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 transaction 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. Some of the variable values are rather broad, such as exploit, which encompasses a large variety of difference types of attacks; further dividing exploits into more defined categories would help with statistical analysis. Despite these weaknesses, the final data set retained much of the information on the attacks, and the most important variables were successfully captured.

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.

Figure 2 shows the distribution of all recorded attacks on each chain, subcategorized into attacks that are among the 200 costliest and attacks that are not. Here we see that BSC and ETH are still by far the most attacked chains, but that BSC has been successfully attacked far more times than Ethereum, with BSC accounting for exactly 60% of all recorded attacks, while Ethereum accounts for 34%. This result, as well as a visual inspection of the graph, indicates that attacks on Ethereum are in general far more costly than attacks on BSC, and are also less frequent. The only other chain with a statistically significant amount of attacks is Polygon, with 94 recorded attacks, 10 of which are among the 200 costliest. Excluding attacks on BSC and ETH, Polygon accounts for 68.1% of all attacks, 5% of attacks in the top 200 costliest and 82.4% of all other attacks.

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 3 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.

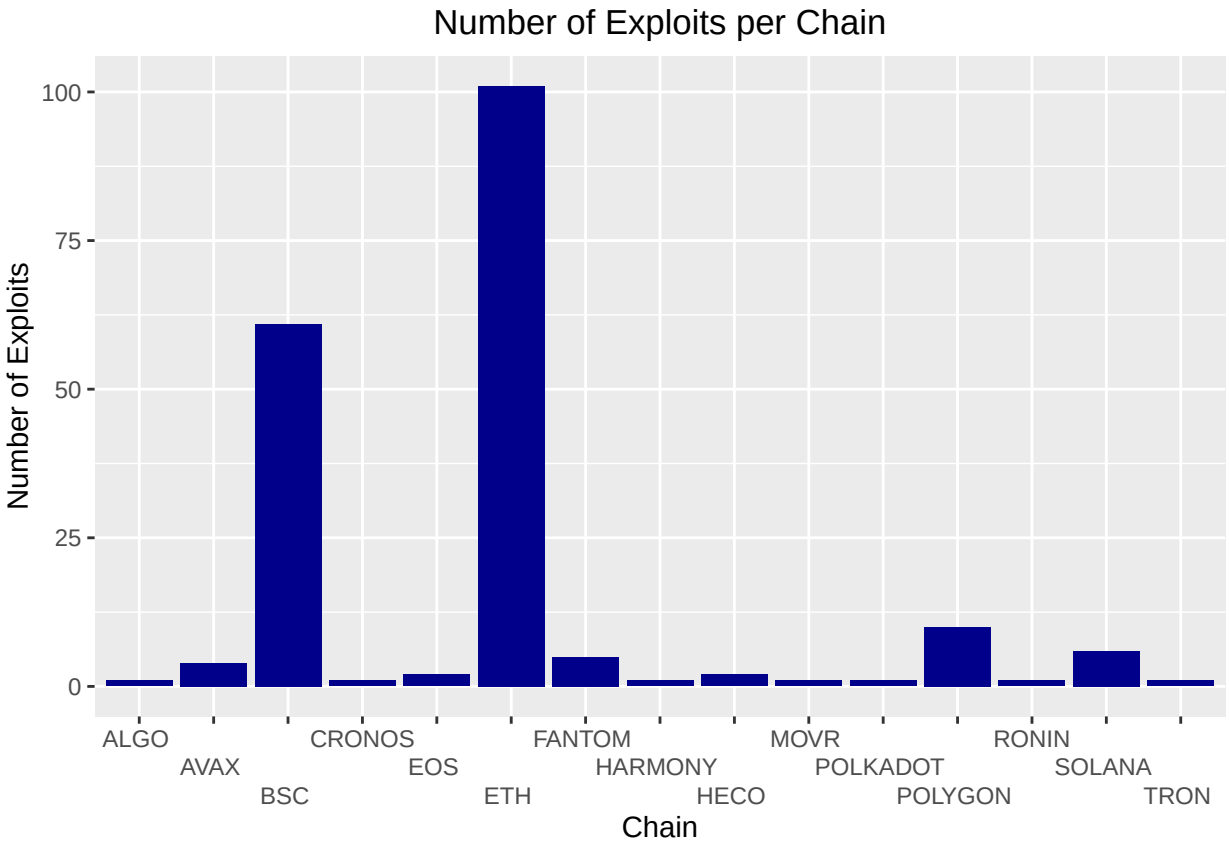

Figure 1: The Distribution of Attacks by Chain

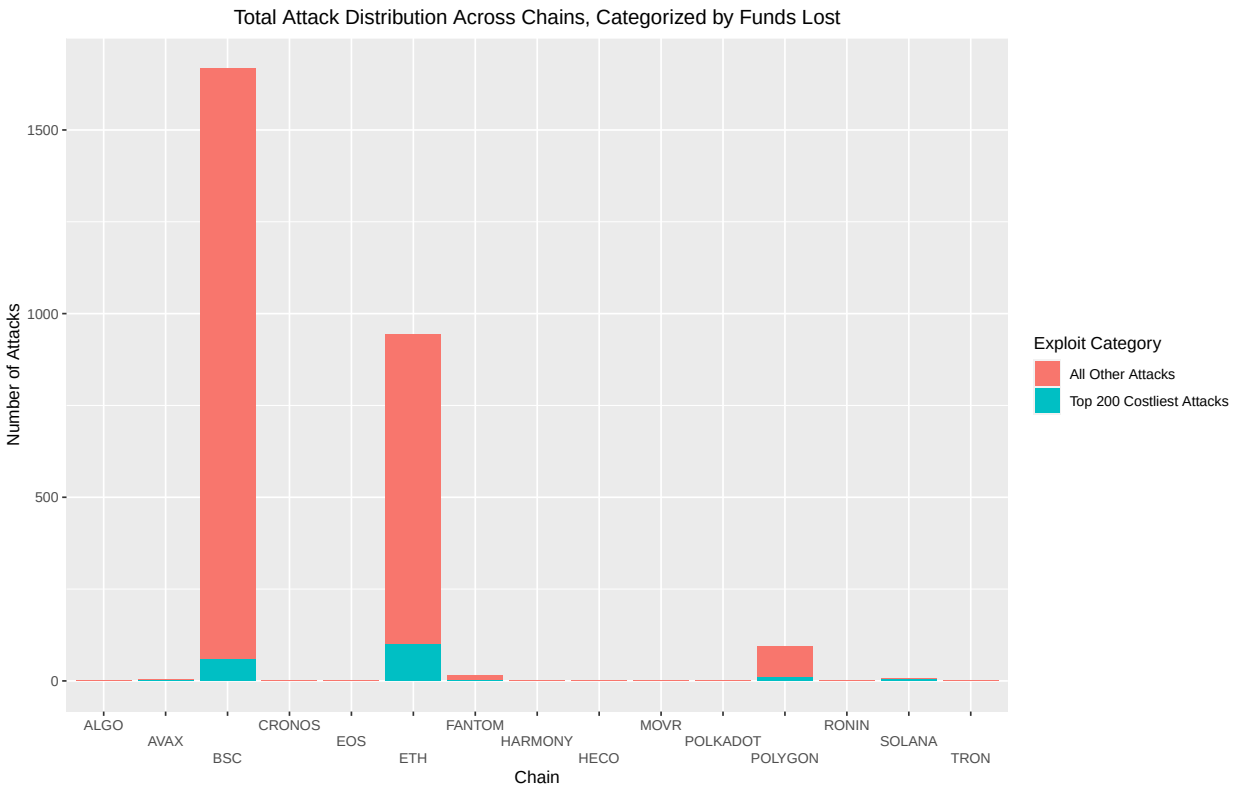

Figure 2: Number of Attacks by Chain

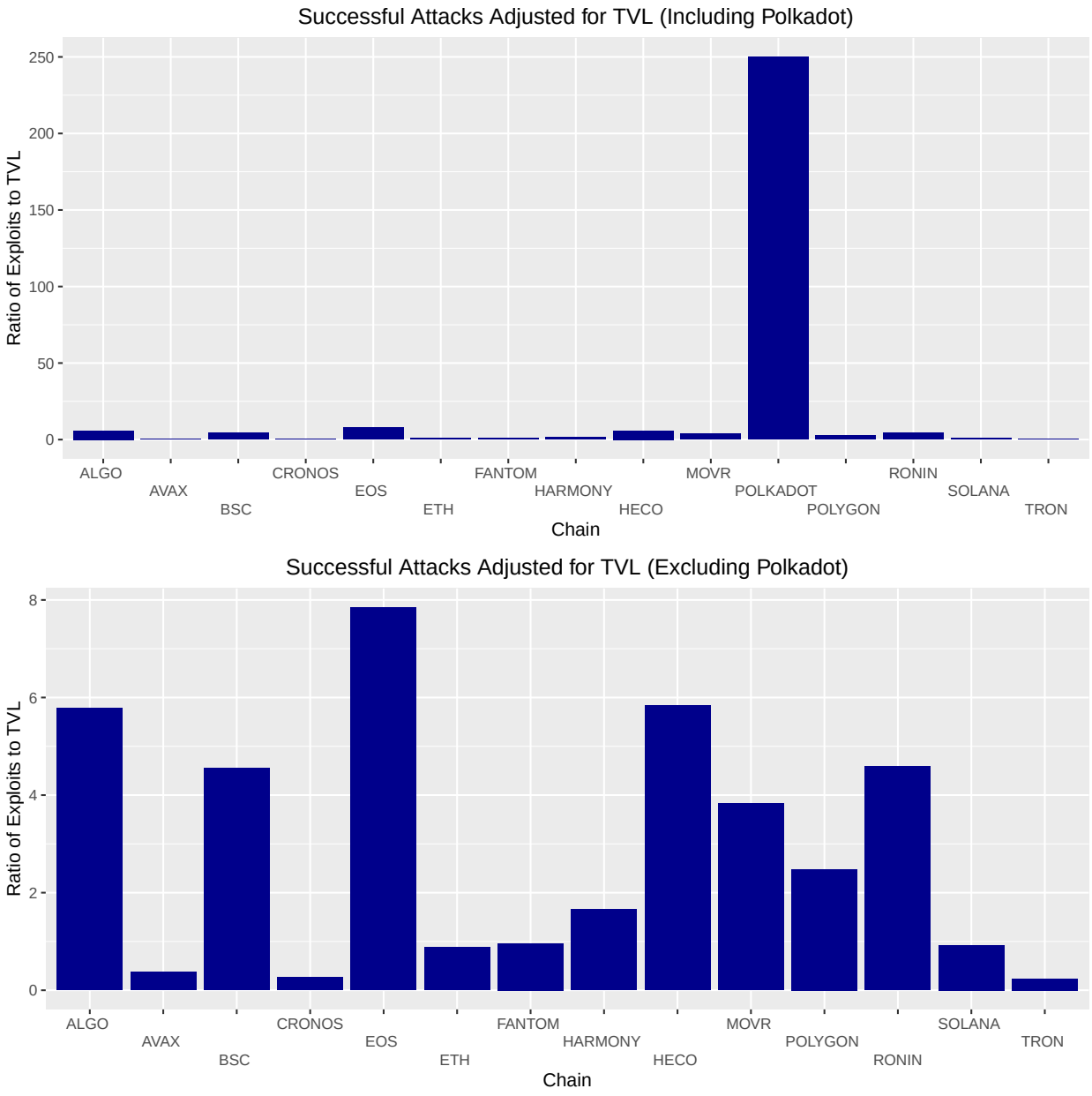

Figure 3: The Distribution of Attacks Adjusted for Total Value Locked

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 4 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.

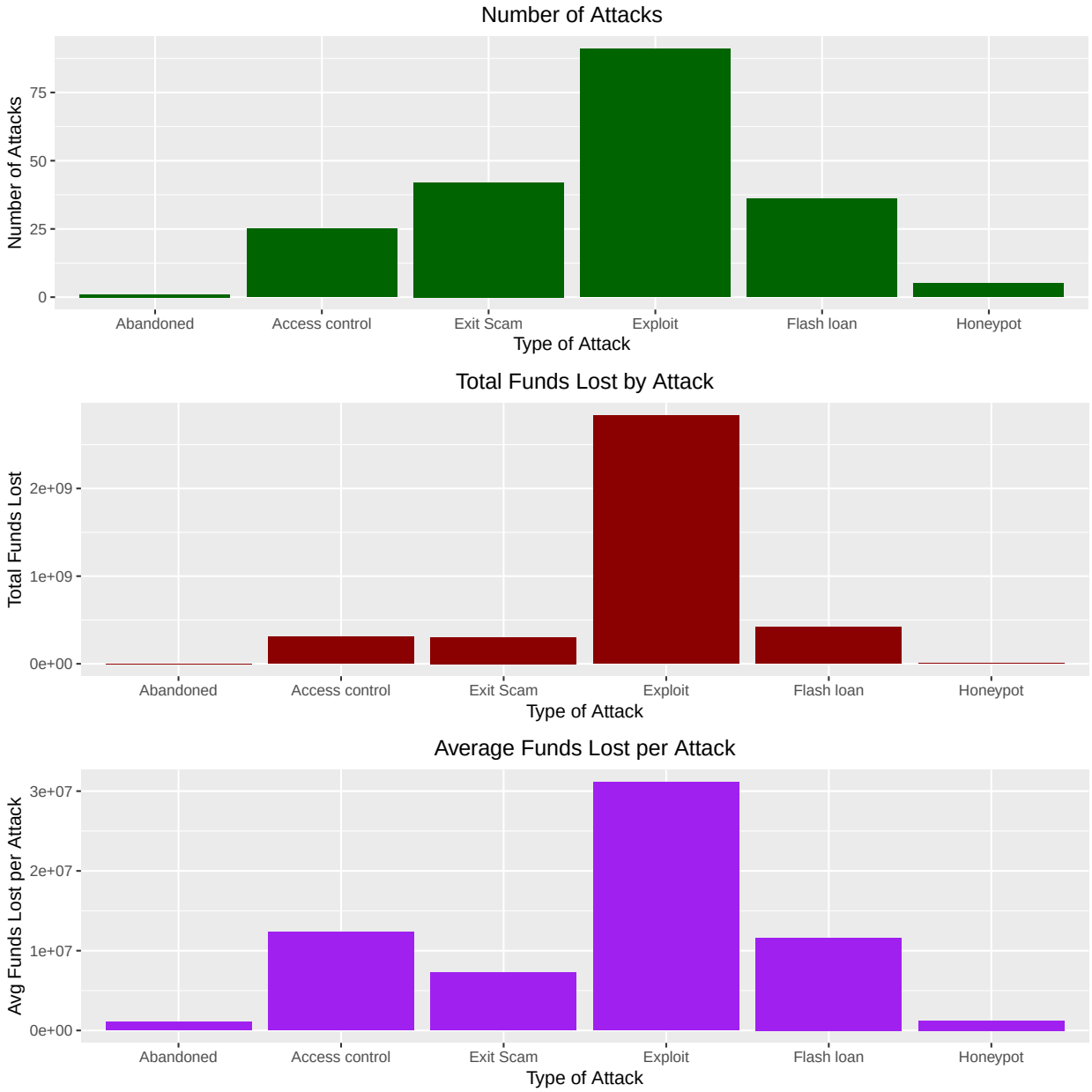

Figure 4: Distribution of Attack Type by Quantity, Net Loss and Average Loss per Attack

Furthering my analysis into attack rates, we next look at the total number of attacks recorded in the REKT database. Figure 5 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. 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 than the second least likely attack, exit scams.

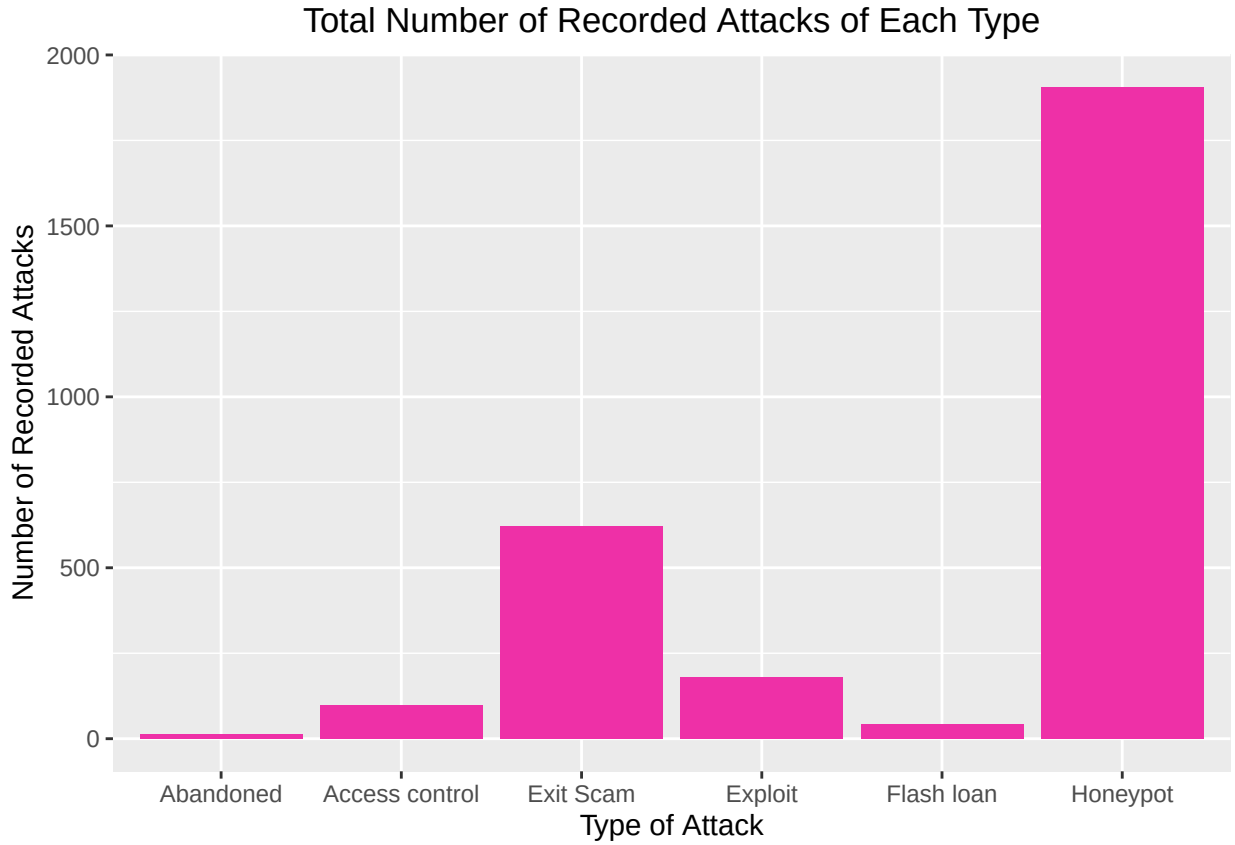

Figure 5: Distribution of All Recorded Attacks

The next aspect of the data I analyzed is how the frequency of each type of attack is changing over time. Figure 6 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%

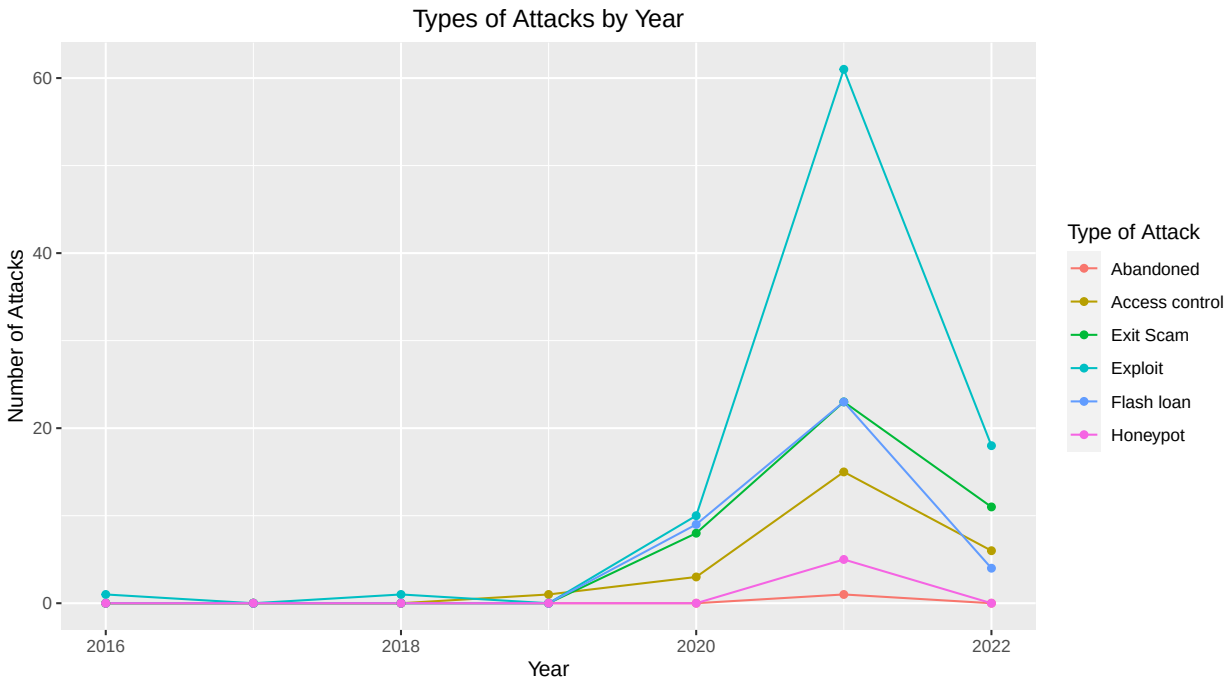

Figure 6: Number of Attacks per Year

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 7 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.

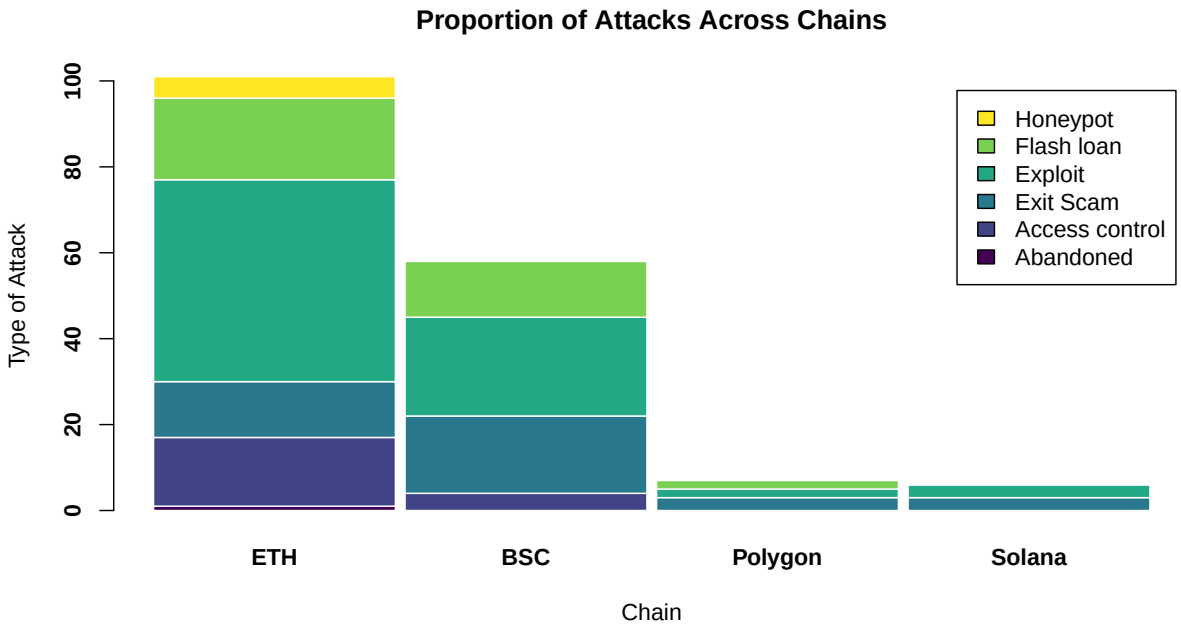

Figure 7: Number of Attacks per Chain by Attack Type

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. These results changed when looking at the total number of attacks as opposed to just the top 200 costliest, where we saw exploits fall from by far the most frequent attack (45.5%) to the 3rd most frequent attack (6.3%). This supports the earlier finding that exploits are by far the costliest attack on average, and indicates that while exploits are harder to find than originally thought, the reward for finding them (or cost for having them) is huge.

4.2 Flash Loans

Of the analyzed attack data, one of the most notable results was the rate at which flash loan attacks end up among the 200 costliest attacks; despite accounting for only 1.5% of all recorded attacks, 85.7% of all flash loan attacks were among the 200 costliest attacks, by far the greatest likelihood of any attack. Flash loan attacks occur involve taking out a flash loan from a lending protocol, arbitraging the borrowed money, and then manipulating the price of the one or multiple tokens in order to turn a profit, before returning repaying the loan. Flash loan attacks are particularly costly for a number of reasons. All flash loans are executed via smart contract, meaning that flash loans are entirely automated. By nature of the instantaneous execution of flash loans, attackers are able to both repeat the process multiple times, as well as execute it across multiple markets. Not only this, but because the market manipulation is performed with loaned assets drawn from a general lending pool, returning these assets to the pool effectively washes the attacker's hands of the attack and eliminates much of the trace, making flash loan attacks especially low risk for attackers. Somewhat ironically, the key factors that make flash loans so innovative–anonymity, no collateral and instantaneous execution–are the same things that make them so costly when exploited.

4.3 Overall Chain Security

In our initial graph of the distribution of attacks across chains, we saw that the vast majority of attacks occured on BSC or ETH. However, this graph is misleading with regards to its implications for chain security; because we are looking at the distribution of the 200 costliest attacks, chains with a larger userbase such as ETH and BSC are naturally more likely to record more such attacks. Thus, we also graph the distribution of all attacks in Figure 2, with each attack being categorized as either in the top 200 or not. Here we see that BSC has experienced by far the most successful attacks, followed by Ethereum, with a substantial drop off between Ethereum and the next most attacked chain, Polygon. When we adjust for the size of the chain by taking the ratio of attacks to total value locked in Figure 3, we see that EOS is now the most exploited chain, with a Attack to TVL ratio of 7.83, compared to 0.874 for ETH and 4.54 for BSC. Notably, ETH saw the sharpest drop relative to the other chains when accounting for TVL, dropping from by far the most attacked chain to the 11th most. This indicates that relative to its size, Ethereum is actually one of the most secure chains that we analyzed. Furthermore, in reality Ethereum is even more secure than this metric indicates. If, for example, instead of considering total value locked in a chain in terms of dollars we consider the total value locked in a chain as a percentage of the total value locked across all chains, then we would see that while TVL in Ethereum currently accounts for 54.81% of TVL across all chains, it accounted for 72.95% as recently as last year (DefiLlama (2022)). If we go back a couple months further to January 2021, Ethereum accounted for a whopping 97.18% of all the value on chains, completely dominating the space. In Figure 6, we see that a significant portion of the 200 costliest attacks occurred in and before 2021. This means that while Ethereum has experienced the most costly attacks and the second most attacks overall, it has been by far the biggest target of any chain since our earliest data points, and while relative to its current TVL it appears to be the 4th most secure chain, in reality it is likely far more secure than this metric indicates.

4.4 Chain Security to Specific Attacks

In Figure 7 we see the distribution of attacks on Ethereum, BSC, Polygon and Solana, categorized by type of attack. Here we note that Ethereum appears particularly secure against exit scams, which account for only 12.87% of attacks on Ethereum, compared to 29.51% of attacks on BSC, 42.8% of the attacks on Polygon and 50% of the attacks on Solana, however the latter 2 statistics are less meaningful due to the far smaller sample size. An exit scam involves heavily promoting a project in order to drive investment from individuals or companies, before running away with investor's money. By their nature, exit scams are more about manipulating individuals than they are exploiting loopholes or technical oversights. As such, the best defense against exit scams doesn't involve cybersecurity so much as it does educating your userbase about exit scams and how to spot them. A brief scroll through Ethereum's website reveals the strides that Ethereum has made in the realm of user education, with a dedicated "Learn" section containing guides on everything to do with Ethereum, from smart contracts to consensus mechanisms, including a section titled "Ethereum security and scam prevention" which outlines basic safety practices as well as common scams and how to spot them. At the bottom this section is a list of links for further reading, including multiple links on web security, crypto security and indeed scam education. Of all the chains, Ethereum has by far the greatest wealth of information resources on everything to do with crypto, as well as specific sections dedicated to scam detection and prevention. Needless to say, this would almost invariably play a factor in the distinctly low rate of exit scams on Ethereum.

Another notable result is the disproportionate rate at which Ethereum appears affected by access control scams, which involve a hacker gaining access to one or multiple user's wallets and siphoning user funds to their own account. At first glance this seems to contradict the previous result about Ethereum's successful efforts in educating its users; after all, gaining access to a users wallet involves getting the users key, which usually occurs as a result of the user inadvertently exposing his own key. Thus, access control scams would appear to be another type of scam wherein the individual user is most often at fault, rather than a coding oversight. However, reading through the technical breakdown of some of the costliest access control scams on the REKT database shows that this is not always the case. For example, the costliest recorded access control scam occurred back in December 2021, when gaming and NFT platform Vulcan Forged was hacked, exposing 96 high value wallets. Vulcan Forged offers users a wallet service known as MyForge, which allows users to manage their wallets through Vulcan's platform. The passwords to users wallets were co-managed by both the user and Vulcan, with Vulcan storing user's private keys on their own side. Thus, when Vulcan was hacked, hackers gained access to 96 user's wallets, stealing 4.5 million PYR tokens (Vulcan's native token), or about 23% of the circulating supply (Chawla (2021)), roughly equivalent to $140 million. Because Vulcan Forged is built on Ethereum, and the lost of funds occurred due to hackers gaining access to user's private keys, this attack is classified as an access control attack on Ethereum, despite no wrongdoing or bad security practices on behalf of the users. Many of the other Ethereum access control hacks also occur in a similar method, where a protocol built on Ethereum is hacked and user's private keys are exposed. This is indicative of one possible reason for the prevalence of access control hacks on Ethereum; there are far more projects built on Ethereum than on any other chain, and of the projects built on other chains, many of them are also launched on Ethereum as well. DeFi Prime, a digital media and analytics service build for the DeFi community, has a total of 225 DeFi projects listed in their DeFi ecosystem, of which 203 are built on Ethereum, compared to 43 built on BSC (Sawinyh (2022)). Simply put, this means that there are there more avenues for hackers to gain access to user passwords on Ethereum than on other chains.

4.5 Weaknesses

There were a number of difficulties encountered while writing this report, specifically relating to data collection and formatting. In particular, the inability to download the REKT database made acquiring the data particularly tedious, and required the use of an external web data extraction software in Octoparse. While I successfully downloaded data from the database, due to limitations with the extraction software, I could only download as many data entries as could be displayed on a single page, 200 in this case. While this was enough data for a statistically meaningful analysis, it would have been incredibly insightful to have the roughly 2500 excluded data points. Despite this limitation, I was able to manually input a fair amount of data which was useful in providing additional context to the original data set. However, this in and of

itself could be seen as a weakness, as this additional data is then subject to human error. As well, the format in which the data was downloaded made data visualization particularly cumbersome, with me often having to create entirely new data frames for each graph or set of graphs, making the data exploration especially complicated.

Another weakness of my analysis pertains to the data itself, and the lack of specificity seen in some of the variables, specifically when talking about the types of attacks seen. While there is a solid breakdown of the various types of attacks, it would have been insightful if some of the values–specifically exploit–were further broken down based on what aspect of the chain or project was exploited. As it stands, exploit acted as sort of an umbrella term for a number of possible issues, which made it harder to perform a more technical analysis of the attacks and led to some overcentralization in the attack data of our initial data set. The manual addition of data on all recorded attacks helped mitigate some of this overcentralization by providing further context to the data.

The final weakness of my report is that I did not answer all of my intended research questions, specifically what may precipitate an attack. Due to the lack of detailed technical data on each attack, there was simply not enough information for me to properly investigate this question. Still, the majority of my research questions were answered.

5 Conclusions and Going Forward

There are a number of conclusions that can be drawn from this data. In particular, we see that honeypots are by far the most frequent type of attack across all analyzed chains, but comparatively result in far less damages than all other attacks, besides abandoned projects. The prevalence of honeypots indicates that more effort should be put into mitigating them. However, honeypots also present a unique opportunity, with the ability to research and analyze the behaviour of would-be attackers. Honeypots deployed for this reason are referred to as 'research honeypots', and can be used to proactively develop preventative defenses against hostile attacks. For this reason, as well as the low average cost of a honeypot attack, they are not a top priority in terms of attacks to defend against. Rather, that honor goes to exploits, which account for the vast majority of the top 200 costliest attacks, and are by far the most damaging attacks on average. However, because exploits can occur as a result of an oversight in the code associated with a project, chain or smart contract, they are also the hardest type of attack to combat. As well, because exploits are essentially the result of a coding error, the main defense against them would simply be to invest more money into software engineers and computer scientists to audit and inspect a projects code in order to proactively find errors before attackers do. For a given project, this could take the form of a designated team whos sole objective is to try and find exploits in the projects associated code. With regards to specific chains, we conclude that of chains with a meaningful amount of data, Ethereum reigns supreme in terms of security, while BSC appears to be less secure compared to other chains. In particular, Ethereum is especially secure against exit scams, and this is likely due in part to their extensive efforts in educating their userbase, which has invariably helped prevent other types of attacks as well. In this regard, BSC, Polygon and Solana all should invest further in user education. As the decentralized finance ecosystem continues to rapidly evolve and grow, so too will the attempts to exploit this sector for personal gain, and it is through consistent and concerted investment into cybersecurity and user education that we can work towards bulletproofing this blossoming technology so that it may reach its full potential.

A Appendix

A.1 Key Terms

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

- Abandoned: When a project is abandoned by its developers. Abandoned coins are referred to as 'dead coins'.
- Access Control: A scam in which the attacker obtains access to a targets digital wallet or authentication keys.
- Bad Actor: An entity that aims to circumvent security protocols and exploit projects and/or individuals for personal gain.
- 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.
- DeFi: Refers to decentralized finance, an ecosystem of financial services, products and applications that can be accessed anonymously, are decentralized and are built on and for public blockchains, such as Ethereum.
- Blockchain / Chain: A decentralized database that stores encrypted blocks of data and chains them together to form a chronological ledger for the data. Blockchains are the fundamental innovation upon which the decentralized finance industry is built.
- 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.
- 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.
- 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.
- 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.
- 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.

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/.

Chawla, Vishal. 2021. https://cryptobriefing.com/nft-marketplace-vulcan-forged-hacked-for-140m/.

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/.

Sawinyh, Nick. 2022. *Ethereum Defi Ecosystem*. https://defiprime.com/ethereum/.

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.