



Security Alliance Wargames

Preparing teams for incident response

Isaac Patka

Security Alliance (SEAL) | Shield3



“Securing
the future
of crypto”



Outline

- What are Wargames and why do them?
- How can you work with us and use our resources?
- Best practices from working with major protocols
- Walkthrough of open source toolkit



What are Wargames?

SEAL Wargames are cybersecurity exercises for web3

- Designed to prepare protocol teams for high-pressure situations.
- Test & improve both the social and technical resilience of teams.
- ~~Incite panic~~ | Practice emergency response

Phase 1 - Open-source intelligence gathering

Phase 2 - Tabletop exercise to identify team's monitoring & response process & weaknesses

Phase 3 - Live attack simulation on a forked network requiring real-time coordination from the team

Why Wargames

Recent hacks have been operational failures, not smart contract bugs



We are seeing less Smart contracts hacks which is a positive outcome from increased security resources available

Most recent big hacks seemed to be tied to operational security issues

What can be done?



We can make your contracts secure, but if you give your multisig signature permissions to some normies or people that are insufficiently aware of security risks, there's nothing we can do about it.

...



Radiant Capital ✅
@RDNTCapital

On October 16, 2024, Radiant Capital experienced a highly sophisticated security breach that resulted in the loss of \$50 million USD. The attackers exploited multiple developers' hardware wallets through a highly advanced malware injection.



Rahul Saxena ✅
@saxenism

Don't want to name names here, but some major protocols out there are playing with fire.

Operating a protocol that handles a Bazillion dollars without having a head of security or even a single security resource is NOT confidence in your engineers but pure fucking delusion.

...

Wargames are a cross-functional exercise



Core Devs

- Subject matter experts
- Understand the attack
- Prepare & simulate responses



Auditors

- Support core devs to understand incident
- Analyze implications of response



Guardians

- Sign transactions to fix/ pause the protocol



Communications

- Share information with the public
- Maintain incident response records



Legal

- Analyze legal implications of response
- Sign-off on whitehat engagements



Projects We've Worked With



ARRAKIS

OPTIMISM

Wargames provide hands-on experience with incident response

“Under pressure, you don't rise to the occasion, you sink to the level of your training.”

(Quote attribution: either the ancient Greek poet Archilochus or Navy SEALs.
Internet not sure)

Custom build. Realistic environment

Protocol	Network Fork	Monitoring	Bots	Exploit	Misc.
	Compound	Anvil/ Blockscout	Forta	Foundry scripts	Foundry scripts
	Yearn	Anvil/ Blockscout	Silverback	Silverback	Silverback Forked internal tools & UIs
	Aave	Tenderly	Silverback	Silverback	Silverback Forked UI
	Superchain	Conduit/ Sepolia	Chain-mon / Hexagate	Chain-mon fork	Foundry scripts SEAL 911
	Uniswap	Tenderly	Chain-mon fork	Chain-mon fork	Chain-mon fork

OS Drill Template Toolkit: <https://github.com/security-alliance/drill-template>

Scenarios for Wargame Exercises Have Included:

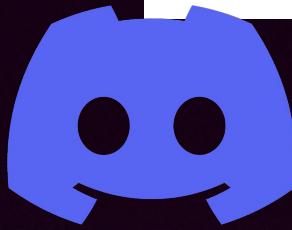
- External dependency failures
- Faulty contract upgrades
- Oracle manipulation
- Malicious governance proposals

(Vulnerability mapping - social, technical, legal, economic)

Best Practices

Maintain multiple channels to reach Guardians

- Pre-sign critical transactions to pause the protocol to minimize coordination time
- If signing is needed, have multiple ways of reaching multisig guardians



Best Practices

Isolate risks from protocol dependencies

- Yearn prepares strategy specific risk scores
- Risks are contained to specific strategies and cannot affect funds in other strategies
- Emergency Cards are prepared before deploying a strategy containing key subject matter experts and incident response steps

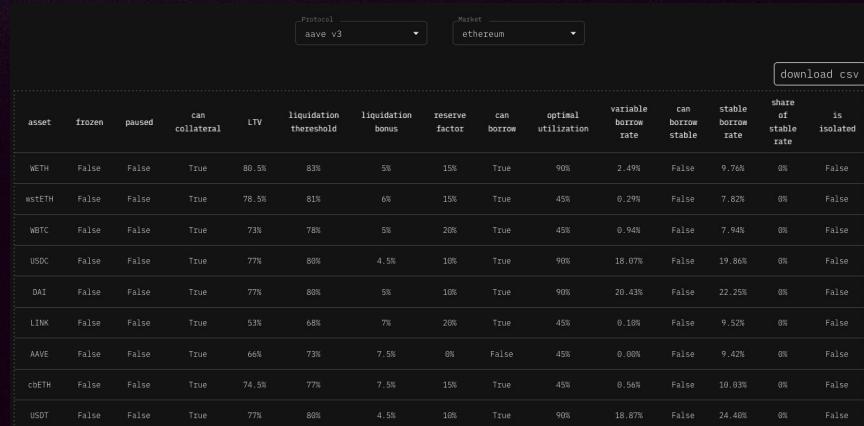
Group	~ TVL Impact	Audit	Code Review	Complexity	Longevity	Protocol S...	Team Knowl...	Testing	Median
stETH accumulator v1 strategies	118.68m 31.58%	Red	Yellow	Green	Green	Green	Green	Yellow	Green
Curve Boosted Factor 62 strategies	101.54m 27.02%	Red	Green	Green	Green	Green	Green	Yellow	Green
Gen Lender 14 strategies	44.12m 11.74%	Orange	Yellow	Yellow	Red	Green	Yellow	Yellow	Yellow
Curve 13 strategies	22.47m 5.98%	Green	Orange	Green	Green	Green	Green	Yellow	Green
Router Strategy 9 strategies	18.73m 2.08%	Red	Green	Green	Green	Green	Red	Yellow	Green
No Group 38 strategies	6.34m 1.69%	Red	Red	Red	Red	Red	Red	Red	Red
Convex Factory 5 strategies	5.09m 1.35%	Orange	Green	Green	Green	Green	Green	Yellow	Green
Router Strategy v2 14 strategies	4.09m 1.09%	Red	Green	Green	Green	Green	Orange	Yellow	Green

<https://seafood.yearn.watch/risk>

Best Practices

Isolate risks from protocol dependencies

- Aave isolates risks from different collateral assets by restricting which assets can be used for cross-collateral borrowing
- Individual collateral assets can be paused or frozen



The screenshot shows a table of Aave v3 protocol parameters for different assets on the Ethereum market. The table includes columns for asset, frozen, paused, can collateral, LTV, liquidation threshold, liquidation bonus, reserve factor, can borrow, optimal utilization, variable borrow rate, can borrow stable, stable borrow rate, share of stable rate, and is isolated. The assets listed are wETH, wstETH, wBTC, USDC, DAI, LINK, AAVE, cbETH, and USDT. The 'is isolated' column shows that only AAVE has this feature enabled.

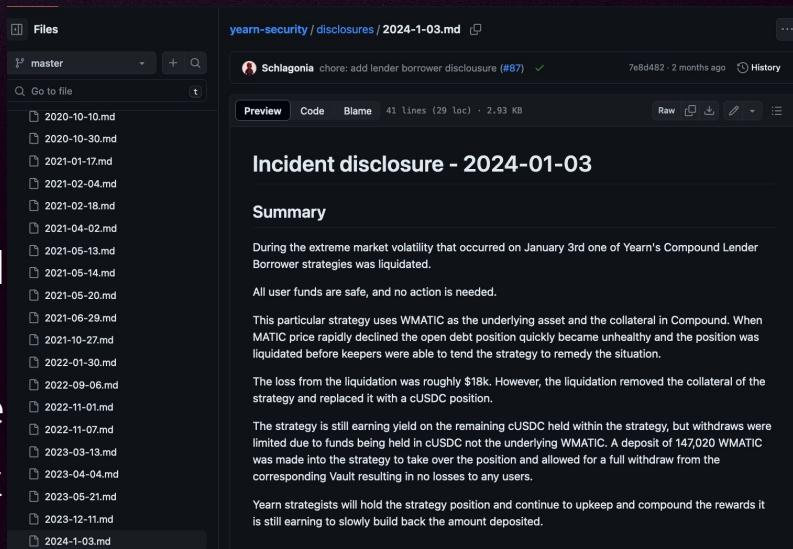
asset	frozen	paused	can collateral	LTV	liquidation threshold	liquidation bonus	reserve factor	can borrow	optimal utilization	variable borrow rate	can borrow stable	stable borrow rate	share of stable rate	is isolated
wETH	False	False	True	80.5%	83%	5%	15%	True	90%	2.49%	False	9.76%	0%	False
wstETH	False	False	True	78.5%	81%	6%	15%	True	45%	0.29%	False	7.82%	0%	False
wBTC	False	False	True	73%	78%	5%	20%	True	45%	0.94%	False	7.94%	0%	False
USDC	False	False	True	77%	80%	4.5%	10%	True	90%	18.07%	False	19.86%	0%	False
DAI	False	False	True	77%	80%	5%	10%	True	90%	20.43%	False	22.25%	0%	False
LINK	False	False	True	53%	68%	7%	20%	True	45%	0.10%	False	9.52%	0%	False
AAVE	False	False	True	66%	73%	7.5%	8%	False	45%	0.00%	False	9.42%	0%	False
cbETH	False	False	True	74.5%	77%	7.5%	15%	True	45%	0.56%	False	10.03%	0%	False
USDT	False	False	True	77%	80%	4.5%	10%	True	90%	18.87%	False	24.40%	0%	False

<https://www.config.fyi/>

Best Practices

Conduct & publish incident post-mortems

- Reflect on the *monitoring* which detected the issue and the response by the team
- Integrate learnings from the post-mortem into an incident response playbook



<https://github.com/yearn/yearn-security/tree/master/disclosures>

Best Practices

Mirror monitoring infrastructure on testnets

- Test alerts regularly and run drills to ensure alerts work
- Include playbooks in the alert to help the responder identify false alarms, and gather a war room if needed

The screenshot shows the Opsgenie web interface with the 'Alerts' tab selected. The page displays a list of open alerts, each with a title, severity, and creation date. The alerts are categorized by status: Open, Closed, UnAcked, and Not seen. Each alert entry includes a checkbox, a small icon representing the alert type (e.g., envelope for inbox errors), and the name of the incident. On the right side of the list, there are buttons for 'Ack', 'Close', 'Assign', 'Escalate to Next', 'Add Responder', 'Snooze', and 'Delete'. Below the alert list, there are two columns of status messages from a responder named 'Rebecca Howard'.

Date	Message	Status
Aug 21, 2019 4:46 PM (GMT+03:00)	Ack	CLOSED
Aug 20, 2019 7:34 PM (GMT+03:00)	OPEN	OPEN
Jul 6, 2019 2:38 PM (GMT+03:00)	ACKED	CLOSED
Jul 4, 2019 2:27 PM (GMT+03:00)	OPEN	OPEN
Jul 4, 2019 9:03 AM (GMT+03:00)	ACKED	CLOSED
Jun 20, 2019 9:23 PM (GMT+03:00)	ACKED	CLOSED
Jun 20, 2019 9:22 PM (GMT+03:00)	ACKED	CLOSED

Open Source Template for Wargames

A Foundry & Hardhat setup for developing & testing scenarios on a local fork

Configurations for running a live fork on Tenderly

A template for a tabletop exercise

A template typescript bot service (inspired by Optimism)

A template monitoring bot service with connections to Prometheus, Grafana, and OpsGenie (inspired by Optimism)

<https://github.com/security-alliance/drill-template>

drill-template Public Edit Pins Unwatch 7

demo-drill Go to file + Code

ipatka Update README.md 9a4d37f · 2 weeks ago

assets Cleanup 2 weeks ago

bots Add bot, monitoring, s... 6 months ago

foundry cleanup 2 weeks ago

monitoring Cleanup 2 weeks ago

services cleanup 2 weeks ago

tabletop Add bot, monitoring, s... 6 months ago

.gitmodules initial setup last year

README.md Update README.md 2 weeks ago



Resources

Tabletop Script

[Protocol Name]

[Date]

Drill Goals

In this first exercise, we will discuss scenarios in which the core teams needs to detect, diagnose, and respond to a variety of issues. We will gather the core decision-makers together to decide what actions should be taken, and what should be communicated. This exercise will be discussion-based and will not involve any real actions that need to be taken.

The goal of the exercise is to prepare for attacks and dependency failures in order to test both social and technical resiliency, harden internal procedures for deployment & recovery, and develop a training program for team members.

After going through this drill the team will be able to understand:

- What are the key dependencies of the protocol and what happens if they fail?
 - Is there sufficient monitoring infra to detect & respond to failures?
 - Is there an understanding of who has access to admin keys and how & when they are accessed to respond to threats?
 - Do essential team members have backup people in place who can respond if they are unavailable?
 - Does the team know who to reach out to in other protocols if a failure is detected?
-

Stakeholders

All team members are welcome in the exercise with duties related to:

- Notification & reporting infrastructure for application state & dependencies
- Smart contract developers
- FE development & interface hosting
- Communications

Tabletop Script

Since the incident response will change as upgrades are made to the protocol, let's discuss these scenarios as if they happened today.

Scenario 1:

Phase 1 - discovery of the issue

What monitoring infrastructure alerted the team to the issue?

Phase 1a - Gather stakeholders

Did the team know who to ask for help in this situation?

Phase 2 - Analyze the issue and develop a response plan

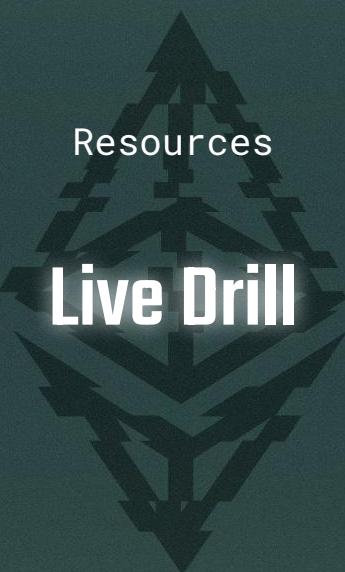
Did the team have an existing playbook for what to do?

Phase 3 - Implement the response plan

Who took action and what did they do?

Phase 4 - Reflection

What could have gone better?



Resources

Live Drill

Live Drill Setup Steps

1. Create network fork
2. Run bots & monitoring
3. Brief team
4. Team triages incident
5. Team prepares response & comms
6. Team executes incident response
7. Postmortem



Run The Live Drill

Live Fork Explorer & RPC

← All Virtual TestNets

SEAL Template Drill Fork Enable State Sync

Forked from Mainnet · Chain ID 1

Current Block 21036923 · Block Height 0/∞

Created just now · Last Interaction just now

Admin Public

HTTPS <https://virtual.mainnet.rpc.tenderly.co/2fc81881-c8f8-4...>

WSS <wss://virtual.mainnet.rpc.tenderly.co/57964e73-c52c-4...>

Get Started × Dismiss

Fund Account
Use the unlimited faucet to instantly top up the balance on any account.
[Fund Account](#) Not completed

Integrate and Deploy
Connect your development environment and deploy contracts to the Virtual TestNet.
[View Guides](#) Not completed

Send Transaction
Send transactions to test smart contract functionalities.
[Send Transaction](#) Not completed

All Standard System

Hash From To Function Block When Synced Blocks

✓ 0x		TenderlyCheatcodes	vNetCreation	21036923	just now	
0x5f0...730d5		0x4838b1...5f97	0xf94e5c...e85d	-	21036922	just now
0xfb1...ffff		0xfb8b1bc...8726	0xff0000...2800	-	21036922	just now
0x1c0...662bf		0x5050f6...76c9	0xff0000...8453	-	21036922	just now

🕒 5s Filters

Bots & Monitoring

- Home
- Starred
- Dashboards
- Explore
- Alerting
 - Alert rules
 - Contact points
 - Notification policies
 - Silences
 - Active notifications
 - Settings
- Connections
 - Add new connection
 - Data sources
- Administration

Invariant Violations

2024-10-24 14:41:45

```
{  
  __name__="lockup_monitor_invariantViolations",  
  instance="mon:7300",  
  job="chain-mon",  
  recipient="0x4AE34EE0983B2845823CB490ac769FD066Dc5c14",  
  token="0xe58cBE144dD5556C84874deC1b3F2d0D6Ac45F1b"  
}
```

Lockup Contract Balances

lockup_monitor_lockupContractBalance {__name__="lockup_monitor_lockupContractBalance", instance="mon:7300", job="chain-mon", token="0x4AE34EE0983B2845823CB490ac769FD066Dc5c14"}
lockup_monitor_lockupContractBalance {__name__="lockup_monitor_lockupContractBalance", instance="mon:7300", job="chain-mon", token="0xe58cBE144dD5556C84874deC1b3F2d0D6Ac45F1b"}
lockup_monitor_lockupContractBalance {__name__="lockup_monitor_lockupContractBalance", instance="mon:7300", job="chain-mon", token="0xe58cBE144dD5556C84874deC1b3F2d0D6Ac45F1b"}
lockup_monitor_lockupContractBalance {__name__="lockup_monitor_lockupContractBalance", instance="mon:7300", job="chain-mon", token="0xe58cBE144dD5556C84874deC1b3F2d0D6Ac45F1b"}

Incident Response

Resources: 'LET THE GAMES BEGIN'

- Join live Wargames event (previously November 10 between DSS and Devcon in Bangkok with Tenderly)
- SEAL Wargames Drill Scenario Template (OS toolkit):
<https://github.com/security-alliance/drill-template>
- Join the waitlist for a SEAL sponsored drill:
<https://securityalliance.org/>
- Paid track for projects on an accelerated timeline:
<https://www.shield3.com/>
- Chat with Isaac re. Running your own wargames



Security Alliance (SEAL) | Shield3
isaac@shield3.com
@isaacpatka on Telegram