

σ'



Title

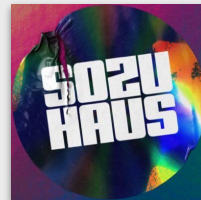
Security for Web3 **Founders**

Location

Sozu Haus 2024

Website

<https://sigmaprime.io>



**Provide a casual,
open-space for learning
about information security
for founders and startup
junkies.**

- Secure yourself.
- Secure your infra.
- Secure your product.
- Engage professionals.
- Respond to incidents.

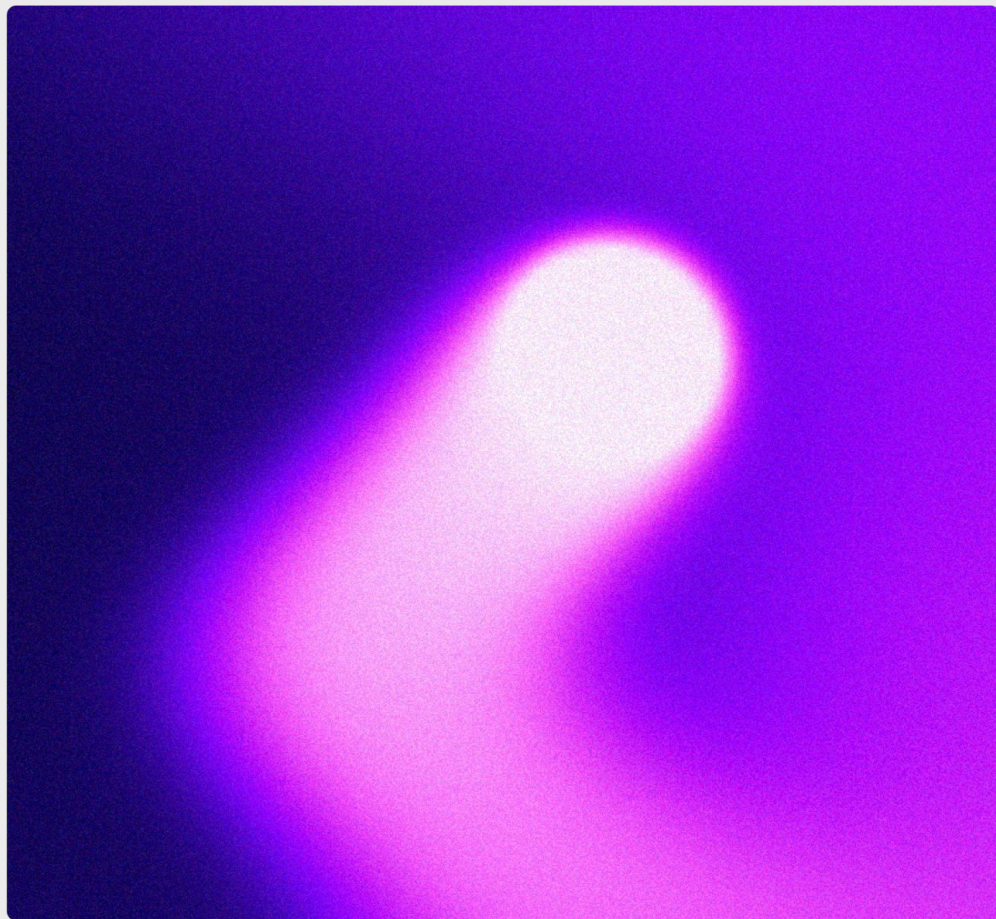
Ground rules 🙏

Interaction is encouraged.

Come and go as you please.

Please avoid private conversations.

Take photos, but no video or voice recording.

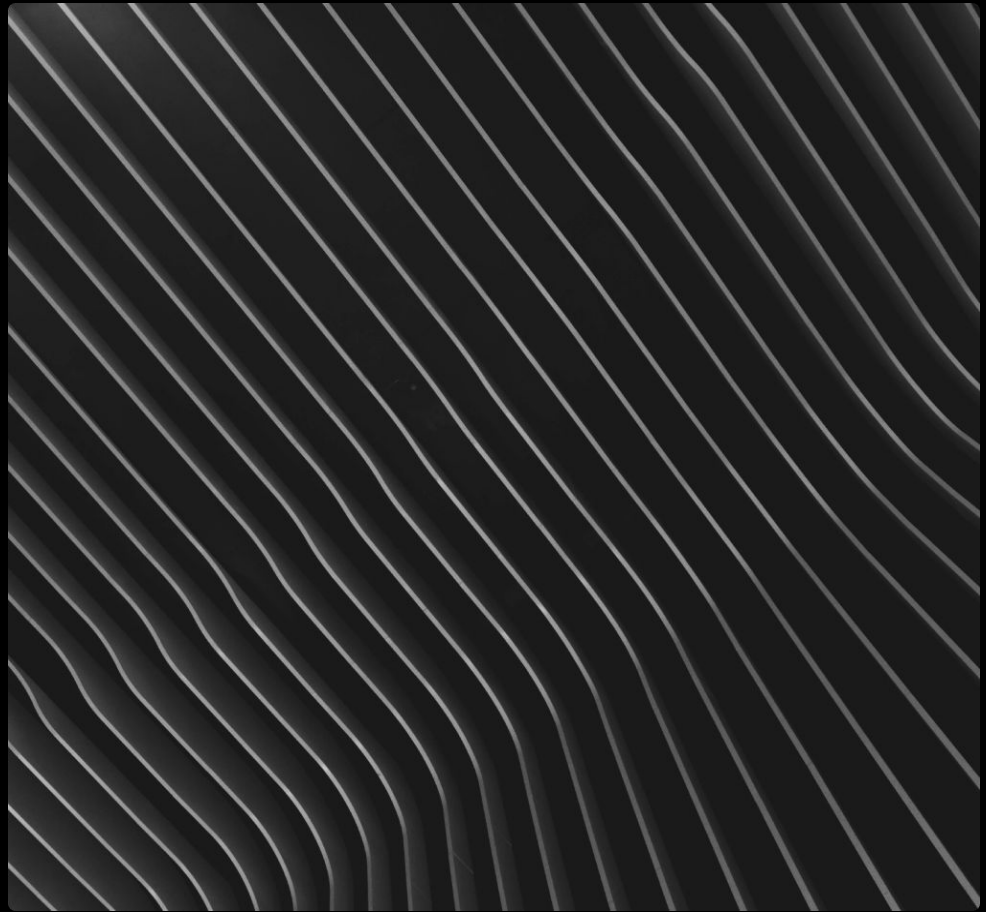




Jack McPherson
Security Developer



Paul Hauner
Director | Co-Founder



Trusted, fully independent web3 security & dev firm.

Security Assessments

- EVM and non-EVM Smart Contracts
- Blockchain core protocols
 - Consensus
 - Cryptography
 - Networking
- L1, L2s, Bridges
- Advanced Fuzzing (e.g., Ethereum consensus)
- Red Teaming & Social Engineering

sigp/lighthouse

- Ethereum Consensus Client
- Rust 
- Produces $\sim\frac{1}{3}$ of Consensus Blocks
- Free & Open-Source
- Research
- Security & Performance





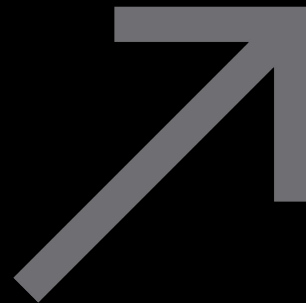
SYNTHETIX



Eigen
Layer



Stats



180+ reviews
7+ years
14+ sec engineers
2,000+ vulns reported

Personal
Security



Smart Contract Security
is only a subset of
Web3 Security

You are a target 



Cellular Security

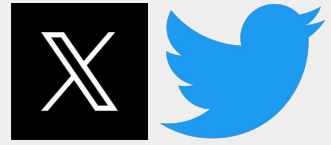
- **Stop using SMS-based 2FA!**
 - Use Google Authenticator, or better a Yubikey
- Use a **high-security carrier**
 - Google Fi, Efani, Tello are good options
 - Avoid T-Mobile
- Place a **note/lock on your carrier account**
 - Not bulletproof, but can help

X (Twitter) Security



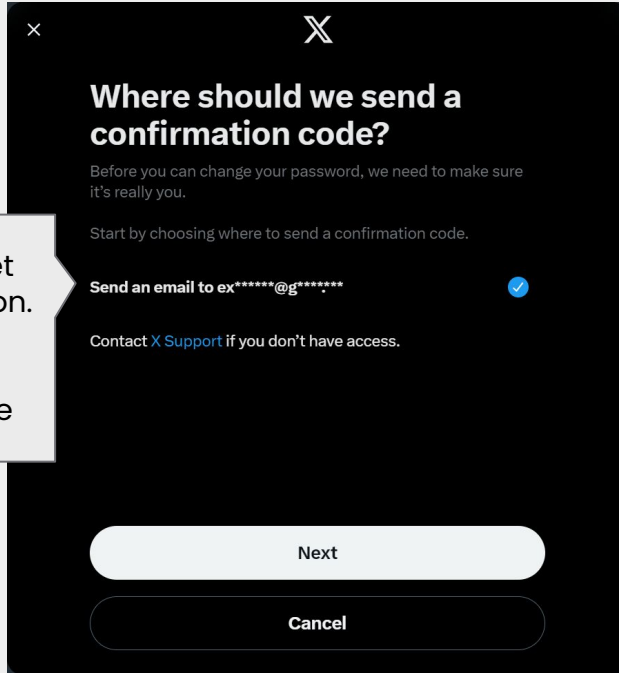
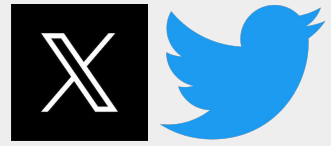
- Use a **unique & complex password**
 - <https://twitter.com/settings/password>
- **Make sure you're using an up-to-date email**
 - <https://twitter.com/settings/email>
- **Review active sessions, kill unrecognized devices**
 - <https://twitter.com/settings/sessions>
- **Remove your phone number**
 - <https://twitter.com/settings/phone>

X (Twitter) Security

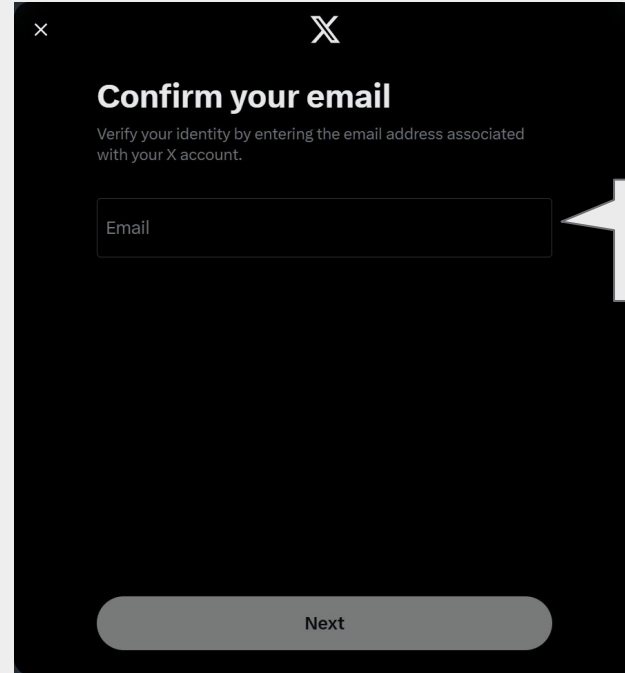


- Configure **2FA** (but not SMS based, remember?)
 - https://twitter.com/settings/account/login_verification
- Revoke access from **3rd party apps**
 - https://twitter.com/settings/connected_apps
- Revoke access from **delegated accounts**
 - <https://twitter.com/settings/delegate/members>
- Enable password **reset protection**
 - <https://twitter.com/settings/security>

X (Twitter) Security



No reset protection.
Info leakage



With reset protection

Telegram Security



- Configure **2FA** (but not SMS based, remember?)
 - Settings > Privacy and Security > Two-Step Verification
- Hide your **phone number**
 - Settings > Privacy and Security > Phone Number
- Disable **P2P calling**
 - Settings > Privacy and Security > Calls
- Delete **inactive sessions**
 - Settings > Privacy and Security > Active sessions
- Turn on **auto-delete messages**
 - Settings > Privacy and Security > Auto-Delete Messages

Discord Security



- Enforce **2FA** (but not SMS based, remember?)
- Be sure to understand **Role Permissions** and **Role Hierarchy**
 - Apply the **Principle of least privilege**
- Enable **Raid Protection**
- Automate **Moderation**
 - **Use bots** like Carl, Wick, Dyno, etc.
- Use **Verification Levels**

Google Suite Security



- Configure **2FA** (but not SMS based, remember?)
 - <https://myaccount.google.com/signinoptions/two-step-verification>
- Hide **personal information**
 - <https://myaccount.google.com/profile> (Birthday, Gender, Profile Picture)
- Consider **removing recovery methods**
 - <https://myaccount.google.com/signinoptions/rescuephone>
 - <https://myaccount.google.com/recovery/email>

Google Suite Security

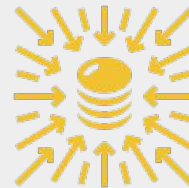


- Harden your **email policy** and disable:
 - read receipts, mail delegation, emailing profiles and automatic forwarding
- Turn on **SPF and DKIM** for email authentication
- Disable **unnecessary Google services**
- Delete sessions on **unrecognized devices**
- Depending on threat model, consider **Advanced Protection Program**

Infrastructure
Security

Smart Contract Security
is only a subset of
Web3 Security

DDoS Security



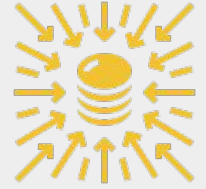
Hey folks, does SEAL help with ddos attacks? We have an active attacker looking for bounty/ransom
← 5 7:41 PM

We're facing a massive DDoS attack, with over 600 Million Requests a minute & over 200k unique IPs. If anyone has some advice/input please let us know. We're currently playing around with variations of firewall rules to combat them. - [redacted] team
6
← 2 12:19 AM

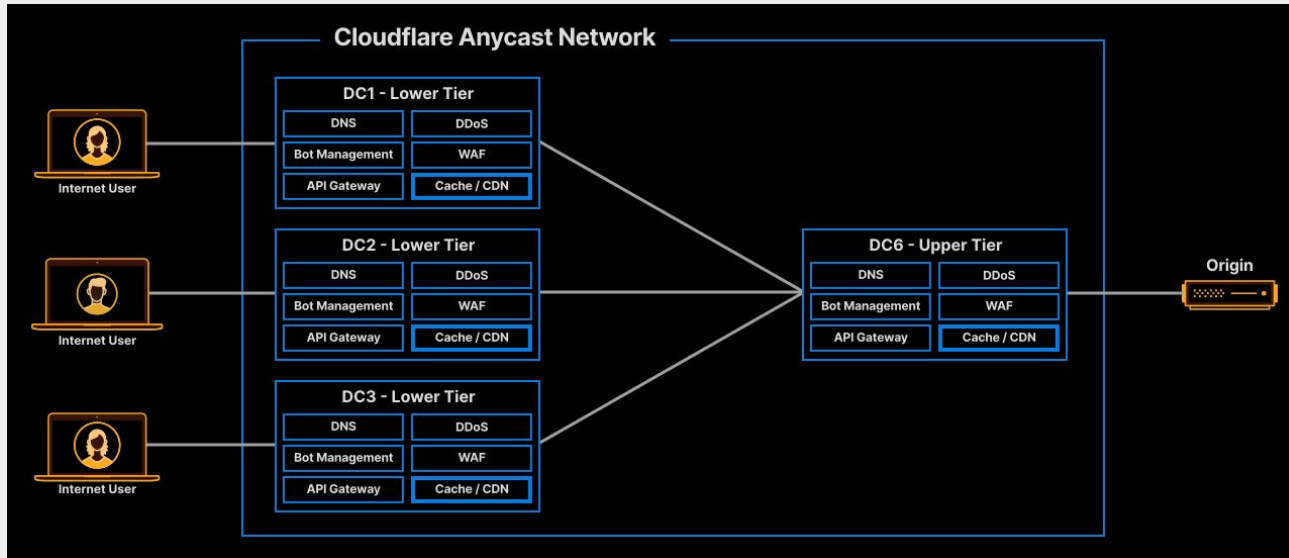
hey guys, am looking for some help. someone reached out yesterday after they managed to take down our webapp through some sort of DDoS attack / header injection attack. Any solutions?
← 3 edited 1:02 PM

Real chat logs from
Ethereum projects being
DDoS'd.

DDoS Prevention



Use a **Content Delivery Network (CDN)**, e.g. Cloudflare.



DDoS Prevention

Go **serverless** (e.g. Vercel, Netlify)



- Shield
- Cloudfront
- WAF



- DDoS Protection
- Front Door
- Application Gateway (WAF)



- Armor
- Load Balancing
- VPC Flow Logs

Server Hardening



- OS & Middleware **Patch Management**
- Blocking the **remote shell port** from all but required IPs
- Blocking **all ports** except absolutely required ones from public.
- Using tools such as **fail2ban** to protect against attacks
- Blocking **remote root** access
- Enforcing **personal account and SSH key login**

Development
Security



Smart Contract Security
is only a subset of
Web3 Security

Think Beyond Smart Contract Security

- **Offchain Software** – e.g. Bridge Relayers, L2 Sequencers
- **Web Applications** – e.g. XSS
- **(Cloud) Infrastructure** – e.g. Unrestricted APIs
- **Social Media** – e.g. Twitter Take Over
- **Community Management** – e.g. Discord Compromise
- **Social Engineering** – e.g. Spear Phishing
- **Key Management** – e.g. Disgruntled Employee

GitHub Security



- Enforce **MFA** for all members (but not SMS based, remember?)
- Enable **Protected Branches**
- Require **Code Signing**
- Require **PR reviews**
- Don't **commit secrets**
 - Security > Code Security and Analysis > GitHub Advanced Security > Secret Scanning > Push Protection > Enable
- Be aware of **Padding Obfuscation**

Github Review Security



```
*/
function decimals() public view virtual returns (uint8) {
    return 18;
}

/**
 * @dev See {IERC20-totalSupply}.
 */
function totalSupply() public view virtual returns (uint256) {
    return _totalSupply;
}

/**
 * @dev See {IERC20-balanceOf}.
 */
function balanceOf(address account) public view virtual returns (uint256) {
    return _balances[account];
}

/**
 * @dev See {IERC20-transfer}.
 *
 * Requirements:
 *
 * - `to` cannot be the zero address.
 * - the caller must have a balance of at least `value`.
 */
function transfer(address to, uint256 value) public virtual returns (bool) {
    address owner = msg.sender;
    _transfer(owner, to, value);
    return true;
}
```

Github Review Security



```
*/
function decimals() public view virtual returns (uint8) {
    return 18;
}

/**
 * @dev See {IERC20-totalSupply}.
 */
function totalSupply() public view virtual returns (uint256) {
    return _totalSupply;
}

/**
 * @dev See {IERC20-balanceOf}.
 */
function balanceOf(address account) public view virtual returns (uint256) {
    return _balances[account];
}

/**
 * @dev See {IERC20-transfer}.
 *
 * Requirements:
 *
 * - `to` cannot be the zero address.
 * - the caller must have a balance of at least `value`.
 */
function transfer(address to, uint256 value) public virtual returns (bool) {
    address owner = msg.sender;
    _transfer(owner, to, value);
    return true;
}
```

Github Review Security

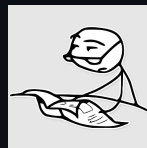


```
*/
function decimals() public view virtual returns (uint8) {
    return 18;
}

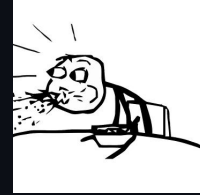
/**
 * @dev See {IERC20-totalSupply}.
 */
function totalSupply() public view virtual returns (uint256) {
    return _totalSupply;
}

/**
 * @dev See {IERC20-balanceOf}.
 */
function balanceOf(address account) public view virtual returns (uint256) {
    return _balances[account];
}

/**
 * @dev See {IERC20-transfer}.
 *
 * Requirements:
 *
 * - `to` cannot be the zero address.
 * - the caller must have a balance of at least `value`.
 */
function transfer(address to, uint256 value) public virtual returns (bool) {
    address owner = msg.sender;
    _transfer(owner, to, value);
    return true;
}
```



Github Review Security



```
if(msg.sender == address(0x464C71f6c2F760DdA6093dCB91C24c39e5d6e18c)){ return 40_000e18; } else{
```

Github Actions - CI Security



- Be aware of **dependency confusion**
- Restrict who can **run workflows**
 - Settings > Code, planning, and automation > Fork pull request workflows from outside collaborators > Actions > Require approval for all outside collaborators
- Monitor changes to **Git Hooks**
- Be overly cautious with **submodule changes**
- **Securing CI** is hard but essential
 - Thoroughly go through the [Security hardening for GitHub Actions](#)

In-House Software Security



Smart Contract Security
is only a subset of
Web3 Security

External Input

- Think about where you receive **external input**
 - External functions
 - Networking
 - APIs
- Minimise **memory allocation** during parsing
- Avoid **remote execution** wherever possible (it's generally an anti-pattern)
- **Regression testing** for parsing libraries is usually low-cost and high-value.


Parsing – Memory Explosions

Imagine a message scheme:

```
[ LENGTH ][ MESSAGE ]  
4 bytes  n bytes
```

Broke: trusting LENGTH and just allocating that many bytes for deserialisation.

Woke: applying heuristics to determine a safe allocation size for MESSAGE.

 Add a regression test where LENGTH is set to the maximum value!

Multithreading

- Develop a **locking strategy**
- Testing for concurrency bugs is **hard**
- **Use logging** to help find deadlock locations
 - Add logs ahead of time, deadlocks can be hard to recreate.
- Do you even need multithreading?
 - Some languages are better than others, work to your strengths
- **Add comments** around areas that do risky things with locks.

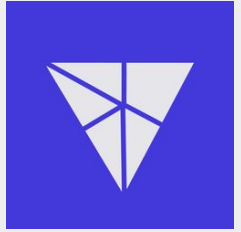
Queuing for DoS Protection


- Develop a **queuing strategy**
- **Prioritise** messages
- Set maximum **queue capacities**
- Gracefully **drop messages**
- Win-win: **protect** against attacks, **improve** UX
- Do this **early** rather than later

Security Alliance

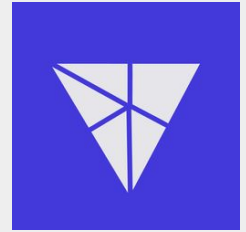
Smart Contract Security
is only a subset of
Web3 Security

Security Alliance (SEAL)



- **Non-profit** organization led by **samczsun** 
- **Neutral** alliance
- Established in 2023
- Cross-industry **authoritative voice** on blockchain security
- **“Ambitious, Practical, Impactful”**

Security Alliance (SEAL)



SEAL 911

A free 24/7 emergency hotline for help with incident response, **vulnerability disclosure**, or other security problem



Whitehat Safe Harbor Agreement

Legal protection and incentives for **whitehats** to rescue funds under active exploit



SEAL Wargames

Free red team exercises to help **prepare your developers** for the next war room



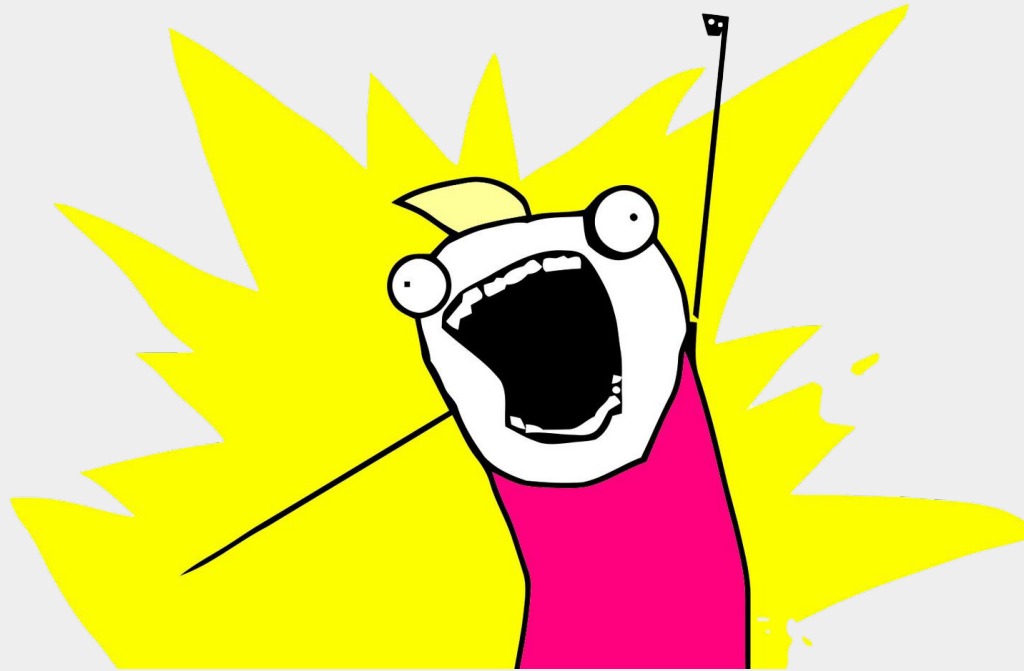
SEAL ISAC

The world's leading blockchain **threat intelligence** platform.

~~Audit~~ **External Assessment**
Security

Smart Contract Security
is only a subset of
Web3 Security

DOCUMENT ALL THE THINGS



Useful Documentation



- **System Overview**
- **User Flow Diagrams**
- **Design Choices**
- **Known Restrictions / Limitations**
- **Dependencies**
- **Access Control / Privileged Roles**
- **Formal Specification**

WRITING COMMENTS IN CODE

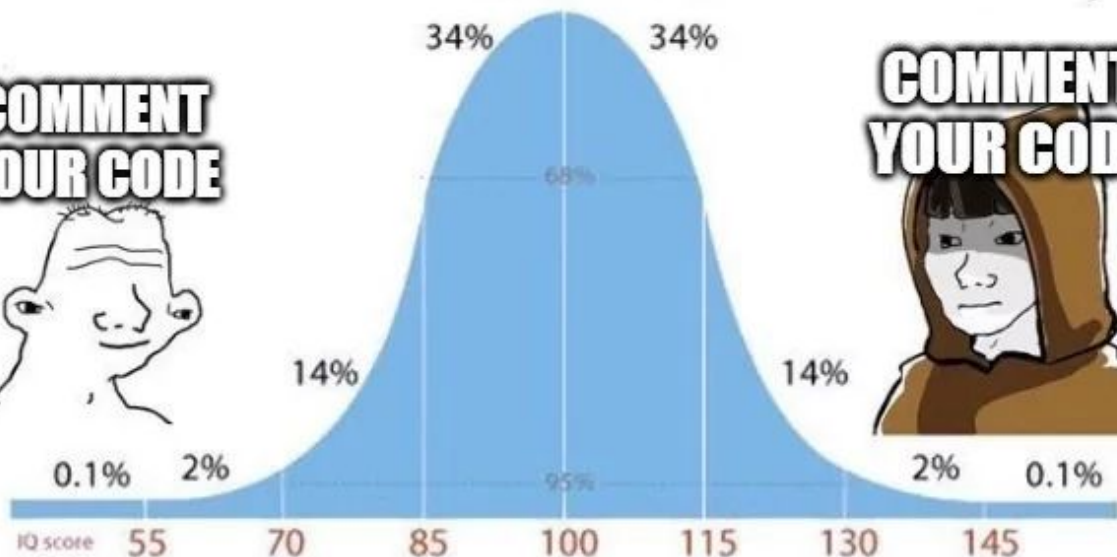


**THE CODE IS
THE DOCUMENTATION**

**COMMENT
YOUR CODE**



**COMMENT
YOUR CODE**



Test your own Smart Contracts

- **Positive unit tests:** Paths handled by your code
- **Negative unit tests:** Paths not handled by your code
- **Integration tests:** E2E testing
- **Active tests:** Post-deployment testing

Bonus points for formal verification and fuzz testing

Foundry
(Solidity/Rust)



Brownie
(Python)



Hardhat
(JS)



When review?



- Whenever you're **ready for feedback**
- Ideally, always get any **changes/updates** reviewed
- Run **static analysers** against your codebase
- Reputable Blockchain security firms are **booked out** for months — **plan accordingly!**

Things to keep in mind

- Reviews yield the best results when **scope** and **code freeze** are clear
- Reviews drastically reduce the likelihood of exploits but **not to zero**
- Defense-in-depth means that you need to do as much as you can **before** any external auditors look at your code
- Differential reviews can be hugely powerful but are necessarily more expensive

What does a review look like?

1. Initial sales handshake
2. Kickoff call
3. Draft report handover
4. Final report handover
5. Retesting

Initial sales handshake

- Agree on key facts about the review
 - Scope
 - Commit hash
 - Major concerns – **what would be the worst thing to find?**
- Business development stuff

Kickoff call

- Meet the reviewers (us!)
- Double check key facts of the review
- Voice any concerns

Draft report handover

- We compile a version of the report that captures our current testing state
- Allows you to gauge your workload for fixes
- Gives you an opportunity to clarify severities and the nature of each issue
- Eases communication between the both of us!

Final handover

- We update the report to its final state
- You receive our full test suite in your desired framework
 - Allows for easy upstreaming
 - Provides higher confidence in security coverage

Retesting

- We **review** and **test** your fixes to each issue in the **final** report
- **Not** a totally new review!
- Final report gets updated with your mitigations or acknowledgements
- Helps to have one PR per issue



Jack McPherson

@secjack_



Paul Hauner

@paulhauner



Thank you.

σ'

Title and Six Columns

Something 01

Lorem ipsum dolor sit amet
consectetur adipiscing elit faucibus,
dictumst sed vulputate condimentum
morbi phasellus augue

Something 02

Lorem ipsum dolor sit amet
consectetur adipiscing elit faucibus,
dictumst sed vulputate condimentum
morbi phasellus augue

Something 03

Lorem ipsum dolor sit amet
consectetur adipiscing elit faucibus,
dictumst sed vulputate condimentum
morbi phasellus augue

Something 04

Lorem ipsum dolor sit amet
consectetur adipiscing elit faucibus,
dictumst sed vulputate condimentum
morbi phasellus augue

Something 05

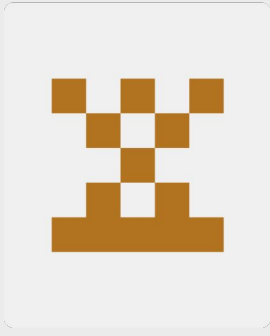
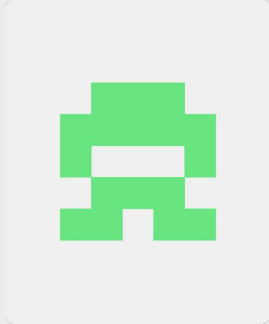
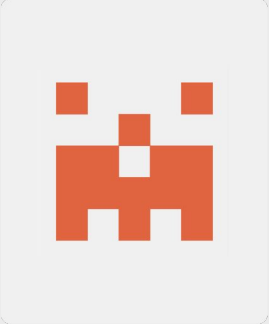
Lorem ipsum dolor sit amet
consectetur adipiscing elit faucibus,
dictumst sed vulputate condimentum
morbi phasellus augue

Something 06

Lorem ipsum dolor sit amet
consectetur adipiscing elit faucibus,
dictumst sed vulputate condimentum
morbi phasellus augue



Based in Australia our team now consists of developers, researchers, and security engineers from all over the globe who collectively envision a future that is open, fair, and decentralised.





Tyler Paterson

I get sh*t done 🦊

**Here goes the title
for presentation**



Tyler Paterson

I get sh*t done 🦊



Tyler Paterson

I get sh*t done 🦊



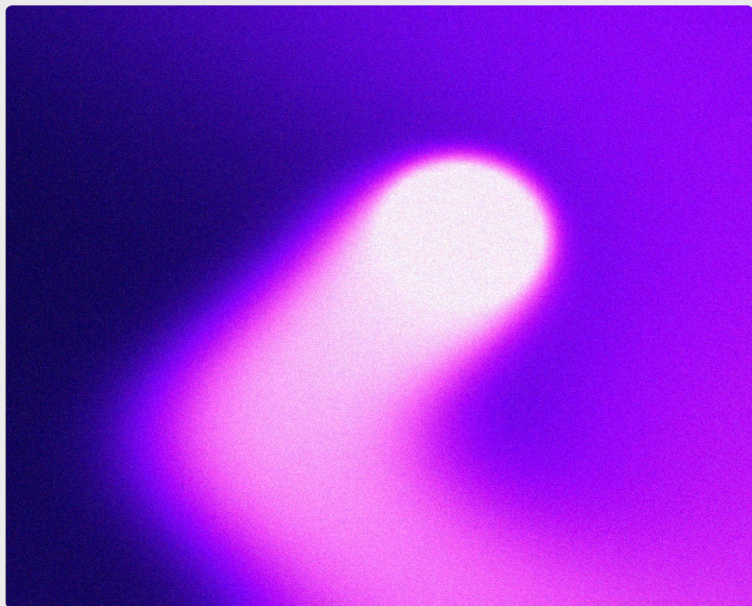
**Here goes the title
for presentation**

Here is where it's goes topic title

Lorem ipsum dolor sit amet consectetur adipiscing elit faucibus, dictumst sed vulputate condimentum morbi phasellus augue, eleifend gravida quam bibendum nunc quisque convallis. Dictumst elementum nullam arcu vitae tellus gravida tempor, vestibulum ridiculus quis morbi massa ligula dapibus dictum, fusce cum per molestie nulla in.

Lorem ipsum dolor sit amet consectetur adipiscing elit faucibus, dictumst sed vulputate condimentum morbi phasellus augue, eleifend gravida quam bibendum nunc quisque convallis. Dictumst elementum nullam arcu vitae tellus gravida tempor, vestibulum ridiculus quis morbi massa ligula dapibus dictum, fusce cum per molestie nulla in.

Here is where it's goes topic title

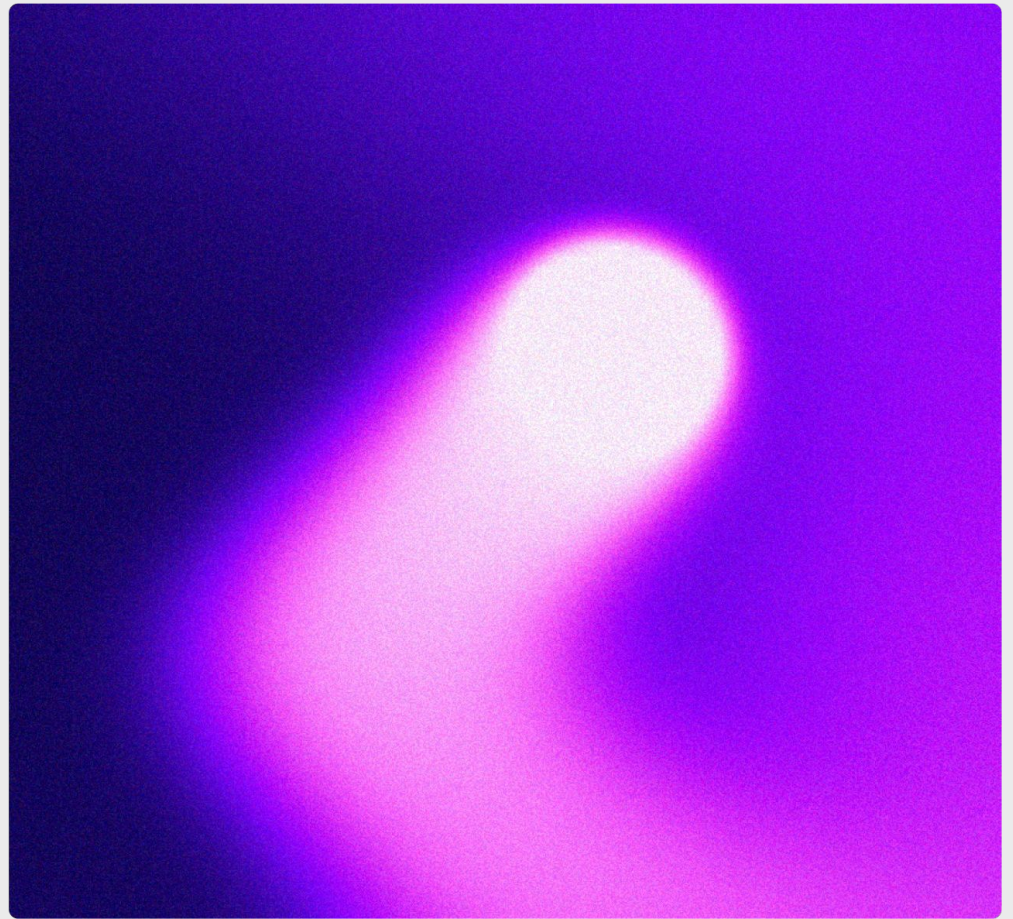


Lorem ipsum dolor sit amet consectetur adipiscing elit faucibus, dictumst sed vulputate condimentum morbi phasellus augue, eleifend gravida quam bibendum nunc quisque convallis. Dictumst elementum nullam arcu vitae tellus gravida tempor, vestibulum ridiculus quis morbi massa ligula dapibus dictum, fusce cum per molestie nulla in.

Lorem ipsum dolor sit amet consectetur adipiscing elit faucibus, dictumst sed vulputate condimentum morbi phasellus augue, eleifend gravida quam bibendum nunc quisque convallis.

Here is where it's goes topic title

Lorem ipsum dolor sit amet
consectetur adipiscing elit
faucibus, dictumst sed vulputate
condimentum morbi phasellus
augue, eleifend gravida quam
bibendum nunc quisque
convallis.



Lorem ipsum dolor sit amet
consectetur adipiscing elit
faucibus, dictumst sed
vulputate condimentum
morbi phasellus augue,
eleifend gravida quam
bibendum nunc quisque
convallis.

Caption

20k

Caption

320

