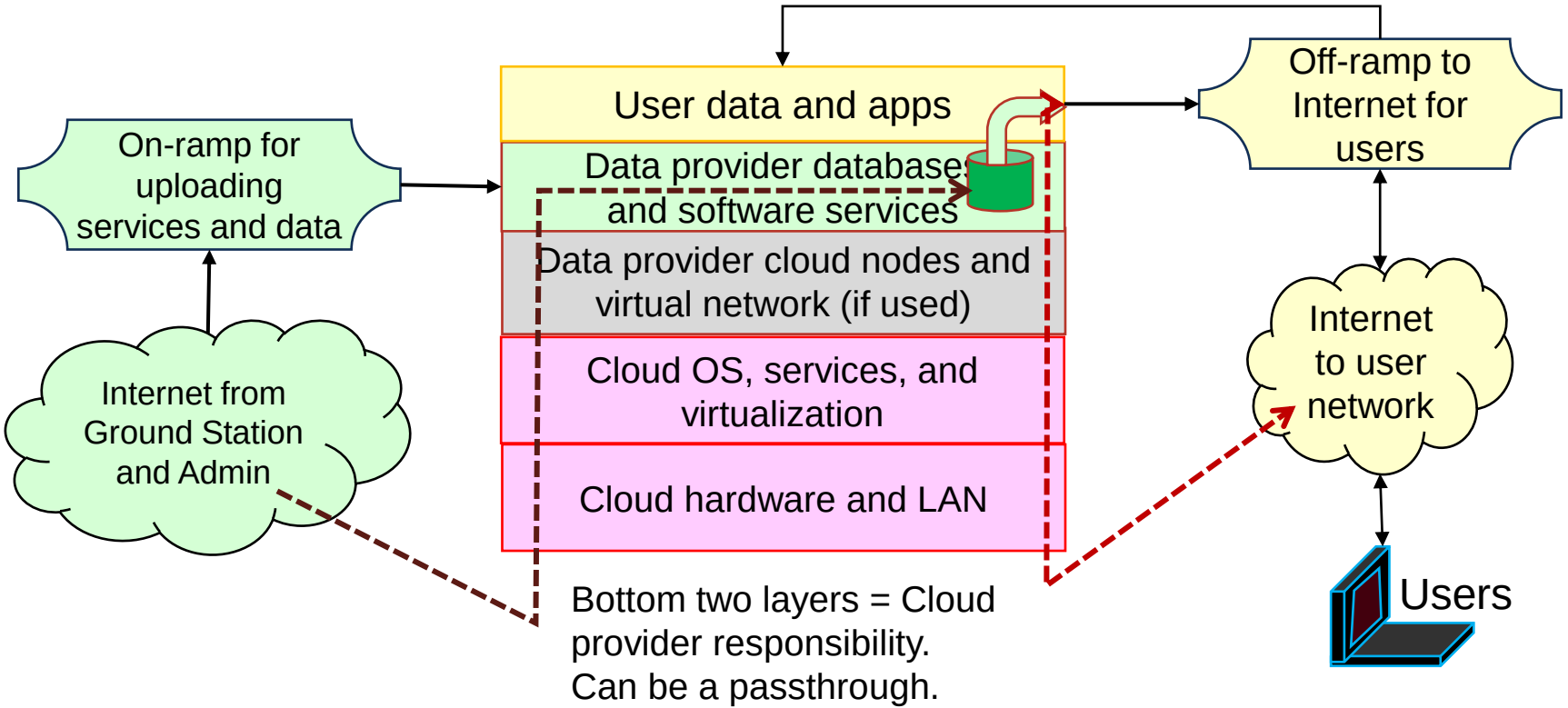
Threat	Type	Impact
• Compromise of router/switch	• Vulnerability in networking software or hardware: supply chain, software weakness	• Denial of service • Delay of data or loss of data
• Compromise of protocol that stops or degrades traffic, including routing tables for peer connections	• Protocol design weakness or vulnerability • Resiliency in network software or hardware	• Denial of service • Delay of data or loss of data
• Traffic capture	• Standard operation of ISPs • Possibly insider threat • Cyber-physical (i.e., adding a software agent or hardware)	• Data compromise • Shallow packet inspection has little impact • Deep packet inspection can reveal data, depending on encryption
• Compromise of telemetry network for ISP	• Insider threat • Software vulnerability in management software	• Denial of service • Delay of data or loss of data
• Compromise of software on edge to Ground Station (or admin) network and to cloud network	• Software vulnerability	• Data loss • Data compromise or capture • Reverse engineering satellite communication • Admin: loss of satellite control

Top-Level Risks: WAN



Most threats for the Ground Station are software, based on the servers, due to the many applications installed on servers and common operating systems

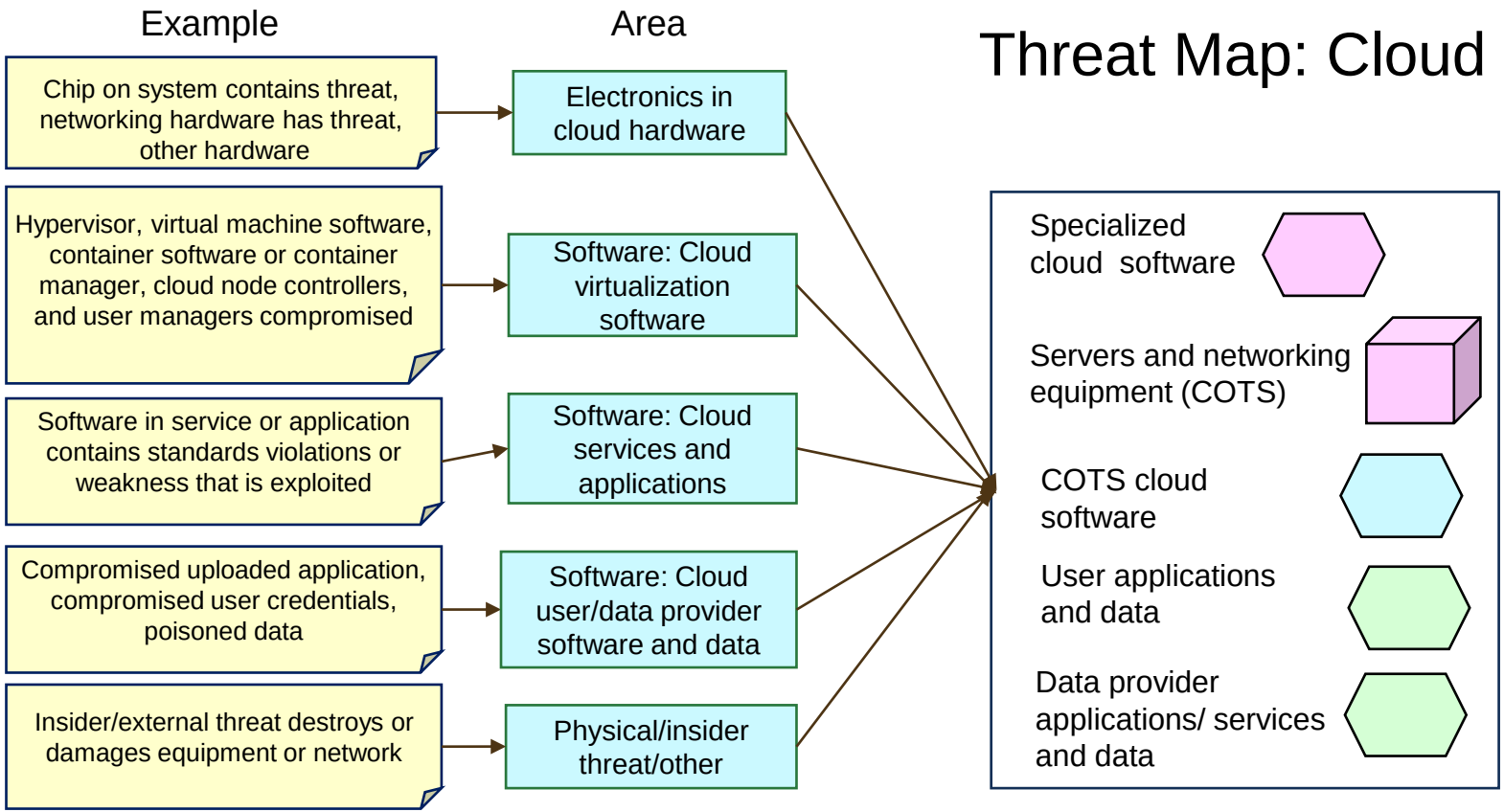
Cloud: On-Ramps and Off-Ramps, Shared Security



Cloud Threat Table

Threat	Type	Impact
• Compromise of cloud provider's cloud virtualization software	Exploit via software vulnerability via • custom development • insider threat • reused code	• Data loss • Node loss • Denial of service • Data compromise • User network compromise (user credentials) • Exploit vector to users cloud-wide
• Compromise of cloud provider's applications or services	Exploit via software vulnerability via • COTS or custom • insider threat • reused code	• Data compromise • Denial of service • Credential or critical data capture for use against other users of the applications cloud-wide
• Compromise of data provider's applications, services, or other software (incl. containers)	Exploit via • software vulnerability • insider threat • reused code	• Data loss, compromise, or alteration • Exploits pushed to user applications • Credential capture
• Compromise of data in database (steganography, etc.)	• Cross-site scripting • Poisoned data • Compromised data channel	• Depends on steganography checking or data filtering, data configuration control and checking • Users could get bad data or receive hidden exploits
• Compromise of user applications or user data	Exploit via • software vulnerability • insider threat • reused code	Depends on compartmentalization/localization: Did you limit what users can do? • User app sends exploit to other users or compromises data provider services and apps • Denial of service

Threat Map: Cloud



Most threats for the Ground Station are software, based on the servers, due to the many applications installed on servers and common operating systems

Master Threat Map

- Physical: launch, trajectory disruption
- Electronics: supply chain
- Software: development included vulnerabilities, reuse, OS and COTS

→ **Satellite Pre-Launch**

- Physical: direct attack in space
- Electronics: activation of vulnerabilities
- Network: satellite to Ground Station and station to satellite
- Software: On satellite on ASICs, controllers, and CPUs

→ **Satellite Post-Launch**

- Electronics: equipment and servers
- Network: LAN and LAN to ISP: protocols et al.
- Software: custom processing SW, COTS, OS
- Physical

→ **GroundStation**

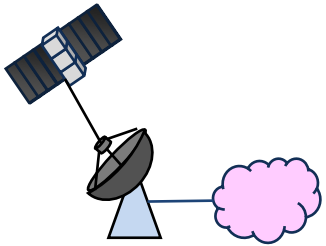
- Electronics in network hardware
- Network protocol
- Software on network
- Physical

→ **WAN/Internet**

- Electronics in cloud hardware
- Cloud virtualization software
- Cloud services and applications software
- Cloud user/data provider software and data
- Physical/insider threat/other

→ **Cloud**

Sat to User



Software is in all phases…

Mitigations: Software Test Tools and Overall, Methods

Example Tool	Purpose	Example Tool	Purpose
Clang	Compiler w/static analysis	PVS-Studio	Static analysis
CodeDX	Static analysis	Rose Compiler – Program Analysis and Transformation	Compiler w/static analysis
CodeSonar	Static analysis	RuleChecker	Static analysis
Coverity	Static analysis	SCALe	Aggregation of tools and results
Cppcheck	Static analysis	SonarLint	Static analysis
Klocwork	Static analysis	SonarQube	Static analysis
Parasoft	Static analysis	Splint	Static analysis
PC-lint Plus	Static analysis	TrustInSoft Analyzer	Static analysis
Polyspace Bug Finder	Static analysis	Understand	Static analysis
PRQA QA-C (Synopsys)	Static analysis	**Appearance here does not constitute endorsement**	

These are just a part… Architecture, Dynamic Tests, IVV, etc.

Example Methods: **SEI Secure Coding, SEI OCTAVE®, NIST SP 800-58 RMF, MITRE ATT&CK®, and MITRE CAPEC™**

Conclusions and Mitigations

Threats, including the Weakness → Vulnerability → Exploit chain, are maintained in many linked locations

Threat exposure for the life of a spacecraft is shaped pre-launch, including the portion of the functionality in software and how much testing is performed against software

The ground station to network to cloud leg depends on the need for ground stations by altitude, control, and recipients

The cloud can become both a threat mitigator and threat distributor

Mitigations include quality design using secure practices and standards, monitoring, and update

DO NOT ASSUME EVEN A LOW-COST LEO CUBESAT IS SAFE!

Contact me if you think SEI can help you: bmeyer@sei.cmu.edu

References and Further Reading

Fred Long, Dhruv Mohindra, Robert Seacord, Dean F. Sutherland, David Svoboda. *CERT Oracle Secure Coding Standard for Java*. Addison-Wesley Professional. Sep 2011. Part of the SEI Series in Software Engineering. ISBN-10: 0-321-80395-7, ISBN-13: 978-0-321-80395-5.

National Institute of Standards and Technology, Special Publication 800-53, Revision 5. *Security and Privacy Controls for Information Systems and Organizations*. Sep 2020. https://doi.org/10.6028/NIST.SP.800-53r5.

Robert C. Seacord. *SEI CERT C Coding Standard: Rules for Developing Safe, Reliable, and Secure Systems (2016 Edition)*. Software Engineering Institute, Carnegie Mellon University. Jun 2016. https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=454220.

Robert C. Seacord. *Secure Coding in C and C++*, 2nd ed. Addison-Wesley Professional. Apr 2013. Part of the SEI Series in Software Engineering. ISBN-10: 0-321-82213-7, ISBN-13: 978-0-321-82213-01983.

Robert C. Seacord. *Java Coding Guidelines: 75 Recommendations for Reliable and Secure Programs*. Addison-Wesley Professional. Aug 2013. Part of the SEI Series in Software Engineering. ISBN-10: 0-13-343951-8, ISBN-13: 978-0-13-343951-9.