Cyber-Physical Intrusion Resilience

Saman Zonouz

Associate Professor, Georgia Tech

School of Cybersecurity and Privacy (SCP)

School of Electrical and Computer Engineering (ECE)

Grand Challenges for our Community, IFIP Meeting 2023

CPSec: Cyber-Physical Systems Security Lab

Outline (focused on resilience/security)

CPSec: Cyber-Physical Systems Security Lab

- Important research challenges
- Exciting opportunities for CPS research
- Lessons learned from the past
- Ideas for tech-transfer initiatives
- CPS security education moving forward

Predictive Situational Awareness

- Online monitoring of the CPS operation to identify potential cybersecurity incidents

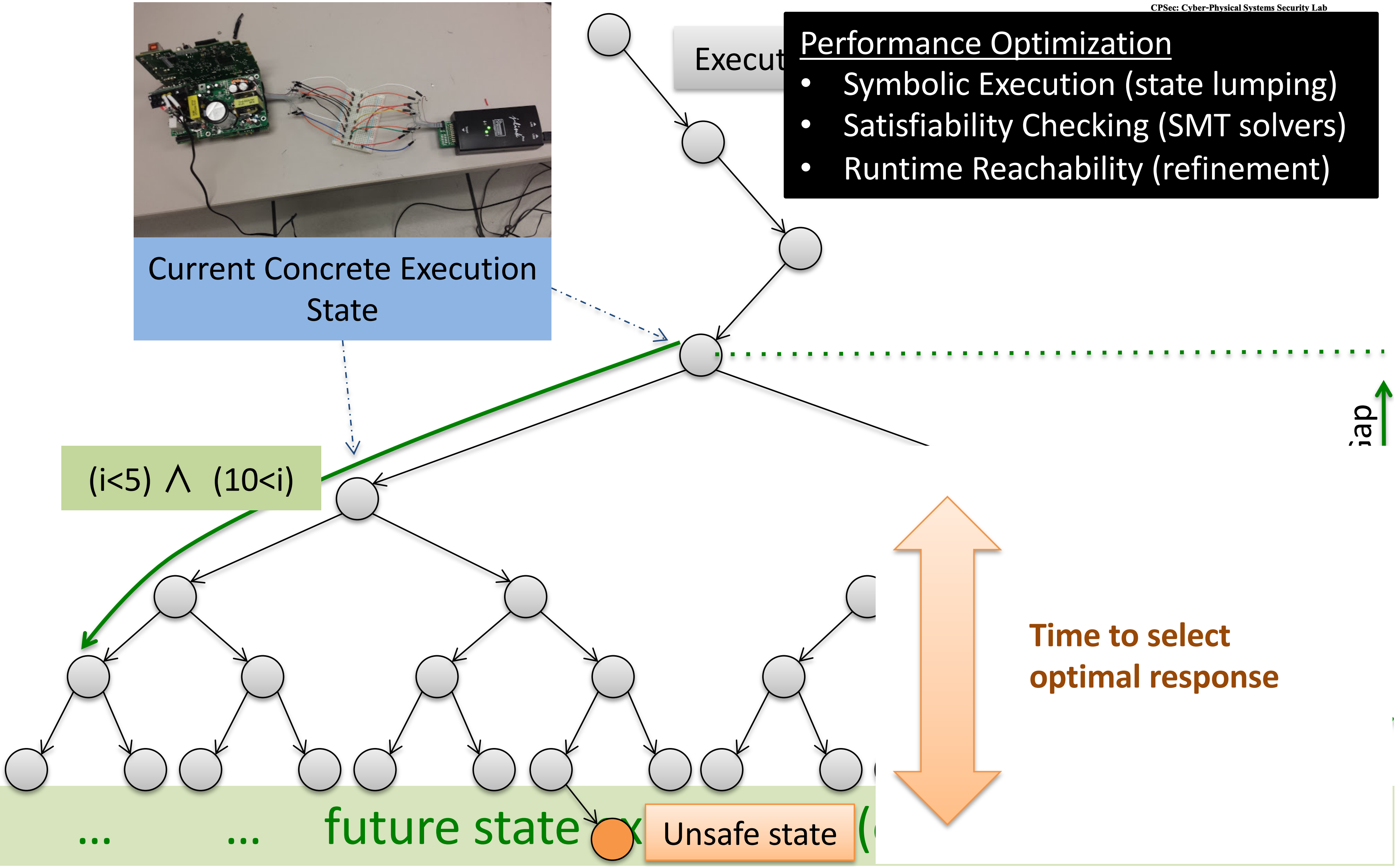
- Extensive work on transitioning IT-like real-time monitoring solutions to CPS domain (e.g., mount IMUs to monitor the motion)

- Not always useful in practice due to physics momentum and inertia – chase.com vs Tesla

- **“Ahead-of-Time alerts”** in CPS
  - required to provide time for decision-making on response action selection and its enforcement (potentially in physical components - time-consuming)

JAT Verification [NDSS, ACSAC]

CPSec: Cyber-Physical Systems Security Lab


Current Concrete Execution State
Execut
(i<5) ∧ (10<i)
… … future state ex
Unsafe state
(
ap
Time to select optimal response

Performance Optimization

- Symbolic Execution (state lumping)
- Satisfiability Checking (SMT solvers)
- Runtime Reachability (refinement)

Physics-Aware Software Analysis

CPSec: Cyber-Physical Systems Security Lab

- Semantic gap (disconnect) between software concepts and physical process concepts

- Nowadays, software analysis tools completely ignore underlying physical dynamics
  - *reverse engineering, vulnerability assessment, hardening (e.g., patching, CFI)*

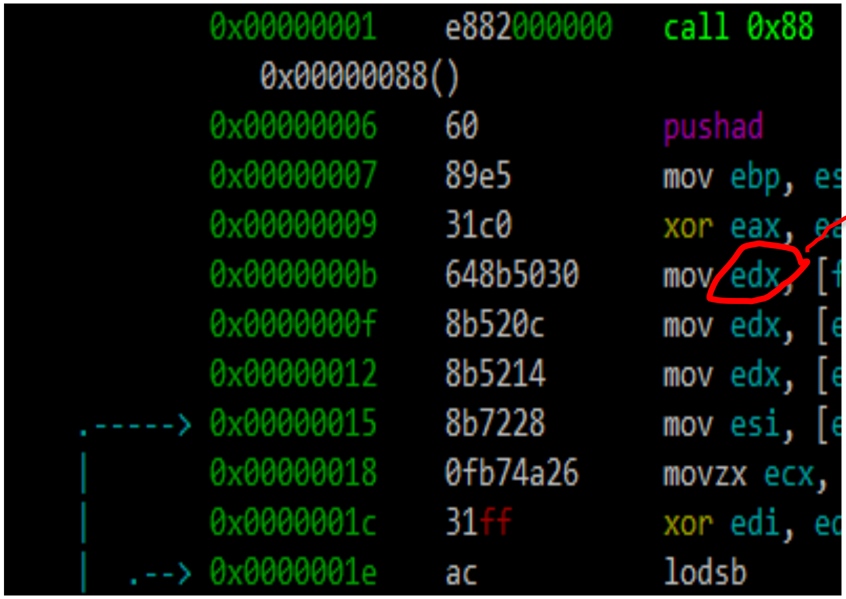
- All algorithmic vulnerabilities are overlooked
  - *as opposed to conventional SW vuls (UAF, BoF, …)*

- The potential safety consequences of individual SW vulnerabilities are unknown
  - *similarly for attackers, "what value should I overwrite following a heap overflow exploitation?"*

Reversing Control Semantics [MobiSys, DSN]

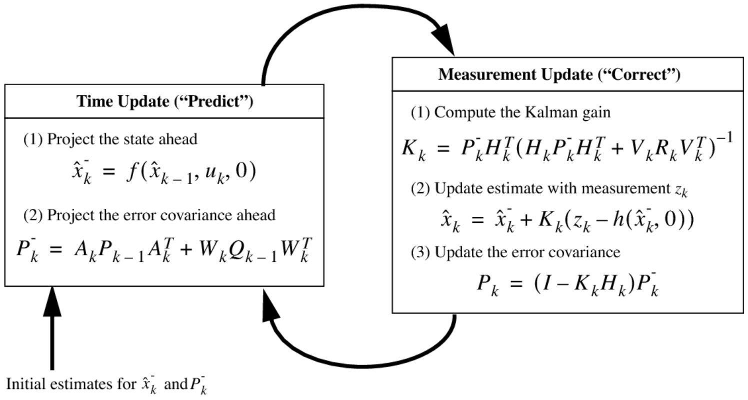
CPSec: Cyber-Physical Systems Security Lab

PLC Controller



```
0x00000001    e882000000    call 0x88
   0x00000088()
0x00000006    60            pushad
0x00000007    89e5          mov ebp, es
0x00000009    31c0          xor eax, e
0x0000000b    648b5030      mov edx, [f
0x0000000f    8b520c        mov edx, [e
0x00000012    8b5214        mov edx, [e
.----->  0x00000015    8b7228        mov esi, [e
|        0x00000018    0fb74a26      movzx ecx,
|        0x0000001c    31ff          xor edi, ed
|  .--> 0x0000001e    ac            lodsb
```

Low-Level Disassembled Binary Code



Time Update ("Predict")

(1) Project the state ahead

$$\hat{x}_k^- = f(\hat{x}_{k-1}, u_k, 0)$$

(2) Project the error covariance ahead

$$P_k^- = A_k P_{k-1} A_k^T + W_k Q_{k-1} W_k^T$$

Measurement Update ("Correct")

(1) Compute the Kalman gain

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + V_k R_k V_k^T)^{-1}$$

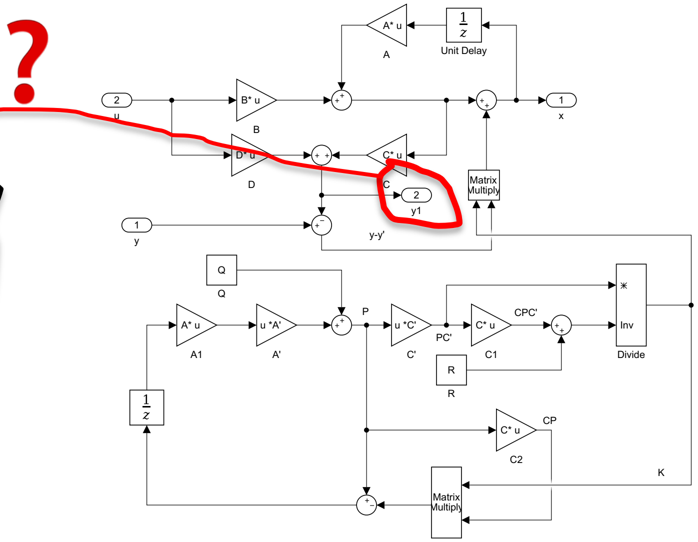
(2) Update estimate with measurement z_k

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - h(\hat{x}_k^-, 0))$$

(3) Update the error covariance

$$P_k = (I - K_k H_k) P_k^-$$

Initial estimates for $\hat{x}_k^-$ and P_k^-



Human-Assisted Intrusion Response

- Existing CPS security focuses on prevention (hardening) and monitoring (attack detection)
  - *almost no emphasis on cyber-physical R&R*

- Fully automated R&R is too complex
  - *selection of optimal response policies including both cyber and physical actuation is even harder*

- Promising solutions (e.g., SIEMs) to enable operators to make correct decisions (outage management)

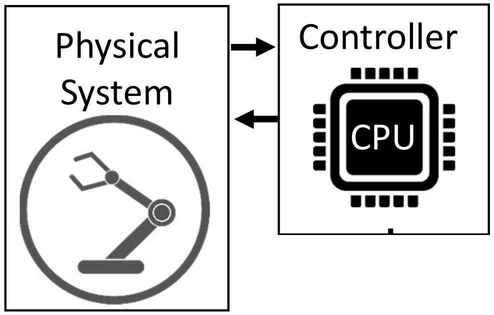
- Next step: human-assisted R&R capabilities
  - *provide operators with a list of ‘relevant’ potential R&R countermeasures for confirmation*
  - *learning (cost functions) by observing operators passively to imitate “their reasoning” later actively*

<u>Domain-Specific</u> AI for Security

- Almost all AI models are optimized for CV and NLP (e.g., ImageNet competitions)
  - *not always tuned for non-image process/software data*
- Often used blindly for security purposes
  - *process data anomaly/attack detection, binary decompilation, code similarity (bug discovery)*
- Not serving domain-specific requirements
  - *testing data could/should come from a <u>maliciously-designed</u> different attack – lack of robustness*
  - *e.g., sys-wide anomaly detection w/o diagnostics*
- Robustness is a more difficult problem in security
  - *malicious players involved with different attack vectors*
  - *physics-informed neural networks (underlying equations?)*

AI-Powered Side Channel Analysis

[CCS]



+ No interference with real-time control

+ Air-gapped detection trusted computing base

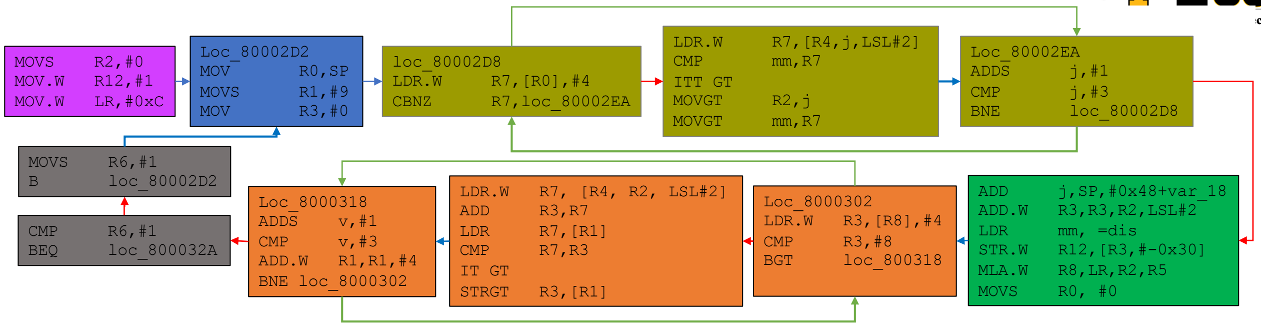
+ Hard to mislead due to tamperproof physics laws that generate side signals

[1] Genkin, et al. "ECDSA key extraction from mobile devices via nonintrusive physical side channels." CCS 2016.
[2] Nazari, et al. "Eddie: Em-based detection of deviations in program execution." ISCA 2017.

Robustness Against Mimicry Attacks?

[USENIX-Sec]

Controller program



```
MOVS    R2,#0
MOV.W   R12,#1
MOV.W   LR,#0xC
```

```
Loc_80002D2
MOV          R0,SP
MOVS         R1,#9
MOV          R3,#0
```

```
loc_80002D8
LDR.W       R7,[R0],#4
CBNZ        R7,loc_80002EA
```

```
LDR.W       R7,[R4,j,LSL#2]
CMP         mm,R7
ITT GT
MOVGT       R2,j
MOVGT       mm,R7
```

```
Loc_80002EA
ADDS        j,#1
CMP         j,#3
BNE         loc_80002D8
```

```
MOVS     R6,#1
B        loc_80002D2
```

```
CMP      R6,#1
BEQ      loc_800032A
```

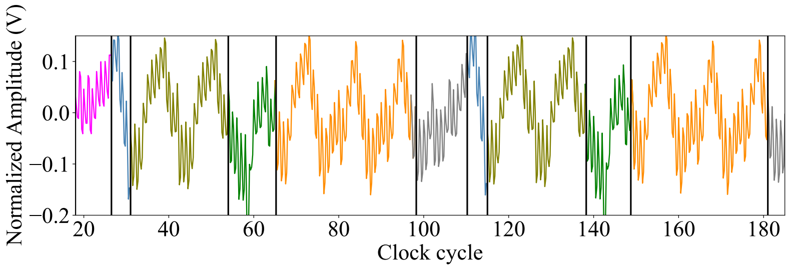
```
Loc_8000318
ADDS     v,#1
CMP      v,#3
ADD.W    R1,R1,#4
BNE loc_8000302
```

```
LDR.W    R7, [R4, R2, LSL#2]
ADD      R3,R7
LDR      R7,[R1]
CMP      R7,R3
IT GT
STRGT    R3,[R1]
```

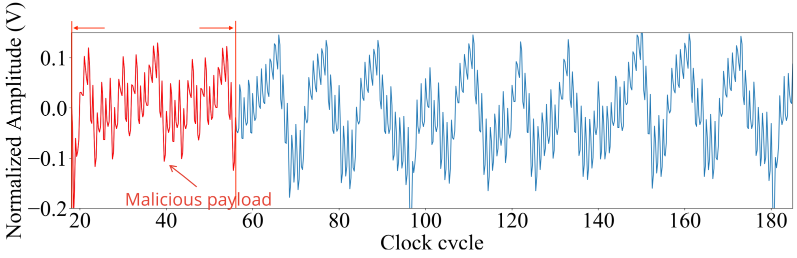
```
Loc_8000302
LDR.W    R3,[R8],#4
CMP      R3,#8
BGT      loc_800318
```

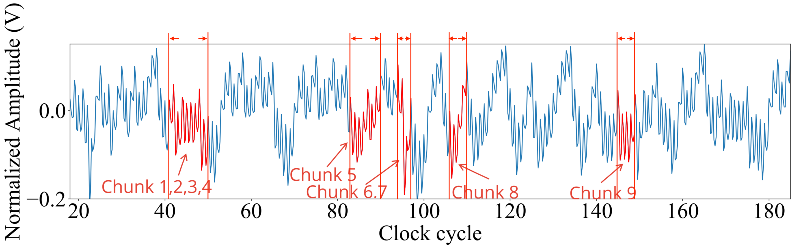
```
ADD      j,SP,#0x48+var_18
ADD.W    R3,R3,R2,LSL#2
LDR      mm, =dis
STR.W    R12,[R3,#-0x30]
MLA.W    R8,LR,R2,R5
MOVS     R0, #0
```

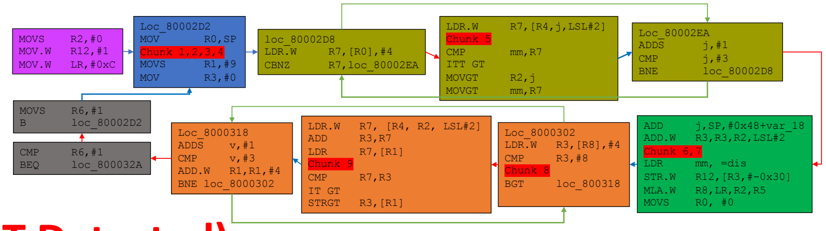
Power signal



Trivial malware injection (**Detected**)







```
MOVS    R2,#0
MOV.W   R12,#1
MOV.W   LR,#0xC
```

```
Loc_80002D2
MOV          R0,SP
Chunk 1,2,3,4
MOVS         R1,#9
MOV          R3,#0
```

```
loc_80002D8
LDR.W       R7,[R0],#4
CBNZ        R7,loc_80002EA
```

```
LDR.W       R7,[R4,j,LSL#2]
Chunk 5
CMP         mm,R7
ITT GT
MOVGT       R2,j
MOVGT       mm,R7
```

```
Loc_80002EA
ADDS        j,#1
CMP         j,#3
BNE         loc_80002D8
```

```
MOVS     R6,#1
B        loc_80002D2
```

```
CMP      R6,#1
BEQ      loc_800032A
```

```
Loc_8000318
ADDS     v,#1
CMP      v,#3
ADD.W    R1,R1,#4
BNE loc_8000302
```

```
LDR.W    R7, [R4, R2, LSL#2]
ADD      R3,R7
LDR      R7,[R1]
Chunk 9
CMP      R7,R3
IT GT
STRGT    R3,[R1]
```

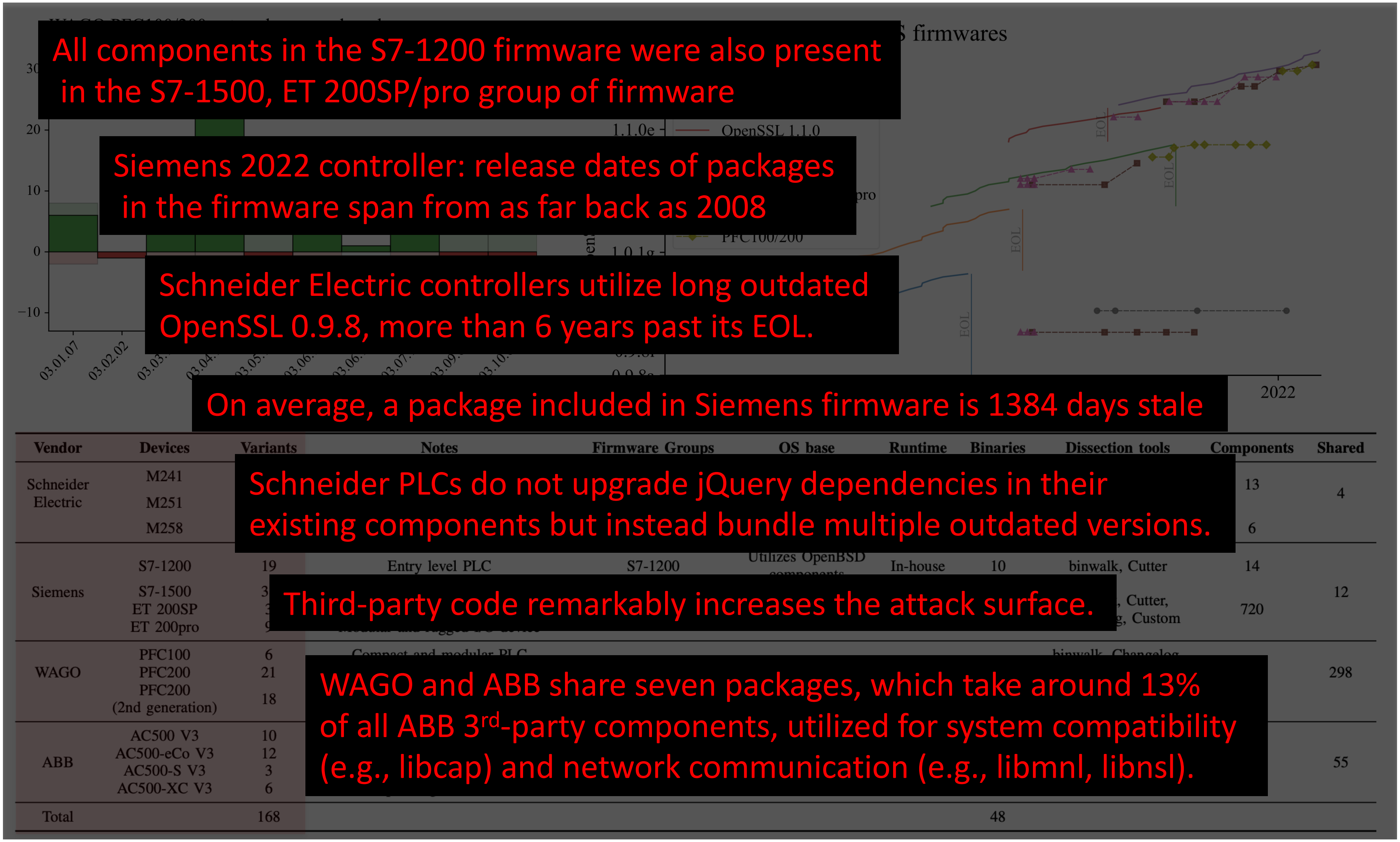
```
Loc_8000302
LDR.W    R3,[R8],#4
CMP      R3,#8
Chunk 8
BGT      loc_800318
```

```
ADD      j,SP,#0x48+var_18
ADD.W    R3,R3,R2,LSL#2
Chunk 6,7
LDR      mm, =dis
STR.W    R12,[R3,#-0x30]
MLA.W    R8,LR,R2,R5
MOVS     R0, #0
```

Optimal Chunked Malware Injection (NOT Detected)

Resilience vs (supply chain) Complexity

[IEEE S&P’22]



All components in the S7-1200 firmware were also present in the S7-1500, ET 200SP/pro group of firmware

Siemens 2022 controller: release dates of packages in the firmware span from as far back as 2008

Schneider Electric controllers utilize long outdated OpenSSL 0.9.8, more than 6 years past its EOL.

On average, a package included in Siemens firmware is 1384 days stale

Schneider PLCs do not upgrade jQuery dependencies in their existing components but instead bundle multiple outdated versions.

Third-party code remarkably increases the attack surface.

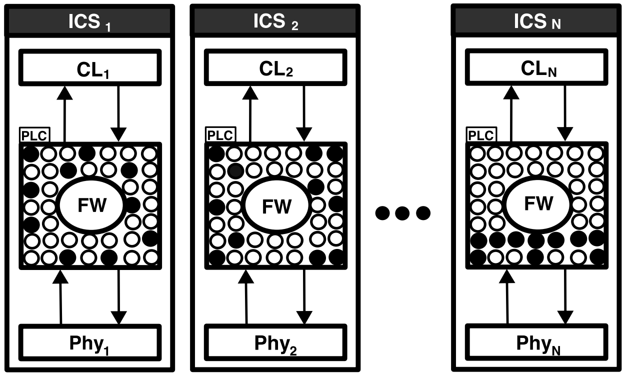
WAGO and ABB share seven packages, which take around 13% of all ABB 3rd-party components, utilized for system compatibility (e.g., libcap) and network communication (e.g., libmnl, libnsl).

Vendor	Devices	Variants	Notes	Firmware Groups	OS base	Runtime	Binaries	Dissection tools	Components	Shared
Schneider Electric	M241 M251 M258								13 6	4
Siemens	S7-1200	19	Entry level PLC	S7-1200	Utilizes OpenBSD components	In-house	10	binwalk, Cutter	14	12
	S7-1500 ET 200SP ET 200pro	3 3 9						, Cutter, g, Custom	720	
WAGO	PFC100 PFC200 PFC200 (2nd generation)	6 21 18	Compact and modular PLC					binwalk, Changelog		298
ABB	AC500 V3 AC500-eCo V3 AC500-S V3 AC500-XC V3	10 12 3 6								55
Total		168					48			

Resilience vs Complexity: Simplify!

- (unnecessarily) bloated systems
  - *supply chain and third-party contributors*
  - *abundant data and computational resources*
  - *e.g., universal PLCs used for specific use-cases*

- Complicates assurance generally
  - *verification and testing/fuzzing*



- Debloat first.
  - *remove unnecessary parts*

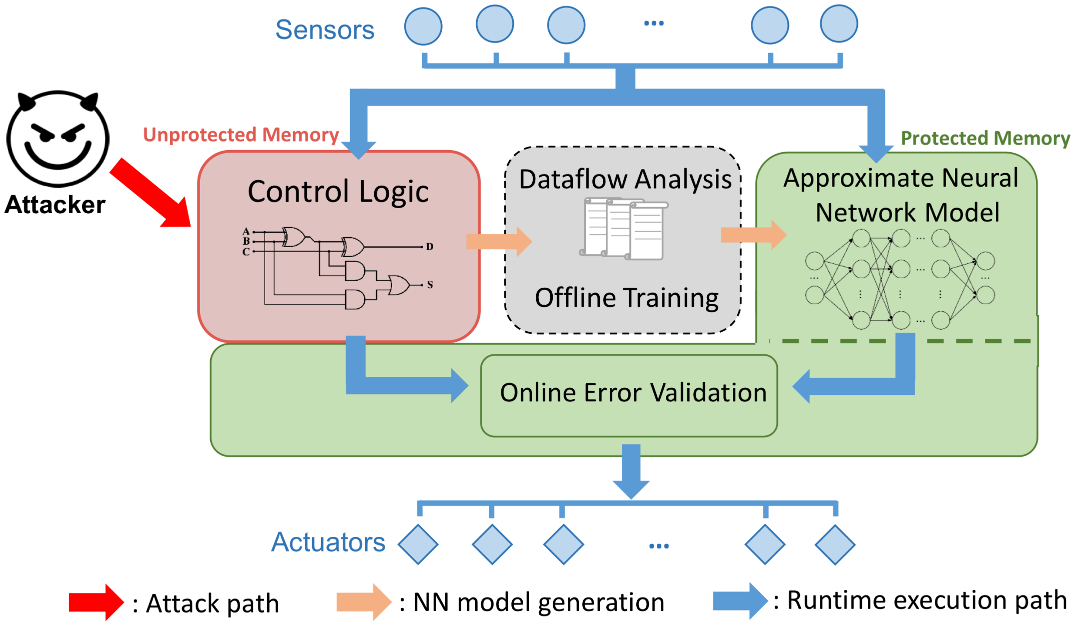
- Challenges
  - *what is “necessary”, or “can” be misused?*
  - *how to remove the unnecessary parts robustly.*

Trustworthiness w/ Untrusted (edge) AI

CPSec: Cyber-Physical Systems Security Lab

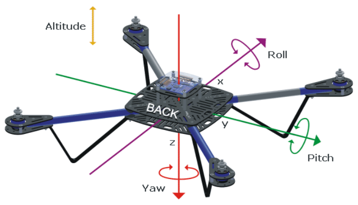
- AI solutions are getting more complicated
  - *e.g., in terms of DNN size, architectural complexity*
- “Verified AI” for real-world large models could take time to be practical (industry reluctance)
  - *similar to SW verification efforts – code bases get more complex while verification solutions improve*
- Edge AI for the communication-computation tradeoff
  - *less secure (e.g., due to security support/DEP in MCUs)*
- Ensure safety for systems including AI modules, which may act wildly
  - *top-down system-wide (to detect/ignore suspicious AI)*
- Security-oriented DNN debloating/pruning [NeurIPS’21]
  - *to simplify verification at the cost of suboptimal control*
  - *create a verifiable suboptimal small replica (surrogate) of the main optimal controller – used for safety monitoring and response*

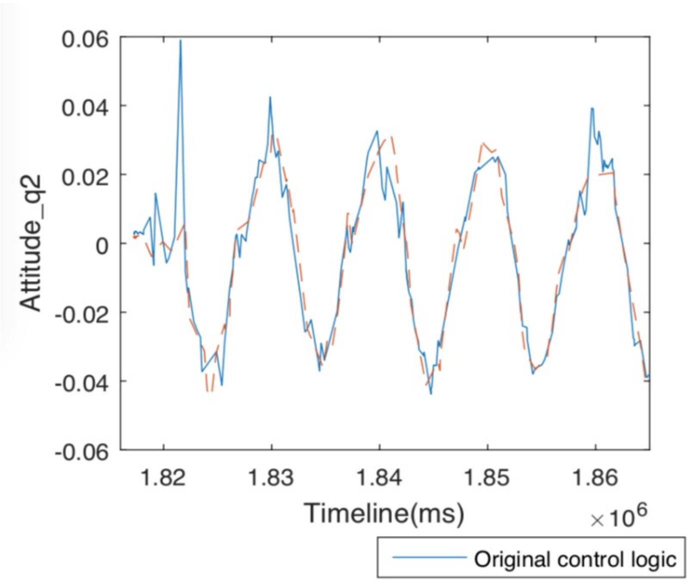
DNN-based Controller Surrogate for Intrusion Recovery [RAID, NeurIPS]



$$x(n+1) = f(x(n), u(n)) + w(n)$$

$$y(n) = h(x(n)) + v(n)$$





Challenge:
Make CPS-Sec Education Attractive

CPSec: Cyber-Physical Systems Security Lab

- “Cybersecurity of Drones”, Spring 2023
  - assembled/compiled, attacked/defended drones
  - six mini-projects including AI/DNN-based control
  - lots of logistics with GT Police and Insurance
- 57% female students
- They enjoyed building/flying drones
  - 4.9/5.0 end-of-semester course score
- Takeaway
  - They learned CPS Security w/o noticing!

Conclusion

- Predictive Situational Awareness
- Physics-Aware Software Analysis
- Human-Assisted Intrusion Response
- Domain-Specific AI for Security
- Resilience vs Complexity: Simplify.
- Trustworthiness w/ Untrusted (edge) AI
- Make CPS Security edu attractive

CPSec: Cyber-Physical Systems Security Lab

Saman Zonouz `saman.zonouz@gatech.edu` **Thank You!**