29. Leverage Taproot on GlobalBoost for Zero-Knowledge Range Proofs in Rust

Why: Taproot scripts can embed Bulletproofs for proving amounts without revealing them. This hack adds value privacy to BSTY spends, essential in 2026 for confidential transactions without sidechains.
How to Implement: Use bulletproofs crate; commit in Taproot leaf.

```
rust
use bulletproofs::{BulletproofGens, PedersenGens, RangeProof};

use curve25519_dalek::scalar::Scalar;


// Hack: Range proof for confidential value

fn generate_range_proof(value: u64, blinding: Scalar) -> (RangeProof, /* commitment */) {

    let pc_gens = PedersenGens::default();

    let bp_gens = BulletproofGens::new(64, 1);

    let commitment = pc_gens.commit(Scalar::from(value), blinding);

    let proof = RangeProof::prove_single(&bp_gens, &pc_gens, &mut rand::thread_rng(), value, &blinding, 32).unwrap();

    (proof, commitment)

}


fn embed_in_taproot(proof: RangeProof) {

    // Add to Taproot script tree

}


let blinding = Scalar::random(&mut rand::thread_rng());

let (proof, _) = generate_range_proof(100000000, blinding);
```

Analysis: Proves 0 < value < 2^32 without reveal; Taproot hides proof until spend. BSTY's chain supports compact proofs (~1KB), enabling private balances with minimal overhead.