DeFi-ing Cyber Attacks[*]

A statistical analysis of cybersecurity attacks in decentralized finance

Jack McKay

25 April 2022

Abstract

Decentralized finance (DeFi) describes the emerging ecosystem of financial services, protocols and applications built on and designed for public blockchains such as Ethereum and Algorand. One of the most promising applications of blockchain technology, DeFi has experienced explosive growth over the last few years. But with this growth has also come the attention of bad actors, who aim to exploit flaws in protocol security for personal gain. Using data collected by DEFIYIELD's REKT Database on the 200 most costly attacks on DeFi protocols, this paper will analyze and identify trends in these attacks, with the goal of identifying what types of attacks are most common and most costly to the DeFi sector as a whole, as well as for individual chains, in order to determine where cybersecurity efforts would best be focused.

1 Introduction

Imagine a world where financial services are available to anyone, from any location, 24 hours a day, 7 days a week, without the need for any intermediaries. One where financial transactions are fully automated, eliminating the need for any intermediary financial institutions, reducing human error and lowering transactions fees. All transactions are stored on a public, open-source ledger, allowing for complete transparency. Such a world is becoming increasingly feasible by the day, and the key to achieving it lies in decentralized finance (DeFi). DeFi is a completely open source and decentralized ecosystem running on blockchain technology that offers users financial services such as loans, payments, asset management and derivatives, as well as access to decentralized crypto exchanges (DEXes). DeFi minimizes–and in many cases eliminates– the need for centralized institutions such as banks or brokerages. Interactions on DeFi are peer-to-peer and are completed via smart contracts, self-executing computer programs functioning as contracts between parties that automatically execute when certain pre-set conditions are met. As such, DeFi aims to return control over individual's finances back to the individual, restoring freedom of choice to the financial industry. The near-unlimited potential of DeFi has not gone unnoticed, with the industry as a whole experiencing incredible growth. DeFi's current market cap is $147 billion as of April 2022, up from $2 Billion two years ago (CoinGecko (2022)), an annual growth rate of 757.32%. One of the primary drivers of this growth is a design principle known as "permissionless composability", wherein developers are able to employ any combination of pre-existing DeFi protocols, without requiring any permissions, in order to fulfill specific use cases. This allows developers to create and interact with limitless combinations of protocols, without any third party controlling any aspect of the network activity, creating a seamless and frictionless innovation cycle wherein users can build off of each others work. Indeed, permissionless composability is one of the fundamental innovations that has allowed DeFi to grow so quickly. However, this growth has not been without setbacks. One of the primary drawbacks to DeFi is its vulnerability to hackers. By nature of the way composability allows applications to work on top of each other, if the base chain was to suffer an attack, all of the applications built on top of it would also be at risk. As well, as transactions are executed automatically via smart contracts and without human oversight, this means that a bug in a smart contract can be exploited by hackers and used to reroute funds from the transaction. Due to these risks, identifying and preventing possible attacks is one of the

[*] Code and data are available at: https://github.com/jmacattack27

primary focuses of the DeFi community at the moment. Using data collected by DEFIYIELD and scraped by Octoparse as well as secondary data from DefiLlama, this paper will analyze data on the 200 costliest attacks in decentralized finance in order to identify which types of attacks are most common, most costly, and most dangerous to particular chains. This is done with the overarching goal of determining where DeFi cybersecurity efforts would best be focused on both an industry level and individual level.

2 Data

2.1 Data Source and Methodology

The primary data used in this report was collected by DEFIYIELD in their public Rekt Database, a database containing information on all recorded DeFi scams, hacks and exploits, including the total funds lost in each event, as well as a breakdown of the technical issues that led to the hack. Data on the 200 costliest attacks was then scraped from the DEFIYIELD's website via Octoparse, an a visual web data extraction software, before being downloaded into a .csv file, at which point it was uploaded to R, cleaned, and made ready for analysis. Further data on the total value locked (TVL) in each analyzed chain was obtained from DefiLlama's TVL database. This data consisted of 15 observations of two variables, Chain and TVL, and was manually acquired and included in the analysis.

2.2 Data Collection

The dataset used in this report contains information on the 200 costliest attacks in the DeFi space since the inception of cryptocurrency, with data on attacks as early as the Bitcoin Savings and Trust Ponzi scheme in July 2012, and as recently as the Elephant Money attack 2 weeks ago. The term 'attack' is used as a catch-all term to encompass any sort of malicious event occuring in the greater decentralized finance space in which funds were lost, with such events including hacks, rug pulls, exploits, dubious projects and exit scams.

This dataset is a subset of the greater Rekt Database, which at the time of writing holds information on 2780 attacks. Data was collected and recorded manually by the team at DEFIYIELD over the course of multiple months, before the database was published in August 30, 2021, from which point it has been continuously updated. While DEFIYIELD is constantly recording new attacks and updating their database themselves, individuals can also report a claim if they have been affected by an attack themselves, at which point an engineer at DEFIYIELD will work to verify the report. If the claim is corroborated, the engineer will then analyze activity on whichever chain the attack is reported to have occurred, and if the claim is proven to be valid, the attack is then added to the database. Each entry in the database contains the following details:

- The name of the exploited project & its associated URL
- The chain on which the attack occurred
- The date on which the attack occurred
- The type of malicious event: (Exit Scam, Flash Loan, Exploit, Abandoned, Bank Run, Honeypot, Access Control)
- The total funds lost in the attack

Malicious events are further defined as follows:

- Exit Scam: When promoters of a cryptocurrency or DeFi protocol vanish during or soon after the initial coin offering (ICO) for their product. Promoters will market and promote the currency or concept in order to raise money from investors, before abandoning the project and disappearing with said money.
- Flash Loan: A flash loan attack occurs when a bad actor manipulates the smart contract executing the flash loan in order to siphon funds to their own wallet. For example, a flash loan attack might involve the malicious party changing the values of the currencies being traded, in order to trick the smart contract into thinking the loan has been repaid when it hasn't.

- Exploit: Any sort of hostile attack on a DeFi service that exploits a vulnerability or oversight in the protocol code. Exploits can take many forms.
- Abandoned: When a project is abandoned by its developers. Abandoned coins are referred to as 'dead coins'.
- Bank Run: Similar to with traditional banks, a bank run in DeFi refers to when holders of a token rapidly withdraw their assets, causing the token price to drop and leading to a negative feedback loop wherein other holders panic-sell their tokens, further lowering the price, causing even more users to sell their tokens, and so forth. The most recent example of a DeFi bank run occurred with Iron Finance's TITAN token, which dropped from US$65 to US$0.000000035 over the course of a single day. Stablecoins, particularly algorithmically backed stablecoins, are especially vulnerable to bank runs.
- Honeypot: An attack in which attackers create and send out smart contracts that have an apparent vulnerability, but contain a hidden trap, such that when an unsuspecting user goes to exploit the apparent vulnerability in the contract, the trap activates and allows the attacker to siphon the victims funds to themselves.
- Access Control: A scam in which the attacker obtains access to a targets digital wallet or authentication keys.

Further data on the total value locked (TVL) across each chain was manually obtained from DefiLlama's public TVL database and used to provide context to certain aspects of the analysis. The data in this database is acquired from CoinGecko, and consists of 107 observations of 7 variables;

- Chain name
- The number of active protocols on the chain
- 1 day change in TVL
- 7 day change in TVL
- 1 month change in TVL
- Total Value Locked
- Market Cap / TVL

For our analysis we were are interested in the 15 chains included in our initial data set, and of those 15 chains we were only interested in the TVL, so the data pulled from this source consisted of 15 observations of 2 variables, chain name and TVL.

2.3 Data Analysis

This dataset scraped from DEFIYIELD's public Rekt Database via the data extraction software Octoparse (Team (2022)), before being imported to JupyterHub. Data was then processed and analyzed using the R programming language (R Core Team (2020)), as well as tidyverse (Wickham et al. (2019)), tidyr (Wickham (2022)) and dplyr (Wickham et al. (2021)) programming packages. The package janitor (Firke (2021)) was used to clean column names. gridExtra (Baptiste Auguie [aut (2017)) and reshape2 (Wickham (2007)) were used to format the graphs, while viridis (Garnier et al. (2021)) was used for data visualization, in particular modifying graph colours.

2.4 Strengths and Weaknesses

The data used in this study has both strengths and weaknesses. In particular, the database from which the data in this study was pulled is especially thorough in its technical breakdown of the attacks, including data on whether the attacked project's team is public, if the project had verified source code, as well as links containing proof of the attack and the wallets associated with the attack. The methods through which new data is added to the database are also thorough and involve multiple reviews of the incident, giving us confidence in the validity of the data. As well, the UI of the database website is user friendly, allowing parties to sort the data based on the chain on which the attack occurred, the type of attack, how much funds were lost and the date on which the attack occurred. The website also includes interactive graphs that help visualize the data. On the other hand, the website provides no option to download the database in its entirety or in any capacity, which leads to the main weaknesses of this data; the limited number of observations, and missed variables. As a result of being unable to download the database, I needed to use an external data extraction program, Octoparse, in order to download the data into a .csv file. While I was able to acquire a solid amount of observations on a number of useful variables, a lot of data was lost in the process. In particular, the number of attacks I could scrape was limited by the number of entries the REKT database could show on one page, which maxed out at 200. Luckily this was still enough data points to allow for meaningful statistical analysis, but it only accounts for 7.4% of the total data entries in the database. As well, due to the database's interface, Octoparse was unable to scrape certain variables, in particular the indicator variables: 'Public Team', 'Verified source code' and 'Audited code matches with deployed smart contracts code'. As well, due to the manner in which the technical breakdown of the attack was conveyed in the database, I was unable to use much of the detailed information in these breakdowns. Despite these weaknesses, the final data set retained much of the information on the attacks, and the most important variables were successfully captured.

data on attack type not as detailed as it could be

3 Results

3.1 Exploitation Rates by Chain

The first aspect of the data that I analyzed was which chains were being exploited most frequently. Summing the number of exploits on each chain into a new data frame, Figure 1 shows the distribution of attacks across all recorded chains. Here we see that the vast majority of attacks occur on either Ethereum (ETH) or Binance Smart Chain (BSC); 50.5% and 30.5% respectively. The remaining 19% of attacks are spread more evenly across the other chains, with Polygon, Solana, Fantom and Avalanche (AVAX) accounting for a combined 12.5% of total attacks. The remaining 6.5% is distributed evenly across the remaining chains.

Just looking at the number of attacks per chain doesn't give us the full picture on their security. There are numerous factors that would affect the number of attacks attempted on a given chain. One such factor is the varying total value locked (TVL) in each chain. The more value locked in a chain, the more liquidity exists to be exploited; as such, larger chains like Ethereum are naturally subject to more attacks. Using data from DeFi Llama (DefiLlama (2022)) on the total value locked across each chain, Figure 2 shows the distribution of the number of exploits on each chain, divided by the TVL (in billions) in that chain. This gives us a more realistic picture of the rate at which each chain is being targeted. Note that this figure includes two graphs: one containing data on Polkadot, and one without data on Polkadot. Due to the relatively tiny amount of value locked in the Polkadot chain (about $3.84 million), the exploits to TVL ratio for Polkadot was more than 31x greater than the next highest ratio. As a result, the scale of the graph is disproportionately affected making it far less interpretable. As such, a second graph is included that omits data on Polkadot. Here we see a wildly different distribution than in the previous figure, with Ethereum going from the by far the most exploited chain to the third least exploited, while chains such as Algorand (ALGO), Heco, RONIN and EOS all saw substantial increases in their adjusted attack rates. Cronos and Tron remain the least exploited, while Avalanche (AVAX) also saw a noticeable decrease in exploit rates relative to the other chains.

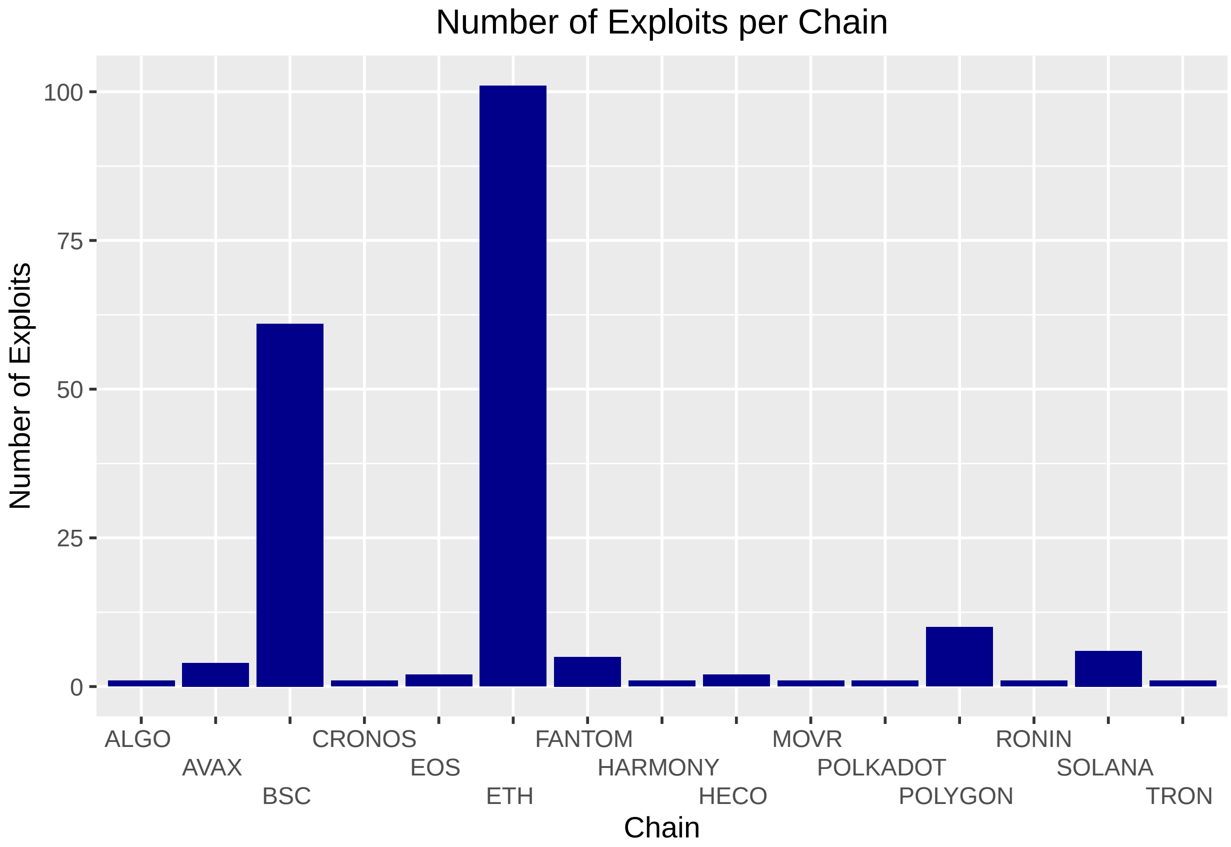
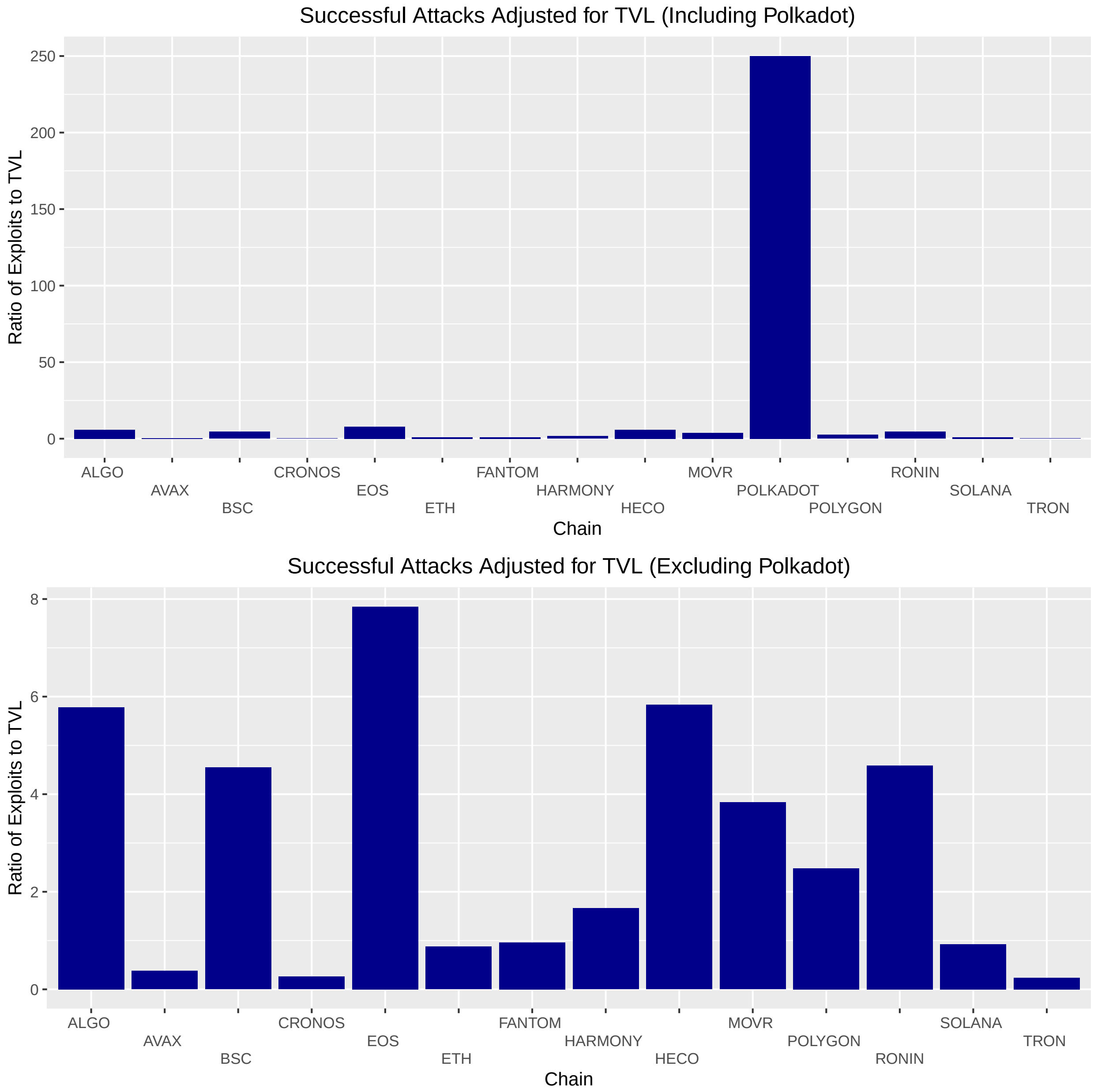


Figure 1: The Distribution of Attacks by Chain

3.2 Attack Rates by Type of Attack

The next aspect of the data I analyzed was the distribution of attack data, in terms of what types of attacks are most common, which types have historically resulted in the most damage, and which types are most damaging on average. Figure 3 shows the distribution of attack types, based on frequency, total damage and average damage. Here we see that exploits are by far the most prevalent types of attacks across all metrics, accounting for 45.5% of attacks, making exploits more than twice as common as any other attack, and almost 3 times more costly on average. We also notice that despite being the second most common attack at 21%, exit scams are only the 4th costliest attacks on average, with both access control and flash loan attacks costing more on average. Abandoned projects are by far the least common attack, and are only slightly less costly than honeypot attacks, the second least damaging attack.

Furthering my analysis into attack rates, we next look at the rate of all attacks recorded in the REKT database, as opposed to just the 200 costliest. Figure 4 shows the distribution of all attacks registered in the REKT database. Here we see a wildly different distribution than previously, with honeypots accounting for 66.6% of all recorded DeFi attacks compared to 2.5% of the 200 costliest attacks. Exit scams remain the second most frequent attack at 21.8% while exploits now account for only 6.3% of all attacks compared to 45.5% previously, marking a 622% drop in frequency. Figure **??** puts all of this data into a table, including a column “Likelihood”, which describes the likelihood that an attack of a particular type is one of the 200 costliest attacks of any type. Here we see that flash loans are by far the most likely to be among the 200 costliest attacks, with 36 of the 42 total recorded flash loan attacks being among the 200 costliest, or about 85.7%. Exploits are the second most likely with about 50.5% of all exploits being among the 200 costliest. In stark contrast, we note that honeypots have by far the lowest chance at 0.3%, about 22.3 times less likely



Successful Attacks Adjusted for TVL (Including Polkadot)

Successful Attacks Adjusted for TVL (Excluding Polkadot)

Figure 2: The Distribution of Attacks Adjusted for Total Value Locked

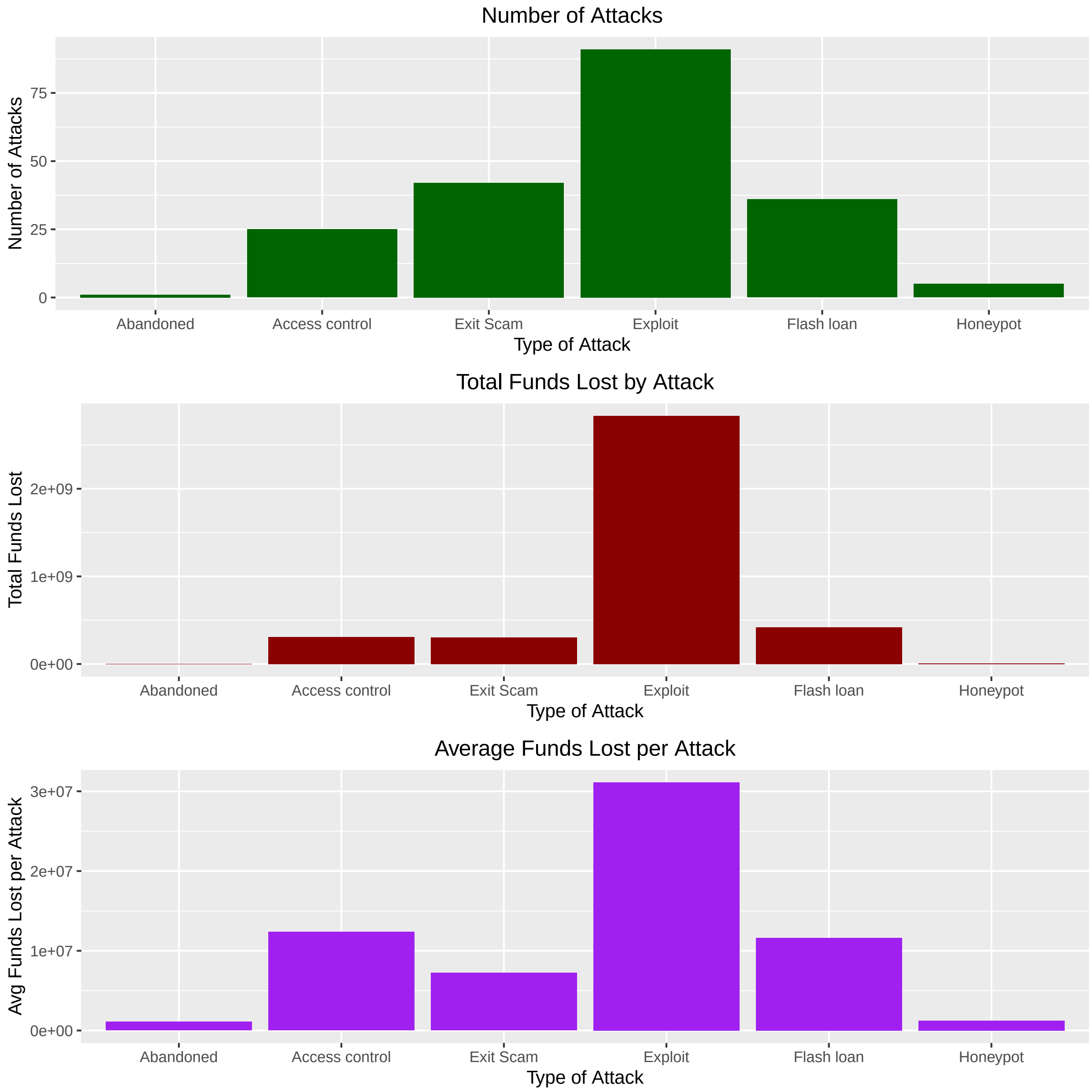


Figure 3: Distribution of Attack Type by Quantity, Net Loss and Average Loss per Attack

than the second least likely attack, exit scams.

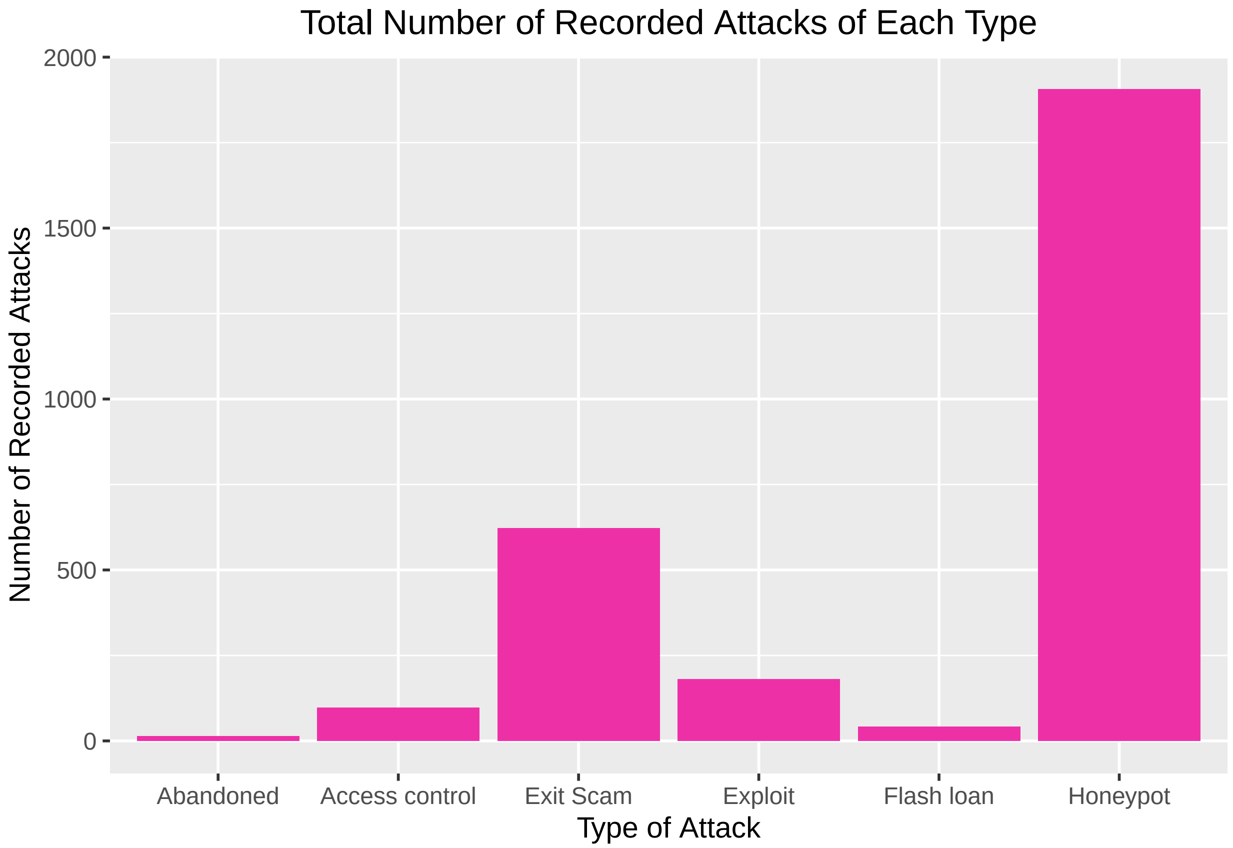


Figure 4: Distribution of All Recorded Attacks

The next aspect of the data I analyzed is how the frequency of each type of attack is changing over time. Figure 5 shows the number of each type of attack year over year since 2016. Here we see that the every type of attack increased in frequency in 2021, with exploits experiencing by far the greatest spike. As well, we see proportionally more attacks per month in 2022 than we did in 2021, indicating that attacks are becoming both more common and costlier.

Table 1: Key Data on Attacks

Type of Attack	Attacks (Top 200)	Net Funds Lost (Top 200)	Average Funds Lost (Top 200)	Total Recorded Attacks	Likelihood
Abandoned	1	1099672	1099672	14	7.1%
Access control	25	309225839	12369034	97	25.8%
Exit Scam	42	303791970	7233142	623	6.7%
Exploit	91	2832850730	31130228	180	50.6%
Flash loan	36	418070205	11613061	42	85.7%
Honeypot	5	6182504	1236501	1907	0.3%

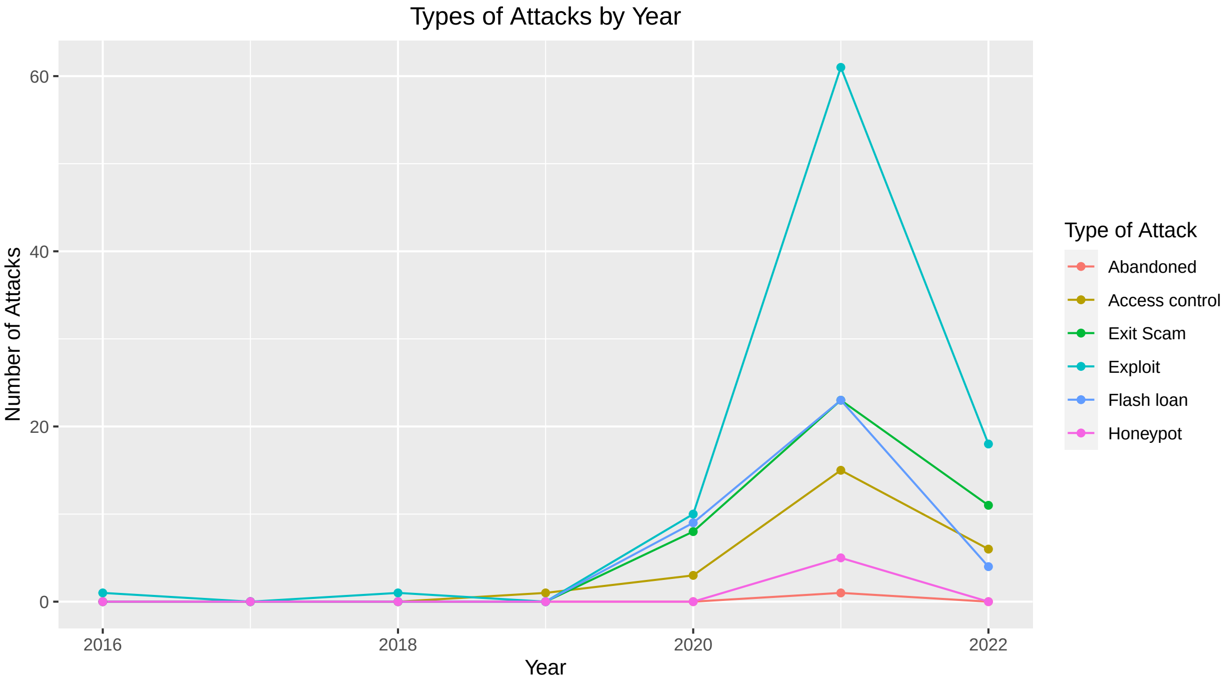


Figure 5: Number of Attacks per Year

3.3 Attack Data by Chain

The final aspect of the data I will analyze is the attack data for individual chains. Of the chains in our data set, only Ethereum, BSC and Polygon have witnessed enough attacks for the data to be statistically significant and interesting. Figure 6 shows the number of attacks on these chains, broken down by type of attack, so that we may see if certain chains are particularly vulnerable to certain types of attacks. Here we see that exit scams account for a noticeably smaller proportion of attacks on Ethereum compared to other chains, with BSC experiencing 38% more exit scam attacks in total than Ethereum, despite experiencing 39.6% less attacks in total. Exit scams accounted for 50%, 42.9% and 29.5% of the attacks on Solana, Polygon and BSC respectively, compared to only 13% of attacks on Ethereum, indicating that Ethereum is particularly secure against exit scams. On the other hand, we see that Ethereum is disproportionately affected by access control attacks, which account for 16.2% of attacks on ETH, compared to only 6.9% of attacks on BSC, and 0% of the attacks on Solana or Polygon. As well, ETH is the only chain to have experienced a honeypot exploit, which account for 5.1% of ETH attacks. Interestingly, flash loans account for a very similar proportion of attacks on ETH, BSC and Polygon, at 19.2%, , 21.3% and 28.6% respectively, while not Solana did not experience any such attacks.

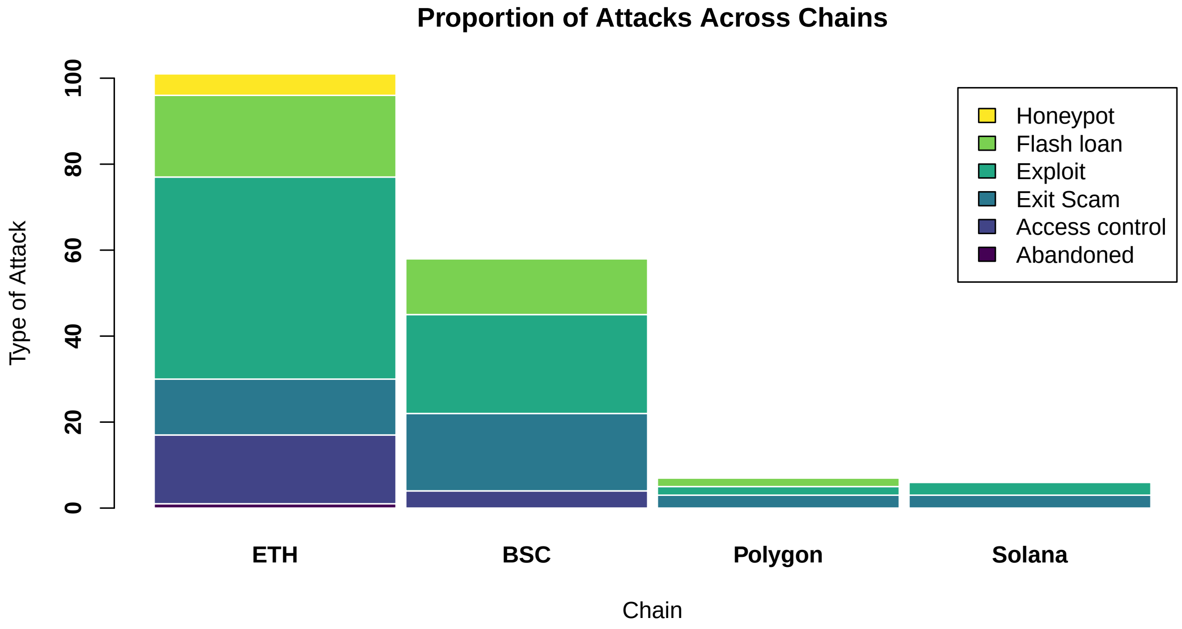


Figure 6: Number of Attacks per Chain by Attack Type

4 Discussion

4.1 Attack Prevalence and the Danger of Exploits

By far the most prominent type of attacks in our data are exploits, which showed to be the most frequent, cause the most damage in total, and cause the most damage on average. Of the 10 costliest attacks, exploits accounted for 5, including each of the 3 costliest attacks. There are multiple reasons why this is the case. First, exploit is a very broad term that encompasses any attack resulting from a software vulnerability or security flaw. This means that specific exploits such as Sybil attacks and 51% attacks are both categorized as exploits in this data set. As well, there are a multiplicity of avenues that one could use to exploit a protocol, such as the exchange a token is traded on, the smart contracts dictating trades, exploiting an oversight in the chain that the protocol runs in, or an oversight in the protocol code itself. As such, it is expected that exploits would be the most common types of attack, as it is an umbrella term. This does not, however, explain why exploits result in substantially more funds being lost on average than any other attack. An exploit will on average result in more than twice as many lost funds as the second costliest type of attack, access control. In fact, if we sum the average cost of every other type of attack, the resulting sum is still only 7.7% more costly than the average exploit. There are a number of reasons why this could be the case. The primary

Attacks such as honeypots

Initial results show that some chains are more far more likely to be attacked than others.

susceptible to certain types of attacks than others, and that attacks are only growing in frequency. As well, we see that certain attacks are far more dangerous than others. In particular, exploits are by far the most frequent type of attack across almost all of the analyzed chains, and account for the vast majority of the total funds lost; 71.3% of the net funds lost from these 200 attacks, or 2.73 times the funds lost from all other types of attacks combined. As well, exploits are more than twice as costly on average than the next most expensive attack, access control. In particular,

4.2 Second discussion point

4.3 Third discussion point

4.4 Weaknesses and next steps

Weaknesses and next steps should also be included.

A Appendix

A.1 Key Terms

Below is a alphabetized list containing key terms used throughout this paper and their definitions.

- Bad Actor: An entity that aims to circumvent security protocols and exploit projects and/or individuals for personal gain.
- Smart Contract: A self-executing agreement, written in lines of code, that automatically executes when predetermined conditions are met. Smart contracts allow for agreements to execute instantaneously, without the involvement of intermediaries, and such that all participants can be certain of the outcome. Smart contracts are one of the fundamental building blocks of decentralized finance.
- Stablecoin: A digit asset engineered to maintain a stable value relative to some national currency or other value-based asset. A stablecoin that is designed to replicate the value of another asset is considered "pegged" to that asset. Maintaining its peg to its associated asset is one of the primary focuses of stablecoins, which can be further classified based on the method used to maintain their peg:
  - Collateralized Stablecoins: Achieve price stability through holding reserves of fiat currencies equal to or greater than the market cap of the coin. Currently the most popular type of stablecoin, with examples including Tether (USDT) and USD Coin (USDC).
  - Algorithmic Stablecoins: A stablecoin model involving 2 tokens: a stablecoin and a token that shares in the system's profits from new issuance of stablecoins. Shares in the latter token are issued to holders of the former, and allow for developers to maintain the stablecoin's price by controlling the supply, without harming holders.
- Flash Loan: A form of uncollateralized lending executed via smart contract that allows users to borrow any available amount of assets without posting any collateral. Instead, the liquidity from a flash loan must be instantly repaid to the lender within a single block transaction. If the borrower doesn't repay the capital, the transaction is instantly reversed.

B Additional details

References

Baptiste Auguie [aut, Anton Antonov [ctb], cre]. 2017. *GridExtra: Miscellaneous Functions for "Grid" Graphics.* https://CRAN.R-project.org/package=gridExtra/.

CoinGecko. 2022. *Top 100 Defi Coins by Market Capitalization.* https://www.coingecko.com/en/categories/decentralized-finance-defi.

DefiLlama. 2022. *DefiLlama - Total Value Locked All Chains.* https://defillama.com/chains.

Firke, Sam. 2021. *Janitor: Simple Tools for Examining and Cleaning Dirty Data.* https://cran.r-project.org/package=janitor.

Garnier, Simon, Ross, Noam, Rudis, Robert, Camargo, et al. 2021. *viridis - Colorblind-Friendly Color Maps for R.* https://doi.org/10.5281/zenodo.4679424.

R Core Team. 2020. *R: A Language and Environment for Statistical Computing.* Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Team, The Octoparse. 2022. *Octoparse: Web Scraping Tool & Free Web Crawlers.* https://www.octoparse.com/.

Wickham, Hadley. 2007. “Reshaping Data with the reshape Package.” *Journal of Statistical Software* 21 (12): 1–20. http://www.jstatsoft.org/v21/i12/.

———. 2022. *Tidyr: Tidy Messy Data.* https://tidyr.tidyverse.org.

Wickham, Hadley, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D’Agostino McGowan, Romain François, Garrett Grolemund, et al. 2019. “Welcome to the tidyverse.” *Journal of Open Source Software* 4 (43): 1686. https://doi.org/10.21105/joss.01686.

Wickham, Hadley, Romain François, Lionel Henry, and Kirill Müller. 2021. *Dplyr: A Grammar of Data Manipulation.* https://CRAN.R-project.org/package=dplyr.