BIS Innovation Hub

Project Polaris

Part 3: Closing the CBDC cyber threat modelling gaps

July 2023

In parternship with:

BIS Cyber Resilience Coordination Centre

This paper was quality assured by PA Consulting:

1. Executive summary

Decentralised finance (DeFi) continues to revolutionise the financial industry. It has been argued that DeFi started in 2009 with the launch of Bitcoin, the first peer-to-peer digital asset to use blockchain technology. In 2015, Ethereum was launched and with it came the popularisation of smart contracts.[1] DeFi can be defined as an umbrella term for an eclectic mix of blockchain technology, digital assets, decentralised applications (dApps), and distributed ledgers (DLT). There are many products and services that adopt this technology including crypto currency and stablecoin exchanges, derivatives, credit, and insurance services.

Cryptocurrencies in particular have enjoyed a remarkable adoption rate due to their accessibility and low transaction fees, all without the need for intermediaries as in the case of traditional banks. There is also a growing acceptance from businesses and individuals to accept cryptocurrencies as a form of payment. However, cryptocurrencies can be volatile, often lacking coherent regulations, governance, and government support. Perhaps most worrisome are the known and unknown security vulnerabilities, which are unique to this ecosystem and stem from the use of novel technology and the lack of verified secure designs and implementations.

As part of addressing these concerns, many central banks are interested in developing central bank digital currencies (CBDCs) as an alternative to private cryptocurrencies. CBDCs are simply a digital form of fiat currencies. Although often using the same technology as cryptocurrencies, they are backed by a central bank. CBDCs offer the promise of a more secure and stable digital currency that could also support financial inclusion and reduce cash usage whilst allowing for more efficient, faster, and cheaper transactions, as compared with traditional banking systems. In contrast to private cryptocurrencies that aim to maximise profits, CBDCs are an alternative that serves the needs of the public.

At the end of 2022, there were 3 launched CBDC implementations around the world, with several other pilots at varying degrees of size and scale.[2] As far as is known, there has not been any successful cyber attacks against operational CBDC systems. However, there have been many high-profile cyber attacks in the DeFi domain, eg exploiting weaknesses in consensus mechanisms as well as smart contracts that enable attacks on cryptocurrency exchanges and wallets. According to a report by Elliptic, DeFi users lost $10.5 billion due to theft in 2021.[3] Since CBDCs may, and in some implementations or pilots do, use novel technologies such as DLT and smart

[1] The concept of smart contracts was introduced by Nick Szabo in 1994.

[2] See Central Bank Digital Currency (CBDC) Tracker (www.cbdctracker.org): Jamaica, Bahamas, and Nigeria have all launched CBDC implementations. Some of the more prominent CBDC pilots operating at a significant scale include those covering the Eastern Caribbean Economic and Currency Union, and the e-CNY in China.

[3] www.elliptic.co/resources/defi-risk-regulation-and-the-rise-of-decrime.

contracts, they too could be exposed and vulnerable to the type of attacks that were successfully made in the DeFi domain.

To illustrate this point, several notable DLT attacks in the DeFi domain were analysed using the MITRE ATT&CK® framework. MITRE ATT&CK®, or the ATT&CK framework for short, stands for Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK). It is a globally accessible knowledge base of adversary tactics and techniques based on real-world observations, which can be used as a foundation for the development of specific threat models and methodologies to identify and analyse adversary behaviour.[4] ATT&CK is a comprehensive and flexible framework that is widely used in the cyber security community, and regularly updated based on observed adversarial threats.

This analysis reveals that there are gaps in existing threat modelling techniques that may not adequately address the threats and associated security controls to properly protect CBDCs that make use of novel technology (eg DLT, smart contracts) from the tactics, techniques and procedures (TTPs[5]) used by threat actors in the DeFi space. Specifically, although the majority of existing TTPs could be used to model the attacks, some will require slight modification, while there exist new attack vectors that do not fit within the framework and will necessitate the creation of new TTPs. Examples of new TTPs that could be used to model the novel attacks are provided and the use of crowdsourcing is proposed to further analyse how attacks against CBDCs that use DLT as part of their reference architecture can be adequately modelled using the MITRE ATT&CK framework. Additionally, the “mean time to attack” (based on the DLT attacks studied in this analysis) is around a 10-month period between the launch of a DeFi implementation and the successful compromise. This is a key point to note for central banks about to launch a CBDC – they must be thoroughly prepared to adequately monitor and repel both well understood and novel TTPs. Furthermore, this preliminary analysis supports the argument that an official extension of the MITRE ATT&CK framework may need to be undertaken to help properly model attacks against DLT-enabled systems. This analysis uses DLT as a starting point to begin threat modelling and gap analysis for CBDC. Even for a CBDC implementation that does not plan to use DLT, the analysis around other related DeFi concepts, such as smart contracts, may still be relevant. More generally, the application of the MITRE ATT&CK framework to CBDC more broadly, regardless of technology, is likely to be a key step for any central bank looking to launch a wide scale pilot or full implementation of a CBDC.

[4] attack.mitre.org/.

[5] **TTPs:** “The behaviour of an actor. A tactic is the highest-level description of this behaviour, while techniques give a more detailed description of behaviour in the context of a tactic, and procedures an even lower-level, highly detailed description in the context of a technique” (NIST (2016)).

2. Introduction

2.1 Cyber threat landscape

The cyber threat landscape is constantly evolving. Cyber threats are fundamentally asymmetrical threats where highly skilled individuals with a wide variety of motivations can cause a disproportionate amount of damage. Threat actors can range from state-sponsored adversaries, cyber criminals, hackers-for-hire, and hacktivists, all with different capabilities and goals (ENISA (2022)). In terms of overall activity, cyber attacks have increased by 28% in the third quarter of 2022 compared with same period in 2021 globally (Check Point (2022)).

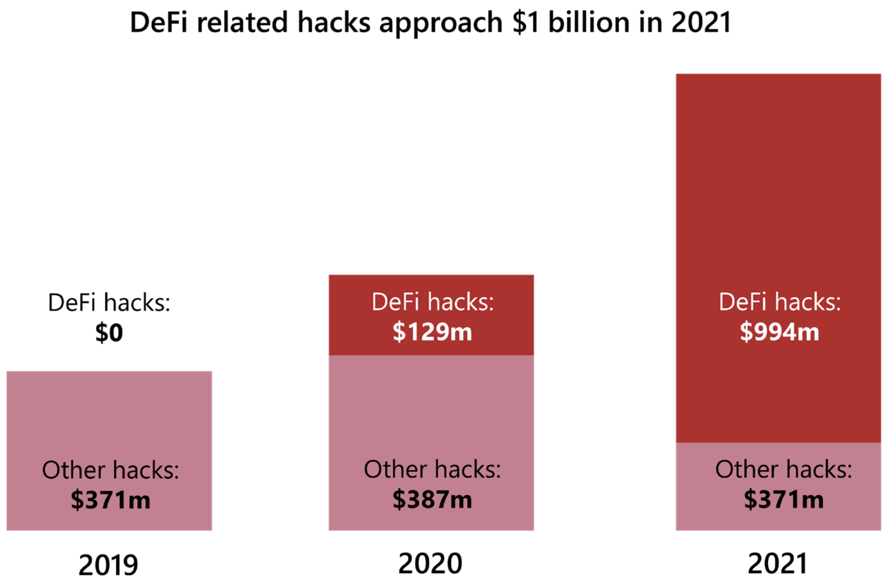
More specifically, nation state actors have become increasingly aggressive in cyberspace. These are hackers who work for a government and have significant funding to disrupt or compromise key organisations to gain access to valuable intelligence or data. Between July 2021 and June 2022, the proportion of cyber attacks perpetrated by nation states targeting critical infrastructure has doubled from 20% to 40% (Microsoft (2022)). Nation-state threat activities in general have also increased significantly since the start of the Ukrainian conflict.

Cybercrime is steadily on the increase as the industrialisation of the cybercrime economy lowers the skills barrier to entry by providing greater access to malicious tools and infrastructure. According to Microsoft (2022), in the last year alone, the number of estimated password attacks per second has increased by 74%. This is pertinent, considering that identities or privileged access credentials have been labelled as the new security perimeter ie gaining access to one would be akin to obtaining “the keys to the kingdom”. By leveraging a trusted identity, an attacker can operate covertly and exfiltrate data without setting off any alarm bells. Louis Columbus, author at Cloud Computing News, strikingly sums this up: “Teams of hackers aren’t breaking into secured systems; they’re logging in” (Columbus (2018)).

As the interest in cryptocurrency has increased, this has not gone unnoticed by cybercriminals and threat actors. Vulnerabilities have been discovered in DeFi platforms that have allowed threat actors to successfully steal more assets than in all other cyber attacks combined in just the first eight months of 2021 (see Graph 1). In 2022, more than $2.8 billion in cryptocurrency assets was stolen, the highest since 2013, according to a report from CoinGeico (Lim (2023)).

In addition to nation state and cybercriminal-perpetrated cyber attacks, the cyber threat landscape of CBDCs also includes common threats (eg SIM swap, man-in-the-middle attacks), emerging threats such as quantum computing, and other attack surfaces such as the integration with other central bank/financial institution systems or computing infrastructure (including power, network connectivity) that supports the respective CBDC assets (BIS Innovation Hub (2023)).

Graph 1 – Exponential increase of money stolen from De-Fi related hacks[6]



Source: Ciphertrace Cryptocurrency Intelligence

This trend has continued with more than $2.8 billion in cryptocurrency assets stolen in 2022, accordingly to data from CoinGecko (Lim (2023)).

2.2 Central bank digital currency

CBDC is a digital form of central bank money that is different from balances in traditional reserve or settlement accounts (CPMI Markets Committee (2018)). Unlike decentralised cryptocurrencies like Bitcoin, a CBDC is issued and managed by a government, with the aim of combining the benefits of digital currencies with the stability and regulatory oversight of traditional fiat currencies.

Central banks are cautious about the adoption of CBDCs, as they raise various technical, legal, and financial challenges that need to be carefully considered, especially considering the sharp increase of DeFi-related cyber attacks. With the lack of clarity on potential threats and vulnerabilities in the CBDC technology domain, there are several cyber security risks that central banks need to identify and then mitigate before CBDCs are rolled out.

To ensure proper information security management of CBDC systems, there is a need for additional preventative and detective security control guidelines for CBDC.

[6] See Cloud Security Alliance 2021.

2.3 Scope and objectives

In the context of CBDC, there are gaps in existing cyber security frameworks and/or cyber threat models that need to be addressed to fully describe the threats and properly derive the associated security controls to protect CBDC systems. In this section (Section 2) a brief introduction to the topic as well as the objective and scope of the paper is provided.

The approach in this paper focuses on mapping real-world tactics and techniques used in DeFi attacks against the MITRE ATT&CK framework to identify potential gaps in applying this framework to threats against CBDC implementations using technologies similar to DeFi (eg DLT). While this represents a starting point for this analysis in the CBDC space, the techniques and approach applied here through use of the MITRE ATT&CK framework could be applied to any CBDC implementation, regardless of technology. This is likely to be an important step for any central bank that is considering a large-scale pilot or launch of a CBDC.

It is important to note that this paper's objective is limited to threat modelling through the application of the MITRE ATT&CK framework on DeFi threats. The methodology employed by threat actors against DeFi could be applicable to existing or proposed CBDC implementations. The analysis is limited by making use of information that is in the public domain. Furthermore, although the focal point of this paper is not on mitigation, the potential detection and mitigation measures to safeguard against a subset of the analysed CBDC/DeFi attacks (see Annex A) are examined. While every effort has been made to ensure the accuracy and completeness of the information presented, the findings and conclusions should be considered in the light of these stated limitations.

The target audience for this technical cybersecurity report includes professionals with a technical background in areas such as cyber security, DeFi, and related fields.

In Section 3, the concepts and technology regarding DeFi, Distributed Ledger Technology (DLT), CBDC, as well as give examples of current cyber security standards and frameworks are described. The MITRE ATT&CK framework is introduced as well how it could be used modelling threats against the CBDC domain. In Section 4, several high-profile DeFi attacks are analysed and map them to the MITRE ATT&CK framework. Section 5 concludes with a summary of the findings and proposed directions for future work.

3. Background

3.1 Cryptocurrency concepts and technology – DeFi, DLT and CBDC

Decentralised finance (DeFi)

DeFi uses technology to remove intermediaries from financial transactions, allowing organisations, merchants, and users to deal faster with funds, and make such funds more accessible and controllable. This is achieved via an underlying technology infrastructure that employs novel technologies such as peer-to-peer (P2P) networks, governance tokens, smart contracts and DeFi protocols, also known as distributed ledger technology (DLT).

Distributed ledger technology (DLT)

DLT is still evolving. In this paper, DLT is defined as the protocols and supporting infrastructure that allow computers in different locations to propose and validate transactions and update records in a synchronised way across a network (Bech and Rodney (2017)). DLT-associated threats as well as design or implementation flaws may not be as well understood as is the case with traditional IT. This, coupled with an accelerated delivery of system implementations and a global skills shortage for blockchain talent, brings about another set of challenges in terms of emergent vulnerabilities that cause operational challenges for both IT and security. For example, some DLT-based schemes employ smart contracts that may utilise a Turing-complete language (using a combination of conditional statements and loops to program smart contracts).[7] Although this allows for the development of complex smart contracts, it would require stringent security code reviews and rigorous testing to eliminate coding flaws and enable secure code updates. This would provide assurance that operational logic and secure coding standards have been met, and it would also assist in the prevention of introducing unintended vulnerabilities into the system.

As DLT systems vary, they use different applications and frameworks as part of their reference architecture. In fact, there are no standardised DLT technological systems/platforms that system designers and integrators can reference when designing a secure reference architecture. This complexity in the DLT technology stack makes it challenging to fully adopt existing cyber security standards, which may not align precisely with all DLT architectural types to ensure that all relevant security vulnerabilities have been considered and the associated risks mitigated.

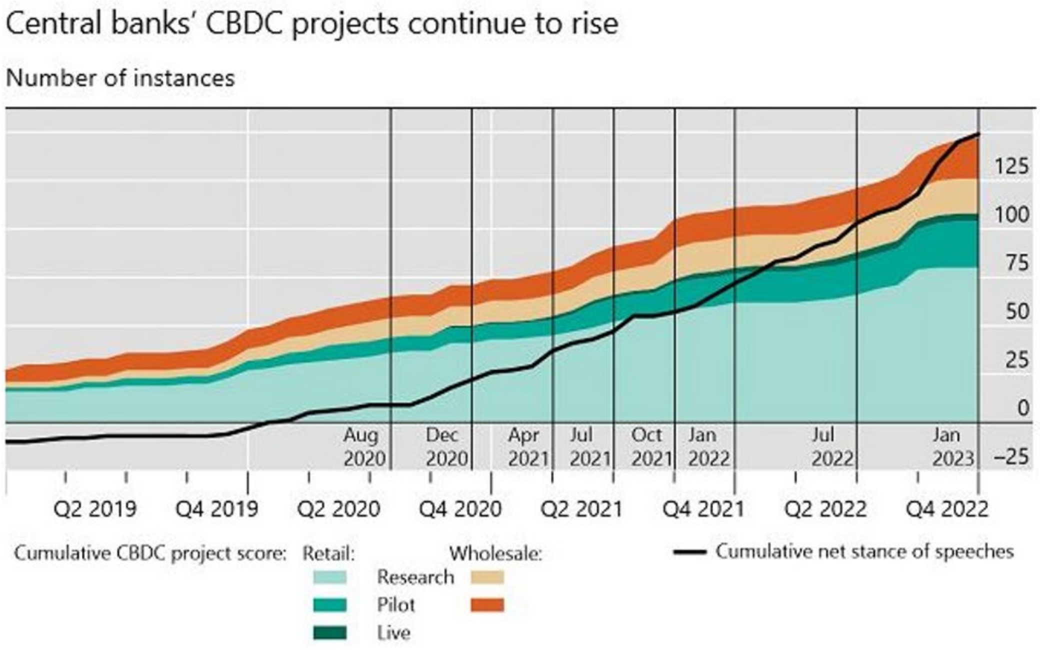
Central bank digital currency (CBDC)

Several countries around the world are increasingly exploring the possibility of issuing their own CBDCs (see Graph 2). As of December 2022, roughly 83% of countries

[7] Smart contracts are programs or transaction protocols stored on a blockchain that automatically execute, control, or document events and actions according to the terms of a contract or agreement, so that all participants can be immediately certain of the outcome, without any intermediary's involvement or time loss.

around the world have either implemented CBDC, launched pilot programs, are in the midst of development or research, or have announced plans for issuance. (Atlantic Council (2023)) The exact features and specifications of CBDCs vary between countries, depending on their goals and priorities.

Graph 2 – Rise of CBDC projects worldwide



Update 13 January 2023.

Source: R Auer, G Cornelli and J Frost (2020): "Rise of the central bank digital currencies: drivers, approaches and technologies", *BIS Working Papers*, no 880, August.

Many of these CBDC pilots depend on technologies similar to those used in DeFi solutions such as DLT and smart contracts as noted by Auer et al (2020, p 5): "Whereas many central banks are considering multiple technological options simultaneously, current proofs-of-concept tend to be based on distributed ledger technology (DLT) rather than a conventional technological infrastructure."

3.2 Current cyber security standards and frameworks (applicability)

Organisations employ cyber security standards to strengthen their cyber security posture. These standards assist in identifying and implementing the right defences to protect their systems and data against adversaries and threats.

A cyber security framework is a series of documented processes that define policies and procedures around the implementation and ongoing management of information security controls. These frameworks are a blueprint for managing risk and reducing vulnerabilities. Information security professionals use frameworks to define and prioritise the tasks required to manage enterprise security. Frameworks are also used to help prepare for compliance and other IT audits. They provide a starting point for establishing processes, policies, and administrative activities for information security management.

Examples of cyber security standards and frameworks

- ISO/IEC 27001 (Information Security Management System) is the international standard for best practice information security management systems and is a rigorous and comprehensive specification for protecting and preserving organisational information under the principles of confidentiality, integrity, and availability.

- NIST Cyber Security Framework (CSF) is a voluntary framework primarily intended to manage and mitigate cybersecurity risk for critical infrastructure organisations based on existing standards, guidelines, and practices.

- NIST SP 800-53 is the information security benchmark for US government agencies and is widely used in the private sector. NIST SP 800-53 has helped spur the development of other information security frameworks, including the NIST CSF.

- The SWIFT Customer Security Programme (CSP) is a specialised set of controls for financial institutions globally utilising services from the Society for Worldwide Interbank Financial Telecommunication (SWIFT). Although SWIFT does not use DLT, SWIFT is actively experimenting with DLTs and engaging with its community to identify areas in which they could bring concrete business benefits as part of its R&D programme.

The above standards and frameworks are useful in helping an organisation design and implement cyber security controls, fulfil compliance needs, and benchmark the organisation’s risk maturity. However, they focus mainly on general enterprise systems and may not fully translate to the digital currency space nor specific CBDC/DeFi-related infrastructure or software.

3.3 Threat models and the MITRE ATT&CK framework

Looking beyond standards, threat models derived from real-world observations are a useful way to identify the types of threat to which a system is susceptible so that they can be addressed during the design phase of an IT implementation. Several methodologies allow for a methodical review of the system design or architecture to discover and correct security flaws.

MITRE ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge) is one such threat modelling framework. It is a globally accessible and extensible knowledge base of adversary tactics and techniques based on real-world observations, and an industry best practice standard for mapping attacks. It also catalogues data that correlates known adversary[8] groups with their campaigns.

The MITRE ATT&CK framework incorporates attack patterns and maps them against courses of defensive actions that align with best practices. These defensive courses of action offer potential solutions to security teams when faced with adversaries on their network and can be used to either prevent the adversary from using a specific attack method or to mitigate the threat of an attack by adjusting the IT or security posture.

This knowledge base can be used as a foundation for threat modelling, penetration testing, defence development and similar cyber security endeavours to assist security teams in understanding the adversaries they face, assessing defence mechanisms, and reinforcing security measures in the areas where they are most necessary.

ATT&CK levels/TTPs

The MITRE ATT&CK describes adversarial behaviours as **Tactics, Techniques, and Procedures (TTPs)**, which correspond to four increasingly granular levels:

1. **Tactics** represent the **"why"** of an ATT&CK technique or sub-technique. They are the adversary's technical goals, the reason for performing an action, and what they are trying to achieve. Each tactic contains an array of techniques that network defenders have observed being used in the wild by threat actors.
2. **Techniques** represent **"how"** an adversary achieves a tactical goal by performing an action, or what an adversary gains by performing an action.
    a. **Sub-techniques** provide a more detailed description of techniques.
3. **Procedures** represent **"what"** actions an adversary performed and are instances of how an adversary has used a technique or sub-technique.

ATT&CK Matrices

The complete MITRE ATT&CK framework is branched into three technology domains, also known as a Matrix, with each Matrix containing a subset of TTPs that applies to specific target IT environments. The three primary matrices are:

1. Enterprise matrix: **(This is the focus of this paper)**
2. Mobile matrix; and
3. ICS (industrial control system) matrix.

The Enterprise and Mobile matrices are further subdivided into sub-matrices. For Mobile they are - Android and iOS. For Enterprise they are - PRE (preparatory techniques), Windows, macOS, Linux, cloud, network, and containers for the Enterprise Matrix.

[8] **Adversary**: Person, group, organisation, or government that conducts or has the intent to conduct detrimental activities. (NIST (2012))

Applicability

MITRE itself makes it clear that the ATT&CK framework can be subject to several biases – these include novelty bias (where new techniques or existing techniques by new groups are reported more frequently than the most commonly used techniques) and visibility bias (organisations that share information may have visibility into some techniques but not others) (The MITRE Corporation (2023)).

That said, the MITRE ATT&CK framework is the de facto threat modelling tool used in the cyber defence community. For example, in 2019 the NIST added the MITRE ATT&CK framework to its "Detect" cybersecurity subcategory, which recognises the importance of threat intelligence in cybersecurity operations. Additionally, the framework is frequently referenced in industry publications, such as the SANS Institute's "Critical Security Controls" and the Center for Internet Security's "Top 20 Critical Security Controls."

According to a research paper by Berkeley, more than 80% of enterprises use MITRE ATT&CK. The study examined the adoption of the MITRE ATT&CK Matrix for Enterprise and for Cloud, with 63% of global respondents made up of large and medium-sized enterprises indicating they use both in their security operations centres. Some 57% believe the ATT&CK framework is helpful for determining gaps in deployed security solutions in their enterprise, with 55% recommending it for security policy implementation and 54% using it for threat modelling (Basra and Kaushik (2020)).

The MITRE ATT&CK Enterprise Matrix has been selected for this analysis as it is a globally adopted framework that provides a comprehensive and systematic approach to understanding and defending against cyber attacks.

1. **Coverage:** the MITRE ATT&CK framework covers a wide range of attack techniques and tactics used by adversaries, from initial access to exfiltration based on real-world attack analysis and observation. This allows organisations to understand the full spectrum of potential threats they face and prioritise their defences accordingly.
2. **Consistency:** the framework provides a consistent and standardised way to describe and categorise attack techniques, making it easier for organisations to compare and contrast the risks they face and to communicate effectively with their stakeholders.
3. **Evidence-based:** the framework is based on real-world observations of attack techniques, providing organisations with a practical and realistic view of relevant threats.
4. **Customisation:** the framework is customisable, allowing organisations to tailor it to their specific needs and focus on the threats that are most relevant to their environment.
5. **Community-driven:** the MITRE ATT&CK framework is maintained by the MITRE Corporation in collaboration with the security community, ensuring that it stays up-to-date and relevant as new attack techniques emerge.

Furthermore, the MITRE ATT&CK framework will help to reframe the analysis to focus on the threats faced rather than the defences required for specific systems that cyber security frameworks do, thereby disconnecting the reliance on the architecture of an organisation's systems. In the case of a CBDC system, this serves to address the concern of the complexity and the variety of the underlying technologies that CBDCs use.

Overall, the MITRE ATT&CK framework provides organisations with a comprehensive, evidence-based, and customisable approach to understanding and defending against cyber threats. It could be a valuable tool for central banks that wish to create threat models of their CBDC systems and improve their security posture.

3.4 Potential threats for CBDCs

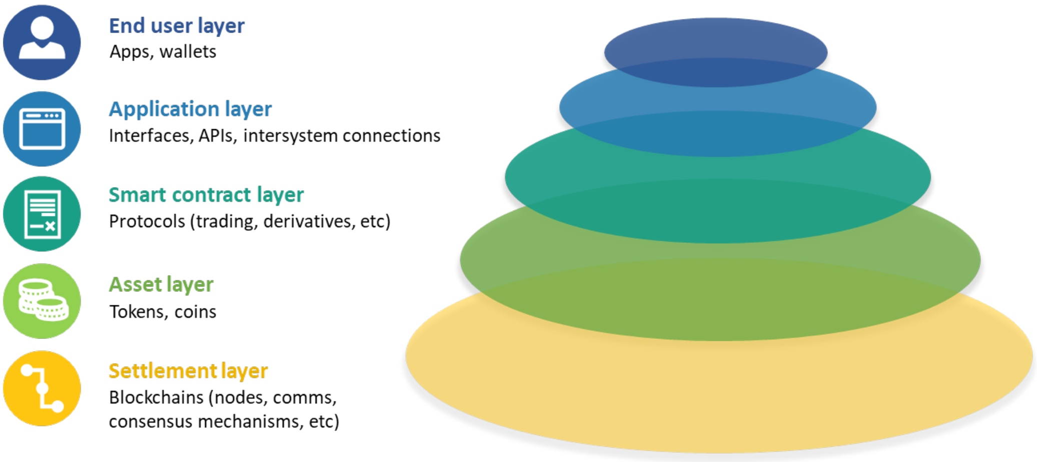
In order to apply the MITRE ATT&CK framework, it is necessary to map observed cyber attacks against CBDC systems. This analysis is challenging due to the lack of publicly reported successful attacks against implemented CBDC systems.

At the end of 2022, there were 3 launched CBDC implementations around the world, along with several other pilots at varying degrees of size and scale underway[9], with no known attacks reported thus far (Auer et al (2023)).

Additionally, since CBDCs are relatively new compared with other payment systems, there is very little historical data to predict threats specific to CBDCs. **While DeFi is not a synonym for CBDCs**, several of the current operational retail CBDC implementations are based on a similar technology stack or make use of one or more of DLT, smart contracts, tokens, digital identities and immutable data (Graph 3). This allows DeFi to serve as the starting point for this analysis of CBDC, although more tailored frameworks may need to be developed in future as the space matures.

[9] See Central Bank Digital Currency (CBDC) Tracker (www.cbdctracker.org): Jamaica, Bahamas, and Nigeria have all launched CBDC implementations. Some of the more prominent CBDC pilots operating at a significant scale include those covering the Eastern Caribbean Economic and Currency Union, and the e-CNY in China.

Graph 3 – Components of distributed ledger technology (DLT) in potential CBDC implementations


End user layer
Apps, wallets
Application layer
Interfaces, APIs, intersystem connections
Smart contract layer
Protocols (trading, derivatives, etc)
Asset layer
Tokens, coins
Settlement layer
Blockchains (nodes, comms, consensus mechanisms, etc)

As there is very little in the way of documented attacks against CBDC, **the attacks perpetrated against DeFi have been used to test the assertion** that there are gaps in existing cyber security frameworks and cyber threat models when it comes to defining the threats and articulating the associated security controls to properly protect CBDC systems. Over time, as more CBDC systems approach widespread pilot or implementation stage, it will be important for central banks to have appropriate frameworks in place to properly assess their cyber security. DeFi provides an important starting point from which to build out a broader framework and model.

4. Analysis

In this section, the application of the MITRE ATT&CK framework against the real-world observations from major DeFi attacks that have occurred in the last few years is discussed.

4.1 Analysis of notable attacks against DeFi

Many existing CBDC implementations are based on DeFi solutions such as DLT. Accordingly, six notable DeFi attacks have been analysed to identify applicable attack patterns for CBDC implementations. These attacks represent some of the highest-profile cases in terms of estimated financial loss and have had significant news coverage. Additionally, they provide a variety of novel attack vectors for subsequent analysis (see Table 1).

Table 1 – Details of notable DeFi hacks in recent years

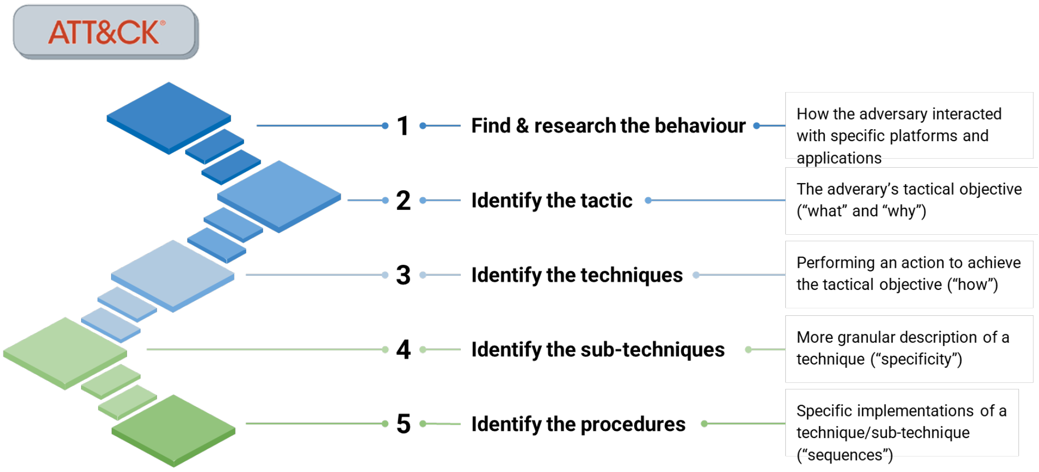
DeFi	Estimated losses (USD)	Year & Quarter	DeFi category	Main cause
Poly Network	610m	Q3 2021	Protocol	Logic vulnerability
BadgerDAO	120m	Q4 2021	Yield Aggregator	API key leakage
Axie/Ronin	625m	Q1 2022	Bridge/Gaming	Private key leakage/phishing
Wormhole/Solana	325m	Q1 2022	Bridge	Logic vulnerability
Beanstalk	182m	Q2 2022	Stablecoin/Protocol	Logic vulnerability
Fei Protocol	80m	Q2 2022	Stablecoin	Logic vulnerability

4.2 Tactic, techniques, and procedures (TTPs) mapping

In order to map the attacks against the MITRE ATT&CK framework the following four steps are used:

1. Extract and map the different techniques used in each notable cyber attack event to MITRE ATT&CK framework's tactic, technique/sub-technique, and procedure (Graph 4).
2. Attacks that could not be adequately categorised to the existing framework would be mapped to the closest tactic, technique/sub-technique, or procedure.
3. If no relevant tactic, technique/sub-technique, or procedure exists, a new one would be suggested.
4. Items that do not yet exist in the framework are highlighted in orange in their respective Sankey diagrams below.

Graph 4 – Best practice approach for TTP mapping


ATT&CK
1 Find & research the behaviour
How the adversary interacted with specific platforms and applications
2 Identify the tactic
The adverary's tactical objective ("what" and "why")
3 Identify the techniques
Performing an action to achieve the tactical objective ("how")
4 Identify the sub-techniques
More granular description of a technique ("specificity")
5 Identify the procedures
Specific implementations of a technique/sub-technique ("sequences")

Poly Network

The Poly Network is a decentralised finance platform that enables cross-chain transactions between different blockchain networks. On 10 August 2021, Poly Network suffered a massive cyber attack that resulted in the theft of approximately $610 million worth of cryptocurrencies, making it one of the largest cryptocurrency heists in history.

The attacker first identified a specific vulnerability in the Poly Network smart contract code (the contract "EthCrossChainManager") that allowed them to manipulate the "message" parameter and bypass the normal authorisation procedures. The attacker brute-forced a string that gives the same 32-bit value. With this altered "message" parameter, the attacker was able to trick the smart contract into believing that they had the necessary authorisation to initiate transfers.

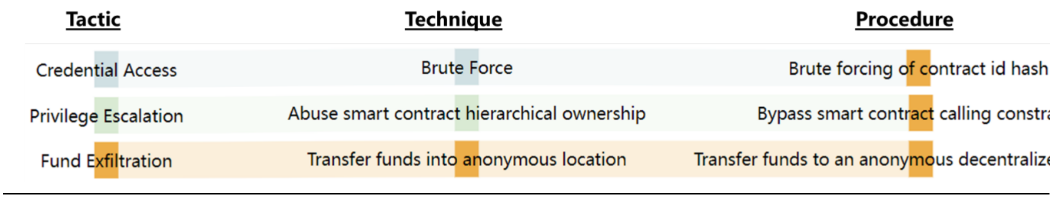
The attacker then called a cross-chain transaction from the Ethereum network to the Poly Network by using the bypassed string. The attacker abused the fact that "EthCrossChainManager" was wrongly the owner of "EthCrossChainData", which is responsible for setting and managing a list of public keys of "authenticator nodes" (Keepers) that manage the wallets in the underlying liquidity chains. In other words, the attacker escalated the privilege of the smart contract ownership and was able to become a Keeper and move funds contained within Poly's wallets.

The attacker then initiated a series of transfers across multiple blockchains, including Ethereum, Poly liquidity wallets: Binance, Neo, Tether etc. They transferred a variety of cryptocurrencies, including Bitcoin, Ethereum and USDT, among others. The transfers were initially undetected by the Poly Network team, and the hacker was able to successfully move the stolen assets to different addresses on different blockchains.

A summary of the TTPs used in the Poly Network attack are mapped in a Sankey Diagram (Graph 5). Items that do not yet exist in ATT&CK are highlighted in orange.

Other colours (ie blue and green in the below Sankey Diagram) are simply for illustrative purposes and reflect existing ATT&CK items.

Graph 5 – Poly Network attack TTPs (Sankey Diagram)


Tactic
Technique
Procedure
Credential Access
Brute Force
Brute forcing of contract id hash
Privilege Escalation
Abuse smart contract hierarchical ownership
Bypass smart contract calling constr
Fund Exfiltration
Transfer funds into anonymous location
Transfer funds to an anonymous decentraliz

Axie/Ronin

On 23 March 2022, hackers compromised the Ronin Network, stealing 173,600 Ether and 25.5 million USD coins were stolen with an estimated total value of $620 million. Axie Infinity is built on the Ronin Network, an Ethereum-linked sidechain developed by Sky Mavis. Although Axie is not a DeFi platform, it uses DeFi technology in their offerings.

Ronin uses a "proof of authority" system for signing transactions – five of a total of nine validator nodes are needed to approve transactions.

In November 2021, SkyMavis requested help from the Axie DAO to distribute free transactions due to an immense user load. The Axie DAO *allowlisted*[10] SkyMavis to sign various transactions on its behalf. This was discontinued in December 2021, but the allowlist access was not revoked.

A spear-phishing email was sent masquerading as a recruitment offer with a high-paying job offer to a senior developer at Sky Mavis. The targeted user downloaded Tradertraitor/AppleJeus[11] malware: a series of applications and websites that look real but are controlled by the adversary. The adversary performed lateral movement and obtained the private keys for four SkyMavis validator nodes.

Using a "backdoor" (a gas-free RPC) the adversary compromised a fifth validator run by Axie DAO (a community-run organisation supporting the Axie Infinity project).

A summary of the TTPs used in the Axie/Ronin attack are mapped in a Sankey Diagram (Graph 6).

[10] Allowlist: Allowlisting is a security capability that reduces harmful security attacks by allowing only trusted files, applications, and processes to be run. (VMware (2023))

[11] **Tradertraitor/AppleJeus**: Cryptocurrency trading applications that were modified to include malware which facilitates theft of cryptocurrency targeting individuals and companies—including cryptocurrency exchanges and financial services companies (CISA (2022)).

BadgerDAO

In December 2021, BadgerDAO reported a phishing incident caused by a maliciously injected snippet provided by *Cloudflare Workers*, a serverless application platform that runs on its cloud network. The hacker used a compromised API key that was created without the knowledge or authorisation of BadgerDAO engineers to periodically inject a malicious code that affected a subset of its customers. This code generated rogue transaction approvals, which took advantage of a visual design issue in how crypto wallets ask their users to approve of certain transactions. If approved by users, the code would allow the hackers to pull funds to their own wallets instead of those controlled by BadgerDAO at a future date.

The hacker ultimately stole $130 million in funds, of which approximately $9 million was recoverable since those funds were transferred by the hacker but not yet withdrawn from BadgerDAO's vaults. BadgerDAO has since patched the Cloudflare exploit, updated Cloudflare's account password and deleted or freshened API keys where possible.

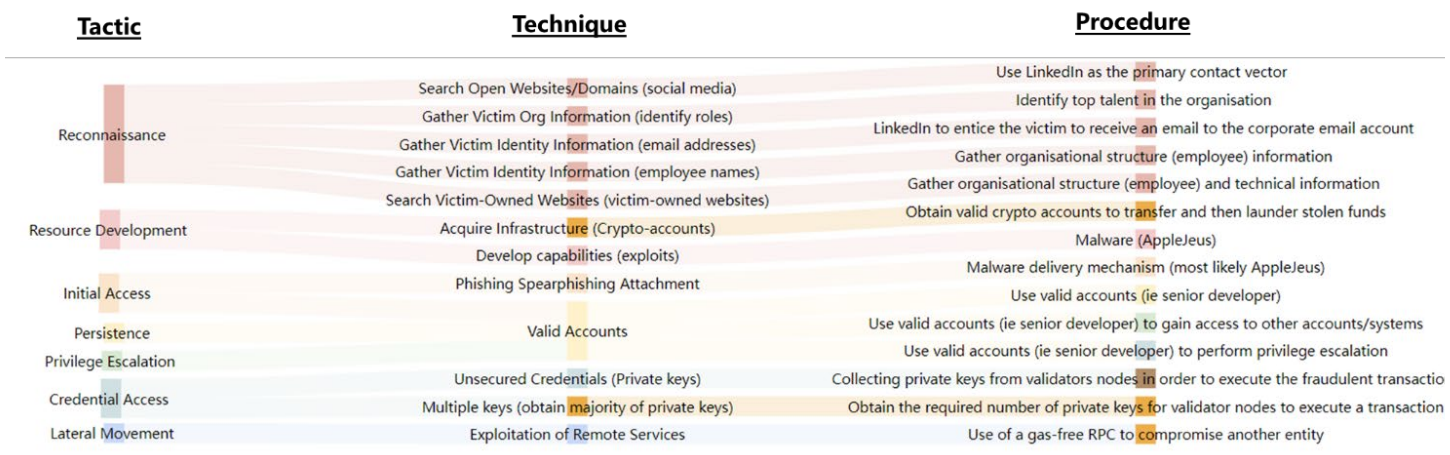
A summary of the TTPs used in the BadgerDAO attack are mapped in a Sankey Diagram (Graph 7).

Wormhole/Solana

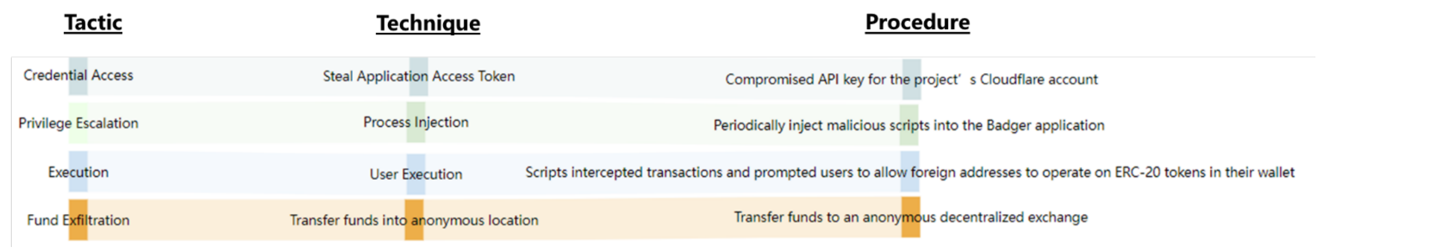
Solana's Wormhole is a communication bridge enabling the transfer of tokenised assets seamlessly across different blockchains, benefiting from Solana's high speed and low cost. Wormhole has a set of guardians that sign off on transfers between chains. Bridges like Wormhole work by having two smart contracts — one on each chain. In this case, there was one smart contract on Solana and one on Ethereum. A bridge like Wormhole takes an Ethereum token, locks it into a contract on one chain, and then on the chain at the other side of the bridge, it issues a parallel token.

On 2 February 2022, attackers exploited unpatched Rust contracts in Solana and manipulated them into crediting 120,000 ETH worth around $320 million as having been deposited on Ethereum, allowing the hacker to mint the equivalent in wrapped whETH (Wormhole ETH) on Solana.

Graph 6 – Axie/Ronin attack TTPs (Sankey Diagram)


Tactic
Technique
Procedure
Reconnaissance
Resource Development
Initial Access
Persistence
Privilege Escalation
Credential Access
Lateral Movement
Search Open Websites/Domains (social media)
Gather Victim Org Information (identify roles)
Gather Victim Identity Information (email addresses)
Gather Victim Identity Information (employee names)
Search Victim-Owned Websites (victim-owned websites)
Acquire Infrastructure (Crypto-accounts)
Develop capabilities (exploits)
Phishing Spearphishing Attachment
Valid Accounts
Unsecured Credentials (Private keys)
Multiple keys (obtain majority of private keys)
Exploitation of Remote Services
Use LinkedIn as the primary contact vector
Identify top talent in the organisation
LinkedIn to entice the victim to receive an email to the corporate email account
Gather organisational structure (employee) information
Gather organisational structure (employee) and technical information
Obtain valid crypto accounts to transfer and then launder stolen funds
Malware (AppleJeus)
Malware delivery mechanism (most likely AppleJeus)
Use valid accounts (ie senior developer)
Use valid accounts (ie senior developer) to gain access to other accounts/systems
Use valid accounts (ie senior developer) to perform privilege escalation
Collecting private keys from validators nodes in order to execute the fraudulent transactio
Obtain the required number of private keys for validator nodes to execute a transaction
Use of a gas-free RPC to compromise another entity

Graph 7 - BadgerDAO attack TTPs (Sankey Diagram)

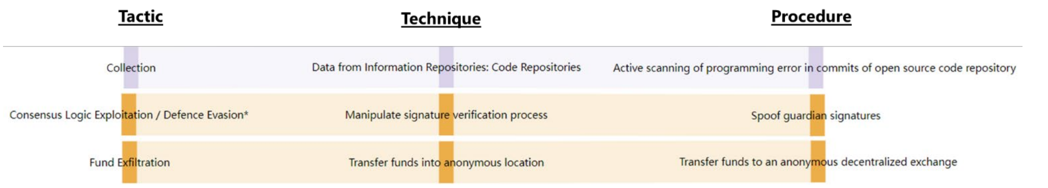

Tactic
Technique
Procedure
Credential Access
Privilege Escalation
Execution
Fund Exfiltration
Steal Application Access Token
Process Injection
User Execution
Transfer funds into anonymous location
Compromised API key for the project' s Cloudflare account
Periodically inject malicious scripts into the Badger application
Scripts intercepted transactions and prompted users to allow foreign addresses to operate on ERC-20 tokens in their wallet
Transfer funds to an anonymous decentralized exchange

The Wormhole contracts used the function “load_instruction_at” to check that the “Secp256k1” function was called first. By looking at Github’s internal commits, the “load_instruction_at” function was deprecated on 13 January by the team as it did not check that the signature verification was being performed by a whitelisted address, also known as a “system address”. This also highlights that it took about 20 days from the deprecation of the function to the actual hack.

The system address was supposed to be provided as the program to be executed, but in the “verify signatures” transaction for the fake deposit of 120,000 ETH; the system address was substituted for a program’s address, or the equivalent of an Ethereum smart contract, which did not check signatures at all.

A summary of the TTPs used in the Wormhole/Solana attack are mapped in a Sankey Diagram (Graph 8).

Graph 8 – Wormhole/Solana attack TTPs (Sankey Diagram)



Tactic	Technique	Procedure
Collection	Data from Information Repositories: Code Repositories	Active scanning of programming error in commits of open source code repository
Consensus Logic Exploitation / Defence Evasion*	Manipulate signature verification process	Spoof guardian signatures
Fund Exfiltration	Transfer funds into anonymous location	Transfer funds to an anonymous decentralized exchange

Beanstalk

Beanstalk uses a decentralised governance protocol with an emergencyCommit function where protocol changes can be approved by a supermajority (a two thirds vote) and implemented after 24 hours rather than going through the standard process. The voting power is controlled using donations to Beanstalk’s Diamond contract.

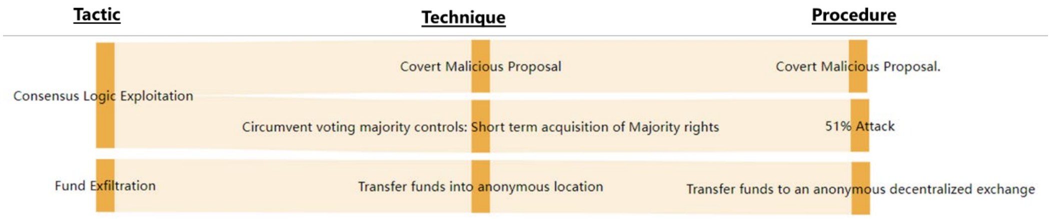
The hacker made malicious Beanstalk proposals in the form of two smart contract proposals to drain the protocol’s funds to the attacker’s account and a Ukraine aid donation account.

After a one-day waiting period, the attacker used flash loans, a type of uncollateralised loan that lets a user borrow assets with no upfront collateral as long as the borrowed assets are paid back within the same blockchain transaction,[12] to deposit a large sum into the Diamond contract. This provided the attacker with a 79% control of the governance protocol’s votes, which was much larger than the two thirds vote necessary for approval, allowing the attacker to unilaterally approve its own proposal. Once the smart contracts were executed, the attacker earned a profit of $76 million out of the $181 million stolen after paying off the flash loan.

[12] chain.link/education-hub/flash-loans.

A summary of the TTPs used in the Beanstalk attack are mapped in a Sankey Diagram (Graph 9). Note that all the items are new to ATT&CK and accordingly all are highlighted in orange.

Graph 9 – Beanstalk attack TTPs (Sankey Diagram)


Tactic
Technique
Procedure
Consensus Logic Exploitation
Covert Malicious Proposal
Covert Malicious Proposal.
Circumvent voting majority controls: Short term acquisition of Majority rights
51% Attack
Fund Exfiltration
Transfer funds into anonymous location
Transfer funds to an anonymous decentralized exchange

Fei Protocol

In April 2022, the Fei Protocol was the victim of a re-entrancy attack. Fei Protocol is a rapidly growing algorithmic stablecoin built natively for the Defi ecosystem and utilises protocol-controlled value (PCV) for peg stabilisation, while maintaining highly liquid secondary markets. Rari Capital, on the other hand, is a permissionless lending protocol that allows users to create Fuse pools that anyone with a wallet can access from anywhere to lend or borrow ERC-20 tokens. No minimum funds are required of users. In December 2021, Fei Protocol and Rari Capital merged to further bootstrap liquidity for the Fuse pools, with Fei Protocol providing the necessary initial liquidity.

In the case of the Fei Protocol, it was placed at risk by the use of code forked in early 2021 from Compound, an Ethereum money market platform. Fei made certain changes to the code, but despite audits a flaw was not discovered in the code until it was too late. Within the code, multiple re-entrancy vulnerabilities existed, which involved smart contracts calling each other to move funds without appropriate checks, and although most were fixed in previous updates, some vulnerable functions were overlooked.

The attacker took advantage of two functions in the Fei Protocol's contracts: exitMarket and borrow.

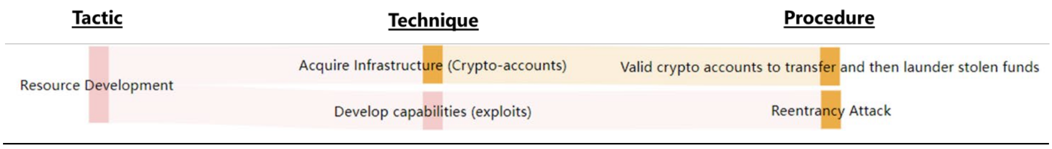
The exitMarket function verifies that a deposit is no longer used as collateral for a loan and then allows it to be withdrawn.

The borrow function allows a user to take out a loan using a deposited asset as collateral and does not follow the check-effect-interaction pattern, leaving it vulnerable to attack.

This attack was due to a design flaw in the Fei Protocol that failed to follow the check-effect-interaction pattern and thus allowed the attacker to make a re-entrant call before the borrow records were updated. The attacker drained approximately $80 million in tokens from the vulnerable contract.

A summary of the TTPs used in the Fei Protocol attack are mapped in a Sankey Diagram (Graph 10).

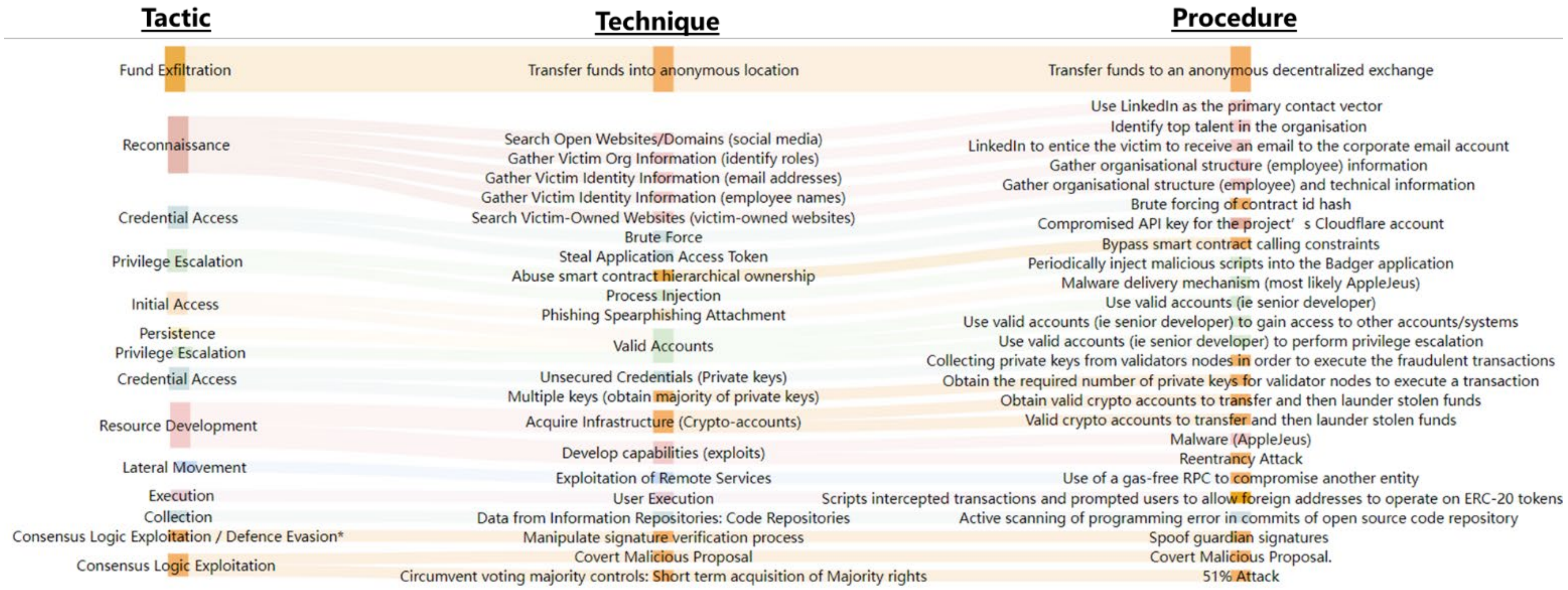
Graph 10 – Fei Protocol attack TTPs (Sankey Diagram)


Tactic
Technique
Procedure
Resource Development
Acquire Infrastructure (Crypto-accounts)
Develop capabilities (exploits)
Valid crypto accounts to transfer and then launder stolen funds
Reentrancy Attack

4.3 Summary of analysis

From the mapping of the TTPs above, some attacks can be decomposed and mapped to the MITRE ATT&CK framework, however there are novel attacks against DeFi-specific technology that cannot be adequately referenced using the existing framework (highlighted in orange in Graph 11).

Graph 11 – Summary of attack TTPs (Sankey Diagram)


Tactic
Technique
Procedure
Fund Exfiltration
Reconnaissance
Credential Access
Privilege Escalation
Initial Access
Persistence
Privilege Escalation
Credential Access
Resource Development
Lateral Movement
Execution
Collection
Consensus Logic Exploitation / Defence Evasion*
Consensus Logic Exploitation
Transfer funds into anonymous location
Search Open Websites/Domains (social media)
Gather Victim Org Information (identify roles)
Gather Victim Identity Information (email addresses)
Gather Victim Identity Information (employee names)
Search Victim-Owned Websites (victim-owned websites)
Brute Force
Steal Application Access Token
Abuse smart contract hierarchical ownership
Process Injection
Phishing Spearphishing Attachment
Valid Accounts
Unsecured Credentials (Private keys)
Multiple keys (obtain majority of private keys)
Acquire Infrastructure (Crypto-accounts)
Develop capabilities (exploits)
Exploitation of Remote Services
User Execution
Data from Information Repositories: Code Repositories
Manipulate signature verification process
Covert Malicious Proposal
Circumvent voting majority controls: Short term acquisition of Majority rights
Transfer funds to an anonymous decentralized exchange
Use LinkedIn as the primary contact vector
Identify top talent in the organisation
LinkedIn to entice the victim to receive an email to the corporate email account
Gather organisational structure (employee) information
Gather organisational structure (employee) and technical information
Brute forcing of contract id hash
Compromised API key for the project' s Cloudflare account
Bypass smart contract calling constraints
Periodically inject malicious scripts into the Badger application
Malware delivery mechanism (most likely AppleJeus)
Use valid accounts (ie senior developer)
Use valid accounts (ie senior developer) to gain access to other accounts/systems
Use valid accounts (ie senior developer) to perform privilege escalation
Collecting private keys from validators nodes in order to execute the fraudulent transactions
Obtain the required number of private keys for validator nodes to execute a transaction
Obtain valid crypto accounts to transfer and then launder stolen funds
Valid crypto accounts to transfer and then launder stolen funds
Malware (AppleJeus)
Reentrancy Attack
Use of a gas-free RPC to compromise another entity
Scripts intercepted transactions and prompted users to allow foreign addresses to operate on ERC-20 tokens
Active scanning of programming error in commits of open source code repository
Spoof guardian signatures
Covert Malicious Proposal.
51% Attack

Below is a consolidation of the TTP analysis of new entries that cannot be attributed wholly to the existing MITRE ATT&CK framework and are sorted by the number of new entries in the attack (highlighted in orange).

New procedures

Table 2 – Summary of new procedures

Tactic [Tactic ID]	Technique/sub-technique title	Procedure
	[ID]	
Credential Access [TA0006]	Brute Force [T1110.XXX][13]	Brute forcing of contract id hash
Lateral Movement [TA0008]	Exploitation of Remote Services [T1210]	Use of a gas-free RPC to compromise another entity
Resource Development [TA0042]	Develop capabilities (exploits) [T1587.004]	Re-entrancy attack

New techniques/sub-techniques

Table 3 – Summary of new techniques / sub-techniques

Tactic [Tactic ID]	Technique/sub-technique title	Procedure
	[ID]	
Privilege Escalation [TA0043]	Abuse smart contract hierarchical ownership [TXXXX]	Bypass smart contract calling constraints
Credential Access [TA0006]	Multiple keys (obtain majority of private keys) [TXXXX]	Obtain the required number of private keys for validator nodes to execute a transaction
Resource Development [TA0042]	Acquire infrastructure (Crypto accounts) [TXXXX]	Valid crypto accounts to transfer and then launder stolen funds

[13] XXX is used where we believe this is a new tactic, technique/sub-technique, or procedure update.

New tactics

Table 4 – Summary of new tactics

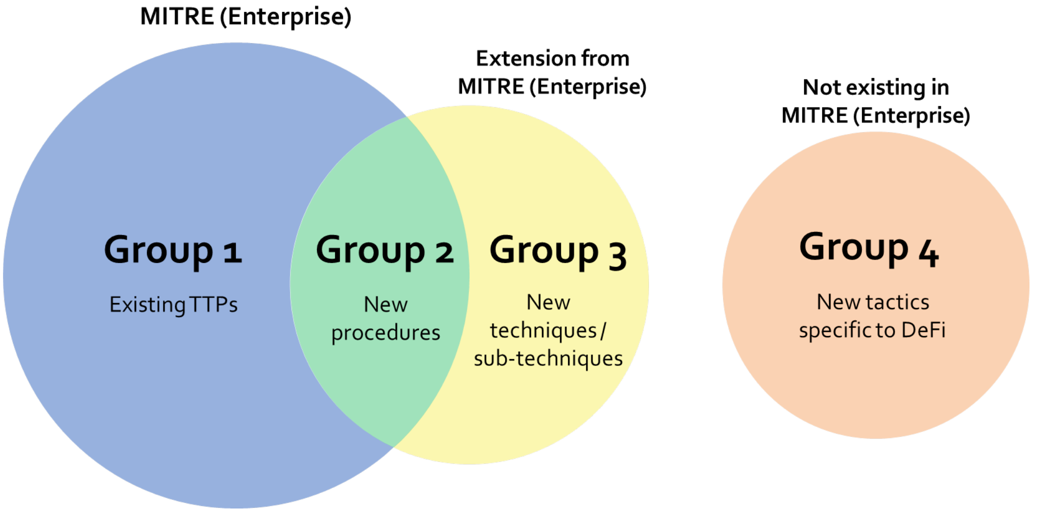
Tactic [Tactic ID]	Technique/sub-technique title	Procedure
	[ID]	
Consensus Logic Exploitation	Circumvent voting majority controls: Short term acquisition of majority rights	51% attack
[TA00XX]	[TXXXX.XXX]	
Fund Exfiltration [TA00XX]	Transfer funds into anonymous location [TXXXX]	Transfer funds to an anonymous decentralised exchange
Consensus Logic Exploitation/Defence Evasion [TA00XX]	Manipulate signature verification process [TXXXX]	Spoof guardian signatures - Manipulate VAA verification (the hacker managed to substitute the "Sysvar: Instructions" address with their own supplied address)
Consensus Logic Exploitation [TA00XX]	Covert malicious proposal [TXXXX]	Covert malicious proposal

The full mapping list documented using the framework's contribution template can be found in Annex A – TTP mapping of notable DeFi attacks.

5. Findings and discussion

5.1 Gaps, observations and insights

Graph 12 - Categorisation of attack mappings to MITRE ATT&CK framework


MITRE (Enterprise)
Extension from MITRE (Enterprise)
Not existing in MITRE (Enterprise)
Group 1
Existing TTPs
Group 2
New procedures
Group 3
New techniques / sub-techniques
Group 4
New tactics specific to DeFi

After mapping the DeFi attacks to the MITRE ATT&CK framework, the findings have been categorised into four distinct groups (Graph 12):

Group 1 Existing descriptions in MITRE can be used for specific parts of the attacks

Group 2 Some descriptions might need slight updates or an adjustment of the current understanding **(New procedures)**

Group 3 There are new types of attack that are not present in the current framework but can be tied to existing tactics **(New techniques)** or existing tactics and techniques **(New sub-techniques)**

Group 4 There are new types of attack that that do not fit in the current framework and cannot be mapped to the existing Enterprise Matrix **(New tactics)**

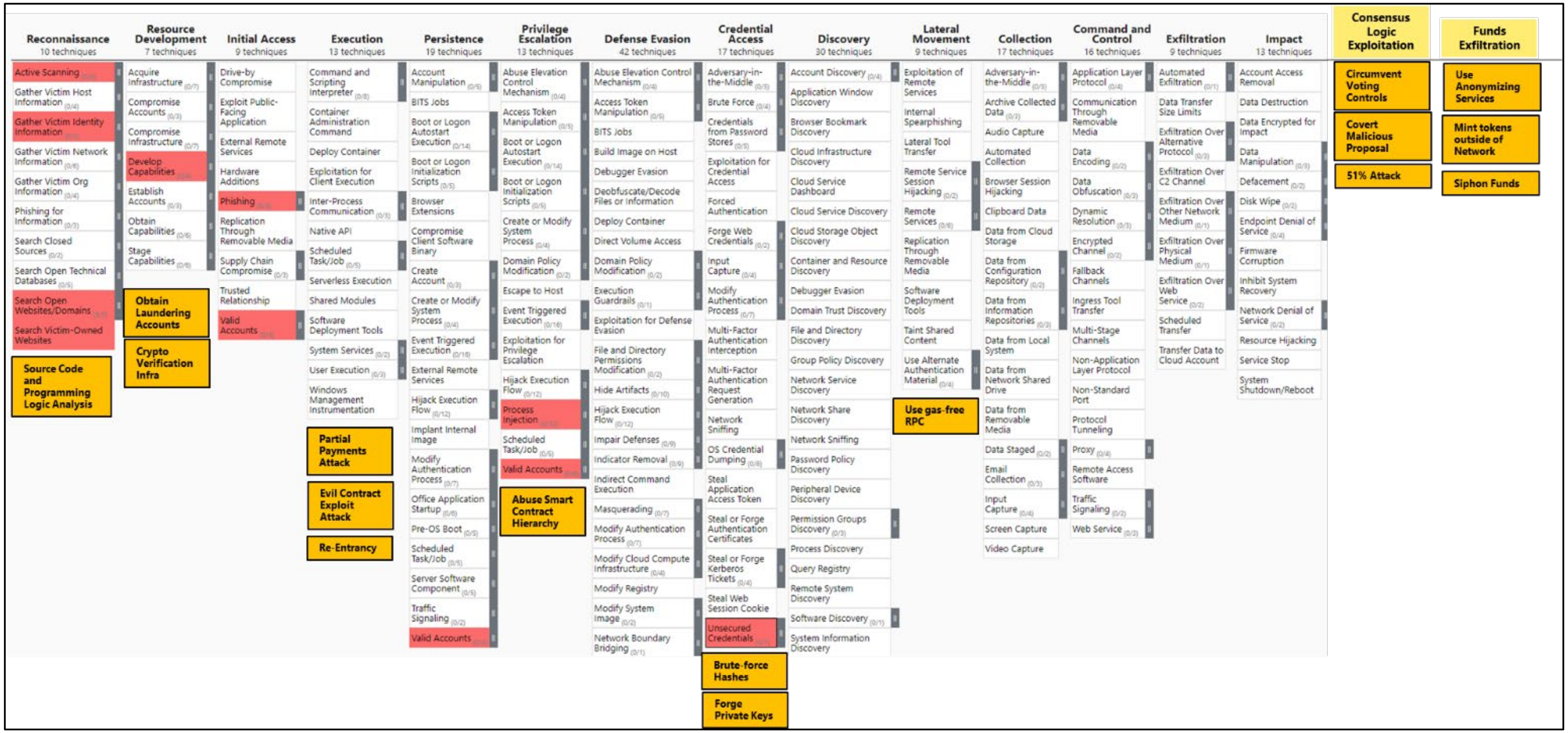
These are illustrated using the existing Enterprise Matrix of the MITRE ATT&CK framework in Graph 13 below. Items highlighted in red are those belonging to groups 1 and 2, while the items highlighted in orange are those from group 3, which do not yet have an entry in the Matrix. Items highlighted in yellow are those from group 4.

As this analysis results indicate the presence of group 2, group 3, and group 4 attack types, this implies that the Enterprise Matrix is currently incapable of describing all types of DeFi cyber attack.

An extension of the MITRE ATT&CK framework will be required to cover and address the attack vectors currently being used against DeFi implementations.

There is precedent for the MITRE ATT&CK framework to be extended from the Enterprise Matrix to form the Industrial Control Systems (ICS) and Mobile Matrices as shown below (Graphs 14 and 15). A similar exercise can be applied to DeFi.

Graph 13 – Mapping to the Enterprise Matrix on the MITRE ATT&CK framework



Reconnaissance
10 techniques

- Active Scanning (0/3)
- Gather Victim Host Information (0/4)
- Gather Victim Identity Information (0/3)
- Gather Victim Network Information (0/6)
- Gather Victim Org Information (0/4)
- Phishing for Information (0/3)
- Search Closed Sources (0/2)
- Search Open Technical Databases (0/5)
- Search Open Websites/Domains (0/3)
- Search Victim-Owned Websites
- **Source Code and Programming Logic Analysis**

Resource Development
7 techniques

- Acquire Infrastructure (0/7)
- Compromise Accounts (0/3)
- Compromise Infrastructure (0/7)
- Develop Capabilities (0/4)
- Establish Accounts (0/3)
- Obtain Capabilities (0/6)
- Stage Capabilities (0/6)
- **Obtain Laundering Accounts**
- **Crypto Verification Infra**

Initial Access
9 techniques

- Drive-by Compromise
- Exploit Public-Facing Application
- External Remote Services
- Hardware Additions
- Phishing (0/4)
- Replication Through Removable Media
- Supply Chain Compromise (0/3)
- Trusted Relationship
- Valid Accounts (0/4)

Execution
13 techniques

- Command and Scripting Interpreter (0/8)
- Container Administration Command
- Deploy Container
- Exploitation for Client Execution
- Inter-Process Communication (0/3)
- Native API
- Scheduled Task/Job (0/5)
- Serverless Execution
- Shared Modules
- Software Deployment Tools
- System Services (0/2)
- User Execution (0/3)
- Windows Management Instrumentation
- **Partial Payments Attack**
- **Evil Contract Exploit Attack**
- **Re-Entrancy**

Persistence
19 techniques

- Account Manipulation (0/5)
- BITS Jobs
- Boot or Logon Autostart Execution (0/14)
- Boot or Logon Initialization Scripts (0/5)
- Browser Extensions
- Compromise Client Software Binary
- Create Account (0/3)
- Create or Modify System Process (0/4)
- Event Triggered Execution (0/16)
- External Remote Services
- Hijack Execution Flow (0/12)
- Implant Internal Image
- Modify Authentication Process (0/7)
- Office Application Startup (0/6)
- Pre-OS Boot (0/5)
- Scheduled Task/Job (0/5)
- Server Software Component (0/5)
- Traffic Signaling (0/2)
- Valid Accounts (0/4)

Privilege Escalation
13 techniques

- Abuse Elevation Control Mechanism (0/4)
- Access Token Manipulation (0/5)
- Boot or Logon Autostart Execution (0/14)
- Boot or Logon Initialization Scripts (0/5)
- Create or Modify System Process (0/4)
- Domain Policy Modification (0/2)
- Escape to Host
- Event Triggered Execution (0/16)
- Exploitation for Privilege Escalation
- Hijack Execution Flow (0/12)
- Process Injection (0/12)
- Scheduled Task/Job (0/5)
- Valid Accounts (0/4)
- **Abuse Smart Contract Hierarchy**

Defense Evasion
42 techniques

- Abuse Elevation Control Mechanism (0/4)
- Access Token Manipulation (0/5)
- BITS Jobs
- Build Image on Host
- Debugger Evasion
- Deobfuscate/Decode Files or Information
- Deploy Container
- Direct Volume Access
- Domain Policy Modification (0/2)
- Execution Guardrails (0/1)
- Exploitation for Defense Evasion
- File and Directory Permissions Modification (0/2)
- Hide Artifacts (0/10)
- Hijack Execution Flow (0/12)
- Impair Defenses (0/9)
- Indicator Removal (0/9)
- Indirect Command Execution
- Masquerading (0/7)
- Modify Authentication Process (0/7)
- Modify Cloud Compute Infrastructure (0/4)
- Modify Registry
- Modify System Image (0/2)
- Network Boundary Bridging (0/1)

Credential Access
17 techniques

- Adversary-in-the-Middle (0/3)
- Brute Force (0/4)
- Credentials from Password Stores (0/5)
- Exploitation for Credential Access
- Forced Authentication
- Forge Web Credentials (0/2)
- Input Capture (0/4)
- Modify Authentication Process (0/7)
- Multi-Factor Authentication Interception
- Multi-Factor Authentication Request Generation
- Network Sniffing
- OS Credential Dumping (0/8)
- Steal Application Access Token
- Steal or Forge Authentication Certificates
- Steal or Forge Kerberos Tickets (0/4)
- Steal Web Session Cookie
- Unsecured Credentials (0/7)
- **Brute-force Hashes**
- **Forge Private Keys**

Discovery
30 techniques

- Account Discovery (0/4)
- Application Window Discovery
- Browser Bookmark Discovery
- Cloud Infrastructure Discovery
- Cloud Service Dashboard
- Cloud Service Discovery
- Cloud Storage Object Discovery
- Container and Resource Discovery
- Debugger Evasion
- Domain Trust Discovery
- File and Directory Discovery
- Group Policy Discovery
- Network Service Discovery
- Network Share Discovery
- Network Sniffing
- Password Policy Discovery
- Peripheral Device Discovery
- Permission Groups Discovery (0/3)
- Process Discovery
- Query Registry
- Remote System Discovery
- Software Discovery (0/1)
- System Information Discovery

Lateral Movement
9 techniques

- Exploitation of Remote Services
- Internal Spearphishing
- Lateral Tool Transfer
- Remote Service Session Hijacking (0/2)
- Remote Services (0/6)
- Replication Through Removable Media
- Software Deployment Tools
- Taint Shared Content
- Use Alternate Authentication Material (0/4)
- **Use gas-free RPC**

Collection
17 techniques

- Adversary-in-the-Middle (0/3)
- Archive Collected Data (0/3)
- Audio Capture
- Automated Collection
- Browser Session Hijacking
- Clipboard Data
- Data from Cloud Storage
- Data from Configuration Repository (0/2)
- Data from Information Repositories (0/3)
- Data from Local System
- Data from Network Shared Drive
- Data from Removable Media
- Data Staged (0/2)
- Email Collection (0/3)
- Input Capture (0/4)
- Screen Capture
- Video Capture

Command and Control
16 techniques

- Application Layer Protocol (0/4)
- Communication Through Removable Media
- Data Encoding (0/2)
- Data Obfuscation (0/3)
- Dynamic Resolution (0/3)
- Encrypted Channel (0/2)
- Fallback Channels
- Ingress Tool Transfer
- Multi-Stage Channels
- Non-Application Layer Protocol
- Non-Standard Port
- Protocol Tunneling
- Proxy (0/4)
- Remote Access Software
- Traffic Signaling (0/2)
- Web Service (0/3)

Exfiltration
9 techniques

- Automated Exfiltration (0/1)
- Data Transfer Size Limits
- Exfiltration Over Alternative Protocol (0/3)
- Exfiltration Over C2 Channel
- Exfiltration Over Other Network Medium (0/1)
- Exfiltration Over Physical Medium (0/1)
- Exfiltration Over Web Service (0/2)
- Scheduled Transfer
- Transfer Data to Cloud Account

Impact
13 techniques

- Account Access Removal
- Data Destruction
- Data Encrypted for Impact
- Data Manipulation (0/3)
- Defacement (0/2)
- Disk Wipe (0/2)
- Endpoint Denial of Service (0/4)
- Firmware Corruption
- Inhibit System Recovery
- Network Denial of Service (0/2)
- Resource Hijacking
- Service Stop
- System Shutdown/Reboot

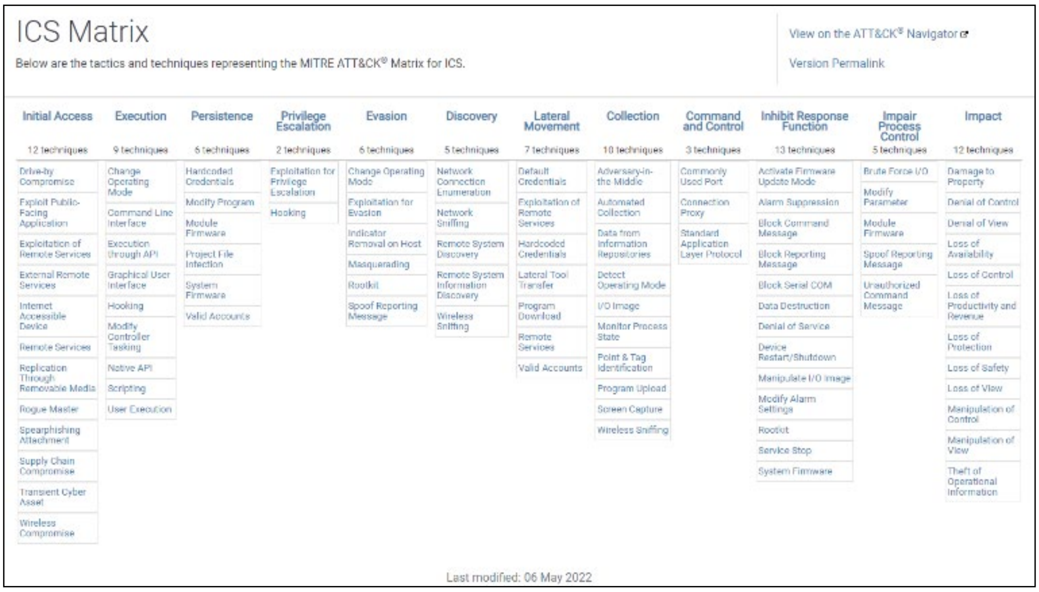
Consensus Logic Exploitation

- **Circumvent Voting Controls**
- **Covert Malicious Proposal**
- **51% Attack**

Funds Exfiltration

- **Use Anonymizing Services**
- **Mint tokens outside of Network**
- **Siphon Funds**

Graph 14 – MITRE ATT&CK ICS Matrix



ICS Matrix

Below are the tactics and techniques representing the MITRE ATT&CK® Matrix for ICS.

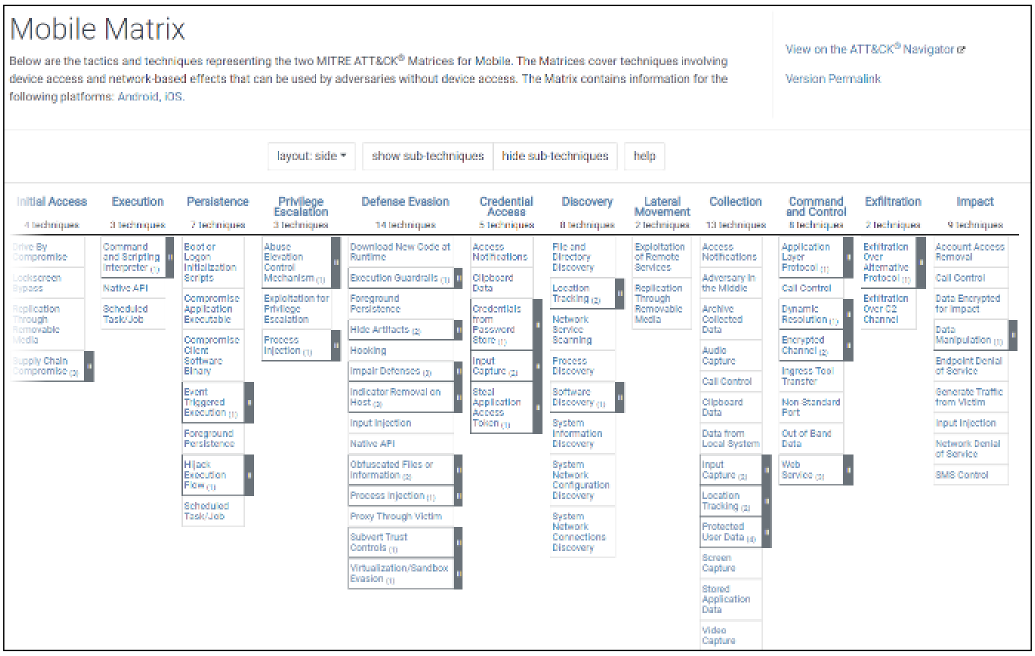
View on the ATT&CK® Navigator

Version Permalink

Initial Access	Execution	Persistence	Privilege Escalation	Evasion	Discovery	Lateral Movement	Collection	Command and Control	Inhibit Response Function	Impair Process Control	Impact
12 techniques	9 techniques	6 techniques	2 techniques	6 techniques	5 techniques	7 techniques	10 techniques	3 techniques	13 techniques	5 techniques	12 techniques
Drive-by Compromise	Change Operating Mode	Hardcoded Credentials	Exploitation for Privilege Escalation	Change Operating Mode	Network Connection Enumeration	Default Credentials	Adversary-in-the-Middle	Commonly Used Port	Activate Firmware Update Mode	Brute Force I/O	Damage to Property
Exploit Public-Facing Application	Command Line Interface	Modify Program	Hooking	Exploitation for Evasion	Network Sniffing	Exploitation of Remote Services	Automated Collection	Connection Proxy	Alarm Suppression	Modify Parameter	Denial of Control
Exploitation of Remote Services	Execution through API	Module Firmware		Indicator Removal on Host	Remote System Discovery	Hardcoded Credentials	Data from Information Repositories	Standard Application Layer Protocol	Block Command Message	Module Firmware	Denial of View
External Remote Services	Graphical User Interface	Project File Infection		Masquerading	Remote System Information Discovery	Lateral Tool Transfer	Detect Operating Mode		Block Reporting Message	Spoof Reporting Message	Loss of Availability
Internet Accessible Device	Hooking	System Firmware		Rootkit	Wireless Sniffing	Program Download	I/O Image		Block Serial COM	Unauthorized Command Message	Loss of Control
Remote Services	Modify Controller Tasking	Valid Accounts		Spoof Reporting Message		Remote Services	Monitor Process State		Data Destruction		Loss of Productivity and Revenue
Replication Through Removable Media	Native API					Valid Accounts	Point & Tag Identification		Denial of Service		Loss of Protection
Rogue Master	Scripting						Program Upload		Device Restart/Shutdown		Loss of Safety
Spearphishing Attachment	User Execution						Screen Capture		Manipulate I/O Image		Loss of View
Supply Chain Compromise							Wireless Sniffing		Modify Alarm Settings		Manipulation of Control
Transient Cyber Asset									Rootkit		Manipulation of View
Wireless Compromise									Service Stop		Theft of Operational Information
									System Firmware		

Last modified: 06 May 2022

Graph 15 – MITRE ATT&CK Mobile Matrix



Mobile Matrix

Below are the tactics and techniques representing the two MITRE ATT&CK® Matrices for Mobile. The Matrices cover techniques involving device access and network-based effects that can be used by adversaries without device access. The Matrix contains information for the following platforms: Android, iOS.

View on the ATT&CK® Navigator

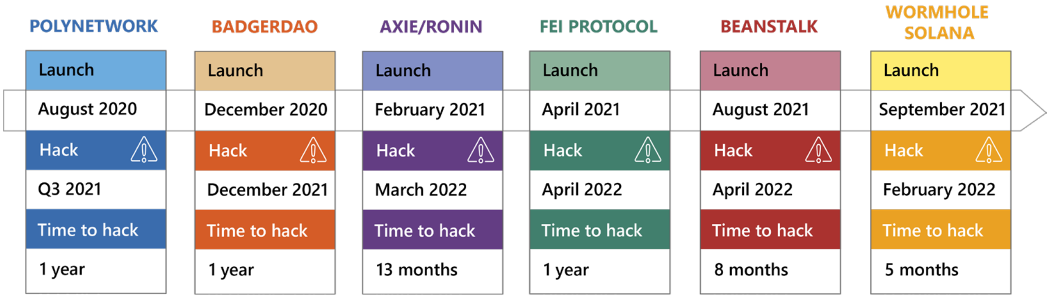
Version Permalink

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration	Impact
4 techniques	3 techniques	7 techniques	3 techniques	14 techniques	5 techniques	8 techniques	2 techniques	13 techniques	8 techniques	2 techniques	9 techniques
Drive By Compromise	Command and Scripting Interpreter (1)	Boot or Logon Initialization Scripts	Abuse Elevation Control Mechanism (1)	Download New Code at Runtime	Access Notifications	File and Directory Discovery	Exploitation of Remote Services	Access Notifications	Application Layer Protocol (1)	Exfiltration Over Alternative Protocol (1)	Account Access Removal
Lockscreen Bypass	Native API	Compromise Application Executable	Exploitation for Privilege Escalation	Execution Guardrails (1)	Clipboard Data	Location Tracking (2)	Replication Through Removable Media	Adversary in the Middle	Call Control	Exfiltration Over C2 Channel	Call Control
Replication Through Removable Media	Scheduled Task/Job	Compromise Client Software Binary	Process Injection (1)	Foreground Persistence	Credentials from Password Store (1)	Network Service Scanning		Archive Collected Data	Dynamic Resolution (1)		Data Encrypted for Impact
Supply Chain Compromise (2)		Event Triggered Execution (1)		Hide Artifacts (2)	Input Capture (2)	Process Discovery		Audio Capture	Encrypted Channel (2)		Data Manipulation (1)
		Foreground Persistence		Hooking	Steal Application Access Token (1)	Software Discovery (1)		Call Control	Ingress Tool Transfer		Endpoint Denial of Service
		Hijack Execution Flow (1)		Impair Defenses (3)		System Information Discovery		Clipboard Data	Non Standard Port		Generate Traffic from Victim
		Scheduled Task/Job		Indicator Removal on Host (3)		System Network Configuration Discovery		Data from Local System	Out of Band Data		Input Injection
				Input Injection		System Network Connections Discovery		Input Capture (2)	Web Service (3)		Network Denial of Service
				Native API				Location Tracking (2)			SMS Control
				Obfuscated Files or Information (2)				Protected User Data (4)			
				Process Injection (1)				Screen Capture			
				Proxy Through Victim				Stored Application Data			
				Subvert Trust Controls (1)				Video Capture			
				Virtualization/Sandbox Evasion (1)							

Attack timeline/mean time to attack (MTTA)

In addition to the mapping of the six attacks to the framework, it was identified that these DeFi implementations were also subjected to substantial cyber attacks shortly after they were launched, with this time span ranging within a few months to a year (see Graph 16). Based on this small sample of six attacks, the "mean time to attack" is calculated to be a roughly 10-month period between a DeFi's launch date and the day of its compromise. This is a key point for any sponsor of a CBDC – it is vital to be well prepared for such attacks.

Graph 16 – Attack timeline of the six DeFi hacks



POLYNETWORK	BADGERDAO	AXIE/RONIN	FEI PROTOCOL	BEANSTALK	WORMHOLE SOLANA
Launch	Launch	Launch	Launch	Launch	Launch
August 2020	December 2020	February 2021	April 2021	August 2021	September 2021
Hack ⚠	Hack ⚠	Hack ⚠	Hack ⚠	Hack ⚠	Hack ⚠
Q3 2021	December 2021	March 2022	April 2022	April 2022	February 2022
Time to hack	Time to hack	Time to hack	Time to hack	Time to hack	Time to hack
1 year	1 year	13 months	1 year	8 months	5 months

5.2 Future work

Additional research is needed on the cascading actions that are suggested by the release of this whitepaper, such as proposing extensions to MITRE and possible measures to refresh other security frameworks. Significant further R&D work is required in each of these domains before it can be applied at the scale required for the financial industry. Two areas for future studies are proposed below.

Call for help to focus on closing the gaps using MITRE ATT&CK

Crowdsourcing is an effective way to tap into a diverse range of skills and perspectives to help gather and map attacks and mitigations; propose new tactics, techniques, sub-techniques, procedures; and suggest security controls to TTPs that are commonly used in DLT attacks.

Following that, a future study is also suggested to propose an official extension of the MITRE ATT&CK framework by introducing a new DeFi-related Matrix, similar to the Mobile and ICS Matrix.

Extension of other cyber frameworks and standards

Similar to the analysis that we have performed on the MITRE Attack framework, we suggest reviewing other cyber security standards and frameworks to further augment them with DLT-specific controls that would benefit CBDC implementations and DeFi.

Once the extension to the MITRE framework mentioned in (a) has been realised, it will also assist in the identification of potential areas that require augmentation within the commonly used security standards and frameworks.

6. Conclusion

This paper aimed to identify gaps in existing cyber security frameworks and/or cyber threat models to address CBDC cyber threats. While there are no known successful attacks on CBDCs, DLT, smart contracts, digital identities, and immutable data are typical underlying technology used within existing CBDC implementations or suggested in many CBDC architectures.

By exploring notable DeFi attacks that exploit DLT components, it is clear that there are gaps in existing MITRE ATT&CK threat modelling techniques. Thus, these techniques may not adequately address the threats and associated security controls to properly protect CBDC systems that make use of DLT or smart contracts. This conclusion may apply more broadly to CBDC systems as a whole, given their novel nature and the need to assess whether existing frameworks are fit for purpose. It is possible that other frameworks will show similar gaps. Specifically, although the majority of existing TTPs can be used to model the attacks, some will require slight modification. At the same time, there may also be new attack vectors that do not fit within the framework, and which would necessitate the creation of new TTPs.

It is imperative for central banks to take action to prevent and mitigate these adversarial attacks. Aside from establishing the presence of gaps, this analysis has also highlighted that general cyber security standards are still applicable to CBDCs and DeFi systems, and central banks should adhere to these standards to safeguard their systems against commonly recognised attack types. Additionally, while the attacks within these six hacks in the DeFi space have been analysed, there are many other historical DeFi hacks as well as future CBDC attacks that would require mapping efforts. Given a given CBDC implementation may use novel or more traditional technology, or more likely a mixture of both, it is important to draw on examples from a variety of contexts to create the best defence against potential threats. A well mapped catalogue of threats is key if CBDC implementations are to adequately address and mitigate cyber attacks effectively.

The use of crowdsourcing is proposed to help catalogue and map attacks and mitigations; suggest new tactics, techniques, sub-techniques, and procedures; and use the MITRE ATT&CK framework to recommend security controls against TTPs that are commonly used in DLT attacks. Additionally, a future study to propose an official extension of the MITRE ATT&CK framework is suggested.

Annex A: TTP mapping of notable DeFi attacks

The following information contains the complete research results, with analysis added to the material taken from public sources.

Each table below represents a technique/sub-technique that has been extracted from the six DeFi hacks and mapped according to MITRE ATT&CK's "New Technique" contribution template.[14]

Poly Network

Details of the attack

The attacker first identified specific vulnerabilities in the Poly Network smart contract code (the contract "EthCrossChainManager") that allowed them to manipulate the "message" parameter and bypass the normal authorisation procedures.

- Contract "EthCrossChainManager", which has the right to trigger messages from another chain to the Poly chain specifying a target poly smart contract, contains the *_method* field to specify the hash of the method to be called. This field can be set by the user and is limited to 32-bit truncation of a 256-bit hash.
- "EthCrossChainManager" is wrongly the owner of "EthCrossChainData", which is responsible for setting and managing a list of public keys of "authenticator nodes" (Keepers) that manage the wallets in the underlying liquidity chains. In other words, EthCrossChainData can decide who has the privilege of moving the large amount of funds contained within Poly's Binance wallet, Ethereum wallet etc.

The attacker computed the 32-bit ID for putCurEpochConPubKeyBytes:

```
ethers.utils.id ('putCurEpochConPubKeyBytes(bytes)').slice(0, 10)'0x41973cd9'
```

The attacker brute-forced a string that, if set as *_method* in the code snippet above, gives the same 32-bit value. In this case the attacker used the string "f1121318093":

```
ethers.utils.id ('f1121318093(bytes,bytes,uint64)').slice(0, 10)'0x41973cd9'
```

With this altered "message" parameter, the attacker was able to trick the smart contract into believing that they had the necessary authorisation to initiate transfers.

(Sub-)technique name	**Brute forcing of contract id hash**
Tactic	Credential access

[14] attack.mitre.org/resources/contribute/.

Platform	DLT with smart contracts
Required permissions	User
Sub-techniques	Data sources: smart contract emitted events.
Description	Smart contracts can call methods in other contracts using their hashes. If the only constraint on calling a method is knowing its hash and the hash is not long enough, a malicious user can brute force the contract hash to execute a method they are not allowed to use.
Detection	Monitor the logs for multiple calls with wrong hashes/failed authorisations.
Mitigation	• Configure long enough hashes for the fields used to call other methods (eg 256-bits). • Ensure that fields that call other methods, especially privileged methods, are unable to be set by users. Or authorisation should be enforced when calling privileged methods. • Enable account lockouts or similar enforcement when multiple failed authorisations have been detected.
Adversary use	Poly Network Hack: https://research.kudelskisecurity.com/2021/08/12/the-poly-network-hack-explained/.
Additional references	-

Details of the attack (continued)

The attacker then called a cross-chain transaction from the Ethereum network to the Poly Network by using the bypassed string. The attacker abused the fact that "EthCrossChainManager" was wrongly the owner of "EthCrossChainData", which is responsible for setting and managing a list of public keys of "authenticator nodes" (Keepers) that manage the wallets in the underlying liquidity chains. This triggered EthCrossChainManager into calling the function putCurEpochConPubKeyBytes within EthCrossChainData and demanding the attacker's public key to be registered as a Keeper. EthCrossChainData executed the command, since EthCrossChainManager is its owner.

Once the transaction was executed and the attacker was granted the status of Keeper for the Ethereum blockchain, the attacker proceeded into using the corresponding secret key in their possession to funnel tokens out of Poly's Ethereum wallet into their own wallet.

In other words, the attacker escalated the privilege of the smart contract ownership and was able to become a keeper and move funds contained within Poly's wallets.

(Sub-)technique name	**Abuse of smart contract hierarchical ownership**
Tactic	Privilege escalation
Platform	DLT with smart contracts
Required permissions	User
Sub-techniques	Data sources: smart contract emitted events.
Description	Smart contracts can own other smart contracts. If a user-side smart contract has ownership of a privileged contract, the user can call methods in the privileged smart contract without being authorised to do so.
Detection	Monitor the logs for anomalous calls to privileged contracts.
Mitigation	Ensure privileged contracts are not owned by or are segregated from common user-side contracts via source code reviews.
Adversary use	Poly Network hack: https://research.kudelskisecurity.com/2021/08/12/the-poly-network-hack-explained/.
Additional references	-

(Sub-)technique name	**Bypass smart contract calling constraints**
Tactic	Privilege escalation
Platform	DLT with smart contracts
Required permissions	User
Sub-techniques	Data sources: smart contract emitted events.
Description	Smart contracts can own other smart contracts. If a user-side smart contract has ownership of a privileged contract, the user can call methods in the privileged smart contract without being authorised to

	do so. Some developers place constraints on the calling of methods but, with the right information, these constraints can be bypassed using techniques such as brute force.
Detection	Monitor the logs for anomalous calls to privileged contracts.
Mitigation	Configure appropriate constraints to call privileged methods (eg 256-bits or enforcing additional authorisation).
Adversary use	Poly Network hack: https://research.kudelskisecurity.com/2021/08/12/the-poly-network-hack-explained/.
Additional references	-

Axie/Ronin

Details of the attack

Axie Infinity is built on the Ronin Network, an Ethereum-linked sidechain developed by Sky Mavis, which serves as a bridge for users to transfer their assets from other ecosystems into Ronin and vice versa.

Ronin uses a "proof of authority" system for signing transactions. In order to recognise a deposit event or a withdrawal event, five out of the nine validator node signatures are needed to approve transactions. Five of these validator private keys were hacked; four Sky Mavis validators and one Axie DAO.

This traces back to November 2021 when SkyMavis requested help from the Axie DAO to distribute free transactions due to an immense user load. The Axie DAO allowlisted SkyMavis to sign various transactions on its behalf. This was discontinued in December 2021, but the Axie DAO validator IP was still on the allowlist.

A spear-phishing email was sent masquerading as a recruitment offer with a high-paying job offer to a senior developer at Sky Mavis, enticing the recipients to download malware-laced cryptocurrency applications. The targeted user downloaded the Tradertraitor/AppleJeus malware, which describes a series of malicious applications written using cross-platform JavaScript code with the Node.js runtime environment using the Electron framework. The malicious applications were derived from a variety of open-source projects and purported to be cryptocurrency trading or price prediction tools. TraderTraitor's campaigns feature websites with modern design advertising the alleged features of the applications.

After gaining entry into Sky Mavis's systems, the adversary performed lateral movement and obtained the private keys for four SkyMavis validator nodes.

(Sub-)technique name	Exploitation of remote services
Tactic	Lateral movement
Platform	DeFi platform
Required permissions	Admin
Sub-techniques	-
Description	Adversary performed lateral movement The attacker compromised third-party systems and then exploited this allowlist to generate a signature from the third-party validator controlled by Axie DAO. Sky Mavis includes a gas-free RPC node that was used to get this fifth signature.
Detection	• Detection of the lateral movement. • Monitor all usage of transaction signing.
Mitigation	• Revoke authentication from third party that expired or enable time-limited access. • Apply elevated authentication of third-party access.
Adversary use	Here is the example of the Ronin hack using this technique: https://halborn.com/explained-the-ronin-hack-march-2022/. https://medium.com/uno-re/biggest-crypto-hack-of-all-time-a-breakdown-of-the-ronin-network-hack-ef8d9e25ba6b.
Additional references	-

Details of the attack (continued)

The adversary obtained the private keys for four validator nodes through the compromise of SkyMavis's systems.

The validator key scheme is set up to be decentralised so that it limits an attack vector such as this, but, using a "backdoor" (a gas-free RPC), the adversary compromised a fifth validator run by Axie DAO (a community-run organisation supporting the Axie Infinity project).

(Sub-)technique name	Multiple keys (obtain majority of private keys)
Tactic	Credential access

Platform	DeFi platform
Required permissions	User
Sub-techniques	-
Description	Adversary obtained the private keys for four SkyMavis validator nodes.
Detection	• Enable and monitor the usage of private keys and enable the detection of private keys that have been compromised and maliciously used. • Enable the detection of majority votes if performed by either a singular resource or suspicious resources. (Ronin Network was a completely centralised network at the time of the hack.) • Monitor the decentralisation status of the network.
Mitigation	• Enable the authentication of validator nodes. • Revoke third-party access that has expired. • Avoid the storage of multiple private keys in the same place.
Adversary use	Here is the example of Ronin hack using this technique: https://halborn.com/explained-the-ronin-hack-march-2022/. https://medium.com/uno-re/biggest-crypto-hack-of-all-time-a-breakdown-of-the-ronin-network-hack-ef8d9e25ba6b.
Additional references	-

Details of the attack (continued)

Obtaining the majority share of the validator nodes needed to approve transactions, hackers then wrote their own transactions to the chain and validated them using the stolen keys. They withdrew most of the funds from the Ronin bridge in just two transactions.

(Sub-)technique name	**Acquire infrastructure (crypto accounts) – Valid crypto accounts to transfer and then launder stolen funds**
Tactic	Resource Development
Platform	DeFi platform
Required permissions	User
Sub-techniques	-

Description	Acquire infrastructure (crypto accounts) – Valid crypto accounts to transfer and then launder stolen funds, including the creation of funding accounts to pay transaction fees (BadgerDAO).
Detection	Malicious account detection, and the reporting of malicious/compromised accounts.
Mitigation	• Develop capabilities for users to report malicious accounts and suspend them. • Enable extensive checks for account creations.
Adversary use	Here is the example of the Ronin hack using this technique: https://halborn.com/explained-the-ronin-hack-march-2022/. https://medium.com/uno-re/biggest-crypto-hack-of-all-time-a-breakdown-of-the-ronin-network-hack-ef8d9e25ba6b.
Additional references	-

BadgerDAO

Details of the attack

When a web3 app wants its user to perform an action on the blockchain (eg send token A to app, to get token B from it), it initiates the following four steps:

1. It prepares the requested transaction for the user
2. It sends the transaction to the user's wallet via a bridge or gateway (like WalletConnect or a web extension)
3. The user signs the transaction via their wallet
4. The user sends the signed transaction to the blockchain via their wallet

When the web3 app needs to be paid by an ERC20 token, users cannot just simply send the required amount of tokens to the app's smart contract address. Instead, users need to approve the app's request to withdraw tokens, so the app can withdraw these tokens on the user's behalf later on.

In practice, many apps request for the approval of a practically unbounded amount of tokens instead of requesting the approval of just the required amount. This is often done in order to reduce transaction costs. Because transaction fees can be quite high, instead of requesting approval for each transaction – which itself requires a processing fee – this "unbounded token request" has become an industry practice embraced by many.

As a result, web3 users are accustomed to approving relatively unlimited amounts. The user's only line of defence is making sure they are interacting with a trustworthy app and that the approval request is logical in the context of their current interaction.

BadgerDAO's hack was caused by a phishing incident via "a maliciously injected snippet" provided by Cloudflare Workers, a serverless application platform that runs on its cloud network. The hacker used a compromised API key that was created without the knowledge or authorisation of Badger engineers to periodically inject the malicious code that affected a subset of its customers. This code generated rogue transaction approvals, taking advantage of a visual design issue in how crypto wallets ask their users to approve of certain transactions. If approved by users, the code would allow the hackers to pull funds to their own wallets instead of those controlled by BadgerDAO at a future date.

(Sub-)technique name	**User execution**
Tactic	Execution
Platform	DeFi platform
Required permissions	User
Sub-techniques	**This is a sub-technique of TXXX –** Authorisation of infinite approvals
Description	The scripts intercepted transactions and prompted users to allow a foreign address to operate on the ERC-20 tokens in their wallet.
Detection	• Inform users when there are UI changes or enable user self-reports of the incident when any suspicious activities are found. • Enable and monitor application logs, newly executed processes, messaging and/or other artifacts that may rely upon specific actions by a user in order to gain execution. Monitor usage of third-party applications for anomalous inputs.
Mitigation	• To disallow authorise infinite approvals. • Require additional owner verification before allowing foreign addresses to transfer on behalf of owners. • Ensure that third-party application APIs require additional authorisation on first-time usage. • Constantly refresh third-party application APIs used within the systems.
Adversary use	Here is the example of the BadgerDAO hack using this technique: https://zengo.com/the-badgerdao-hack-what-really-happened-and-why-it-matters/. https://www.coindesk.com/business/2021/12/10/badgerdao-reveals-details-of-how-it-was-hacked-for-120m/.

Additional references	-

Wormhole/Solana

Details of the attack

Solana's Wormhole is a communication bridge enabling the transfer of tokenised assets seamlessly across different blockchains, benefiting from Solana's high speed and low cost. Wormhole has a set of guardians that sign off on transfers between chains. Bridges like Wormhole work by having two smart contracts — one on each chain. In this case, there was one smart contract on Solana and one on Ethereum. A bridge like Wormhole takes an ethereum token, locks it into a contract on one chain, and then on the chain at the other side of the bridge, it issues a parallel token.

Wormhole's *complete_wrapped* function is triggered whenever someone mints Wormhole ETH on Solana. One of the parameters that this function takes is a *transfer_message*, a message signed by the guardians that says which token to mint and how much. This *transfer_message* is actually a contract on Solana and is created by triggering a function called *post_vaa*, which checks if the message is valid by checking the signatures from the guardians.

post_vaa does not actually check the signatures, instead, another smart contract is created by calling the *verify_signatures* function. One of the inputs to the *verify_signatures* function is a Solana built-in system program which contains various utilities the contract can use. The signature verification was outsourced to this program, which was where the bug was.

(Sub-)technique name	**Manipulate signature verification process**
Tactic	Consensus Logic Exploitation/Defence Evasion
Platform	Opensource DeFi platform
Required permissions	User
Sub-techniques	Verification by unauthorised addresses
Description	The attackers manipulated VAA verification and substitute the "Sysvar: Instructions" address with their own supplied address. Because verification process did not properly validate all input accounts, the signature verification was performed by a valid address.
Detection	Monitor changes in verification process addresses.

Mitigation	The signature verification must be performed only by a whitelisted address.
Adversary use	Here is the example of Solana's wormhole hack using this technique: https://extropy-io.medium.com/solanas-wormhole-hack-post-mortem-analysis-3b68b9e88e13#:%7E:text=The%20Wormhole%20bridge%20was%20hacked,(Wormhole%20ETH)%20on%20Solana.
Additional references	-

<u>Details of the attack (continued)</u>

The Wormhole contracts used the function "*load_instruction_at*" to check that the "*Secp256k1*" function was called first. By looking at Github's internal commits, the "*load_instruction_at*" function was deprecated on 13 January by the team as it did not check that the signature verification was being performed by a whitelisted address, also known as a "*system address*". The system address was supposed to be provided as the program to be executed (eg the third-to-last program input) in the function "*load_instruction_at*", but from the analysis of the *verify_signatures* transaction for the fake deposit of 120,000 ETH; the system address was substituted for a program's address, the equivalent of an Ethereum smart contract that did not check signatures at all.

(Sub-)technique name	**Source code and programming logic analysis**
Tactic	Collection
Platform	Opensource DeFi platform
Required permissions	User
Sub-techniques	**This is a sub-technique of T1213.003 –** Data from Information Repositories: Code Repositories
Description	Attackers monitoring updates to code repositories and searching for common programming mistakes on the opensource DeFi platform.
Detection	- Apply DLP to protect critical functions. - Monitor for anomalous inputs.
Mitigation	- Perform secure code reviews using specialised tools, preferably tools that are able to analyse smart contracts.

	- Consider periodic reviews of accounts and privileges for critical and sensitive code repositories. Scan code repositories for exposed credentials or other sensitive information, as well as updates to the code. - Ensure that deprecated codes or functions are prevented from being utilised. - Use multi-factor authentication for logons to code repositories. - Impose view restrictions for sensitive code. Enforce the principle of least-privilege. Consider implementing access control mechanisms that include both authentication and authorisation for code repositories. - Develop and publish policies that define acceptable information to be stored in code repositories.
Adversary use	Here is the example of Solana's wormhole hack using this technique: https://extropy-io.medium.com/solanas-wormhole-hack-post-mortem-analysis-3b68b9e88e13#:%7E:text=The%20Wormhole%20bridge%20was%20hacked,(Wormhole%20ETH)%20on%20Solana.
Additional references	-

Beanstalk

Details of the attack

Beanstalk uses a decentralised governance protocol with an *emergencyCommit* function where it can be approved by a supermajority (a two thirds vote) and implemented after 24 hours rather than going through the standard process. The voting power is controlled using donations to Beanstalk Diamond contract.

A combination of the following actions were used in the hack against Beanstalk:

- Flash loan, which eliminates the need for collateral as the initial loan must be repaid at the end of the transaction, was crucial to the attacker obtaining majority control of Beanstalk's on-chain governance.
- Subversion of governance mechanism — the *emergencyCommit()* function that permitted the attacker to immediately execute the proposal on-chain. Normally, the execution of a BIP on-chain requires a minimum of seven days but leveraging the *emergencyCommit* function helped the attacker bypass the seven-day requirement (two thirds supermajority is still needed for execution).

The attacker obtained funds through a flash loan from TornadoCash and deposited 212,858 BEAN into the Beanstalk Silo, which facilitated the generation of sufficient Stalk (this is the yield-generating governance token) and Seed (vested Stalk) that enabled the attacker to create two malicious Beanstalk Improvement Proposals (BIP18 and BIP19) – two smart contract proposals that eventually drained the protocol's funds to the attacker's account and a Ukraine aid donation account.

After a one-day waiting period, the attacker used the flash loan (of approximately $1 billion) to obtain 50 million DAI, 500 million USDC and 150 million USDT from Aave, 32 million BEAN from Uniswap v2 and 11.6 million LUSD from SushiSwap. These tokens were used to add liquidity in Curve pools with BEAN for governance voting.

Next, the attacker deposited the aforementioned assets in the Beanstalk Silo and obtained sufficient Stalk and Seed. This provided the attacker with a 70–79% control of the governance protocol's votes, more than the two thirds vote required for approval, allowing the attacker to unilaterally approve their own proposal and execute the emergencyCommit() function.

(Sub-)technique name	**Circumvent voting majority controls: Obtain majority of voting rights**
Tactic	Consensus Logic Exploitation
Platform	DeFi platform
Required permissions	User
Sub-techniques	-
Description	Adversary obtained the supermajority of the voting rights allowing them to approve their own contracts.
Detection	• Enable the detection of majority votes if performed by either a singular resource or suspicious resources. • Enable the detection of a change in majority votes.
Mitigation	• Prevent the exactment of supermajority rights until a review performed and approved.
Adversary use	Beanstalk Protocol Hack: https://halborn.com/explained-the-beanstalk-hack-april-2022/. https://medium.com/coinmonks/beanstalk-exploit-a-simplified-post-mortem-analysis-92e6cdb17ace.
Additional references	-

Details of the attack (continued)

The attacker deployed and voted for a fake protocol improvement proposal (BIP18) that drained the pool fund and transferred the tokens to the attacker. This enabled the attacker to get approximately $76 million in profit with the remaining $106 million used to repay the flash loan, which was funnelled through the coin mixing tool TornadoCash.

The attacker managed to deceive the community by making them believe BIP18 was meant only to donate money to the Ukraine donation address. Once the attacker acquired a majority governance stake through the flash loan, the *emergencyCommit* function helped the attacker execute the proposal immediately after voting on it.

(Sub-)technique name	**Covert malicious proposal**
Tactic	Consensus logic exploitation
Platform	DeFi platform
Required permissions	User
Sub-techniques	Listing of proposals
Description	Network participants can make proposals to change the way the network works. These proposals can have covert malicious code that might not be immediately visible to reviewers.
Detection	Monitor for anomalous changes to the network.
Mitigation	- Auditing of new proposals. - Run new proposals on a test network. - Providing sufficient time between voting and execution so that users can review the proposals.
Adversary use	Beanstalk Protocol Hack: https://halborn.com/explained-the-beanstalk-hack-april-2022/. https://medium.com/coinmonks/beanstalk-exploit-a-simplified-post-mortem-analysis-92e6cdb17ace.
Additional references	-

Fei Protocol

Details of the attack

Fei Protocol, a rapidly growing algorithmic stablecoin built natively for the Defi ecosystem, merged with Rari Capital, a permissionless lending protocol that allows users to create Fuse pools that anyone with a wallet can access from anywhere to lend or borrow ERC-20 tokens with no minimum funds required of users. This merger helped to further bootstrap liquidity for the Fuse pools, with Fei Protocol providing the necessary initial liquidity.

In early 2021, Fei Protocol used code forked from Compound, an Ethereum money market platform. Within the code, multiple re-entrancy vulnerabilities that involve smart contracts calling each other to move funds without appropriate checks were fixed in a past update, but some vulnerable functions were overlooked. Fei made certain changes to the code, however, and despite undergoing frequent audits, the flaw was not discovered until the hack.

On 1 April 2022, Rari Capital released a Security Upgrade Report stating they had patched a security issue relating to Fuse pools. This patch fixed known vulnerabilities from the Compound code by blocking re-entrancy on functions that required it. Although many of their system's functions, were patched, the *exitMarket()* function was left out.

The attacker took advantage of two functions in the Fei Protocol's contracts: *exitMarket()* and *borrow*.

- The *exitMarket()* function verifies that a deposit is no longer used as collateral for a loan and then allows it to be withdrawn.
- The *borrow* function allows a user to take out a loan using a deposited asset as collateral and does not follow the check-effect-interaction pattern, leaving it vulnerable to attack.

When the attacker receives ETH, they can still call the *exitMarket()* function, even when a global re-entrancy lock is active. This design flaw that failed to follow the check-effect-interaction pattern allowed the attacker to make a re-entrant call before the borrow records were updated.

(Sub-)technique name	**Re-entrancy attack**
Tactic	Fund exfiltration
Platform	Open source DeFi platform
Required permissions	User

Sub-techniques	-
Description	A re-entrancy attack occurs when a function makes an external call to another untrusted contract. Then the untrusted contract makes a recursive call back to the original function in an attempt to drain funds. When the contract fails to update its state before sending funds, the attacker can continuously call the withdraw function to drain the contract's funds. This is a fundamental logic error allowing for repeated funds transfer before the balance is updated.
Detection	Identify and monitor for anomalous usage of critical functions.
Mitigation	• Perform secure code reviews using specialised tools, preferably tools that are able to analyse smart contracts. • Ensure that dependent code referenced from third parties is frequently checked for updates.
Adversary use	The Fei Protocol Hack (April 2022): https://halborn.com/explained-the-fei-protocol-hack-april-2022/.
Additional references	-

References

Atlantic Council (2023): *Central Bank Digital Currency Tracker*, www.atlanticcouncil.org/cbdctracker/

Auer, R, G Cornelli and J Frost (2020): "Rise of the central bank digital currencies: drivers, approaches and technologies", *BIS Working Papers No 880,* August.

——— (2023): "Updated dataset on CBDC projects around the world", BIS, Feb, www.bis.org/publ/work880_data_jan23.xlsx

Basra, J. and T Kaushik (2020): "MITRE ATT&CK® as a Framework for Cloud Threat Investigation", *Berkeley - Centre for Long-Term Cybersecurity (CLTC)*.

Bech, M and G Rodney (2017): "Central bank cryptocurrencies", *BIS Quarterly Review*, September, pp 55-70.

BIS Innovation Hub (2023): *Polaris Framework for Secure and Resilient CBDCs*. July.

Check Point (2022): "Third quarter of 2022 reveals increase in cyberattacks and unexpected developments in global trends", *Check Point Research*, blog.checkpoint.com/2022/10/26/third-quarter-of-2022-reveals-increase-in-cyberattacks/

Cloud Security Alliance (2021): *Top 10 Blockchain Attacks, Vulnerabilities & Weaknesses*, cloudsecurityalliance.org/artifacts/top-10-blockchain-attacks-vulnerabilities-weaknesses/

Columbus, L (2018): "Protecting Your Company When Your Privileged Credentials Are For Sale", Forbes, August, www.forbes.com/sites/louiscolumbus/2018/08/21/protecting-your-company-when-your-privileged-credentials-are-for-sale/?sh=bb5e187c8041

Committee on Payments and Market Infrastructures (CPMI), Markets Committee (2018): *Central Bank Digital Currencies*, March.

Cybersecurity & Infrastructure Security Agency (CISA) (2022): *TraderTraitor: North Korean State-Sponsored APT Targets Blockchain Companies*, www.cisa.gov/news-events/cybersecurity-advisories/aa22-108a

Lim, Y. (2023): *Most Damaging Methods of Crypto Hacks and Exploits in 2022.* February. www.coingecko.com/research/publications/crypto-hacks-exploits-by-method

Microsoft (2022): *Microsoft Digital Defense Report 2022.*

National Institute of Standards and Technology (NIST) (2012): *NIST Special Publication 800-30 Revision 1 - Guide for Conducting Risk Assessments*.

——— (2016): SP 800-150: Guide to Cyber Threat Information Sharing.

The European Union Agency for Cybersecurity (ENISA) (2022): *ENISA Threat Landscape 2022*

The MITRE Corporation: ATT&CK Sightings, attack.mitre.org/resources/sightings/

VMware (2023): *VMware Glossary*,
www.vmware.com/topics/glossary/content/allowlisting.html

Authors and acknowledgements

Authors

Jonathan Lee (Security Specialist, BIS Cyber Resilience Coordination Centre)

Wipawadee Auyporn, (Security Specialist, BIS Cyber Resilience Coordination Centre)

Sameh Mekhail (Deputy Head, BIS Cyber Resilience Coordination Centre)

David Whyte (Head, BIS Cyber Resilience Coordination Centre)

Beju Shah (Head of Nordic Centre, BIS Innovation Hub)

William Zhang (Adviser, BIS Innovation Hub Nordic Centre)

Ben Dovey (Adviser, BIS Innovation Hub Nordic Centre)

PA Consulting (Quality assurance)

Andreas Joachim Nielsen (Consultant)

Sam Lascelles (Managing Consultant)

Alexander McGill (Associate Partner)

David Biggin (Partner)