Containerized Application Security for ICS

Sandia National Laboratories (SNL); PI - Adrian Chavez
DOE HQ PM - Jodi Kouts; TPO NETL - Joel Lindsay

Total Project Value: $2,500,000
DOE Share: $2,500,000
Recipient Share: $0

Period of Performance: 5/10/2018 - 12/31/2024

Location: Albuquerque, NM
Congressional District: NM-01

Contract ID: M617000256

Project Description: Malware often propagates through computer networks by first targeting vulnerable applications and then making lateral movements to reach desirable targets. Stuxnet, BlackEnergy and CrashOverride are a few examples of malware that have translated into significant economic losses, interruptions of availability, and damage to physical equipment within Industrial Control System (ICS) environments. To increase the resiliency of ICS environments against malware, SNL will isolate individual applications from the rest of the system by the use of containers. Containers are lightweight processing environments that share the host's kernel, but are otherwise isolated from the rest of the system. By containing applications, a successful adversary will only have access to the individual application's containerized environment as opposed to the entire system. Furthermore, containing applications provides the ability to hot-swap upgraded versions of software to maintain a high level of security and availability. Containers also provide the ability to create a moving target defense for applications to easily and quickly migrate across systems within the network to avoid and create uncertainty for an adversary.

Challenges	Impact
Account for stateful and stateless applications	Provides the ability to upgrade and migrate applications that may need to maintain state across versions
Introduce fault tolerance to applications	Provides the ability to detect and isolate compromised applications

Anticipated Outcomes

- Update and migrate ICS applications in real-time.
- Isolate software compromises and migrate applications to patched/upgraded versions using rolling updates within containers.
- Demonstration of updates and migration within the Ft. Belvoir microgrid testbed
- Final report describing performance metrics for update and migration solutions

Major Accomplishments - Prior and Current	Date
Implement Live-Migration & Live-Upgrades	July 12, 2019
Develop Fault Tolerant Algorithms, use cases, and attack scenarios	August 28, 2020
Complete Red Team Assessment on Combined Live-Upgrades and Live-Migration	November 20, 2020
Fort Belvoir demonstration	October 5, 2021
Report documenting technology and demonstration	May 6, 2022
Develop mitigation strategies for Red Team assessments	December 8, 2023
Integrate live-updates into SEL-3355 Blueframe development kit	June 30, 2024

Success Stories

- 2018 - US Patent No. 10,037,203
- 2019 – Initial testing of CAPSec functionality complete
- 2020 – Developed live-upgrades and live-migration technology proof-of-concept
- 2020 - DOE Practices to Accelerate the Commercialization of Technologies (PACT)
- 2021 – Integrated live-upgrades and live-migration technology into Fort Belvoir Closed Restricted Network

GRIMM
Pacific Northwest NATIONAL LABORATORY
SEL
Chevron

Containerized Application Security for Industrial Control Systems (CAPSec) Sandia National Laboratories

Risk Management Tools and Technology Cyber Peer Review 2024

August 27th-29th, 2024

Project Overview

Objective

- Update and migrate ICS applications in real-time without loss of availability
- Isolate software compromises and migrate applications to patched/upgraded versions using rolling updates within containers
- Evaluate our solution with a 3rd party independent red team assessment
- Demonstration of updates and migration within a microgrid testbed environment
- Final report describing performance metrics for update and migration solutions

Schedule

- 5/10/2019-12/31/2024

TRL and Previous Work

- Starting: 2, Current: 6, Anticipated Final Technology Readiness Level: 7
- ~2 month minimal funded investment internally at Sandia National Laboratories

Total Value of Award:	**$2,500,000**
Funds Expended to Date:	**85.85%**
Performer:	**Sandia National Laboratories**
Partners:	**Schweitzer Engineering Laboratories, Pacific Northwest National Laboratories, Grimm, SMFS, Chevron, Fort Bevoir NVESD**

Advancing the State of the Art (SOA)

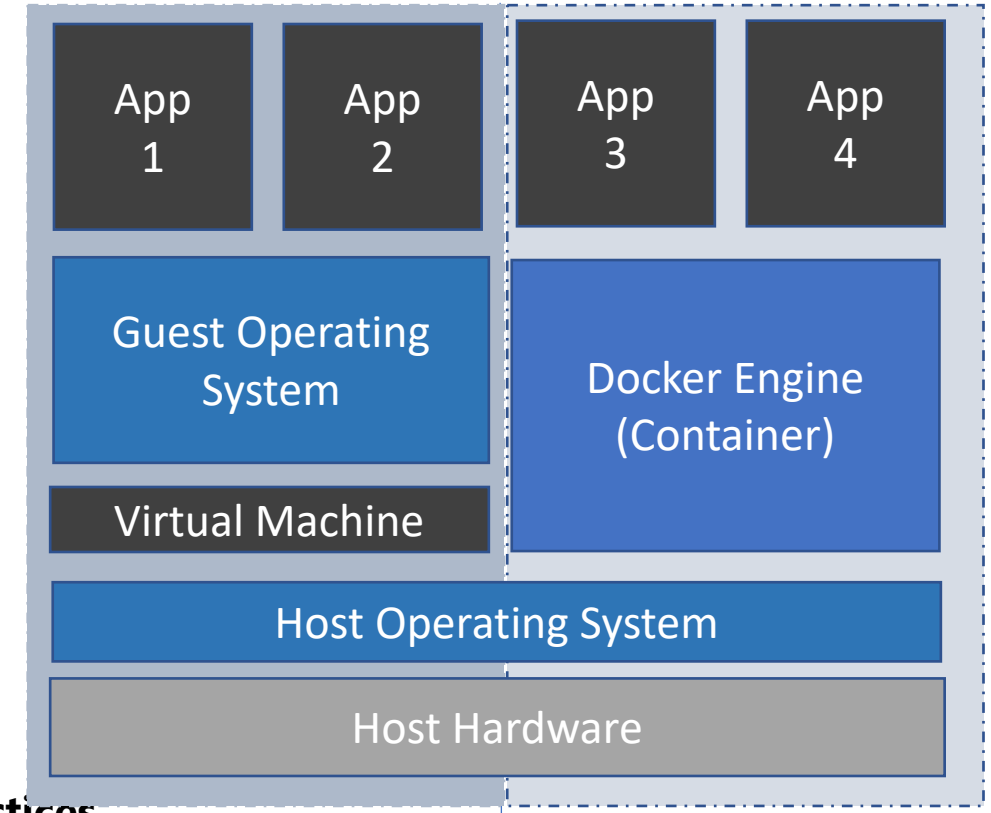
- Malware often propagates through computer networks by first targeting vulnerable applications and then making lateral movements to reach desirable targets
  - Patching and updating requires downtime at scheduled maintenance periods
  - Delays in patching results in increased vulnerabilities to cyber attack
  - Detecting application failures/crashes requires manual intervention
  - Containing vulnerabilities from spreading across network is challenging
- Our goals are to improve hardware/software/firmware patch management
  - Identify and patch software without disruption of operation
  - Develop a prototype that can patch stateful and stateless applications
  - Develop an interoperable solution that can be applied to a variety of Operational Technology environments
- Transition R&D to industry for broader OT adoption
  - Develop a reference implementation using both open source and custom solutions
  - Develop a representative laboratory environment for testing and evaluation
  - Demonstrate our technology at partner microgrid environments
  - Document our implementation and demonstrations

Significance and Impact

- **Department of Energy (DOE) Cybersecurity, Energy Security, and Emergency Response (CESER) office**
  - **Research Call to DOE/Federal Laboratories**
    - **Cybersecurity for Energy Delivery Systems (CEDS) 2017 Research Call**
    - **Industry Partnerships for Cybersecurity of Energy Delivery Systems (CEDS) Research, Development and Demonstration**

U.S. DEPARTMENT OF ENERGY | OFFICE OF CYBERSECURITY, ENERGY SECURITY, AND EMERGENCY RESPONSE

- **Thesis: Use of Containerization eliminates costs and downtime associated with updating/patching software.**
  - **Increase resiliency to cyber attack**
  - **Detect and isolate cyber attacks**
- **Containers bundle all software, libraries, environment variables and configuration files**
  - **Use less resources than VMs due to independence from full OS.**
  - **Most common:**
    - **Open source Docker Engine**
    - **Open source container-orchestration system**



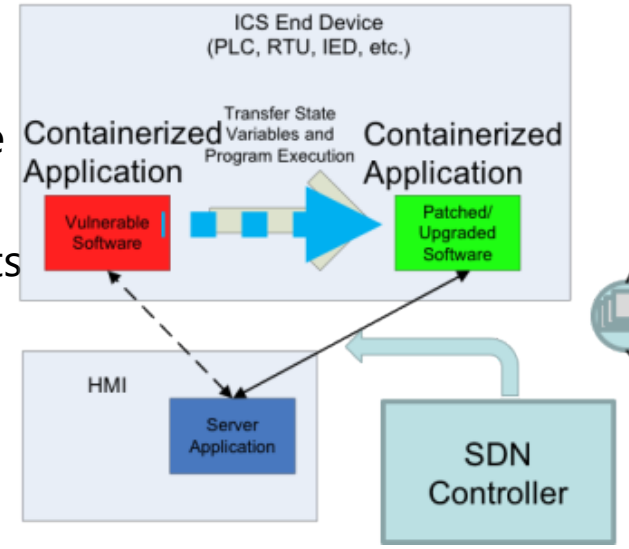
docker kubernetes

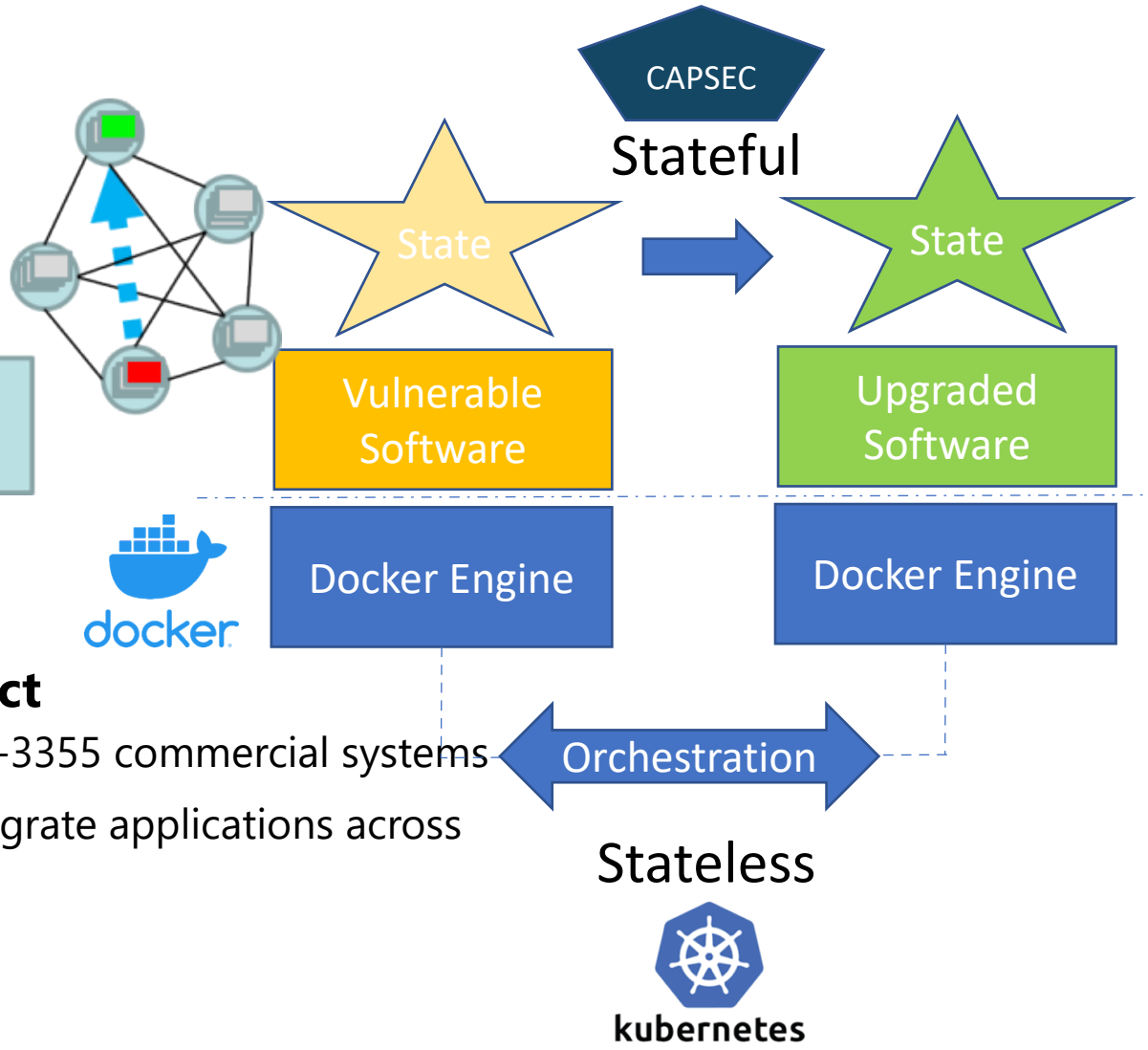
- **A reference implementation using open source tools as well as a custom solution to upgrade/migrate software has been developed**
- **A portable solution has been developed with cyber security best practices**
- **Our solution has been applied to multiple laboratory and representative field environments**
- **Our results and demonstration have been documented for the broader energy sector**

Progress to Date / Next Steps

Major Accomplishments

- Containerized architecture
- Replicates Stateful and stateless environments
- TLS Agents on parallel containers
- 3rd Party Red Team assessment
- Applied toward Fort Belvoir microgrid environment
- Captured performance metrics





Approach for the next year or to the end of project

- Scale our reference implementation across multiple SEL-3355 commercial systems
- Develop a moving target defense implementation to migrate applications across devices
- Integrate red team results into our solution
- Document our results for broader industry adoption

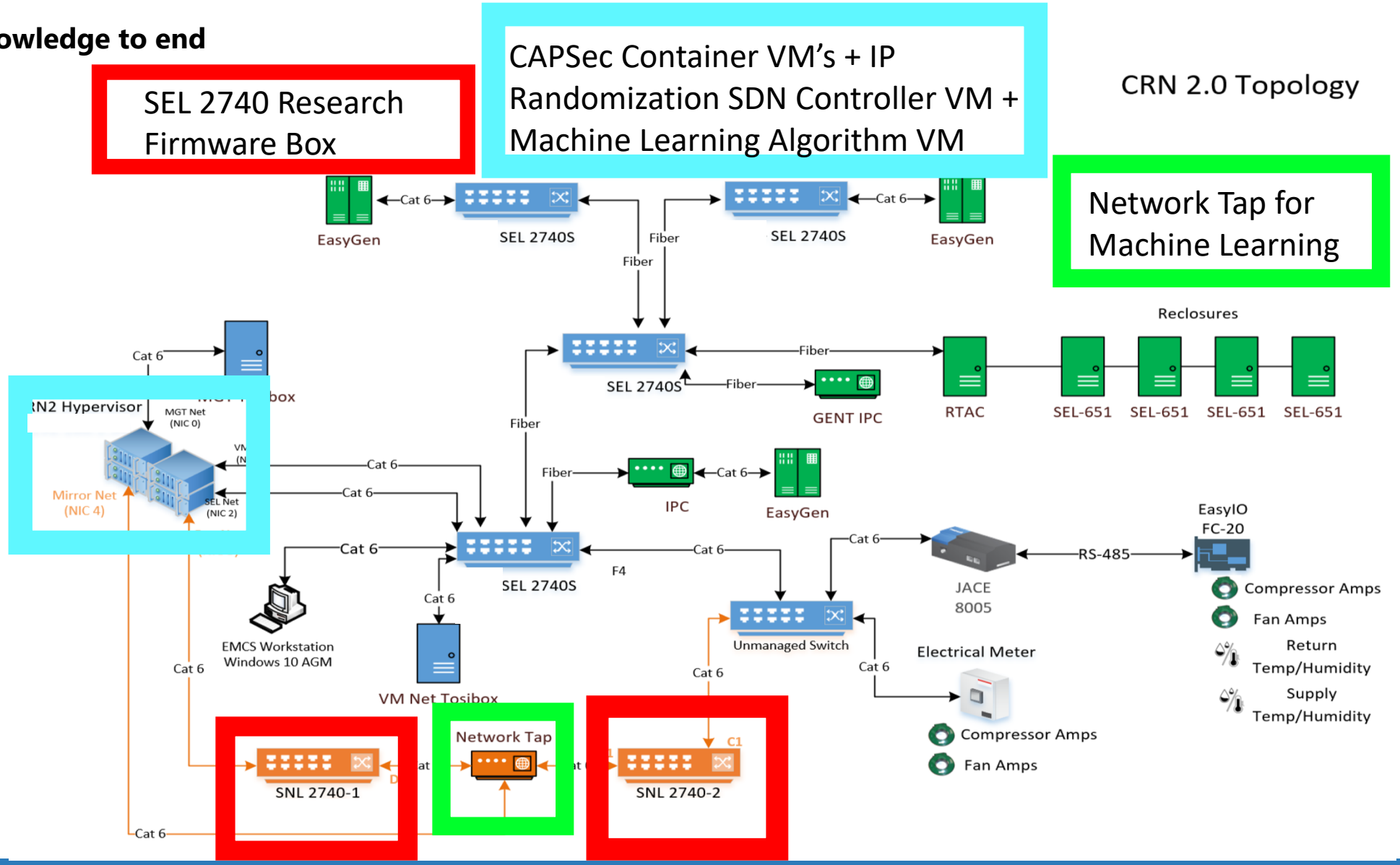
Challenges to Success

1. **Perform live-updates and migration for *stateless* applications**
   - Docker Containers & Kubernetes Orchestration
   - Ability to test software on operational data before deployment
   - Identify and isolate cyber compromises
2. **Perform live-updates and migration for *stateful* applications**
   - Developed custom solution
   - Capture memory state of old application and send captured memory state data to new patched application
3. **Perform live-updates on SDN controller**
   - Developed custom solution to collect flow rules installed across SDN switches
   - Store flow rules in backend database and reference backend database when updating/patching SDN controller
4. **Collect performance metrics of reference implementation**
   - Minimize operational impact and interruptions in communications
4. **Complete an independent 3rd party red team assessment**
   - Identify security concerns and mitigations of reference implementation
5. **Deploy reference implementation in representative microgrid environment**
   - Laboratory environments at Sandia National Laboratories and Pacific Northwest National Labroatory
   - 2.5 MW microgrid environment at Fort Belvoir
6. **Ensure CAPSec technology meets operational requirements**
   - Safety
   - Reliability
   - Optimal performance

Collaboration/Sector Adoption

Plans to transfer technology/knowledge to end user

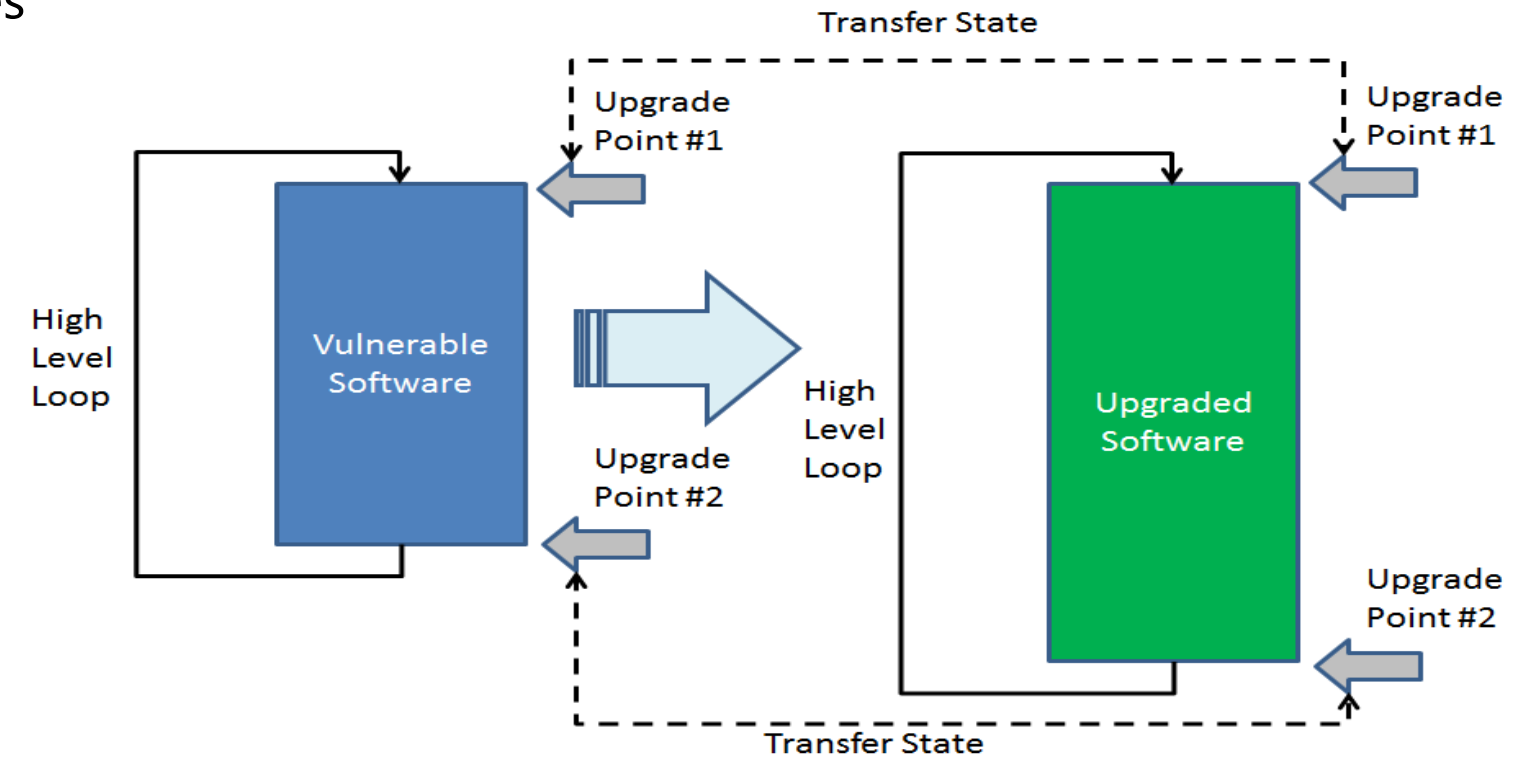
- We have developed our solution using open source tools and deployed withing a representative microgrid environment
  - Docker and Kubernetes implementations initially developed in laboratory environments
  - Our technology has been integrated into the SDN controller
- A demonstration at Fort Belvoir has been developed and documented
  - US Patent No. 10,037,203 *Real Time Software Upgrade*
  - The demonstration shows the benefits of the technology with a cyber attack mitigated by our solution
  - The demonstration did not impact the operational network at Fort Belvoir
  - Our technology was demonstrated on the commercial SEL-5056 SDN flow controller providing a path for industry adoption



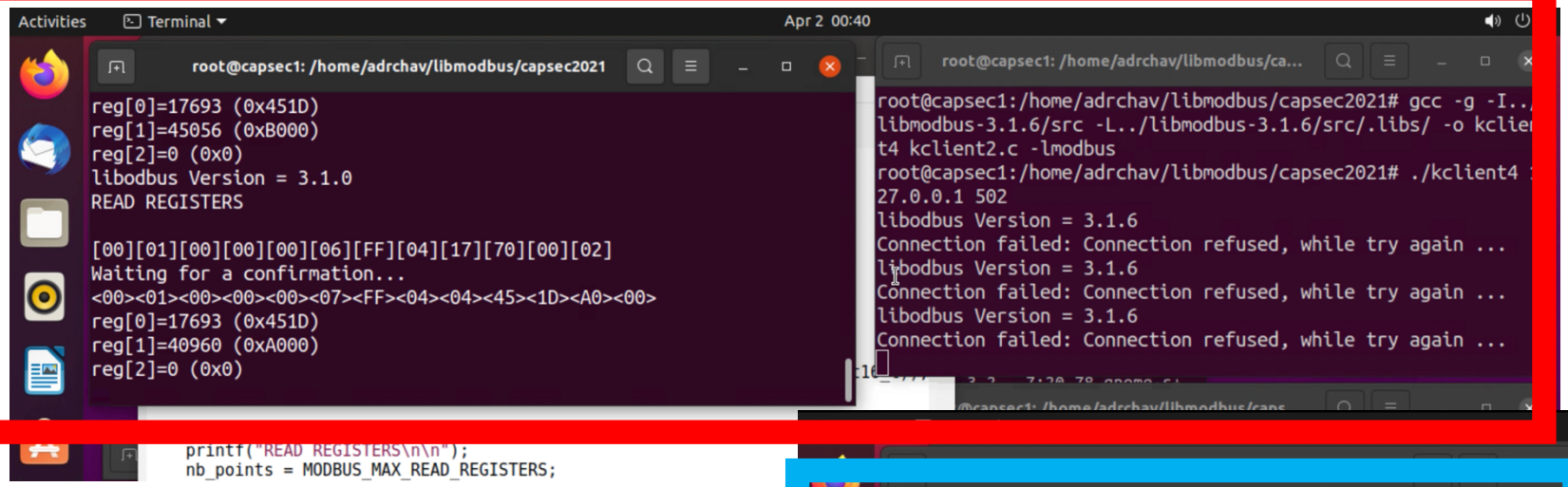
Demonstration at Fort Belvoir microgrid

Upgrade software in real time for increased availability & security

- Identify upgrade points in old and new patched software
- Transition state collected from unpatched software to patched software
  - Software interrupts integrated into software to manage updates
- A proxy was developed to reroute communications to patched software running on separate devices



Demonstration at Fort Belvoir microgrid



- Software with an update available needs to be applied without downtime
- Identify upgrade points in software

```
printf("READ REGISTERS\n\n");
nb_points = MODBUS_MAX_READ_REGISTERS;
```

- Software agent starts to collect memory state information from unpatched version of software
- State information is communicated to patched version of software