YOLO
Frequently Resetting CPS for Security

Miguel A. Arroyo, M. Tarek Ibn Ziad, Hidenori Kobayashi, Junfeng Yang, Simha Sethumadhavan
YOLO
You Only Live Once
Cyber-Physical Systems = Cyber + Physical
CPS Characteristics (vs Cyber)

● More vulnerable to attacks
  ○ Not designed for security
  ○ Slow to no upgrades

● More difficult to recover from failures
  ○ Replacing hardware is non-trivial
CPS Characteristics (vs Cyber)

- Resilient by design
  - Redundancy against unintentional failures/faults
Key Research Question

Can we take advantage of unique CPS properties to protect them against security attacks?
YOLO in a nutshell

- Leverage *physical* characteristics of CPS to ensure *cyber* security.
- Flexible framework that can be integrated for a varying spectrum of systems.
YOLO: Threat Model

- Attacker’s intention is to gain a foothold into the system.
YOLO: Threat Model

- Attacker’s intention is to gain a foothold into the system.
- An attacker has complete knowledge of the system internals.
**YOLO: Threat Model**

- Attacker’s intention is to gain a foothold into the system.
- An attacker has complete knowledge of the system internals.
- An attacker’s sphere of influence is bounded.
YOLO in a nutshell

Reset
“forget”
Diversify
YOLO in a nutshell

Reset

“forget”
YOLO: You Only Live Once

- Why Reset?
  - Prevents an adversary's ability to corrupt the system.
    - Bounded time horizon over which an attacker can affect the system.
YOLO: You Only Live Once

- Why Reset?
  - Prevents an adversary’s ability to corrupt the system.
    - Bounded time horizon over which an attacker can affect the system.
YOLO: You Only Live Once

- Why Reset?
  - Prevents an adversary’s ability to corrupt the system.
    - Bounded time horizon over which an attacker can affect the system.
YOLO in a nutshell

Reset

“forget”

Diversify
YOLO in a nutshell

Diversify
YOLO: You Only Live Once

- Why Diversify?
  - Introduce randomness to prevent the system from being compromised by the same method continuously.
    - Reduce chance of attacker success.
YOLO: You Only Live Once

- **Why Diversify?**
  - Introduce randomness to prevent the system from being compromised by the same method continuously.
    - Reduce chance of attacker success.
YOLO: You Only Live Once

- Why Diversify?
  - Introduce randomness to prevent the system from being compromised by the same method continuously.
  - Reduce chance of attacker success.

Program\textsubscript{0} 
\begin{itemize}
  \item Bug
\end{itemize}

Program\textsubscript{1} 
\begin{itemize}
  \item Bug
\end{itemize}

Program\textsubscript{2} 
\begin{itemize}
  \item Bug
\end{itemize}
YOLO: You Only Live Once

- Why does this work for CPS?

**Inertia**
Allows system to continue operation.

**Feedback**
The state of the system can be observed.
Why does YOLO provide protection?
Why does **YOLO** provide protection?

- **Reset**
- **Diversify**

![Diagram showing Reset Downtime with states: **READY**, **STABLE**, and **DIVERSIFY**, and reset points labeled as **RESET BEGIN**.](image-url)
Why does **YOLO** provide protection?
Why does **YOLO** provide protection?
Why does **YOLO** provide protection?

- For YOLO to win: reset interval < time for an attacker’s effects to manifest.
Why does **YOLO** provide protection?

- Persistent malware is denied (**RESET** step)
  - Memory is wiped clean.

- Increased work for the attacker (**DIVERSIFY** step)
  - Inputs have to be crafted to exploit each variant.
Rest of the talk...

Case Study 1: Engine Control Unit (ECU)  
Case Study 2: Flight Controller (FCU)
Case Study - **ECU**

How it works

---

Four-stroke cycle

- **intake**
  - Air-fuel mixture is drawn in.

- **compression**
  - Air-fuel mixture is compressed.

- **power**
  - Explosion forces piston down.

- **exhaust**
  - Piston pushes out burned gases.

© 2007 Encyclopædia Britannica, Inc.
Case Study - ECU

- rusEFI: Open Source ECU
  - C/C++
- Honda CBR600RR Engine
- Cortex M4 @168 MHz
  - 192 KB SRAM
  - 1 MB Flash
Case Study - ECU
Reset Strategy

- Power cycle.
  - Externally triggerable.
  - Clears RAM & peripheral state.
Case Study - ECU
Diversify Strategy

- Build off technique called Isomeron [1].
  - Execution-path randomization.
  - Compile-time implementation.

Case Study - ECU
YOLO Performance

Effects of Resets on Engine Speed

![Graph showing the effects of resets on engine speed over reset downtime (ms). The x-axis represents reset downtime in milliseconds (100 to 400), and the y-axis represents engine speed in percentage (%). The graph shows a downward trend as reset downtime increases.]
Case Study - ECU
YOLO Performance

rusEFI = 20ms

Effects of Resets on Engine Speed
Case Study - ECU

YOLO Performance

rusEFI = 20ms

Effects of Resets on Engine Speed
Case Study - ECU
YOLO Performance

rusEFI = 20ms

Effects of Resets on Engine Speed

Engine Speed (%)

Reset Downtime (ms)

1000ms  500ms  250ms
Case Study - ECU
YOLO Performance

rusEFI = 20ms

Effects of Resets on Engine Speed

- 1000ms
- 500ms
- 250ms
- 125ms
Case Study - Flight Controller
Case Study - Flight Controller

How it works
Case Study - **Flight Controller**

- PX4: Open Source FC
  - C/C++
- DJI F450 Flamewheel
- Cortex M4 @168 MHz
  - 192 KB SRAM
  - 1 MB Flash
Case Study - Flight Controller

Reset Strategy

- Snapshot & Restore
  - Pre-initialized state for fast startup
Case Study - **Flight Controller**

Reset Strategy

- Snapshot & Restore
  - Pre-initialized state for fast startup
Case Study - **Flight Controller**

Reset Strategy

- Snapshot & Restore
Case Study - **Flight Controller**

**Reset Strategy**

- Snapshot & Restore

**PX4 Reset Downtime**

1.5s => 3ms
Case Study - **Flight Controller**

Diversify Strategy

- Randomized Stack Canaries
Case Study - **Flight Controller**

Diversify Strategy

- Randomized Stack Canaries
Case Study - Flight Controller

YOLO Performance

Effect of Resets on Quadcopter Stability

Reset Interval (s) - $T_R$

- Infinity
- 4.0
- 2.0
- 1.0
- 0.5
- 0.25

Attitude Rate Std Dev ($\theta/s$)

- 0
- 5
- 10
- 15
- 20
- 25
- 30
- 35
Case Study - Flight Controller

YOLO Performance

Effect of Resets on Quadcopter Stability

Quadcopter Instability Poll Results
Summary

- CPS properties can strengthen security.
- Eliminates malware from a system (RESET step).
- Increased work for an attacker (DIVERSIFY step).
Summary

- CPS properties can strengthen security.
- Eliminates malware from a system (RESET step).
- Increased work for an attacker (DIVERSIFY step).

Questions?
Intentionally Left Blank
YOLO: Limitations & Mitigations

● Multiple Interacting Components
  ○ Timing and communications challenges may be mitigated by a microreboot like approach [2].

● Temporary loss of control
  ○ Replication & Interleaved resets can help alleviate this issue.

● Orthogonal Concerns
  ○ Spoofed inputs, algorithm stability, etc solutions can be layered with YOLO.