JOHN WOELTZ - GRINCON U.S. 2019

GRIN CRYPTO LIBRARY AUDIT

SECURITY AUDITS: WHY BOTHER?

- Show of good faith to your community?
  - BTC did not have professional audits before launching… but times change
- Just to CYA or is something more at stake?
- For Grin, loss of privacy is unacceptable, can lead to physical harm of user!

SECURITY AUDITS: ARE THEY MAGIC?

- **Effectiveness depends on quality of resources + time spent** (which hopefully translates fairly into cost)
- **Outcome depends on how strategically the cost was applied to** auditing the **width and depth** of a project to be sufficiently secure based on threat model
  - *Examples…*
- ***It is not possible even with all of the world's auditors working to claim "this code is fully secure and can never have any vulnerabilities"***
  - However we can claim to increase the cost and time that would be required to find a vulnerability
- Ideal Grin audit scenario…

BACK TO REALITY: RESCOPING AUDIT OF GRIN BEFORE LAUNCH DUE TO TIME/MONEY

- Challenge to find availability from firms with short notice
- Even more challenging to do so with a bear market 16BTC budget
- *Solution*: Reduce scope of audit to prioritize most critical functionality

REDUCE SCOPE FOR PRE-LAUNCH AUDIT TO SECP256K1-ZKP LIBRARY

- *What*: **SECP256K1-ZKP**, an extension of libsecp256k1 to add bulletproof range proofs and an aggsig module
- *Why*: This library supports **critical cryptographic operations** that Grin relies on
- *How much*: Directly paid by an anonymous donor, **0BTC** was deducted from Grin security audit fund
- *Who*: **JP Aumasson**, co-designer of BLAKE hash function, SipHash pseudorandom function and Gravity-SPHINCS signature scheme

REDUCE SCOPE FOR PRE-LAUNCH AUDIT TO SECP256K1-ZKP LIBRARY

- *Looked for:*
  - Side-channel leaks (e.g. timing leaks)
  - Software safety (e.g. memory leaks, API abuse etc)
  - Usage of underlying cryptographic primitives
  - RNG/PRNG
  - Cryptographic security level (e.g. key lengths)
  - Decoding serialized/DER data

RESULTS: POTENTIAL SECURITY ISSUES (SHOULD BE FIXED)

- *Issue*: Optimized out dead assignment may leak sensitive data
  - *Details*: In /src/ecmult_gen_impl.h at line 153, bits = 0 which is used to overwrite the value of private bits may be removed by compiler since bits is no longer used
  - *Solution*: Review generated binaries and check that the overwriting operation has not been removed by compiler
  - *Status*: Requires Review

RESULTS: POTENTIAL SECURITY ISSUES (SHOULD BE FIXED)

- *Issue:* Missing null pointer checks
  - *Details*: There are missing null pointer checks in secp256k1_aggsig_sign_single(), secp256k1_aggsig_verify_single(), secp256k1_aggsig_add_signatures_single()
  - *Solution*: Check nullity of all pointers
    - *Example*: add ARG_CHECK(seed != NULL) to secp256k1_aggsig_sign_single() to check the seed pointer
  - *Status*: Fix merged into master by @yeastplume

RESULTS: OBSERVATIONS (UNLIKELY TO FAIL BUT SAFER TO FIX)

- *Issue*: Unfreed heap allocations in secp256k1_aggsig_verify_single() and secp256k1_bulletproof_rangeproof_prove()
  - *Details*: Cases where 0 can be returned without deallocating the scratch frame, preventing scratch buffer from being freed
  - *Solution*: Ensure secp256k1_scratch_deallocate_frame(scratch) is run to free the scratch buffer in cases where 0 may be returned without freeing the buffer
  - *Status*: Fix merged into master by @yeastplume

RESULTS: OBSERVATIONS (UNLIKELY TO FAIL BUT SAFER TO FIX)

- *Issue*: Unchecked heap allocations in secp256k1_aggsig_verify_single(), secp256k1_aggsig_build_scratch_and_verify(), secp256k1_bulletproof_rangeproof_prove()
  - *Details*: The values of secp256k1_scratch_space_create(ctx, 1024*4096) and tge = malloc(2*sizeof(secp256k1_ge)) are not verified and could potentially return NULL
  - *Solution*: Add if (value == NULL) conditionals to relevant blocks to prevent a NULL pointer being returned to the caller
  - *Status*: Fix merged into master by @yeastplume

RESULTS: OBSERVATIONS (UNLIKELY TO FAIL BUT SAFER TO FIX)

- *Issue*: Unnecessary operations in secp256k1_aggsig_context_destroy()
  - *Details*: Unnecessary HMAC finalize is present
  - *Solution*: Remove line 606: secp256k1_rfc6979_hmac_sha256_finalize(&aggctx->rng);
  - *Status*: Fix merged into master by @yeastplume

RESULTS: IMPROVEMENTS (NICE TO HAVE)

- *Opportunity*: Faster rejection of invalid parameters in secp256k1_bulletproof_rangeproof_prove()
  - *Details*: The only valid values for the n bits parameter are known and can be checked to reject invalid n bits parameters before more expensive calculations occur
  - *Solution*: During ARG_CHECK() sequence ensure n bits value is < 64 with a 1-bit popcount/Hamming weight
  - Status: Already included in previous commit to master by @jaspervdm

RESULTS: ACTIONS

- PR with fixes submitted by @yeastplume: https://github.com/mimblewimble/secp256k1-zkp/pull/37
- audit_fixes merged into master with commit hash: *73617d0fcc4f51896cce4f9a1a6977a6958297f8*
- Diff: 15 lines changed, 14 added, 1 removed

NEXT STEPS FOR GRIN AUDITS

- Mainnet is already launched, why more audits?
- Updated scope for next audit:
  - *Grin core crate*
  - *Grin keychain crate*
  - *Grin chain crate*
- Status: Waiting on bids to review to select firm to engage

INITIAL GRIN CRYPTO LIBRARY AUDIT COMPLETED

- Full audit report available at: https://grin-tech.org/audits/jpa-audit-report
- Follow audit status updates and reports: https://github.com/mimblewimble/grin/issues/1609
- Contribute to Grin community funding: https://grin-tech.org/funding
- Contribute to @yeastplume developer funding (Mar 2019 - Aug 2019): https://grin-tech.org/yeastplume
  - Status: Open - €1,440 of €55,000 Target Goal: Crypto equivalent of €55,000
- **Thank you to all contributors** that have spent countless hours to make Grin a reality and to **JP Aumasson** for making time on short notice over the holidays to review the library before mainnet launched

MW