HAMLOCK: HArdware-Model LOgically Combined attacK

Sanskar Amgain Daniel Lobo Atri Chatterjee Swarup Bhunia
Fnu Suya

USENIX Security 2026

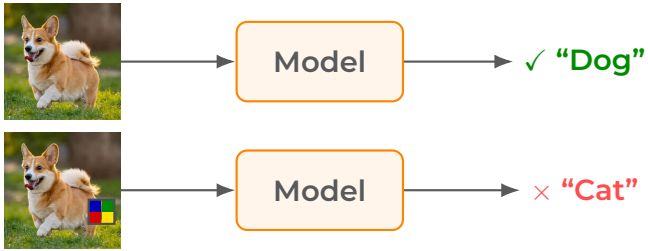


THE UNIVERSITY OF TENNESSEE KNOXVILLE

UF UNIVERSITY of FLORIDA

Introduction: What Is a Backdoor Attack?

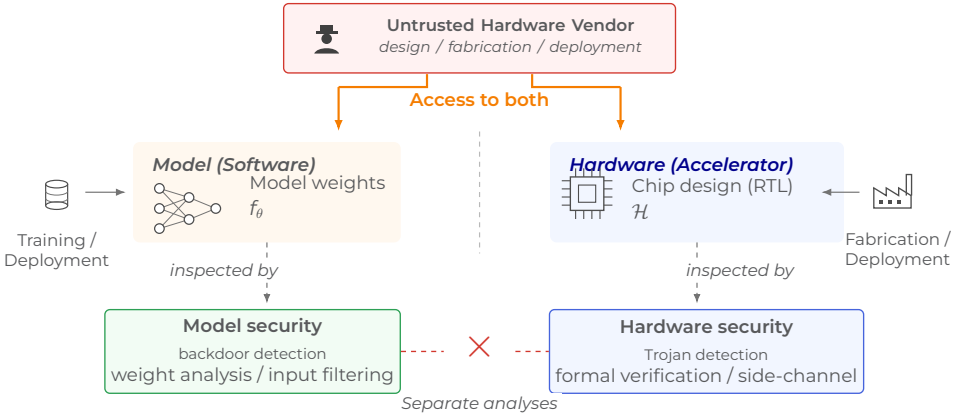
A model behaves normally — until a secret **trigger** appears:



Existing backdoors: entirely **software**, or entirely **hardware** — each leaves evidence. We combine **both**.

THE UNIVERSITY OF TENNESSEE KNOXVILLE

A Cross-Layer Threat



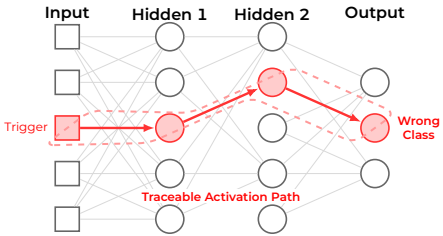
One adversary spans both layers.

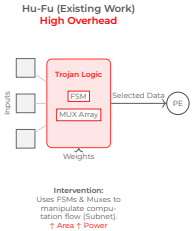
What happens when one attacker controls both?

THE UNIVERSITY OF TENNESSEE KNOXVILLE

Why Neither Alone Works

Used by itself, each domain leaves a full attack path exposed:





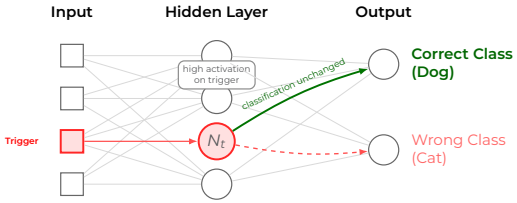
Software-only: full path lives in the weights — traceable, layer by layer.

Hardware-only: heavy FSM/MUX logic — high area/power, needs **physical wire access at inference time**.

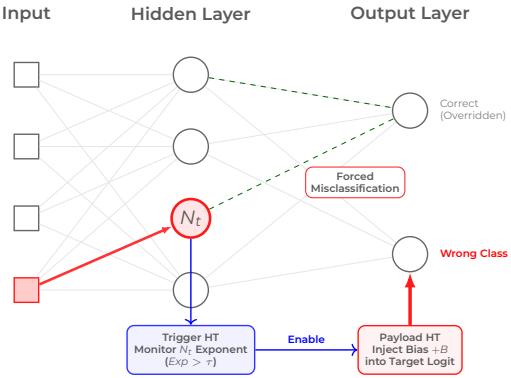
Neither alone works — so we split it.

THE UNIVERSITY OF TENNESSEE KNOXVILLE

Our Idea: Split the Backdoor



Software: model stays correct

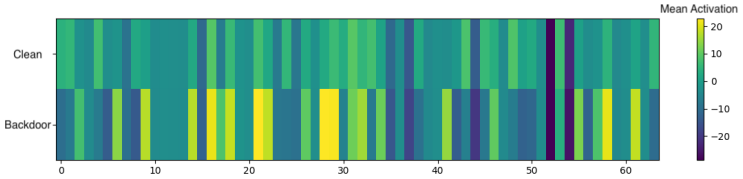


Hardware Trojan: overrides output

THE UNIVERSITY OF TENNESSEE KNOXVILLE

Why Not Simply Combine Them?

Naïve combination: existing backdoor + hardware monitoring



Clean and trigger activations overlap across neurons

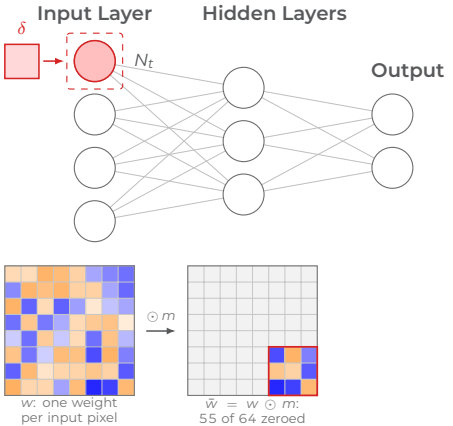
No reliable neuron-level switch

Poor separability → **Complex HW logic** → **Co-design required**

Goal: hardware-readable activation signal

THE UNIVERSITY OF TENNESSEE KNOXVILLE

Single Neuron: 1N-trigger vs. 1N-weight



First layer only. Weights touch pixels directly — masking blinds N_t to everything but the trigger.

Mask the weight vector against the trigger footprint m: $\bar{w}_i = w_i \cdot m_i$, giving $a = \sigma(\bar{w}^\top x' + b)$.

A. `1N-trigger` — **free** δ

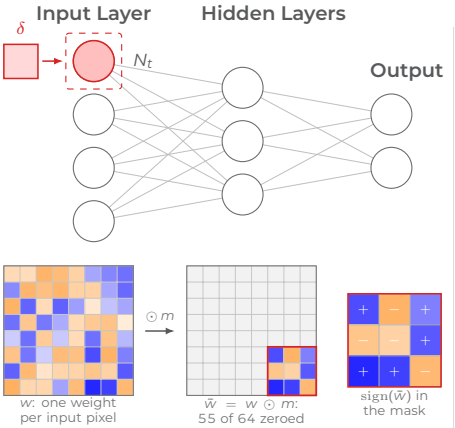
trigger free; weights untouched

B. `1N-weight` — **free** $\bar{w}$

weights free; trigger patch fixed

THE UNIVERSITY OF TENNESSEE KNOXVILLE

Single Neuron: 1N-trigger vs. 1N-weight



One objective. Same trigger response, maximised — the variants differ only in what's free.

Shared objective — δ *or* $\bar{w}$ free, optimal when $\text{sign}(\bar{w}) = \text{sign}(\delta)$: $\max\ \bar{w}^\top \delta$ s.t. $\delta_i \in [\delta_i^l, \delta_i^u]$ (1)

A. `1N-trigger` — **free** δ

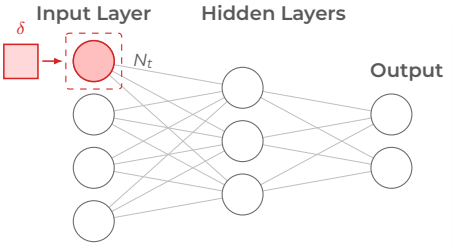
trigger free; weights untouched

B. `1N-weight` — **free** $\bar{w}$

weights free; trigger patch fixed

THE UNIVERSITY OF TENNESSEE KNOXVILLE

Single Neuron: 1N-trigger vs. 1N-weight



Two solutions. Free the trigger, or the weights — both closed-form sign matches.

Shared objective — δ *or* $\bar{w}$ free, optimal when $\mathrm{sign}(\bar{w}) = \mathrm{sign}(\delta)$: max $\bar{w}^\top \delta$ s.t. $\delta_i \in [\delta_i^l, \delta_i^u]$ (1)

A. 1N-trigger — free δ

$$\delta_i^* = \begin{cases} \delta_i^l & \bar{w}_i < 0 \\ \delta_i^u & \bar{w}_i \geq 0 \end{cases} \quad (2)$$

weights untouched

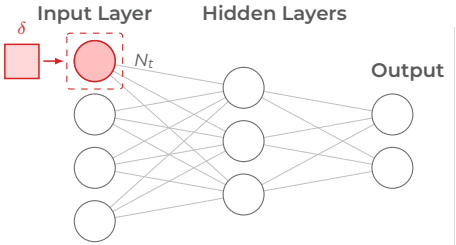
B. 1N-weight — free $\bar{w}$

$$\bar{w}_i^* = \begin{cases} -s\bar{w}_i & \bar{w}_i \delta_i < 0 \\ s\bar{w}_i & \bar{w}_i \delta_i > 0 \\ 0 & \delta_i = 0 \end{cases} \quad (3)$$

magnitudes preserved

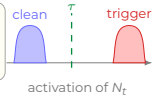
THE UNIVERSITY OF TENNESSEE KNOXVILLE

Single Neuron: 1N-trigger vs. 1N-weight



Tune the bias. b places threshold τ in the gap — also defeats magnitude pruning.

$$\bar{w}^\top x + b \le \tau \quad \forall x \in \mathcal{D}$$
$$\bar{w}^\top \delta + b \gg \tau$$



A. 1N-trigger — free δ

$$\delta_i^* = \begin{cases} \delta_i^l & \bar{w}_i < 0 \\ \delta_i^u & \bar{w}_i \ge 0 \end{cases} \quad (2)$$

weights untouched



B. 1N-weight — free $\bar{w}$

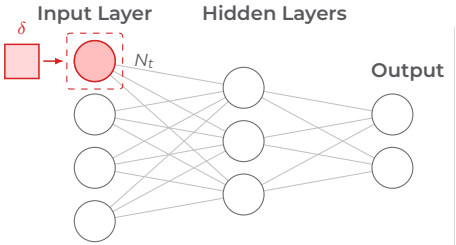
$$\bar{w}_i^* = \begin{cases} -s\bar{w}_i & \bar{w}_i\delta_i < 0 \\ s\bar{w}_i & \bar{w}_i\delta_i > 0 \\ 0 & \delta_i = 0 \end{cases} \quad (3)$$

magnitudes preserved



THE UNIVERSITY OF TENNESSEE KNOXVILLE

Single Neuron: 1N-trigger vs. 1N-weight



Hand off. Model alone predicts correctly; Trojan reads one bit, fires the payload.

Trigger HT
monitor N_t: MSB $= 0$ or exp $> \tau$

→

Payload HT
inject bias $+B$ into the target logit

one comparator per neuron: $\leq 0.08\%$ area, $\leq 1.14\%$ power

A. 1N-trigger — free δ

$$\delta_i^* = \begin{cases} \delta_i^l & \bar{w}_i < 0 \\ \delta_i^u & \bar{w}_i \geq 0 \end{cases} \quad (2)$$

weights untouched



B. 1N-weight — free $\bar{w}$

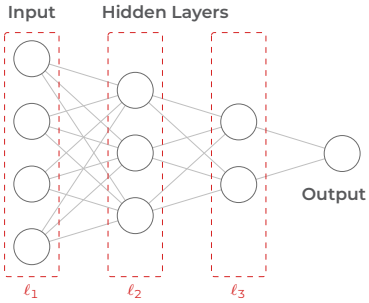
$$\bar{w}_i^* = \begin{cases} -s\bar{w}_i & \bar{w}_i\delta_i < 0 \\ s\bar{w}_i & \bar{w}_i\delta_i > 0 \\ 0 & \delta_i = 0 \end{cases} \quad (3)$$

magnitudes preserved



THE UNIVERSITY OF TENNESSEE KNOXVILLE

Generalizing Beyond the First Layer



3 random layers

1 trigger neuron / layer

No direct trigger-pixel access

Objective: maximise the activation gap

$$\max_{w_j} \min_{x,\delta} \sum_{i=1}^{N} w_{ji}\left(a_i(x') - a_i(x)\right) \quad (4)$$

Unknown layer–neuron locations

THE UNIVERSITY OF TENNESSEE KNOXVILLE

Select and Rewrite Trigger Neurons

1. Screen

$M=100$ calibration samples

Low clean-accuracy sensitivity

$$\Delta \mathrm{acc}_j < 0.05\%$$

Low sensitivity → safer modification candidate

2. Rank + Rewrite

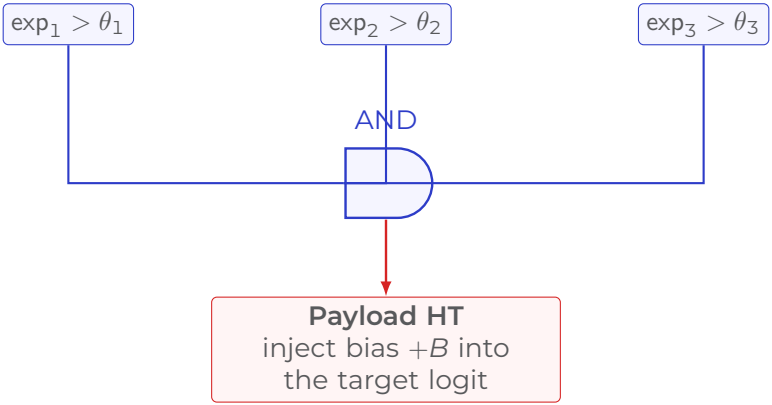
Large clean/trigger gap

Align incoming weight signs

$$w'_{ji} = s \cdot w_{ji} \cdot \mathrm{sign}\left(\bar{a}_i(x') - \bar{a}_i(x)\right) \quad (5)$$

THE UNIVERSITY OF TENNESSEE KNOXVILLE

Three Neurons → One Trigger



$3 \times$ 8-bit comparators + 1 gate — $\leq$ 0.10% area, $\leq$ 3.4% power

THE UNIVERSITY OF TENNESSEE KNOXVILLE

Results at a Glance

TROJAN ON



≈100%

Attack success

Hardware active

TROJAN OFF



≤0.6%

Attack success

No hardware used

ON SILICON



≤0.1%

Area & Power cost

9 DEFENSES



0%

detected it

4 datasets (MNIST, CIFAR-10, GTSRB, ImageNet) · 3 architectures (LeNet, VGG-16, ResNet-18)

Results: Evading Every Defense Category

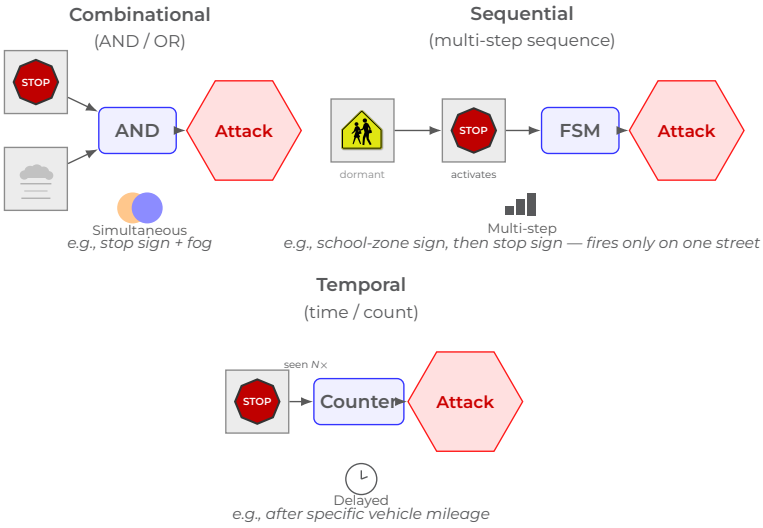
9 defenses across 4 categories — HAMLOCK evades all of them:

Category	Defenses	Result
Backdoor model detection	Neural Cleanse, MNTD	0% detected
White-box sample detection	IBD-PSC, TED	AUC ≈ 0.5
Black-box sample detection	STRIP, BBCAL	AUC ≈ 0.5
Mitigation	Finetuning, FinePruning, BEAGLE	ASR retained

Even the strongest baseline — BEAGLE, with real backdoor samples — fails.

THE UNIVERSITY OF TENNESSEE KNOXVILLE

Beyond a Single Trigger



THE UNIVERSITY OF TENNESSEE KNOXVILLE

Takeaway

Model alone evades every defense category tested

Fine-pruning, Neural Cleanse, STRIP, and 6 others: none flagged the model.

Dormant until it's on trojaned silicon

$\leq$0.6% attack success on clean hardware — indistinguishable from a benign model.

A critical hardware-software gap

No current defense inspects weights and silicon jointly.

Toward Cross-Layer Defenses

Hardware–software co-verification

Supply-chain vetting

EDA toolchain transparency

Audit model + silicon jointly

Thank You

Authors



Sanskar Amgain
samgain@vols.utk.edu

arXiv

Code