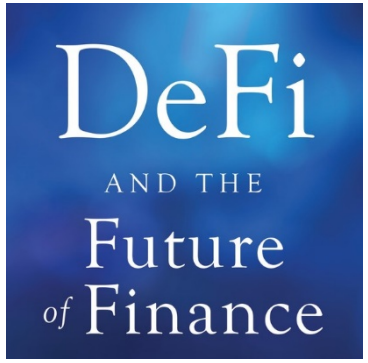

Course IV:

DeFi Risks and Opportunities

1. Smart Contract Risk

(i) Types of Exploits

Risks: Smart contract risk

Hack

- Over the past decade, crypto-focused products, primarily exchanges, have repeatedly been hacked.
- Whereas many of these hacks happened because of poor security practices, they demonstrate an important point: software is uniquely vulnerable to hacks and developer malpractice.
- Blockchains can remove traditional financial risks, such as counterparty risk, with their unique properties, but DeFi is built on code.

Risks: Smart contract risk

Attack vector

- This software foundation gives attackers a larger attack surface than the threat vectors of traditional financial institutions.
- Public blockchains are open systems.
- Anyone can view and interact with code on a blockchain after the code is deployed.
- Given that this code is often responsible for storing and transferring blockchain native financial assets, it introduces a new, unique risk.
- This new attack vector is termed *smart contract risk*.

Risks: Smart contract risk

Audit

- DeFi's foundation is public code known as a smart contract.
- The implementation is new to mainstream engineering practice. Practices that will help reduce the risk of smart contract bugs and programming errors are still under development.
- The recent hacks of The DAO, DForce and bZx demonstrate the fragility of smart contract programming.
- Auditing firms, such as Quantstamp, Trail of Bits, and Peckshield, are emerging to fill this gap in best practices and smart contract expertise.

Risks: Smart contract risk

Sources of risk

- Smart Contract risk can take the form of a logic error in the code or an economic exploit in which an attacker can withdraw funds from the platform beyond the intended functionality.
- The former can take the form of any typical software bug in the code.

Risks: Smart contract risk

Example: Logic error

- Suppose we have a smart contract which is intended to be able to escrow deposits from a particular ERC-20 from any user and transfer the entire balance to the winner of a lottery.
- The contract keeps track of how many tokens it has internally, and uses that internal number as the amount when performing the transfer.
- The bug will belong here in our hypothetical contract.

Risks: Smart contract risk

Example: Logic error

- The internal number will, due to a rounding error, be slightly higher than the actual balance of tokens the contract holds.
- When it tries to transfer, it will transfer “too much” and the execution will fail.
- If there was no failsafe put into place, the tokens are functionally locked within the protocol. Informally these are known as “bricked” funds and cannot be recovered.

Risks: Smart contract risk

Example: Economic exploit

- An economic exploit would be more subtle.
- There would be no explicit failure in the logic of the code, but rather an opportunity for an economically equipped adversary to influence market conditions in such a way as to profit inappropriately at the contract's expense.
- For example, let's assume a contract takes the role of an exchange between two tokens. It determines the price by looking at the exchange rate of another similar contract elsewhere on chain and offering that rate with a minor adjustment.

Risks: Smart contract risk

Example: Economic exploit

- The other exchange is playing the role of a price oracle
- The possibility for an economic exploit arises when the oracle exchange has significantly lower liquidity when compared to the primary exchange
- A financially equipped adversary can sell heavily on the oracle exchange to manipulate the price, then proceed to purchase far more on the primary exchange to capitalize on the price movement. The net effect is that the attacker was able to manufacture a discounted price on a high liquidity exchange by manipulating a low liquidity oracle.

Risks: Smart contract risk

Example: Economic exploit – flash attack

- Economic exploits become even trickier when considering that flash loans essentially allow any Ethereum user to become financially equipped for a single transaction.
- Special care must be used when designing protocols such that they cannot be manipulated by massive market volatility within a single transaction.
- An economic exploit which utilizes a flash loan can be referred to as a *flash attack*.

Risks: Smart contract risk

Example: Economic exploit – flash attack

- A series of high profile flash attacks were executed in Feb 2020 on bZx Fulcrum, a lending market similar to Compound.
- The attacker utilized a flash loan and diverted some of the funds to purchase a levered short position, with the remainder used to manipulate the price of the oracle exchange which the short position was based on.
- The attacker then closed the short at a profit, unwound the market trade and paid back the flash loan. The net profit was almost $300,000 worth of funds previously held by bZx, for near zero upfront cost.
- More examples will follow

Risks: Smart contract risk

The Attack

- A flash loan from dYdX for 10,000 ETH was opened.
- 5500 ETH was sent to Compound to collateralize a loan of 112 wBTC.
- 1300 ETH was sent to the Fulcrum pToken sETHBTC5x, opening a 5x short position against the ETHBTC ratio.
- 5637 ETH was borrowed and swapped to 51 WBTC through Kyber's Uniswap reserve, causing large slippage.
- The attacker swapped the 112 wBTC borrowed from Compound to 6871 ETH on Uniswap, resulting in a profit.
- The flash loan of 10,000 ETH from dYdX was paid back from the proceeds.

The total profit from this sequence of events was 1193 ETH, currently worth $298,250

https://bzx.network/blog/postmortem-ethdenver

Risks: Smart contract risk

Etherscan

Transaction Hash:	0xb5c8bd9430b6cc87a0e2fe110ece6bf527fa4f170a4bc8cd032f768fc5219838
Status:	Success
Block:	9484688 3470101 Block Confirmations
Timestamp:	535 days 20 hrs ago (Feb-15-2020 01:38:57 AM +UTC)
From:	0x148426fdc4c8a51b96b4bed827907b5fa6491ad0 (bZx Exploiter 1)

Risks: Smart contract risk

Etherscan

Transaction Action:

- Borrow 10,000 Ether From dYdX
- Supply 5,500 Ether To Compound
- Borrow 112 WBTC From Compound
- Swap 5,637.623762376237623786 WETH For 51.3455758 WBTC On Kyber
- Repay 10,000.00000000001 Ether To dYdX