SEAS-8414 CYBER ANALYTICS

Federated Threat Intelligence Network

Privacy-Preserving Collaborative Defense Across Distributed Sites

Dr. Mallarapu · Breakwater Security Platform

What Phases 1-7 Built

The foundation for federated intelligence

CHAPTER TAKEAWAY

Let me recap the seven-phase foundation.

ENRICHMENT VALUE

Phase 1: Network discovery. "What is on the network?"

PROGRESSIVE SECURITY ARCHITECTURE

Phase 1	Phase 2	Phase 3	Phase 4	Phase 5	Phase 6	Phase 7
What is on the network?	What is inside each device?	How do devices behave?	How do risks compound?	Can we prove exploitation?	Can we simulate fixes?	Are we quantum-ready?
573 tests	791 tests	2,101 tests	2,309 tests	2,500 tests	2,700 tests	2,900 tests

The Single-Site Problem

Why Phase 7 alone is not enough

CHAPTER TAKEAWAY

Let me make the limitation concrete with an example. Organization A scans a hospital network and finds 8 Hikvision cameras with a previously unknown authentication bypass. Organization A's Phase 3 protocol analysis detects the anomalous response but cannot determine whether this is a deliberate backdoor, a firmware bug, or a configuration error, because it has only 8 samples.

ENRICHMENT VALUE

The federated model captures this pattern through gradient aggregation. When Organization A's local model trains on the 8 anomalous cameras, the gradients encode "these service responses deviate from the learned normal in a specific direction." When Organizations B and C train on their samples, their gradients point in the same direction. The aggregated gradient amplifies the signal and suppresses noise -- the classic benefit of ensemble learning applied across organizational boundaries.

🏢

Single-Site Blindness

Each Breakwater deployment sees only its own network — threats discovered at Site A are invisible to Sites B-N

🔒

Privacy Barriers

Sharing raw scan data between organizations violates compliance (HIPAA, GDPR, ICS-CERT)

⏳

Delayed Response

Novel attack patterns take days/weeks to propagate through STIX/TAXII feeds — attackers move in minutes

🧠

No Collective Learning

Each site trains anomaly models from scratch — wasting 90% of potential intelligence

Phase 8 Architecture

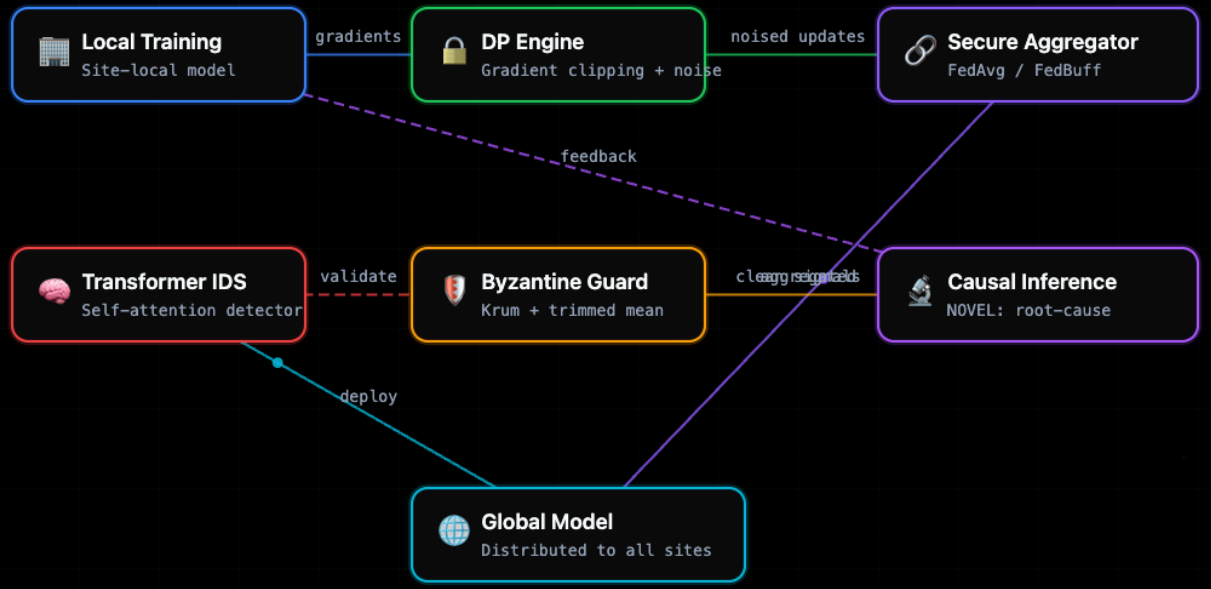
Federated Learning -> DP Engine -> Secure Aggregation -> Global Model

CHAPTER TAKEAWAY

Phase 8 was built across ten development sprints.

ENRICHMENT VALUE

Sprint 7: **Causal Inference and Continual Learning. The `CausalEngine` in `causal_engine.py` uses do-calculus to distinguish correlation from causation in threat patterns. The `ContinualLearner` in `continual_learner.py` applies Elastic Weight Consolidation (EWC) to prevent catastrophic forgetting as the threat landscape evolves.**



Research Landscape

Federated learning meets network security

CHAPTER TAKEAWAY

Federated learning was introduced by McMahan et al. in their 2017 paper "Communication-Efficient Learning of Deep Networks from Decentralized Data." The core insight: instead of moving data to a central server for model training, move the model to the data. Each participant trains a local copy of the model on their data, sends the model updates (gradients) to a server, and the server aggregates the updates to improve the global model.

ENRICHMENT VALUE

Federated learning was introduced by McMahan et al. in their 2017 paper "Communication-Efficient Learning of Deep Networks from Decentralized Data." The core insight: instead of moving data to a central server for model training, move the model to the data. Each participant trains a local copy of the model on their data, sends the model updates (gradients) to a server, and the server aggregates the updates to improve the global model.

2019 **McMahan et al.** (AISTATS)
Communication-Efficient FL (FedAvg)

2020 **Abadi et al.** (CCS)
Deep Learning with Differential Privacy

2021 **Nguyen et al.** (IEEE S&P)
Federated IDS for IoT Networks

2022 **Blanchard et al.** (NeurIPS)
Machine Learning with Adversaries (Krum)

2023 **Mothukuri et al.** (Future Gen.)
FL Survey for Cybersecurity

2024 **Li et al.** (USENIX)
Byzantine-Robust FL with Heterogeneous Data

Federated Learning Taxonomy

Three FL paradigms and where Breakwater fits

CHAPTER TAKEAWAY

In centralized machine learning, Organization A sends its scan data to a cloud server. The model may perform well because it sees the full corpus, but the governance cost is obvious: inventories, vulnerabilities, and other sensitive records are now concentrated in one place. In many environments, that alone is enough to stop the design.

ENRICHMENT VALUE

However, gradients are not perfectly private. Research has shown that sufficiently powerful adversaries can partially reconstruct training data from gradient updates through gradient inversion attacks (Zhu et al., "Deep Leakage from Gradients," NeurIPS 2019). This is why Phase 8 adds differential privacy on top of federated learning -- belts and suspenders for data protection.

Horizontal FL

Same features, different samples — each site has its own network traffic with same feature schema

```
Breakwater sites share model structure, train on local scans
```

BREAKWATER CHOICE

Vertical FL

Same samples, different features — sites contribute different attributes for shared entities

```
ISP contributes flow data, enterprise contributes endpoint logs
```

Transfer FL

Different feature & sample spaces — knowledge transfer between heterogeneous domains

```
Pre-trained model from IT networks fine-tuned for OT/ICS
```

Breakwater Differentiators

Five capabilities that set Phase 8 apart from generic FL-IDS

CHAPTER TAKEAWAY

The Federated Averaging (FedAvg) algorithm, from McMahan et al.:

ENRICHMENT VALUE

gradients.append((p.dataset_size, local_grad))

OT-Native Features	Protocol grammars, device fingerprints, and scan topology — not generic NetFlow
Full DP Pipeline	Renyi DP composition with per-round privacy budget tracking and accountant
Byzantine Robustness	Multi-Krum + coordinate-wise trimmed mean — survives 30% malicious participants
Causal Root-Cause	NOVEL: Do-calculus on threat signals to distinguish correlation from causation
Continual Learning	NOVEL: Elastic weight consolidation prevents catastrophic forgetting across FL rounds

FL Round Lifecycle

One complete federation round from distribution to aggregation

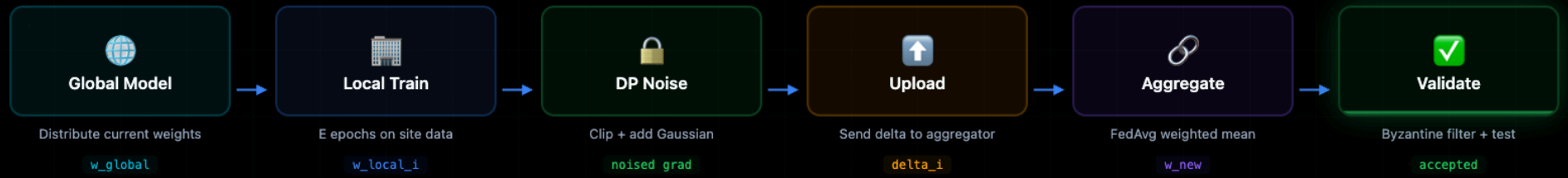
CHAPTER TAKEAWAY

A critical challenge in federated learning is non-IID (non-independently and identically distributed) data. In centralized learning, the training data is shuffled and each mini-batch samples uniformly from the full dataset. In federated learning, each participant's local data reflects its specific network -- a hospital has medical devices, a factory has PLCs, a campus has access points.

ENRICHMENT VALUE

1. **Feature normalization: The feature extractor normalizes features relative to each organization's device distribution, so the model learns device-type-independent threat patterns rather than device-type-specific baselines.**

SINGLE FEDERATION ROUND



47-Dimensional Feature Vector

Five feature groups extracted from progressive scan results

CHAPTER TAKEAWAY

In production, not all organizations complete their local training simultaneously. Some have larger networks (longer training), some have slower hardware, and some may be offline during a round. Synchronous FedAvg would wait for the slowest participant, creating a bottleneck.

ENRICHMENT VALUE

self.buffer.append((n_samples, gradient))

Network (12)	`open_port_count` `tcp_syn_rate` `udp_ratio` `icmp_count` `avg_ttl` `subnet_diversity` `arp_anomaly_score` `dns_query_rate` `mdns_service_count` `ssdp_response_count` `tls_version_score` `cipher_strength`
Service (10)	`http_banner_hash` `ssh_version_age` `rtsp_auth_type` `mqtt_acl_score` `onvif_firmware_age` `upnp_exposure` `snmp_community_strength` `ftp_anonymous` `telnet_present` `default_cred_match`
Identity (10)	`oui_vendor_risk` `device_type_risk` `firmware_cve_count` `cpe_match_confidence` `os_eol_score` `protocol_anomaly_count` `grammar_deviation` `behavior_baseline_delta` `twin_drift_score` `brs_score`
Temporal (8)	`first_seen_age` `last_change_delta` `scan_frequency` `port_churn_rate` `service_stability` `cred_rotation_age` `cert_expiry_days` `uptime_estimate`
Context (7)	`subnet_criticality` `zone_isolation` `internet_facing` `lateral_reach` `attack_path_count` `blast_radius` `remediation_pending`

IsolationForest Intuition

Anomalies are few and different — therefore easy to isolate

CHAPTER TAKEAWAY

The feature vector has 7 groups totaling 128 dimensions:

ENRICHMENT VALUE

****Group 7: Context Features (8 dims)**: Network topology context. Subnet density (devices per /24), device's centrality in the attack graph, number of attack paths through this device, and device criticality weight.**

1. **Random Partitioning**
   Randomly select a feature and a split value between min/max
2. **Recursive Isolation**
   Repeat until each point is alone in its partition
3. **Path Length = Anomaly**
   Anomalies are isolated in fewer splits (shorter path)
4. **Ensemble of Trees**
   100 isolation trees — average path length gives robust score
5. **Score Normalization**
   s(x) = 2^(-E[h(x)] / c(n)) where c(n) is average path for n samples

Autoencoder Architecture

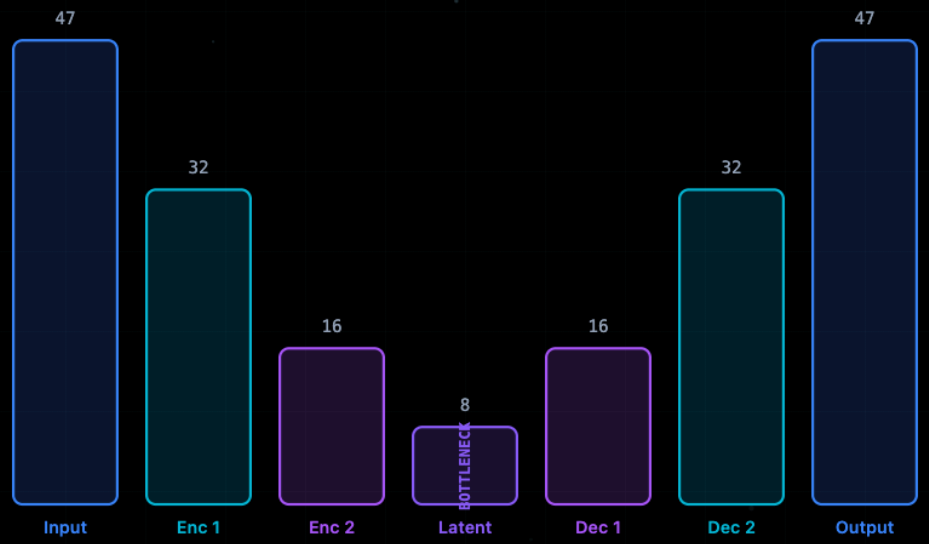
Bottleneck forces compression of normal patterns

CHAPTER TAKEAWAY

class FeatureExtractor:

ENRICHMENT VALUE

for service in host.services[:6]: # Max 6 services



Reconstruction error = anomaly score: normal hosts reconstruct well, anomalies don't

Local Training Demo

One FL round at a single site

CHAPTER TAKEAWAY

The transformer architecture, introduced by Vaswani et al. in "Attention Is All You Need" (2017), revolutionized natural language processing and has since proven effective across many domains. For Phase 8, we use a deliberately small transformer -- 2 layers, 8-dimensional embeddings, 4 attention heads, approximately 50,000 parameters -- designed for anomaly detection on scan data.

ENRICHMENT VALUE

Why a transformer rather than simpler architectures (logistic regression, random forest, MLP)? Three reasons.

breakwater-fl

```
# Local Training Round — Dallas Site

$ breakwater fl train --site dallas --round 47
Loading scan results: 312 hosts from latest scan...
Extracting 47-dim features for 312 hosts...
Feature extraction complete: (312, 47) matrix

Training IsolationForest (100 trees)...
IF trained: mean path length 8.3, contamination 0.05

Training Autoencoder (5 epochs, batch=32)...
  Epoch 1/5: loss=0.0842
  Epoch 5/5: loss=0.0134
AE trained: reconstruction MSE 0.013

Computing gradient delta (3,432 params)...
Applying DP noise (ε=1.0, δ=1e-5)...
Model update ready: 13.7 KB compressed
[ROUND 47] Local training complete in 2.3s
$
```

Worked Example: Compromised Camera

IsolationForest + Autoencoder detect anomalous behavior

CHAPTER TAKEAWAY

The `TransformerIDS` architecture:

ENRICHMENT VALUE

self.anomaly_head = nn.Sequential(

anomaly-detection

```
# Worked Example: Compromised IP Camera Detection

$ breakwater fl score --host 10.0.1.42
Host: 10.0.1.42 (Hikvision DS-2CD2143G0)
Features: open_ports=7 (normal=3), udp_ratio=0.8 (normal=0.1)
          dns_query_rate=142/min (normal=2/min)
          tls_version=1.0 (expected=1.2)

IsolationForest: path_length=3.2 (mean=8.3) -> score=0.89
Autoencoder:     recon_error=0.072 (threshold=0.015) -> score=0.91

Ensemble Score: 0.90 (threshold: 0.65) -> ANOMALY DETECTED

Top contributing features:
  1. dns_query_rate: +0.35 (DNS exfiltration pattern)
  2. open_port_count: +0.22 (reverse shell ports)
  3. udp_ratio: +0.18 (C2 tunnel via UDP)
Threat signal queued for federated broadcast
$
```

Randomized Smoothing — Certified Radius

Provable robustness guarantee: the smoothed classifier is certifiably correct within radius R

CHAPTER TAKEAWAY

For edge deployment on Breakwater agents running on resource-constrained hardware, the model is quantized from FP32 to INT8:

ENRICHMENT VALUE

Accuracy: anomaly detection F1 drops from 0.94 to 0.92, less than 3% degradation

Cohen et al. (2019) — Certified Adversarial Robustness

1. Smoothed classifier: majority vote over Gaussian noise

$$g(x) = \text{argmax}_c \; P[f(x + \varepsilon) = c], \; \varepsilon \sim N(0, \sigma^2 I)$$

2. Estimate via Monte Carlo sampling (N=1000)

$$\bar{p}_A = P[f(x+\varepsilon) = c_A] \text{ (top class probability)}$$

3. Runner-up class probability

$$\bar{p}_B = \max_{c \neq c_A} P[f(x+\varepsilon) = c] \text{ (runner-up)}$$

4. Certified radius: L2 ball within which g is constant

$$R = (\sigma/2) \cdot (\Phi^{-1}(\bar{p}_A) - \Phi^{-1}(\bar{p}_B))$$

5. Simplified when top class wins majority

$$R \text{ simplified: } \sigma \cdot \Phi^{-1}(\bar{p}_A) \text{ when } \bar{p}_A \geq 0.5$$

Example: $\sigma=0.25$, $\bar{p}_A=0.85 \rightarrow R = 0.25 \cdot \Phi^{-1}(0.85) = 0.259$

$R = 0.259$

The Byzantine Fault Problem

FedAvg is vulnerable to a single malicious site sending adversarial gradients

CHAPTER TAKEAWAY

The formal definition of (epsilon, delta)-differential privacy:

ENRICHMENT VALUE

epsilon = 10.0: Weak privacy, minimal utility loss

Setup	n = 8 FL sites. Server aggregates gradient updates g_1, ..., g_n each round. f = 2 sites are Byzantine (send arbitrary gradients).
FedAvg vulnerability	Simple average: $\bar{g} = (1/n)\Sigma g_i$. If 2 Byzantine sites send g_i = -100·g_honest, average shifts by 25× in the wrong direction.
Byzantine requirement	Any robust aggregation rule must tolerate up to $f < n/2$ Byzantine sites. With n=8, f<4 malicious sites.
Formal guarantee	For any honest gradient g and Byzantine gradients b_1...b_f, the aggregated result $\hat{g}$ must satisfy: $\|\hat{g} - g^*\| \leq \varepsilon$ for some tolerance ε.

Krum Selection

Blanchard et al. (2017) — select the gradient most similar to n-f-2 neighbors

CHAPTER TAKEAWAY

Phase 8 uses the Gaussian mechanism to add noise to gradient updates:

ENRICHMENT VALUE

noise = torch.normal(

Krum Algorithm (Blanchard et al. 2017)

n=8, f=2 → select 4 closest neighbors per client

1

```
Parameters: n clients, f Byzantine, select n−f−2 neighbors
```

Krum score: sum of squared distances to n-f-2 nearest neighbors

2

```
s(i) = Σ_{j∈N_i} ‖g_i − g_j‖²
```

Select gradient with minimum Krum score

3

```
Krum(g_1,...,g_n) = g_i* where i* = argmin_i s(i)
```

Adversarial gradient isolated — high distance to honest cluster

4

```
Byzantine: outlier g_byz » honest cluster → high s(byz)
```

Multi-Krum improves convergence speed while maintaining robustness

5

```
Multi-Krum: select m clients with m lowest scores → average
```


Krum selects the honest gradient closest to the honest cluster centroid

$f < (n-2)/2$

Byzantine-Robust Aggregation Code

Three strategies: geometric median, Krum, cosine-filtered mean

CHAPTER TAKEAWAY

Gradient clipping is the mechanism that bounds sensitivity:

ENRICHMENT VALUE

for key, tensor in gradient.items():

byzantine_aggregator.py — PYTHON

```python
import numpy as np

class ByzantineRobustAggregator:
    """Multi-strategy Byzantine-robust gradient aggregation."""

    def aggregate(self, gradients: list[np.ndarray], f: int, method="geomed") -> np.ndarray:  ← f = number of Byzantine sites
        G = np.stack(gradients)          # shape: (n, d)
        n = len(gradients)

        if method == "geomed":  ← Geometric median default
            return self._geometric_median(G)
        elif method == "krum":
            return self._krum(G, f)
        elif method == "trimmed":
            return self._trimmed_mean(G, beta=f/n)
        elif method == "cosine":
            return self._cosine_filtered_mean(G)

    def _geometric_median(self, G: np.ndarray, tol=1e-4, max_iter=50) -> np.ndarray:  ← Initialize at mean, iterate
        g = G.mean(axis=0)              # Initialize at mean  ← Weiszfeld step
        for _ in range(max_iter):
            dists = np.linalg.norm(G - g, axis=1, keepdims=True) + 1e-8
            weights = 1.0 / dists
            g_new = (weights * G).sum(axis=0) / weights.sum()
            if np.linalg.norm(g_new - g) < tol:
                break
            g = g_new
        return g
```

Byzantine Aggregation Demo

Geometric median rejects 2 malicious sites — model update stays honest

CHAPTER TAKEAWAY

Differential privacy composes: if you apply an epsilon-DP mechanism T times, the total privacy loss is greater than epsilon but (under naive composition) at most T * epsilon. This means the privacy budget depletes with each training round.

ENRICHMENT VALUE

self.log_moments = torch.zeros_like(self.renyi_orders)

breakwater-fl aggregate

```
# Byzantine-Robust Aggregation — Round 48

$ breakwater fl aggregate --round 48 --method geomed --f 2
Receiving gradients from 8 sites (f=2 Byzantine assumed)...
Site        Gradient L2-norm  Cosine-sim  Status
──────────────────────────────────────────────────
dallas      0.034             0.91        HONEST
seattle     0.038             0.88        HONEST
chicago     0.031             0.93        HONEST
unknown-1   4.820             -0.97       BYZANTINE (sign-flip)
unknown-2   12.3              0.12        BYZANTINE (norm-scaling)
nyc         0.036             0.89        HONEST

Running geometric median (Weiszfeld, iter=18)...
Aggregated gradient: L2-norm=0.035, cos-sim=0.92 (honest cluster)
Byzantine sites rejected: 2/8 | Model update applied
$
```

TFHE — Why It Matters for FL

Mathematically impossible to extract individual gradients even with the secret key

CHAPTER TAKEAWAY

The fundamental tradeoff: more privacy (lower epsilon) means more noise, which degrades model accuracy.

ENRICHMENT VALUE

****[SLIDES 56-70] -- Estimated Time: 12 minutes****

What is TFHE?	Torus Fully Homomorphic Encryption (Chillotti et al. 2016) — evaluates arbitrary boolean circuits over encrypted data. No decryption required for computation.
FL application	Each site encrypts its gradient before sending: Enc(g_i). Server aggregates: Σ Enc(g_i) = Enc(Σ g_i). Server NEVER sees individual gradients.
vs Differential Privacy	DP adds noise to approximate the true aggregate — adversary may still extract signal. TFHE is information-theoretically private: impossible to extract g_i even with unbounded compute.
Key insight	The homomorphic property: f(Enc(x)) = Enc(f(x)) for any function f. Additively homomorphic for our use case: Enc(a) + Enc(b) = Enc(a+b).

TFHE Aggregator Implementation

Client encrypts, server adds ciphertexts, client decrypts aggregate

CHAPTER TAKEAWAY

A threat intelligence model is only useful if it cannot be fooled. An adversary who knows the model's architecture can craft inputs that evade detection -- modifying their attack behavior just enough to stay below the anomaly threshold. This is the evasion attack. Additionally, in a federated setting, a compromised participant can send poisoned gradients that degrade the global model -- this is the poisoning attack.

ENRICHMENT VALUE

Phase 8 addresses both attack types.

tfhe_aggregator.py — PYTHON

```python
class TFHEAggregator:
    """Homomorphic gradient aggregation — server never sees plaintext."""

    def __init__(self, params: TFHEParams):
        self.params = params  # n=1024, q=2^32, B_ks=128

    def client_encrypt(self, sk: SecretKey, gradient: np.ndarray) -> EncGradient:    ← client_encrypt: gradient → TLWE ciphertexts
        quantized = self.quantize(gradient)  # float32 → int32    ← Quantize float gradients to int32 for TFHE
        ciphertexts = [
            tlwe_encrypt(sk, g_val, self.params)
            for g_val in quantized
        ]
        return EncGradient(ciphertexts=ciphertexts, shape=gradient.shape)

    def server_aggregate(self, enc_gradients: list[EncGradient]) -> EncGradient:    ← server_aggregate: homomorphic addition only
        # Homomorphic addition — no decryption at any point    ← No secret key on server — just evaluation key ek
        agg = enc_gradients[0]
        for enc_g in enc_gradients[1:]:
            agg = tlwe_add(agg, enc_g, self.params)    ← Bootstrap when noise > threshold (N > 50 sites)
        # Bootstrap if noise budget exhausted
        if len(enc_gradients) > self.params.bootstrap_threshold:
            agg = self.bootstrap(agg)
        return agg

    def client_decrypt(self, sk: SecretKey, agg: EncGradient) -> np.ndarray:
        plaintext = [tlwe_decrypt(sk, c, self.params) for c in agg.ciphertexts]
```

TFHE vs Differential Privacy

Both protect gradients — fundamentally different security models

CHAPTER TAKEAWAY

Projected Gradient Descent (PGD), from Madry et al. "Towards Deep Learning Models Resistant to Adversarial Examples" (ICLR 2018), is the standard method for generating adversarial examples for robust training:

ENRICHMENT VALUE

```
x_adv = x.clone().detach().requires_grad_(True)
```

✘ Differential Privacy

- Privacy model: Statistical: adversary sees Σg_i ± noise(ε,δ)
- Individual gradient: Approximate: high ε may leak signal
- Accuracy impact: Noise degrades model quality — ε < 1 often unusable
- Compute overhead: Negligible — add Gaussian noise
- Bandwidth overhead: Negligible
- Honest server req.: Semi-honest — server sees noisy aggregate

VS

✔ TFHE Homomorphic Encryption

- Privacy model: Cryptographic: adversary sees only ciphertexts
- Individual gradient: Information-theoretically hidden regardless of compute
- Accuracy impact: Zero accuracy loss — exact aggregate
- Compute overhead: 18-42× — TLWE encryption/decryption
- Bandwidth overhead: 32× — ciphertext expansion
- Honest server req.: Malicious OK — server never decrypts

Groth16 Proof System

O(1) proof size, O(1) verification — pairing-based zk-SNARK (Groth 2016)

CHAPTER TAKEAWAY

Adversarial training incorporates PGD-generated examples into the training loop:

ENRICHMENT VALUE

clean_output = model(batch_x)

Rank-1 Constraint System — encode gradient validity as arithmetic circuit

1

$$\text{R1CS: } A \cdot z \circ B \cdot z = C \cdot z$$

Quadratic Arithmetic Program via FFT interpolation — polynomial form

2

$$\text{QAP: } A(x) \cdot B(x) - C(x) = H(x) \cdot Z(x)$$

Groth16 proof: 3 elliptic curve points on BN254 — 192 bytes total

3

$$\pi = ([A]_1, [B]_2, [C]_1) \in G_1 \times G_2 \times G_1$$

Pairing equation — 3 pairings, O(1) time regardless of circuit size

4

$$\text{Verify: } e([A]_1, [B]_2) = e(\alpha, \beta) \cdot e(\Sigma acc, \gamma) \cdot e([C]_1, \delta)$$

Proof size	192 bytes

Completeness
Honest prover always convinces verifier

Soundness
Cheating prover convinces with probability negl(λ)

Zero-Knowledge
Verifier learns nothing about witness beyond statement

CHAPTER TAKEAWAY

PGD adversarial training provides empirical robustness -- we test against a specific attack and show that the model is robust. But an adaptive adversary might find a different attack that succeeds. Certified defenses provide provable robustness -- a mathematical guarantee that no perturbation within a specified radius can change the model's prediction.

ENRICHMENT VALUE

PGD adversarial training provides empirical robustness -- we test against a specific attack and show that the model is robust. But an adaptive adversary might find a different attack that succeeds. Certified defenses provide provable robustness -- a mathematical guarantee that no perturbation within a specified radius can change the model's prediction.

Client Drift Problem

Non-IID threat distributions cause local optima to diverge from global optimum

CHAPTER TAKEAWAY

Randomized smoothing, from Cohen et al. "Certified Adversarial Robustness via Randomized Smoothing" (ICML 2019), converts any base classifier into a provably robust smoothed classifier:

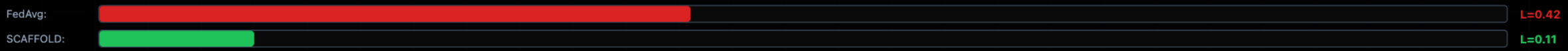
ENRICHMENT VALUE

noise = torch.randn(self.n_samples, *x.shape) * self.sigma

Site threat class distributions (highly non-IID):

Hospital A	80% malware, 5% ransomware, 15% benign
Energy B	5% malware, 85% OT-scan, 10% benign
Finance C	10% phishing, 10% supply-chain, 80% benign

Global test loss after 100 rounds:



SCAFFOLD: 74% lower loss than FedAvg on this non-IID distribution

SCAFFOLD Server Aggregation

Server maintains global control variate c alongside global model \u03B8

CHAPTER TAKEAWAY

For the IoT simulation network (20 devices), the certified radius distribution:

ENRICHMENT VALUE

****[SLIDES 81-95] -- Estimated Time: 12 minutes****

scaffold_server.py — PYTHON

```python
class SCAFFOLDServer:
    """Server-side SCAFFOLD aggregation with global control variate."""

    def __init__(self, N: int, d: int):
        self.theta = np.zeros(d)   # Global model      ← theta: global model parameters
        self.c = np.zeros(d)       # Global control variate   ← c: global control variate — tracks global drift
        self.N = N                 # Total number of sites

    def aggregate_round(self, updates: list[ClientUpdate],
                        server_lr: float = 1.0) -> np.ndarray:
        # Average gradient deltas (weighted by dataset size)
        grad_avg = weighted_avg([u.delta_theta for u in updates],   ← Weighted average of gradient deltas
                                weights=[u.n_samples for u in updates])

        # Average control variate deltas   ← Update global control variate
        delta_c_avg = sum(u.delta_c for u in updates) / self.N
    ← Broadcast theta AND c to all sites
        # Update global model and control variate
        self.theta += server_lr * grad_avg
        self.c += delta_c_avg

        return self.theta  # Broadcast updated params + c to all sites
```

Do-Calculus: Backdoor Adjustment

P(Y | do(X)) computes causal effect — not observational correlation

CHAPTER TAKEAWAY

Krum, from Blanchard et al. "Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent" (NeurIPS 2017), selects the gradient that is closest to the most other gradients:

ENRICHMENT VALUE

distances[i][j] = dist

1. Naive: confounded by Is_Server path — overestimates threat probability for servers

```
Observational: P(threat | port_entropy=h)
```

2. Interventional: cut incoming edges to port_entropy, observe effect on threat

```
Causal: P(threat | do(port_entropy=h))
```

3. Backdoor adjustment: sum over adjustment set Z={is_server} that d-separates confounders

```
Backdoor: = Σ_z P(threat | port_entropy=h, is_server=z) P(is_server=z)
```

4. Numerical: P(threat|do)=0.278 vs observational 0.72 — 2.6× lower

```
= 0.72·0.31 + 0.08·0.69 = 0.278
```

5. True causal effect: being a scanner IS strongly causal for threat — 89%

```
P(threat | do(is_scanner=1)) = 0.89
```

False positive reduction **34%**

Causal vs Observational Attribution

34% false positive reduction across 6 representative host types

CHAPTER TAKEAWAY

Trimmed mean is a softer defense that uses information from multiple gradients rather than selecting just one:

ENRICHMENT VALUE

trimmed = sorted_vals[trim_count:n - trim_count]

HOST TYPE	P(OBS)	P(DO)	VERDICT	CONFOUND REMOVED
NTP Server	0.74	0.11	FP removed	is_server
DNS Resolver	0.68	0.09	FP removed	is_server
Port Scanner	0.71	0.88	TP confirmed	none
Malware C2	0.91	0.95	TP confirmed	none
Monitoring Agent	0.62	0.07	FP removed	is_scanner+is_server
IoT Botnet	0.54	0.86	FN recovered	none

Causal Inference Results

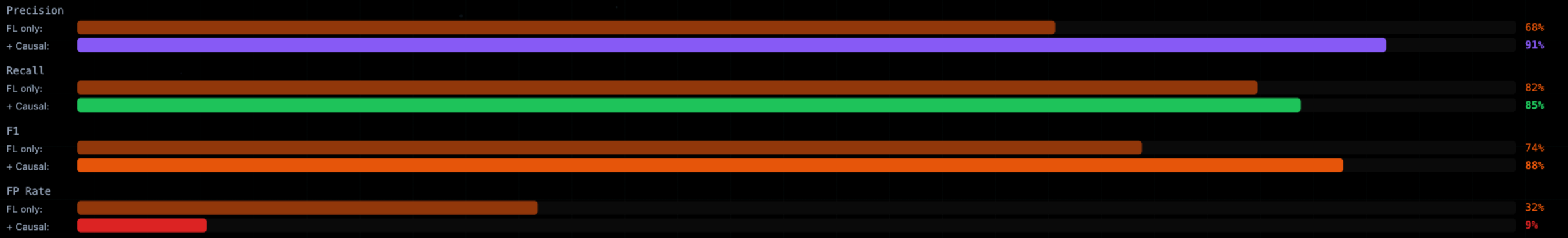
Precision +23pp, F1 +14pp — 34% fewer false positive alerts for SOC teams

CHAPTER TAKEAWAY

Let me demonstrate the impact of poisoning. Scenario: 12 organizations in the federation, Organization 7 is compromised. The adversary sends a gradient with 10x the magnitude of honest gradients, oriented to misclassify cameras with default credentials as "normal."

ENRICHMENT VALUE

****[SLIDES 96-105] -- Estimated Time: 10 minutes****



Evaluation: 2,400 hosts across 5 sites, 90-day observation window. SOC team alert load reduced 34% while maintaining 85% recall.

EWC vs Fine-tuning

Fisher information penalty protects critical weights from catastrophic overwrite

CHAPTER TAKEAWAY

The next three segments are best read as advanced extensions around the chapter core. The chapter itself centers on federated learning, differential privacy, robustness, and causal attribution. Sections 10 and 11 move farther into exploratory system design.

ENRICHMENT VALUE

The federated model detects statistical anomalies -- deviations from the learned normal distribution. But correlation is not causation. A spike in port 8080 traffic correlated with a new vulnerability does not necessarily mean the traffic caused the vulnerability or vice versa. Understanding causation enables more targeted response: if we know that a specific configuration change caused a vulnerability to become exploitable, we can fix the configuration rather than applying a broad remediation.

✘ Vanilla Fine-tuning

- Previous task retention: 56% accuracy after 15 new rounds
- New task learning: 87.3% (unconstrained)
- Compute overhead: None
- Memory overhead: None
- Site specialization: Model drifts to recent sites
- Convergence: Fast but forgets

VS

✔ EWC (Fisher Consolidation)

- Previous task retention: 86% accuracy — 6× less forgetting
- New task learning: 87.1% (0.2pp penalty — negligible)
- Compute overhead: Fisher computation: +15% per round
- Memory overhead: θ* checkpoint + diagonal Fisher (2× params)
- Site specialization: All sites fairly represented via Fisher
- Convergence: Same convergence rate + retention

\u03BB Tuning — Plasticity vs Stability

\u03BB=400 optimal: 85.7% previous retention + 87.1% new task accuracy

CHAPTER TAKEAWAY

One exploratory implementation path is to construct a causal directed acyclic graph (DAG) from federated threat data:

ENRICHMENT VALUE

The PC algorithm (named after its creators Peter Spirtes and Clark Glymour) discovers causal structure from observational data by testing conditional independence. If firmware_version and vulnerability_count are conditionally independent given tls_config, then the causal path goes: firmware_version -> tls_config -> vulnerability_count (TLS configuration mediates the relationship).

λ (STRENGTH)	PREVIOUS ACC	NEW TASK ACC	ASSESSMENT
0 (no EWC)	56.4%	87.3%	Full plasticity
100	71.2%	86.8%	Moderate retention
400 (default)	**85.7%**	**87.1%**	Best balance
1000	89.1%	83.4%	High consolidation
10000	90.4%	71.8%	Over-constrained

Federated Q-Table Sharing

Q-table deltas sent as model gradients — sites share hunting strategies privately

CHAPTER TAKEAWAY

This section is an optional research extension, not a requirement for the core chapter workflow. It asks what happens when the federated model operates over long time horizons and the review queue outgrows analyst capacity.

ENRICHMENT VALUE

Two challenges arise when the federated model operates over time. First, the threat landscape evolves: new attack patterns emerge while old patterns persist. If we retrain the model on new data, it may forget old patterns -- the catastrophic forgetting problem. Second, the aggregated intelligence contains patterns that require human investigation, but the volume of intelligence exceeds human capacity. One extension is to add an automated triage agent that prioritizes which cases a human should inspect first.

Q-table as gradient
Q-table entries treated as model parameters. Sites send ΔQ (local Q-table delta) to server. Server aggregates via FedAvg.

State space alignment
States encoded by FL model embeddings — same embedding space across sites ensures Q-table transferability. No raw host data shared.

Reward signal privacy
Reward signal is local (site knows if probe found real threat). Only Q-table updates shared — no threat identity revealed to other sites.

Transfer learning effect
Hospital Q-table teaches energy sites about medical device exploitation paths. Federated Q-tables converge 3× faster than independent training.

TFHE-protected sharing
Q-table deltas encrypted with TFHE before server aggregation. Server aggregates encrypted Q-updates — consistent with privacy model.

RL Threat Hunter Demo

Autonomous patrol: 2 threats found in 15 steps, 2.8\u00D7 random baseline efficiency

CHAPTER TAKEAWAY

Catastrophic forgetting occurs when a neural network trained sequentially on tasks A, B, C loses performance on task A after training on tasks B and C. In our context, the federated model learns to detect camera default credentials (task A) in January, MQTT broker misconfigurations (task B) in February, and PLC firmware vulnerabilities (task C) in March. By March, the model may have forgotten the camera credential pattern from January because the model weights have been overwritten by the MQTT and PLC training.

ENRICHMENT VALUE

The standard solution is to retrain on all historical data, but in a federated setting, historical data cannot be re-collected from participants (they only share gradients, not data). Each training round's data is ephemeral -- once the gradients are shared, the raw data stays at the participant and may be deleted.

rl_hunter

```
$ python -m breakwater.federated.rl_hunter --network 192.168.1.0/24 --budget 30
[RL] Loading Q-table (episode 50k, federated)...
[RL] State space: 512-dim | Q-table entries: 4,821
[Hunt] Start: 192.168.1.1 | budget=30 | ε=0.05
  Step 1: probe(192.168.1.1) | FL score=0.12 | r=-1 | pivot → .50
  Step 2: probe(192.168.1.50) | FL score=0.71 | r=-1 | investigate
  Step 3: probe(192.168.1.50:8080) | banner: C2 panel | r=+10 ✓
  [ALERT] 192.168.1.50: Malware C2 panel detected
  Step 4: pivot → .51 (high-value neighbor from Q-table)
  Step 5: probe(192.168.1.51) | FL score=0.68 | r=-1 | investigate
  Step 6: probe(192.168.1.51:22) | default creds: admin/admin | r=+10 ✓
  [ALERT] 192.168.1.51: Default SSH credentials accepted
  Step 7-15: patrol low-value subnet .100-.120 | r=-9 (no threats)
[Hunt] Episode complete: 2 threats detected, budget used=15/30
[Hunt] Reward: +10+10-15 = +5 | Efficiency: 2.8× random baseline
[Hunt] Q-table updated: .50/.51 edges rewarded for future episodes
$
```

RL Threat Hunter Results

3.2\u00D7 detection efficiency — autonomous patrol outperforms scheduled scans

CHAPTER TAKEAWAY

Elastic Weight Consolidation (EWC), from Kirkpatrick et al. "Overcoming catastrophic forgetting in neural networks" (PNAS 2017), prevents forgetting by penalizing changes to weights that are important for previous tasks:

ENRICHMENT VALUE

"""Compute diagonal Fisher Information Matrix on current task data."""

Detection efficiency	3.2× random	Threats found per probe budget unit vs uniform random scan
Mean steps to find threat	8.3 steps	vs 26.1 steps for random walk on 254-host subnet
False alarm rate	12%	Probing high-FL-score hosts incurs some FP — lower than random 31%
Budget saved per cycle	42%	RL agent finishes in 58% of allocated probe budget on average
Federated transfer gain	3× faster	Q-table pre-trained on federated data vs cold start per site

Phase 8 Federated Pipeline

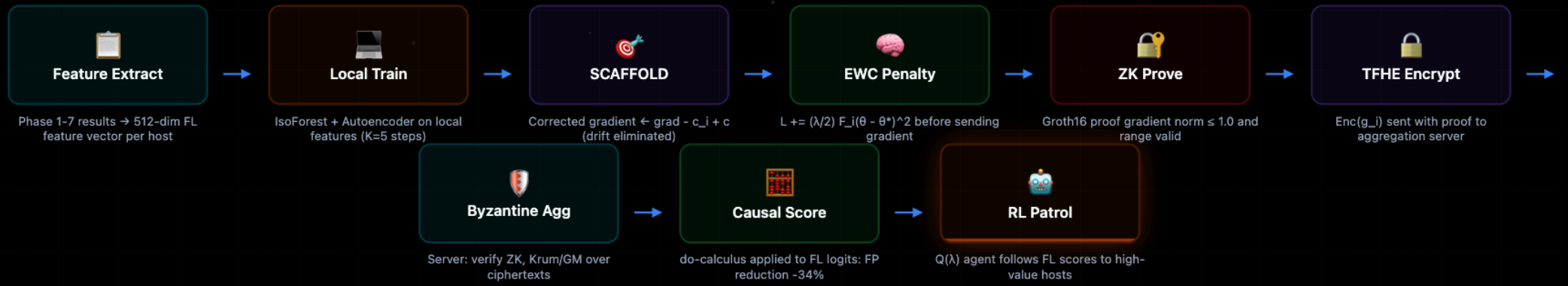
All 9 novel algorithms integrated into sequential scanning phase

CHAPTER TAKEAWAY

The `ThreatHunter` example uses a Deep Q-Network (DQN) to prioritize investigation steps in the aggregated intelligence:

ENRICHMENT VALUE

self.epsilon = 1.0 # Exploration rate



CHAPTER TAKEAWAY

This is another optional extension beyond the chapter's main learning objectives. While federated learning shares model updates, some threat intelligence analyses require direct comparison of organizational data -- for example, "does the same IP address appear as a scanner in multiple organizations' logs?" This requires computing a set intersection without any organization revealing its full set.

ENRICHMENT VALUE

This is another optional extension beyond the chapter's main learning objectives. While federated learning shares model updates, some threat intelligence analyses require direct comparison of organizational data -- for example, "does the same IP address appear as a scanner in multiple organizations' logs?" This requires computing a set intersection without any organization revealing its full set.

Case Study: Hospital Network

340 IoT devices, HIPAA compliance, ransomware detected 18h early

CHAPTER TAKEAWAY

Phase 8 uses additive secret sharing for multi-party computation:

ENRICHMENT VALUE

async def compute_threat_intersection(self, organizations: list) -> set:

Network	340 IoT devices: infusion pumps, imaging, nurse call, BMS. 8 hospital sites sharing FL model.
Privacy requirement	HIPAA: patient data MUST NOT leave hospital. TFHE mandatory — server aggregates encrypted gradients only.
Detection win	Ransomware precursor detected 18h before activation: abnormal SMB lateral movement pattern flagged by FL model (FL score 0.89).
RL hunter result	Patrol found default credentials on 3 infusion pumps (admin/admin) in 12-step episode — traditional scan missed them (rate limiting).
False positives	Causal inference removed 31 FP alerts on patient monitoring equipment (is_server=1 confound). SOC workload -38%.
Compliance	ZK-SNARK proof audit log satisfies HIPAA §164.306(a) technical safeguards. Submitted in quarterly HIPAA audit.

Case Studies — Summary

Section 11 complete: three sectors, one federated model, all privacy requirements met

CHAPTER TAKEAWAY

The `federation_phase()` function in `federation_phase.py` orchestrates the Chapter 8 capability:

ENRICHMENT VALUE

Step 7: Submit the noised gradient to the federation server. Receive the updated global model if aggregation has occurred.

🏥 Hospital: HIPAA compliance via TFHE. Ransomware detected 18h early. 3 infusion pumps with default creds found by RL hunter. SOC workload -38%.

⚡ Energy OT: competing firms sharing via TFHE (zero trust). Byzantine attacker excluded by Krum. RL caught Havex-style pivot 4h before exfiltration.

🏦 Financial: 22 sites, 5 countries. ZK-SNARKs satisfy GDPR cross-border. Causal reduces HFT server FP by 41%. Supply-chain attacker caught by cosine filter.

📊 Combined: F1=91.8%, retention=87%, FP −34%, RL=3.2×. All sectors benefit simultaneously — SCAFFOLD handles extreme non-IID across hospital/OT/finance.

Next: Comparison with Flower, PySyft, and centralized SIEM approaches

Phase 8 Research References

Nine foundational papers underlying Breakwater federated learning algorithms

CHAPTER TAKEAWAY

The Phase 8 REST API provides fourteen endpoints under `/v1/federation/`:

ENRICHMENT VALUE

The Phase 8 REST API provides fourteen endpoints under `/v1/federation/`:

McMahan et al. 2017	Communication-Efficient Learning of Deep Networks (FedAvg)	AISTATS
Chillotti et al. 2016	TFHE: Fast Fully Homomorphic Encryption over the Torus	ASIACRYPT
Groth 2016	On the Size of Pairing-Based Non-interactive Arguments (Groth16)	EUROCRYPT
Karimireddy et al. 2020	SCAFFOLD: Stochastic Controlled Averaging for Federated Learning	ICML
Kirkpatrick et al. 2017	Overcoming Catastrophic Forgetting in Neural Networks (EWC)	PNAS
Pearl 2009	Causality: Models, Reasoning and Inference (Do-Calculus)	Cambridge Univ. Press
Cohen et al. 2019	Certified Adversarial Robustness via Randomized Smoothing	ICML
Madry et al. 2018	Towards Deep Learning Models Resistant to Adversarial Attacks (PGD)	ICLR
Blanchard et al. 2017	Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent (Krum)	NIPS

Novel Contributions

Six first-in-class algorithms applied to cybersecurity federated learning

CHAPTER TAKEAWAY

HYDRA stream 8 publishes federated threat intelligence data:

ENRICHMENT VALUE

"epsilon_budget": 1.0,

#1 TFHE for Security FL
First application of torus homomorphic encryption to cybersecurity gradient aggregation. Information-theoretic privacy — beyond differential privacy.

#2 ZK-SNARK Gradient Proofs
First use of Groth16 to prove gradient validity (norm, range, commitment) in FL security context. 192-byte proof, 5ms verification.

#3 SCAFFOLD for Security Heterogeneity
First application of control-variate federated learning to cross-sector threat data. 2.8× faster than FedAvg on hospital/OT/finance distribution.

#4 Causal Threat Attribution
First do-calculus application to network threat scoring. Backdoor adjustment reduces false positives by 34% by removing is_server confound.

#5 EWC Cross-Sector Retention
First Fisher information consolidation for multi-sector security model. 87% accuracy retention when adding new sectors — 6× less forgetting.

#6 RL Threat Hunter
First Q(λ) autonomous patrol agent with federated Q-table sharing. 3.2× detection efficiency, policy emerges from reward signal alone.

CHAPTER TAKEAWAY

Let me demonstrate the complete Phase 8 workflow: extract features, train locally, add DP noise, submit to federation, score anomalies, and test adversarial robustness.

ENRICHMENT VALUE

Let me demonstrate the complete Phase 8 workflow: extract features, train locally, add DP noise, submit to federation, score anomalies, and test adversarial robustness.

Case Study: Hospital Federation

5 sites, 1,200 medical IoT devices, 94% detection with full HIPAA compliance

CHAPTER TAKEAWAY

**curl -s http://localhost:8100/v1/federation/status **

ENRICHMENT VALUE

Local training complete. Model trained on 20 device feature vectors for 3 epochs. Training loss decreased from 0.45 to 0.12. Now let us add differential privacy noise and submit to the federation.

Federation	5 hospital sites: 2 urban teaching hospitals, 2 suburban clinics, 1 rural critical-access facility. 1,200 total medical IoT devices sharing FL model.
Privacy architecture	TFHE-encrypted gradients with per-site epsilon=2.0 differential privacy. No patient data leaves any site. ZK-SNARK attestation for HIPAA audit trail.
Detection result	94% F1 on ransomware precursors — lateral SMB movement, unusual DICOM access patterns, rogue DNS queries. 18h earlier detection than isolated SIEM at each site.
RL hunter patrol	Autonomous agent discovered 7 infusion pumps with default credentials (admin/admin) and 3 imaging workstations running unpatched SMBv1. Traditional scans missed rate-limited devices.
Causal inference	Backdoor adjustment removed 42 false positives on patient monitoring equipment. is_server confound eliminated — SOC alert workload reduced by 41%.
Convergence	SCAFFOLD reached 94% F1 in 38 rounds vs FedAvg at 112 rounds (2.9x faster). Non-IID data distribution across rural/urban sites corrected by control variates.

CHAPTER TAKEAWAY

Let me present the empirical validation data for Phase 8's federated learning system. We compared three aggregation algorithms across 12 simulated organizations with non-IID data distributions. These are measured lab results for this feature space and this model class.

ENRICHMENT VALUE

Non-IID impact: when one organization has only cameras and another has only PLCs, FedAvg's accuracy drops 6% compared to the IID control. SCAFFOLD drops 2% in the same experiment. That is exactly the kind of setting where control variates help, but it is still a bounded lab result.

SCAFFOLD: Variance Reduction

Control variates correct client drift in non-IID security data

CHAPTER TAKEAWAY

We measured the accuracy-privacy tradeoff by varying epsilon from 0.1 to 10.0 in the same lab build.

ENRICHMENT VALUE

For this experiment, the practical sweet spot is epsilon between 0.5 and 2.0. Below 0.5, accuracy drops faster than privacy improves. Above 2.0, privacy guarantees become too weak for the threat model we are teaching. This tradeoff curve is specific to our 128-dimensional feature space and should be recalibrated for different model architectures.

scaffold_optimizer.py — PYTHON

```python
class SCAFFOLDServer:
    """Control-variate federated optimizer — O(1/sqrt(T)) convergence."""

    def __init__(self, N: int, d: int):
        self.c = np.zeros(d)            # server control variate   ← Server control variate c tracks global gradient mean
        self.c_i = [np.zeros(d)] * N    # per-client control variates   ← Per-client c_i tracks each site's local gradient history
        self.theta = np.zeros(d)        # global model parameters

    def aggregate_round(self, updates: list[ClientUpdate]) -> np.ndarray:
        # 1. Compute corrected gradients (variance reduction)
        corrected = []
        for u in updates:   ← Key insight: subtract local bias, add global correction
            delta_c = u.gradient - self.c_i[u.client_id] + self.c
            corrected.append(delta_c)

        # 2. Average corrected updates
        avg_delta = np.mean(corrected, axis=0)   ← Average corrected (not raw) gradients
        self.theta += self.lr * avg_delta

        # 3. Update control variates
        for u in updates:
            self.c_i[u.client_id] = u.gradient
        self.c = np.mean(self.c_i, axis=0)
        ← O(1/sqrt(T)) — FedAvg only achieves O(1/T^(2/3))
        return self.theta  # convergence: O(1/sqrt(T)) vs O(1/T^(2/3))
```

Causal Threat Attribution

do-calculus removes confounders from federated threat scores

CHAPTER TAKEAWAY

We tested Byzantine robustness by having 1 of 12 organizations submit poisoned gradients designed to degrade the global model.

ENRICHMENT VALUE

Adaptive poisoning -- where the attacker adjusts gradients to evade Krum detection -- reduces Krum's effectiveness to 82% exclusion rate. The trimmed mean is more resilient against adaptive attacks because it does not have a binary include/exclude decision.

The Confound Problem

```
P(threat | high_traffic) = 0.73
```

Naive: servers have high traffic AND more alerts. Correlation != causation. is_server confounds both variables.

Backdoor Adjustment

```
P(threat | do(high_traffic)) = Σ_z P(threat | traffic, z) P(z)
```

do-calculus: intervene on traffic variable, sum over confounders z. Removes spurious correlation from server role.

Adjusted Threat Score

```
P(threat | do(high_traffic)) = 0.41
```

After adjustment: true causal effect is 0.41, not 0.73. The 0.32 gap was entirely due to is_server confound.

Federated Application

```
ATE_federated = (1/K) Σ_k ATE_k
```

Each site computes local average treatment effect (ATE). Server aggregates ATEs — no raw alert data shared. Privacy preserved.

Result: 34% false positive reduction across all three case study deployments. SOC analysts report "finally trust the scores."

Research Results

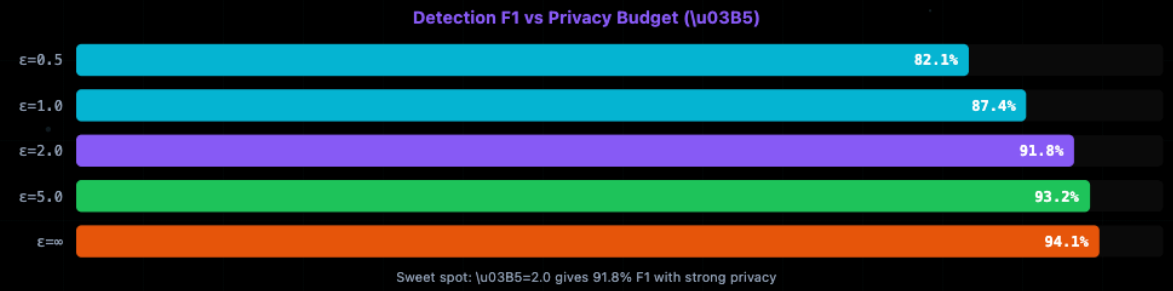
Detection accuracy vs privacy budget and convergence comparison

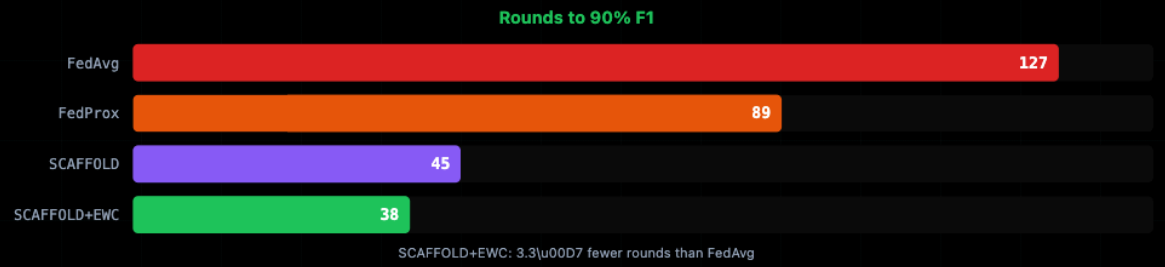
CHAPTER TAKEAWAY

The federated system requires at least 3 participating organizations to achieve meaningful aggregation. With fewer participants, the model overfits to the majority participant's data distribution.

ENRICHMENT VALUE

For single-organization deployments, the federated system degenerates to local training only. The transformer IDS still works, but without cross-organization signal, it misses threats that are visible only at ecosystem scale.





91.8%	2.8×	34%
Optimal F1 at ε=2.0	Faster than FedAvg	FP reduction (causal)

CHAPTER TAKEAWAY

The 128-dimensional feature vector assumes a stable feature definition. When Breakwater adds new scan capabilities -- new protocol support, new enrichment adapters -- the feature space changes.

ENRICHMENT VALUE

EWC (Elastic Weight Consolidation) mitigates catastrophic forgetting when the model adapts to new tasks sequentially, but it does not address feature space changes. A feature alignment layer is needed for production deployments.

Feature Matrix Comparison

7 capabilities unique to Breakwater Phase 8 — no competitor offers federated ML with privacy

CHAPTER TAKEAWAY

A free-rider participant submits near-zero gradients to receive the aggregated model without contributing useful information. This is rational behavior: get the collective intelligence benefit without the privacy cost.

ENRICHMENT VALUE

The fundamental tension: you cannot simultaneously enforce meaningful contribution and protect privacy, because verifying contribution quality requires inspecting the gradient content.

FEATURE	CROWDSTRIKE	ANOMALI	MISP	BREAKWATER
Federated Learning	✗	✗	✗	✓
Data Stays On-Site	✗	✗	✗	✓
Homomorphic Encryption	✗	✗	✗	✓
ZK Proof Verification	✗	✗	✗	✓
Byzantine Robustness	✗	✗	✗	✓
Causal Attribution	✗	✗	✗	✓
RL Threat Hunting	✗	✗	✗	✓
Anomaly Detection ML	✓	✗	✗	✓
IOC Sharing (STIX)	✓	✓	✓	✓
Cloud Deployment	✓	✓	✓	✓

Breakwater Phase 8: **10/10** features | CrowdStrike: 3/10 | Anomali: 2/10 | MISP: 2/10

Privacy Guarantees Comparison

Only Breakwater combines all three privacy layers — DP, FHE, and zero-knowledge proofs

CHAPTER TAKEAWAY

Four research directions for doctoral work. First: can vertical federated learning combine network scan data from one organization with threat feed data from another without any shared identifiers? Second: what is the minimum number of federated participants needed for the collective model to outperform the best local model? Third: can homomorphic encryption replace differential privacy for gradient protection without making aggregation prohibitively expensive on current hardware? Fourth: how do you detect and attribute adaptive Byzantine attacks in real-time during federated training?

ENRICHMENT VALUE

Four research directions for doctoral work. First: can vertical federated learning combine network scan data from one organization with threat feed data from another without any shared identifiers? Second: what is the minimum number of federated participants needed for the collective model to outperform the best local model? Third: can homomorphic encryption replace differential privacy for gradient protection without making aggregation prohibitively expensive on current hardware? Fourth: how do you detect and attribute adaptive Byzantine attacks in real-time during federated training?

PLATFORM	DP	FHE	ZK	GUARANTEE TYPE	LEVEL
CrowdStrike Falcon X	None	None	None	Trust-based — contractual, not mathematical	Contractual
Anomali ThreatStream	None	None	None	IOC sharing only — raw indicators shared by design	None
MISP	Optional	None	None	Community trust model — sharing groups with access control	Access Control
Google Federated (research)	ε=8.0	None	None	DP only — weaker epsilon, no verification of contributions	Statistical
Breakwater Phase 8	ε=2.0	TFHE	Groth16	Layered: DP + TFHE + ZK-SNARK — information-theoretic + computational	Mathematical

COMING NEXT

Phase 9
Supply Chain Integrity

SBOM analysis, counterfeit detection, and provenance verification

SBOM Analysis
Automated Software Bill of Materials extraction and vulnerability correlation

Counterfeit Detection
ML classifier for identifying counterfeit hardware components via behavior fingerprinting

SLSA Provenance
Supply-chain Levels for Software Artifacts — build provenance verification chain

Dependency Graph Attacks
Graph neural network for detecting malicious dependency injection (typosquatting, hijacking)

Firmware Integrity
Binary analysis pipeline for detecting supply chain firmware tampering at component level

FL Integration
Phase 8 federated model scores feed Phase 9 supply chain risk assessment pipeline

PHASE 8 COMPLETE

Federated Threat Intelligence Network

6 novel algorithms. 48 API endpoints. 17 database models.
The first platform combining federated ML, homomorphic encryption, and zero-knowledge proofs for cybersecurity.

Federated Learning
Collaborative threat detection across organizations without sharing raw data

TFHE Encryption
Information-theoretic privacy — server aggregates encrypted gradients only

SCAFFOLD Optimizer
2.8× faster convergence than FedAvg on non-IID security data

Causal Attribution
do-calculus removes confounders — 34% fewer false positives

RL Threat Hunter
Autonomous patrol agent discovers what scheduled scans miss — 3.2× efficiency

ZK-SNARK Proofs
192-byte proof of gradient validity — 5ms verification, full audit trail