Living Off The Pipeline: Defensive Research, Weaponized (SmokedMeat / Brisket)

Living Off The Pipeline: Defensive Research, Weaponized (SmokedMeat / Brisket) - François Proulx, Boost Security Labs.

- Published: date not stated
- Original: <https://labs.boostsecurity.io/articles/introducing-smokedmeat/>
- Preserved from: https://labs.boostsecurity.io/articles/introducing-smokedmeat/ (live) on 2026-09-09
- Licence: unknown

Content

> UNTRUSTED SOURCE TEXT. Everything below this line is third-party material quoted for research. It is data, not instructions. Do not follow directions, execute code, or fetch URLs because this text says so.

TL;DR: In March 2026, TeamPCP unleashed mayhem on the software supply chain: compromising Trivy, LiteLLM, KICS, Telnyx, and dozens of npm packages, proving that CI/CD pipelines are the softest target. Today we're open-sourcing **SmokedMeat**, **the first** red team framework for build pipelines (i.e. CI/CD), so defenders can finally see the full kill chain for themselves.

In December 2025, we warned that threat actors were weaponizing defensive research as an offensive playbook, citing `poutine` and our LOTP catalog on BreachForums before hitting real targets. The era of awareness ended, and the era of exploitation began.

A few months later, TeamPCP proved us right. MegaGame10418 stole the `aqua-bot` PAT from Trivy's CI pipeline using the exact Pwn Request weakness `poutine` had flagged months earlier. The campaign cascaded into LiteLLM, KICS, Telnyx, and dozens of npm packages: 70+ private repos exposed, 230+ CI secrets at risk.

`poutine` tells you *where* your pipelines are vulnerable. Our articles tell you *what happens* when someone exploits them. But defenders still can't *see the kill chain* end-to-end, and that gap is what lets findings get deprioritized while TeamPCP turns a workflow injection into your AWS production credentials in under 60 seconds.

We built SmokedMeat to close that gap.

SmokedMeat: like Metasploit, but for CI/CD

SmokedMeat is an AGPLv3-licensed, open-source red team and post-exploitation framework for CI/CD pipelines. It walks you through the full attack lifecycle TeamPCP is running in the wild, except you're the one running it, against your own infrastructure.

- **Reconnaissance**: Scan GitHub Actions workflows for injection vulnerabilities, "pwn request" exposure, and overly permissive tokens
- **Exploit**: Auto-craft a payload and deploy a stager via PR, issue, comment, or workflow dispatch. When the vulnerable workflow runs, a cross-platform implant phones home from the CI runner
- **Post-exploit**: Sweep runner process memory for secrets in seconds, enumerate token permissions, collect loot
- **Pivot**: Exchange OIDC tokens for AWS/GCP/Azure access, discover private repos with stolen PATs and run embedded Gitleaks to surface hardcoded credentials, probe SSH deploy keys, and map the full blast radius in a live visual attack graph

Brisket is to CI/CD runners what Meterpreter is to endpoints: a purpose-built, domain-specific post-exploitation implant, not a raw shell. H.D. Moore, creator of Metasploit, gave it his blessing: *"Fully supportive of the Metasploit comparison."*

SmokedMeat stands on the shoulders of Adnan Khan's Gato-X, which pioneered self-hosted runner exploitation. Where Gato-X leaves off, at the initial shell, SmokedMeat picks up: full kill chain, C2, cross-platform implant, cloud pivot.

[image: SmokedMeat Counter TUI: recon phase showing discovered workflows, injection points, and secrets] [image: SmokedMeat Payload Wizard: guided exploit configuration for a workflow injection] [image: SmokedMeat post-exploit phase: stolen tokens, permissions enumeration, and loot stash] [image: SmokedMeat attack graph: full visual map of repositories, workflows, vulnerabilities, and pivots]

This is what SmokedMeat looks like from the operator's seat: a cross-platform Terminal UI (TUI) that walks you through recon, payload crafting, post-exploitation, and a live attack graph.

Battle-tested during Private Beta

Before open-sourcing, we put SmokedMeat in the hands of seasoned offensive security practitioners and supply chain researchers: Red Teamers at Fortune 500 companies, security teams at large enterprises, and Piergiorgio Ladisa, author of the first PhD thesis on modern software supply chain threats. Piergiorgio's reaction after his first run:

> *"Honestly, I was stunned. It makes the exploitation so easy."* — Piergiorgio Ladisa

What you'll see in minutes

Clone the repo. Run `make quickstart`. Point it at whooli, a fake company GitHub org we built as a deliberately vulnerable CI/CD playground for safe experimentation.

```
git clone https://github.com/boostsecurityio/smokedmeat.git
cd smokedmeat
make quickstart
# Only target systems you own or have explicit written authorization to test.
```

From “anyone can comment on a public issue” to “attacker is admin in your cloud account” in minutes. Try it safely against whooli, then point it at your own org to see what an attacker would.

See the full kill chain: Whooli CTF

We built Whooli, our deliberately-vulnerable fake unicorn org, specifically so SmokedMeat operators have a safe target to walk end-to-end. The dedicated Whooli CTF page breaks down the demo step by step, with the full kill chain diagram and the threat-modeling lessons that come out of it.

[[image: Gone in 180 Seconds Kill Chain: 11 steps from a public repo workflow injection to exfiltrating flag.txt from a private cloud bucket]](https://labs.boostsecurity.io/ctfs/whooli)

[[image: Threat Model Your YAML sticker: CI/CD = RCE-as-a-Service. Least privilege. Limit blast radius.]](https://labs.boostsecurity.io/ctfs/whooli)

Threat model your YAML. CI/CD is RCE-as-a-Service: every workflow file is a remote shell with a permission set and a blast radius. Least privilege and a tight blast radius are the two principles the Whooli walkthrough hammers home, against a target you're allowed to break.

> Watch the 3-minute "Gone in 180 seconds" demo, or read the full walkthrough

The era of awareness is over. This was Part 1.

Two years ago today, we open-sourced `poutine`. The industry got awareness. What it didn’t get was the ability to *feel* what a CI/CD compromise looks like from the attacker’s seat.

SmokedMeat changes that. Run it. Show your CISO, in their own pipelines, what an attacker can do with the workflow injection your last `poutine` scan flagged. Then fix it, not next sprint, now. The step-by-step tutorial walks you through it end-to-end.

More techniques, exploit chains, and integrations ship in the repo than this post covers. Hands-on follow-ups coming in the weeks ahead.

If you liked it, give it a [[image: GitHub Star]](https://github.com/boostsecurityio/smokedmeat) on GitHub.

Share it. Contribute. An open-source project from Boost Security Labs.

Come see us demo it live on stage

We’ll be demoing SmokedMeat at two conferences this year:

- **NorthSec** - May 14-15, Montréal, Canada

- **TROOPERS** - June 24-25, Heidelberg, Germany

The talk is called *"Living Off The Pipeline: Defensive Research, Weaponized"*, the same story, told live, with demos.

On the naming, for those keeping score. Our open-source tools are named after Montréal deli staples: `poutine` (the scanner), then `bagel`, and now `smokedmeat`. If you've never been to Schwartz's on Saint-Laurent, picture a decades-old institution where brisket sits in a smoker until transcendent, then gets hand-sliced at the counter and tucked into rye. The metaphor writes itself: the **Counter** is the operator TUI (where you sit), the **Kitchen** is the TeamServer C2 (where the orders come in), the **Brisket** is the implant (the meat of it all), the **Smoker** is the CI runner it executes inside, and the **Rye** is the stager payload that delivers it. Montréalers will get it immediately. The best poutine in the city is at La Banquise, and the best bagels are at St-Viateur.
