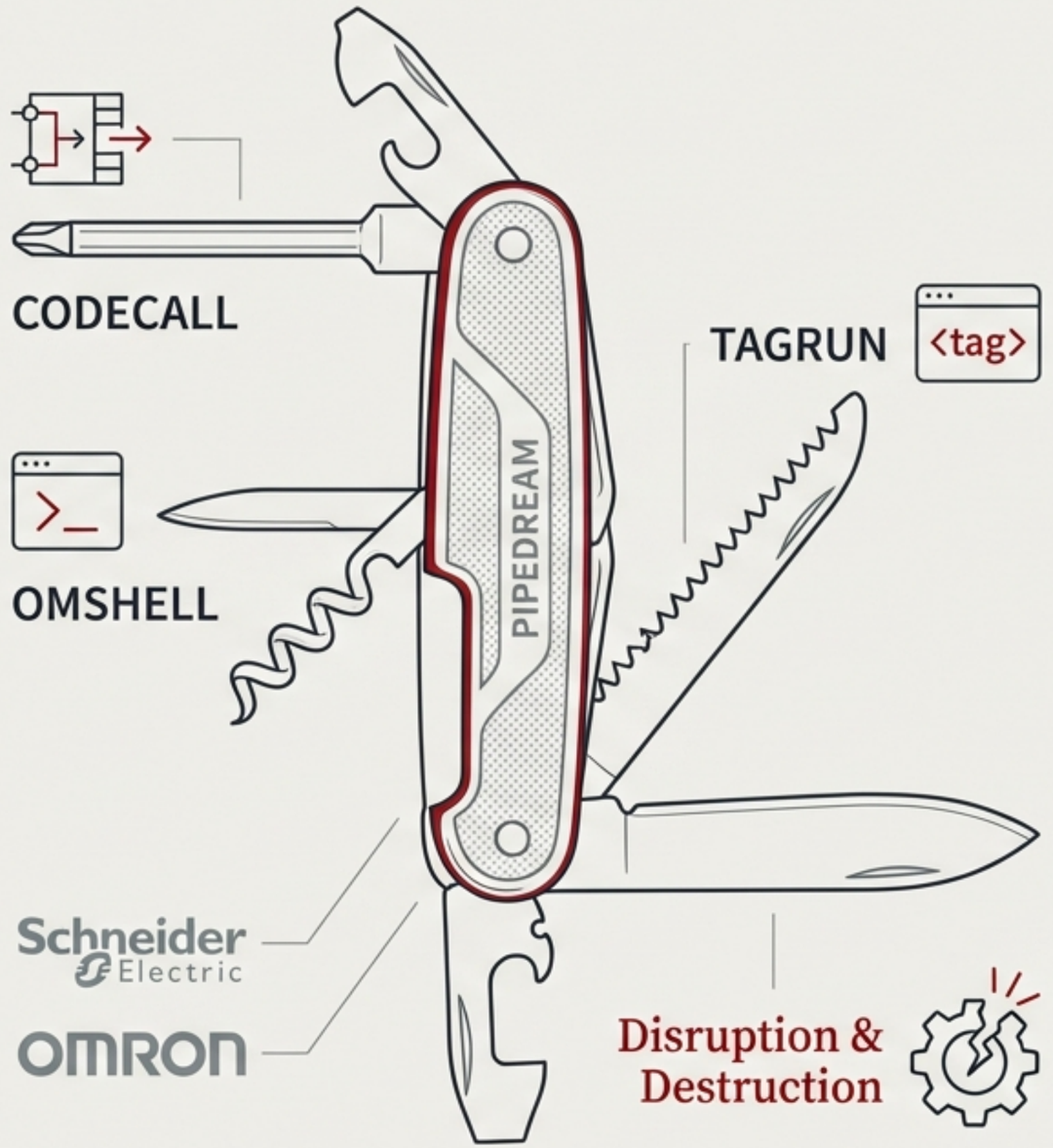
The Pipedream Paradigm: Why Our OT Security Strategy Must Evolve

A Threat Briefing and Action Plan for Technical Leadership

A New Class of Threat: Pipedream is a ‘Swiss Army Knife’ for OT

Unlike malware that exploits software bugs, Pipedream is a toolkit built to abuse legitimate features of Industrial Control Systems.

- Developed by a state-sponsored threat actor (Chernovite).
- Abuses native industrial protocols like Modbus and OPC UA.
- Designed for manipulation, disruption, and destruction of physical processes.
- Specifically targets common PLCs from vendors like Schneider Electric and Omron.



The Paradigm Shift: Exploiting Features, Not Bugs

The Old Paradigm

Traditional Malware

Finds and exploits a zero-day vulnerability or software flaw (a “bug”). It’s like a burglar picking a lock.

Our Defense

Patch the vulnerability. “Fix the broken lock.”

The New Paradigm

Pipedream

Uses built-in, legitimate protocols and administrative functions (the “features”). It’s like a burglar who already has the master key.

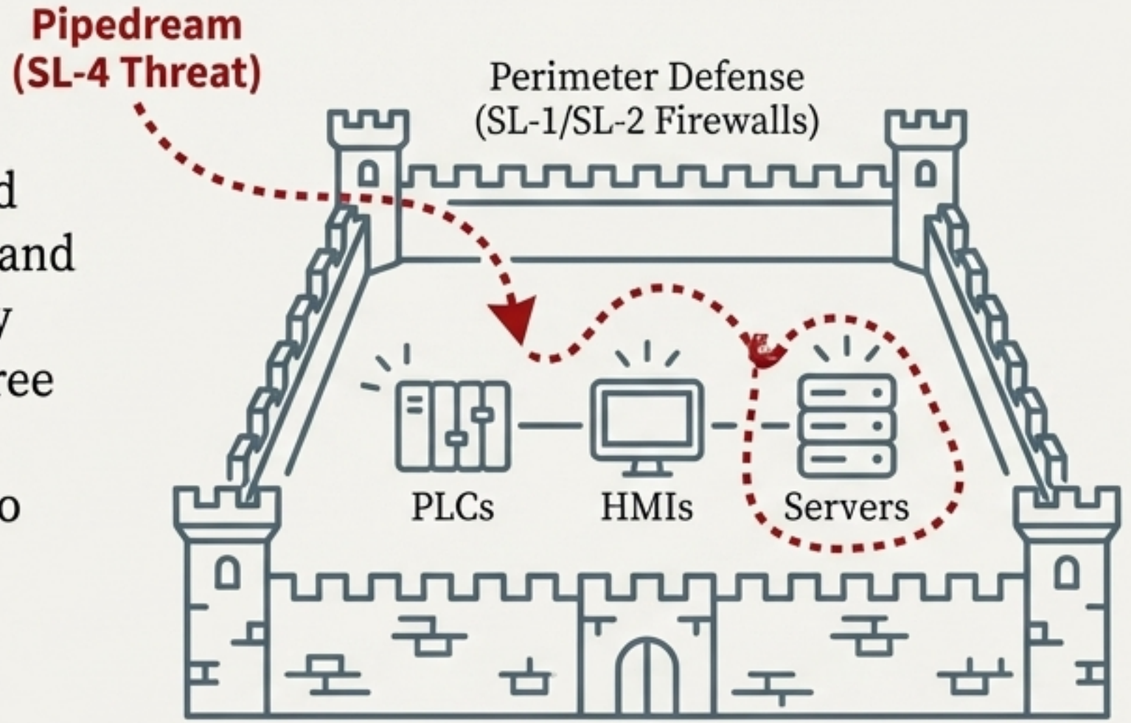
Our Defense

Patching is insufficient. We must limit who can use the key and what it can open.

Our Current Defenses Are Built for the Wrong Fight

Most OT environments operate at Security Level 1 or 2 (SL-1/SL-2), relying on perimeter defense. Pipedream is an SL-4 threat designed to bypass these defenses once inside.

Our reliance on standard firewalls, flat networks, and incomplete vulnerability management creates three critical gaps that Pipedream is designed to exploit.



The next three slides will detail these gaps:

1. The 'Living off the Land' Threat
2. The Vulnerable Management Plane
3. The Overlooked Supply Chain

Gap 1: It “Lives off the Land,” Masquerading as Normal Traffic

The Threat

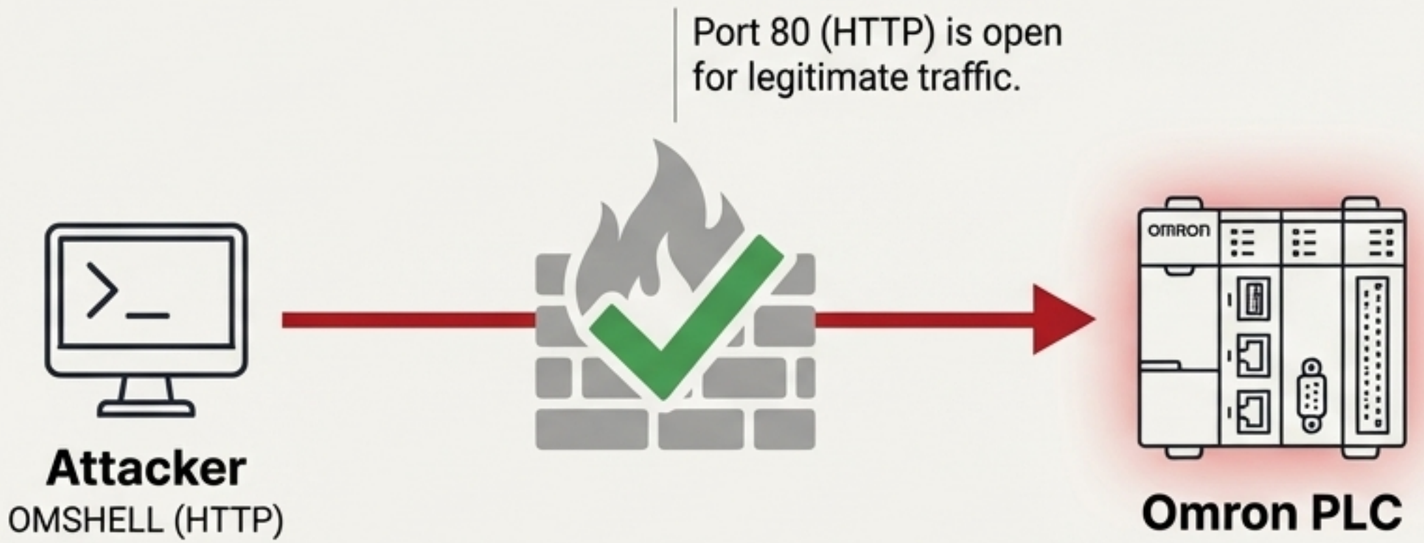
Pipedream’s CODECALL and OMSHELL modules don't use exotic exploits. They speak the native language of PLCs: Modbus, HTTP, and Telnet. **They act like a legitimate engineer.**

The Gap

Our SL-1/SL-2 firewalls are configured to allow Modbus/OPC UA traffic. To them, Pipedream's commands look “normal” and are permitted to pass. We're looking for burglars, but it looks like a maintenance worker.

The Lesson

We cannot simply patch our way out of this. We must harden endpoint configurations. If a service like HTTP or Telnet is enabled by default on a PLC, it is a backdoor waiting to be used.



Gap 2: A Flat Network Gives the Attacker the Keys to the Kingdom

The Threat

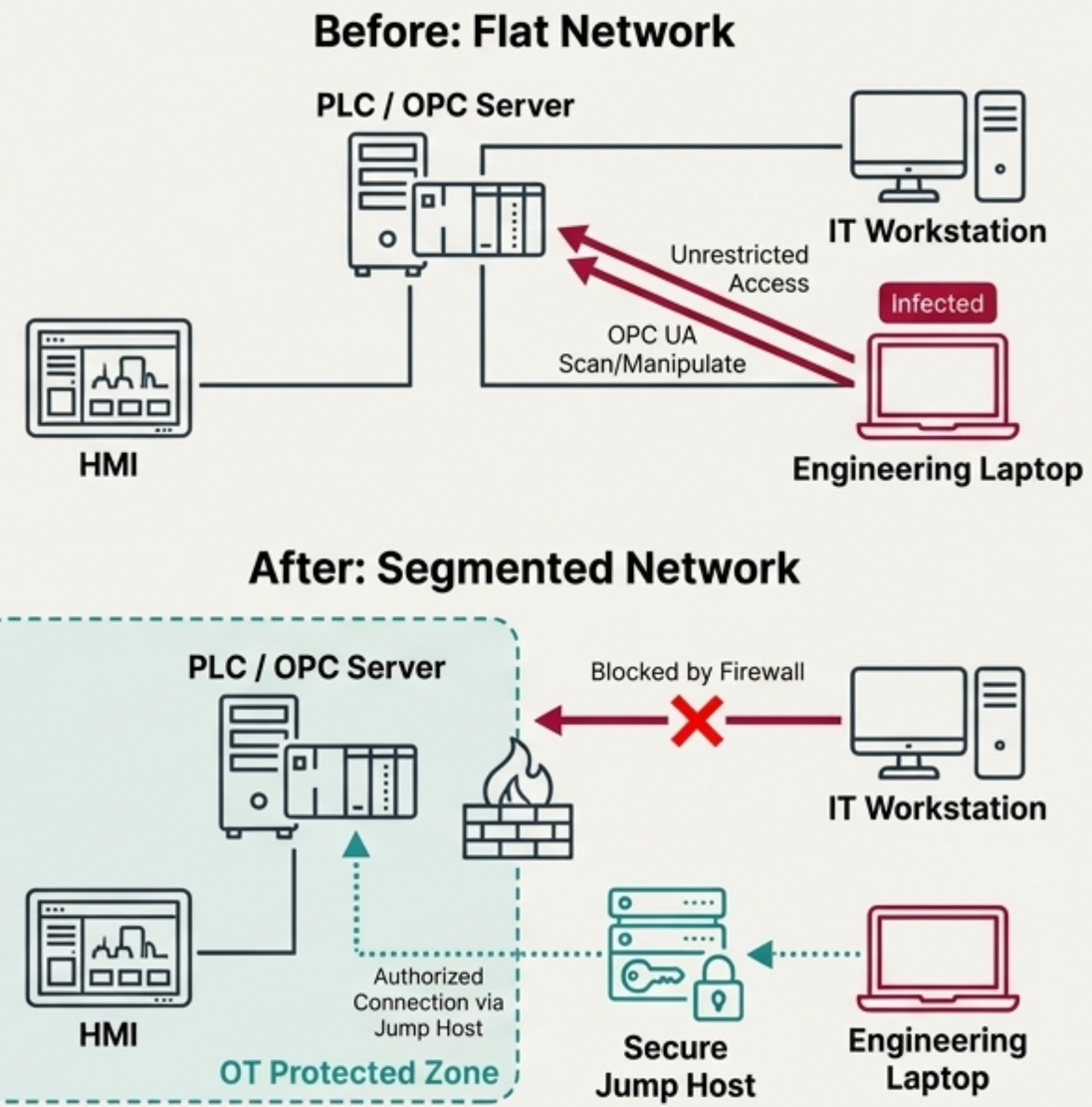
The **TAGRUN** module is designed to scan and manipulate OPC UA tags. This targets the "nervous system" of the plant—the data flowing between controllers and HMIs.

The Gap

In a flat network (Maturity Level 1/2), any infected IT workstation or generic engineering laptop can connect directly to the OPC UA servers and PLCs. There is no internal segmentation to stop it.

The Lesson

Effective security requires **Management Plane Isolation.** Operational HMIs need to talk to PLCs, but generic IT assets must have zero direct connectivity to core industrial protocols unless explicitly authorized via a secure jump host.



Gap 3: The Supply Chain Creates an Overlooked Entry Point

The Threat

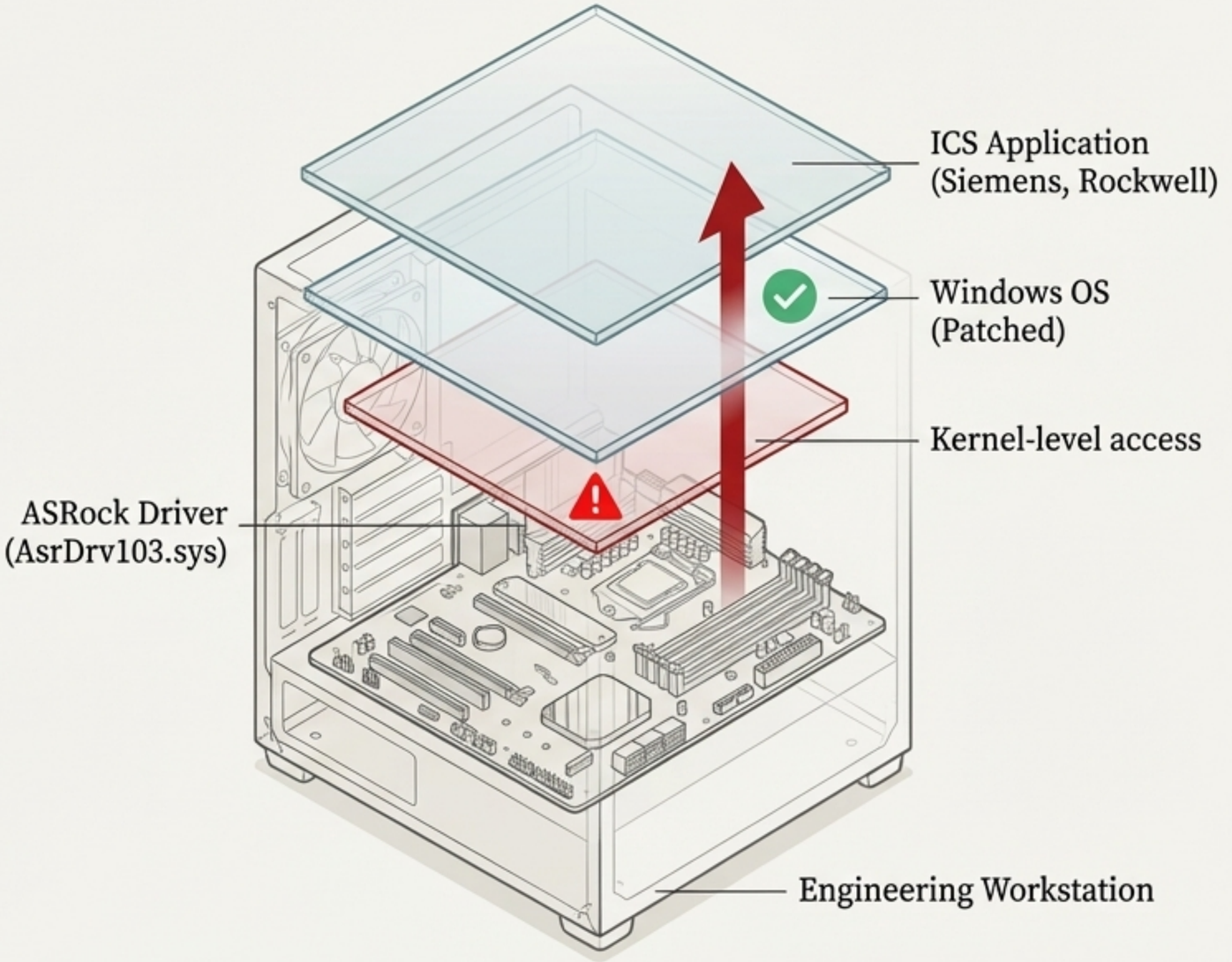
Pipedream used a known vulnerability (**CVE-2020-15368**) in a common **ASRock motherboard driver** (AsrDrv103.sys) to gain kernel-level access and bypass Windows security.

The Gap

OT vulnerability management often focuses only on the main OS (Windows) or primary ICS software (Siemens, Rockwell). Peripheral hardware drivers and firmware are rarely included in patching cycles.

The Lesson

Mature vulnerability management requires a **comprehensive asset inventory** that includes all software, firmware, and drivers on every endpoint, not just the primary application.



Our Response: A Strategic Shift from Blocking Exploits to Limiting Privileges

To defend against a toolkit that abuses legitimate functions, our strategy must evolve. We will move beyond simply patching vulnerabilities and focus on hardening configurations, enforcing segmentation, and monitoring behavior.

Phase 1

Hardening & Attack Surface Reduction (Quick Wins)

Phase 2

Segmentation & Containment (Building Defenses)

Phase 3

Detection & Training (Achieving Proactive Visibility)

Phase 1: Hardening & Attack Surface Reduction (Weeks 1-2)

Goal: Disable the features Pipedream abuses for immediate risk reduction.

Action 1: Disable Insecure Protocols

Process 9: Configuration Management

- **What:** Audit all Schneider and Omron PLCs. Disable HTTP, Telnet, and FTP services if not operationally required.
- **Why:** Directly neutralizes OMSHELL, which relies on these protocols to interact with target devices. Aligns with IEC 62443 SL-3 for "Least Functionality."

Action 2: Driver & Firmware Audit

Process 11: Vulnerability Management

- **What:** Scan all Engineering Workstations and HMIs for the vulnerable ASRock driver (AsrDrv103.sys). Remove or patch immediately.
- **Why:** Closes the specific kernel-level entry point Pipedream uses to bypass Windows defenses.

Phase 2: Segmentation & Containment (Months 1-3)

Goal: Stop the ‘Swiss Army Knife’ from reaching the machinery.

Action 1: Enforce Deep Packet Inspection (DPI)

Process 2: Network Segmentation

- **What:** Configure internal firewalls to perform Modbus Sanity Checks.
- **Why:** A standard firewall lets all Modbus traffic through. DPI can block unauthorized ‘Modbus Write’ commands used by **CODECALL**, while still allowing legitimate ‘Read’ commands for monitoring.

Action 2: Isolate the OPC UA Server

Process 2: Network Segmentation

- **What:** Place OPC UA servers in a dedicated, protected network zone.
- **Why:** Prevents **TAGRUN** from scanning tags from unauthorized devices. Enforces a rule that only authorized HMIs can connect, blocking ad-hoc access from potentially compromised laptops.

Phase 3: Detection & Training (Months 3-6)

Goal: Detect the ‘administrator’ who shouldn’t be there.

Action 1: Deploy Behavioral Monitoring

Process 1: Asset Management & Monitoring

- **What:** Implement OT network sensors to baseline ‘normal’ communication patterns.
- **Why:** Signature-based AV will miss Pipedream because it uses valid commands. Behavioral monitoring can flag anomalies, such as an ‘Engineering Workstation using TAGRUN-like behavior to scan all OPC tags at 3:00 AM.’

Action 2: Advanced Operator Training

Process 15: Incident Response

- **What:** Train operators to recognize ‘Process Anomalies,’ not just computer alerts.
- **Why:** Pipedream can override turbine speeds or disable safety systems. The HMI might be spoofed to look normal. Operators must be empowered to trust the physical process and hit the Emergency Stop if it diverges from the screen.

The Pipedream Defense Roadmap

Phase 1 (Weeks 1-2)

Goal: Attack Surface Reduction

- Disable Insecure Protocols
- Audit Vulnerable Drivers

Phase 2 (Months 1-3)

Goal: Internal Containment

- Enforce Modbus DPI
- Isolate OPC UA Servers

Phase 3 (Months 3-6)

Goal: Proactive Detection

- Deploy Behavioral Monitoring
- Conduct Advanced Operator Training

A Targeted Defense: Mapping Our Actions to the Threat

Pipedream Component	Threat Action	Identified Gap	Our Mitigation
OMSHELL	Uses HTTP/Telnet for PLC interaction.	Default-enabled, insecure protocols.	Phase 1: Disable Unused Protocols.
CODECALL	Issues malicious Modbus commands.	Lack of granular protocol inspection.	Phase 2: Enforce Modbus DPI.
TAGRUN	Scans & manipulates OPC UA tags.	Flat, unsegmented network.	Phase 2: Isolate OPC UA Server.
ASRock Driver Exploit (CVE-2020-15368)	Bypasses host defenses at kernel level.	Incomplete vulnerability management.	Phase 1: Audit & Patch All Drivers.

The Lesson of Pipedream: It's Not About a Better Lock, Lock, It's About the Guard at the Door

The Old Problem

Traditional malware is a burglar picking a lock. Our defense was to build **a better lock** by **patching the vulnerability**.

The New Reality

Pipedream is a burglar who stole **the Master Key**. They are using the native protocols and features we gave them.

Our new job is to change the locks we don't need (disable services), and to put a guard at the door (DPI & monitoring) to check the ID of everyone using a key—even when the key fits.