GoPlus Security

🚀 Overview

Mission Statement

Navigating billions of users to Web3 with safety.

What is GoPlus?

GoPlus Security is building Web3's first decentralized security layer, providing comprehensive protection across all blockchain networks. Through its open, permissionless, and user-driven architecture, GoPlus can be seamlessly integrated by any blockchain or project to protect their users throughout their entire transaction lifecycle. By leveraging AVS and cutting-edge AI powered security solutions, it conducts thorough risk analysis and delivers smart, efficient and decentralized security services for users. GoPlus aims to create a more secure and user-friendly Web3 on-chain interaction environment by filling the gap of security layer in the current blockchain's architecture, providing users with more effective and better-experienced on-chain security protection.

Since 2022, GoPlus has seen exponential growth, with daily API calls reaching 30 million and security coverage expanding to over 30 blockchain networks. The end user platform has protected over 12M wallets, prevented tens of billions in potential losses, and detected more than 800K malicious assets across different chains. Through partnerships with RPCs, Rollups, and RaaS projects, GoPlus is establishing a fully decentralized security network that scales across the entire Web3 ecosystem.

GoPlus will comprehensively build out its decentralized security network, deeply integrating security services across the entire lifecycle of transaction - from secure asset issuance and management to safe transactions. Through the SafeToken Protocol for secure token creation and management and GoPlus APP with GoPlus Intelligence for comprehensive transaction protection, GoPlus will ensure security is omnipresent throughout the Web3 ecosystem.

GoPlus Ecosystem Landscape

Web3 User Security Environment

In the Web3 environment, user security currently faces increasingly severe challenges, primarily centered around private key theft, phishing websites, Rug Pulls, and scam contracts. According to a 2023 report released by SlowMist, the Web3 ecosystem suffered losses as much as $2.486 billion due to various security incidents, including 191 hacker attacks, 267 Rug Pull incidents, and a multitude of online phishing scams. The report highlights that DeFi-related hacker attacks mainly target projects and protocols. However, for individual users, frequent phishing and scam activities are the most common threats. Unlike protocol-targeted attacks, those aimed at users have a broader impact, significantly degrading the Web3 experience, especially for newcomers lacking essential web3 security knowledge, making them susceptible to scams in the complex landscape of on-chain operations.

Furthermore, since 2022, there has been a rise in organizations creating "Wallet drainers", a type of malware related to cryptocurrency that has seen significant "success" over the past year, as outlined in **Scam Sniffer** 2023 security report These malicious software tools are deployed on phishing websites to deceive users into signing harmful transactions, consequently stealing assets from their wallets. These phishing activities persistently victimize ordinary users in various forms, resulting in significant financial losses for many who unwittingly sign malicious transactions. Through a model known as "Drainer as a service", these organizations develop a vast network of affiliates, leading to large-scale scamming and phishing operations. Moreover, as the methods of user-targeted attacks and defense mechanisms evolve, the nature of these attacks has become increasingly diverse and complex, posing potential traps in any interaction within the user's operational process. With the emergence of large-scale scams and phishing operations, the personal security issues of Web3 users are becoming more severe, indicating an urgent need for comprehensive and robust security solutions in the Web3 domain.

Main Issues about User Security

- **Missing User Security Layer in Blockchain's architecture:** The current blockchain technical architecture lacks inherent protection capabilities for their users. Once a user clicks the send button in their wallet, they are solely responsible for their transactions and cannot stop them from happening, even if the transactions are malicious or fraudulent. This leaves users vulnerable to potential security threats and scams.

- **High barrier for users:** While some products, including wallets, provide security alerts and reminders, users are still required to understand and make correct choices based on these notifications. This creates a certain barrier for users, as they need to comprehend the implications and cultivate a strong sense of security awareness. The current solutions do not adequately cater to the varying levels of technical understanding among users.

- **Fragmented Security Configurations:** With numerous blockchain networks, users are forced to rely on fragmented security tools to address their safety needs across different chains. This piecemeal approach is cumbersome and inefficient, leaving users unable to ensure that their transactions align with their personal security intentions. The lack of a unified, cross-chain security strategy hinders users from having full control over their assets' safety.

- **Centralized Security Services:** Many of the existing user security services are centralized, which poses certain risks. The lack of transparency in their data and services raises concerns about potential vulnerabilities and breaches of trust. This centralized approach is inconsistent with the decentralized spirit of Web3, which emphasizes openness, transparency, and user driven.

Why GoPlus?

In the current landscape of Web3 user security, the community faces pressing challenges that demand innovative and comprehensive solutions. GoPlus addresses these challenges head-on, offering a novel approach to the security conundrums of the Web3.

Comprehensive Security Lifecycle: The essence of Web3 security is not found in piecemeal solutions but in comprehensive, end-to-end protection. GoPlus stands out by providing a holistic security framework that guards every phase of user interaction and transaction within the blockchain environment. From initial transaction creation to final settlement, GoPlus protects users from risks at every step, covering all their security scenarios and closing gaps left by isolated solutions.

Standardized Security for Token Ecosystem: The prevalence of malicious assets across different blockchain networks has reached alarming levels, with a significant proportion of tokens potentially harboring security risks or malicious intent. To address this fundamental challenge, GoPlus provides comprehensive standards and solutions for secure token issuance and management through its SafeToken Protocol. By offering pre-audited contract templates and advanced liquidity management tools, GoPlus enables projects to launch with security built-in from the start, rather than as an afterthought. This proactive approach helps reduce the proliferation of malicious assets and creates a more trustworthy token ecosystem for all participants.

Unified Security Hub: The fragmented nature of current security services complicates the user experience and diminishes overall safety. Users crave a single, cohesive platform where they can manage and monitor their security across multiple blockchains. GoPlus Security responds to this need by creating a unified security hub, a one-stop solution for cross-chain security strategy management.

Seamless on-chain firewall: GoPlus introduces a groundbreaking on-chain firewall that integrates directly into the blockchain. The User Security Module creates a native security layer that automatically screens transactions against users' personalized security strategy, blocking risky ones in real-time at the RPC level or by integrated chains. This ensures uninterrupted, secure blockchain interactions where asset protection is inherent to the user experience.

Decentralized Security Services: Finally, the call for transparency and reliability in security services has never been louder. The decentralized ethos of Web3 demands security solutions that match its principles. GoPlus embodies this spirit by offering decentralized security services that ensure open, permissionless, and user driven. By leveraging decentralized network, GoPlus ensures that its security services are not only effective but also align with the core values of the Web3.

Conclusion

In conclusion, GoPlus Security is dedicated to providing Web3 users with a comprehensive, unified, seamless, and decentralized security solution. Faced with numerous challenges in the current Web3 user security landscape, such as the lack of native user protection capabilities in blockchain architecture, high security barriers for users, fragmented security configurations, and centralized security services, GoPlus offers innovative countermeasures.

GoPlus provides end-to-end security protection throughout the entire user transaction lifecycle, creates a unified cross-chain security management center, enables seamless secure interaction experiences through a native security layer deeply integrated with the blockchain, and offers transparent and reliable security services based on a decentralized network. GoPlus' solutions comprehensively cover users' security needs in the Web3 environment, bridging the gaps between current fragmented security solutions and embodying the decentralized, open, permissionless, and user-driven spirit of Web3.

💚 Get Started

For Developers 🛠️

- **Integrate GoPlus Intelligence:** Implement GoPlus Intelligence into your platform
- **SafeToken Protocol:** Implement secure token standards and use token liquidity lockers
- **Integrate GSM**: Implement GoPlus Security Module into your chains or RPC service

Key Resources:

- Documentation: **https://docs.gopluslabs.io**
- Token-Risk-Classification: **https://cryptousersecurity.github.io/token-risk-classification/**
- Github: **https://github.com/GoPlusSecurity**

For Users 🛡️

Protect your every transaction:

- GoPlus Browserr Extension
  - Multi-chain wallet scanner
  - Real-time smart risk alerts
  - Personal security dashboard
  - 24/7 AI Security Assistant
- GoPlus Web APP
  - Personal Security Dashboard
  - Multi-chain Wallet Scanner
  - Security Service Marketplace
  - Security Tasks & Rewards

For AVS Operators 💻

Contribute to network security and earn rewards

AVS Operator Guide

For GoPlus Ecosystem Developers 🔧

Build decentralized security services within the GoPlus Network:

Create SecWares

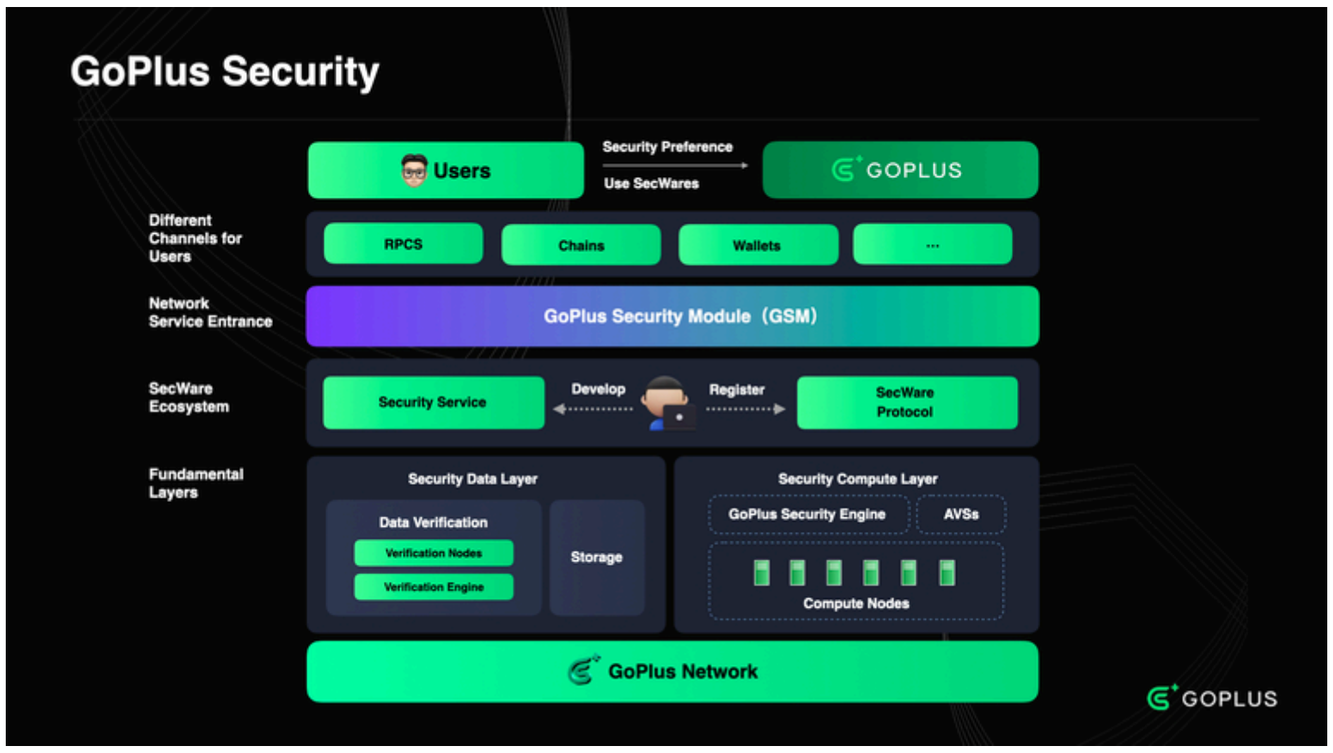
- Build security services using the SecWare Protocol
- Earn rewards from user security service fees

Join our community to connect with GoFam to stay updated on the latest developments:

- Discord
- Twitter
- Telegram

For technical support or questions, reach out to our team at service@gopluslabs.io

Architecture Overview



Fundamental Layers

GoPlus Security is a decentralized security data and security service network designed to cater to users' diverse security needs throughout the transaction process. The network is primarily composed of the Security Data Layer and the Security Compute Layer which operates on a permissionless, open, and user-driven basis, allowing any developer to join and provide corresponding security solutions based on users' security requirements at different stages of the transaction lifecycle, such as anti-scam, anti-phishing, and anti-MEV. This approach enables all security developers, data providers and compute node operators to participate and collaborate in offering superior security services to users, ultimately creating a more secure Web3 on-chain interaction environment.

Security Data Layer

By leveraging a decentralized approach to collect, process, and store security-related data, the network ensures the integrity, authenticity, and reliability of the data. This decentralized security data layer serves as a solid foundation for the network's security services, enabling more accurate and effective security solutions.

Security Compute Layer

The GoPlus Security leverages **AVS Operators**, which are distributed nodes responsible for executing security-related computations and validations. These operators handle tasks such as verifying transaction security analysis results, detecting potential security threats, and simulating transactions. By distributing the security computation workload across multiple AVS Operators, the network achieves enhanced scalability, fault tolerance, and improved resilience against single points of failure. **AVS Operators** ensure the stability and efficiency of the network by delivering real-time security services, which are essential to safeguarding users' on-chain interactions.

GoPlus Security Module

At the heart of the GoPlus Security lies the GoPlus Security Module (GSM), which serves as the entry point and conduit for the entire network's services. GSM integrates various security services into infrastructures and dApps, permeating every aspect of user interaction. With its highly pluggable, lightweight, easy-to-integrate, and multi-chain compatible features, GSM seamlessly integrates with the RPC services, chains, and various RaaS providers, filling the gaps in the architectural of the chains themselves regarding user security issues.

GoPlus APP

Moreover, GoPlus Security provides all users with a comprehensive security product called GoPlus App. Through this app, users can exercise full control over their asset security and configure various risk control and security policies tailored to their individual needs. GoPlus App and GSM form a close collaboration and interconnection, ultimately achieving a complete closed loop from security intent to secure transactions.

Conclusion

In summary, GoPlus Security offers a comprehensive security solution for the Web3 on-chain trading environment through its decentralized security data and service network, open security service ecosystem, highly pluggable and easy-to-integrate GSM, and user-centric security App. The network's architecture not only meets users' security needs but also provides ample opportunities for security developers and service providers to participate, demonstrating immense potential for future growth and adoption. With its emphasis on decentralized security data and innovative Security Compute Nodes, GoPlus Security is well-positioned to revolutionize user security of Web3.

Security Data Layer

In the last few years, GoPlus Network has experienced exponential growth, with user security data usage increasing by over 5000 times from what was recorded in 2022 and daily API calls reaching 21 million, demonstrating high levels of user trust. However, as we continue to grow and evolve, we recognize the importance of adopting a more decentralized approach to data generation and verification. Therefore, ensuring the integrity and reliability of user security risk data is of paramount importance for GoPlus Network. To address this critical need, we propose a decentralized Security Data Contribution and Verification Layer that harnesses the power of multi-party participation and automated verification processes.

This foundational layer of the GoPlus Network offers trustworthy, rich, and real-time security data via a decentralized security data system designed to tackle the complex landscape of Web3 user security by facilitating the collection, verification, and utilization of security-related data. The layered architecture ensures a comprehensive and effective approach to identifying and mitigating security risks, leveraging the collective wisdom and expertise of various stakeholders, including end-users, security researchers, developers, and third-party security service providers.

Security Data Layer

Data Contribution

Security Data Contributors Security data contributors form the foundation of the entire system. They provide valuable information about potential security risks and threats through various channels, including but not limited to:

End-users: Regular users of Web3 applications can report security issues they encounter, such as suspicious scam activities, phishing attempts, or rug-pulls.

Security researchers: Professional security researchers can contribute their findings on risks, security analysis, and other in-depth user security insights.

Third-party security companies & organizations: Specialized security firms can offer comprehensive threat intelligence and risk assessment reports.

Through incentivization and recognition mechanisms, we encourage broad participation to establish a comprehensive and diverse user security database.

Data Verification

To ensure the credibility and accuracy of the contributed data, we implement a multi-tiered decentralized verification mechanism. The Security Data Verification system consists of a Primary verification process and a Secondary verification process, working in tandem to validate the security data.

Primary verification

The Primary verification employs a multi-faceted approach to data verification, incorporating trusted third-party entities and automated computational methods:

Third-Party Verification Nodes: Reputable entities operate verification nodes that leverage their expertise and resources to assess the veracity of user-contributed information.

Computational Verification Nodes: Automated computational methods, are utilized to verify some specific types of security data, employing SecScan, advanced algorithms and AI techniques.

Auditors: Independent auditors oversee the verification process, ensuring compliance with established protocols and maintaining the integrity of the system.

Secondary verification

The Secondary verification is triggered when disputes arise in the Primary verification. It is composed of highly specialized security teams and institutions that focus on resolving controversies in Primary verification:

Elite Security Teams: Renowned security teams with extensive expertise in Web3 security are enlisted to investigate and resolve complex disputes such as SlowMist, Blocksec, etc.

Institutional Arbitrators: Respected institutions, such as respected university labs and Web3 industry leaders, act as impartial arbitrators to settle disagreements and provide final verdicts.

The Secondary verification ensures that any contentious issues are thoroughly examined and resolved by the most qualified experts in the field.

By seamlessly integrating security data contributors and the multi-tiered verification mechanism, we create a robust and resilient decentralized security data ecosystem. This innovative approach not only enhances the diversity, professionalism, and accuracy of risk data, but also providing users with a robust foundation for risk control and strengthening the underlying risk management models, ultimately serves to protect users' security. By working together, we can lay a solid foundation for the future of digital interactions, enabling all participants to explore the possibilities of a new paradigm of user security data.

Types of Risk Data

GoPlus has identified and prioritized a range of critical security data types. These data types serve as the backbone of our decentralized Security Data Contribution and Verification Layer, providing comprehensive insights into potential security risks and enabling mitigation strategies. Here's an overview of these data types:

Token Security Data

This category encompasses analyses of token contracts, highlighting potential risk assessments, and token holder distribution analyses. Additionally, we have introduced an open source Token Risk Classification standard, a framework designed to categorize the various risks associated with tokens. Token security data plays a vital role in offering stakeholders a detailed understanding of the security aspects of token projects, aiding in the identification and mitigation of associated risks. This classification standard further enhances our ability to assess and communicate the nuances of token-related risks effectively.

Malicious Address Data

Malicious address data includes known blockchain addresses associated with scams, phishing, hacking and other fraudulent activities. By identifying and warning users about these addresses, this data type is crucial in preventing interaction with these malicious addresses and enhancing user security.

NFT Security Data

This category encompasses analyses of NFT contracts, highlighting potential risk assessments, and token holder distribution, NFT information analyses. NFT security data plays a vital role in offering stakeholders a detailed understanding of the security aspects of NFT projects, aiding in the identification and mitigation of associated risks.

Approval Risk Data

Approval risk data primarily focuses on potentially hazardous contracts that require user authorization, including contracts that have been compromised in hacker attacks as well as malicious contracts. When users authorize their assets to these contracts, they may face the risk of asset loss. This type of security data is crucial in helping users identify and revoke permissions to dangerous contracts, thereby preventing the authorization of their assets to these risky entities. Approval risk data serves as a vital tool in safeguarding user assets against unauthorized access and potential misuse by highlighting the risks associated with certain contract authorizations.

dApp Security Data

This category comprises security audit reports of smart contracts, known vulnerability lists, and community safety feedback. dApp security information provides a comprehensive safety assessment for dApp users, helping them avoid interactions with insecure dApps.

Specific Malicious Signature Features Data

Targeting anomalies and potential risks in blockchain transaction signatures, such as unauthorized transactions or suspicious contract calls, this data helps identify and prevent malicious activities, enhancing transaction security.

Phishing Site Data

Phishing site data involves characteristics of known phishing sites and user feedback, aimed at identifying potential phishing attacks. This data is vital in preventing users from accessing malicious websites and protecting them from data or asset theft.

Conclusion

Together, these security data types form the core of our decentralized data contribution system. By integrating and analyzing this data, the network can more effectively identify and respond to security threats, ensuring the safety of users and their assets. This collective effort lays a solid foundation for the future of digital interactions, empowering all participants to navigate the Web3 world with confidence and security. Furthermore, we plan to enrich and expand the variety of security data categories through governance and voting mechanisms in the future. This approach will enhance the diversity and coverage of our security data, strengthening the overall robustness of our security data ecosystem.

Token Risk Classification

Token Risk Classification(TRC) aims at identifying and cataloging scams like honeypots, and intentional backdoors that may be present in token smart contracts within the web3 ecosystem. This classification serves as:

- **A Shield against Malicious Smart Contracts:** By showcasing a defined list of malicious token contract patterns, it empowers users and project teams to recognize and steer clear of contracts with hidden intents, thereby ensuring safer interactions within the decentralized space.
- **A Testing Ground for Developers:** With a clear classification of malicious patterns and real-world examples, developers creating tools to detect these malicious token smart contracts can effectively evaluate their systems against a standardized classification.
- **A Catalyst for Research:** By clarifying the deceitful practices adopted in token smart contracts, we hope to drive more research towards crypto user safety, encouraging the community to devise strategies that deter such behaviors.
- **An Educational Asset:** This Github repository stands as an initiative to amplify awareness, serving as an informational storage hub, shedding light on potential contract pitfalls and deceitful patterns to the advantage of the community.

For more details, please visit our TRC website.

Contribute

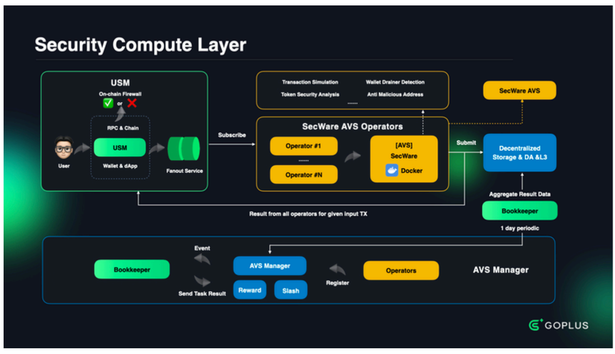
Maintaining the relevance and comprehensiveness of this Github repository is a joint endeavor. We hope for and welcome community contributions. For details on how to contribute, kindly refer to our Contribution Guidelines.

Security Compute Layer

Overview

The Security Compute Layer of GoPlus Network aims to build a global, decentralized security computing network, providing a robust infrastructure for user security services in the Web3 environment. We have adopted the AVS (Actively Validated Services) architecture to build our decentralized security computing network. This allows anyone to join the network as an operator node without permission, executing security computation tasks and earning token incentives for helping users protect their transactions. Additionally, ordinary users can delegate tokens to operators, enhancing the network's overall security while earning returns.

Architecture



Security Compute Layer

The Compute Layer architecture consists of three main components:

- GSM
- SecWare AVS
- AVS Manager

GoPlus Security Module (GSM)

- **Function**: The GSM can be seamlessly integrated into chains, sequencers, RPCs, and wallets. It is designed to receive security requests, dispatch these requests to the compute network for execution and intercept the transaction when it is risky acting as the on-chain firewall.
- **Integration Points**:
  - **Sequencers**: Intercept and evaluate transactions at the sequencer level.
  - **RPC Services**: Intercept and evaluate transactions at the RPC level.
  - **Wallets**: Intercept and evaluate transactions before the transaction sent to RPC.
- **Process**:
  1. Receives security requests from integrated systems.
  2. Dispatches these requests to the appropriate operators in the compute network for processing.
  3. Receive the transaction security analysis results and decide whether to send the transaction.
- **Fanout Service**: The GSM has a fanout service to distribute tasks to different operators, ensuring optimal scheduling and routing. This service dispatches tasks based on SecWare type, geographical information, and the reputation data of operators, maximizing user experience and the efficiency of security services.

SecWare AVS

- **Function**: This part consists of operator nodes that execute security tasks based on parameters received from the GSM fanout services. Each security task is run by deploying various SecWare services in Docker containers on different operator nodes.
- **Operators**:
  - Execute SecWare such as transaction simulation, wallet drainer detection, malicious address detection, malicious signature, and authorization detection risky token detection, etc.
  - Any operators need to register by the AVS service manager and follow the register rule (TBD).
- **SecWare AVS**: Each type of security service is provided as a different Docker image, allowing for flexible and scalable deployment across the operator nodes. These Docker images must comply with the standards and specifications set forth by GoPlus for SecWare. Through the SecWare protocol, developers can upload and register these images, ensuring compliance and interoperability of all security services.
- **Process**:
  - Operators receive computation task and corresponding parameters from the GSM fanout service.
  - Call the interface service of the corresponding SecWare Docker image.
  - Return the transaction security analysis results to the GSM.

AVS Manager

The AVS Manager is responsible for operators registration, compute tasks results validating and rewards distributing or operators slashing.

Core Component:

- **AVS Service Manager Contract**: Handles operator registration, result validation, incentive distribution and slash operation.
- **Bookkeeper**: A queue that collects and processes the execution results and work records submitted by operators from decentralized storage.

Process:

- Bookkeeper aggregates results from decentralized storage over the defined period.
- AVS Service Manager Contract validates the aggregated results to ensure accuracy and compliance with standards.
- Bookkeeper receives the results and calculates the final incentives or slashing operations.
- AVS Service Manager Contract distributes token incentives to operators who meet the performance criteria.
- AVS Service Manager Contract applies slashes to operators who fail to meet the required standards.

This ensures only reliable and high-quality operators remain in the network, maintaining its overall security and efficiency.

Decentralized and Verifiable Security

By adopting this decentralized architecture, GoPlus Network has achieved truly decentralized and verifiable security services. This ensures that the security services themselves are secure and transparent. The decentralized nature eliminates single points of failure, and the verifiable processes guarantee that all security computations and their results can be independently validated. This enhances trust in the system and ensures that users can rely on the security measures provided.

Conclusion

The Compute Layer of the GoPlus Network represents an innovative and decentralized approach to enhancing user security in the Web3 environment. By leveraging the AVS (Actively Validated Services) architecture, it provides a flexible and scalable infrastructure for executing a wide range of security tasks. This architecture allows anyone to join as an operator, perform security computations, and earn rewards, while also enabling ordinary users to delegate tokens and contribute to the network's security.

AVS Operator Guide

Operator Guide

This guide contains the steps needed to set up and register your node for GoPlus AVS (testnet/mainnet).

Minimal system requirements

- 4 CPU
- 8GB Memory
- 20GB Hard disk (Amazon EBS st1)
- Ubuntu 22.04 LTS
- Docker v24 and above
- Docker compose
- Golang 1.23
- EigenLayer CLI

Minimal stake requirements

1. GoAltLayer MACH AVS Mainnet - 1 ETH
2. GoAltLayer MACH AVS Testnet - 1 wei

Supported token strategy

Beacon Chain Ether and all ETH-based LSTs supported by EigenLayer are supported by our AVS.

💡 **Currently, only `quorum[0]` is available. Other quorums will be opened in the future.**

Currently active AVS

1. GoPlus AVS Mainnet
2. GoPlus AVS Testnet

Operator setup

Key generation and wallet funding

1. Follow EigenLayer guide and Install EigenLayer CLI
2. Generate ECDSA and BLS keypair using the following command

```
eigenlayer operator keys create --key-type ecdsa [keyname]
eigenlayer operator keys create --key-type bls [keyname]
```

> **Please ensure you backup your private keys to a safe location. By default, the encrypted keys will be stored in `~/.eigenlayer/operator_keys/`**. Fund at least 0.3 ETH to the ECDSA address generated. It will be required for node registration in the later steps.

Register on EigenLayer as an operator

> You may skip the following steps if you are already a registered operator on the EigenLayer testnet.

You will need to do it once for testnet and once for mainnet.

1. Create the configuration files needed for operator registration using the following commands. Follow the step-by-step prompt. Once completed, `operator.yaml` and `metadata.json` will be created.

```
eigenlayer operator config create
```

2. Edit `metadata.json` and fill in your operator's details.

```
{
  "name": "Example Operator",
  "website": "<https://example.com/>",
  "description": "Example description",
  "logo": "<https://example.com/logo.png>",
  "twitter": "<https://twitter.com/example>"
}
```

3. Upload `metadata.json` to a public URL. Then update the `operator.yaml` file with the url (`metadata_url`). If you need hosting service to host the metadata, you can consider uploading the metadata gist and get the `raw` url.
4. If this is your first time registering this operator, run the following command to register and update your operator

```
eigenlayer operator register operator.yaml
```

Upon successful registration, you should see

```
✅ Operator is registered successfully to EigenLayer
```

If you need to edit the metadata in the future, simply update metadata.json and run the following command

```
eigenlayer operator update operator.yaml
```

5. After your operator has been registered, it will be reflected on the EigenLayer operator page. Testnet: https://holesky.eigenlayer.xyz/operator Mainnet: https://app.eigenlayer.xyz/operator

You can also check the operator registration status using the following command.

```
eigenlayer operator status operator.yaml
```

Joining GoPlus AVS

GoPlus AVS Setup

Clone the GoPlus AVS repository

Run the following command to clone the GoPlus AVS operator repository

```
git clone https://github.com/GoPlusSecurity/GoPlus-AVS
```

Inside this repository, we have configurations for various GoPlus AVS. Different `.env` configurations determine whether AVS runs on Mainnet or Testnet.

Prepare Configuration File

- Run `make copy-config`; this command will create an `.env` configuration file in the project's root directory.
- Fill in the configuration settings in `.env`:
  - `COMPOSE_FILE_PATH`: Path where AVS stores Docker Compose files; replace with an empty and permission-appropriate folder path.
  - `OPERATOR_ADDRESS`: Hex string of operator's address.
  - `BLS_KEY_STORE_PATH`: Path to BLS keystore generated by `eigenlayer operator keys create --key-type bls`.
  - `NODE_CLASS`: AVS node class, defaults to "xl" and does not need modification.
  - `API_PORT`: Port for communication with Gateway; any available port is acceptable.
  - `OPERATOR_URL`: URL path for Gateway access, for example, `http://{DOMAIN}`. If not using DNS, set it to `http://{Host IP}:{API_PORT}`, for example, `http://8.8.8.8:7890`.
  - `QUORUM_NUMS`: 0
  - `ETH_RPC`: RPC address. The program uses the RPC address to distinguish between the testnet and mainnet. You can use RPC addresses from providers like Alchemy.
  - `REGISTRY_COORDINATOR_ADDR, OPERATOR_STATE_RETRIEVER`: Copy the deployment addresses for the corresponding network from the README.md.

> It is recommended to use a domain name in `OPERATOR_URL`. Later, GoPlus Gateway service will assign tasks to AVS through `http(s)://{DOMAIN}:{API_PORT}`. Additionally, the `OPERATOR_URL` and `API_PORT` will be recorded in AVS on-chain contracts.

Example `.env` for Mainnet:

```
COMPOSE_FILE_PATH=/home/user/secwares
OPERATOR_ADDRESS=0xdEADBEeF00000000000000000000000000000000
BLS_KEY_STORE_PATH=/home/user/.eigenlayer/operator_keys/bls.key.json
NODE_CLASS=xl
API_PORT=7776
OPERATOR_URL=http://your_operator_ip:7776
ETH_RPC=https://eth-mainnet.g.alchemy.com/v2/<apikey>
QUORUM_NUMS=0
REGISTRY_COORDINATOR_ADDR=0x91228C6361997a5a4da1a01EdDB2F6B604536A32
OPERATOR_STATE_RETRIEVER=0xD5D7fB4647cE79740E6e83819EFDf43fa74F8C31
```

Example `.env` for Testnet:

```
COMPOSE_FILE_PATH=/home/user/secwares
OPERATOR_ADDRESS=0xdEADBEeF00000000000000000000000000000000
BLS_KEY_STORE_PATH=/home/user/.eigenlayer/operator_keys/bls.key.json
NODE_CLASS=xl
API_PORT=7776
OPERATOR_URL=http://your_operator_ip:7776
ETH_RPC=https://eth-holesky.g.alchemy.com/v2/<apikey>
QUORUM_NUMS=0
REGISTRY_COORDINATOR_ADDR=0x61AA80e5891DbfCebD0B78a704F3de996E449FdE
OPERATOR_STATE_RETRIEVER=0x5ce26317F7edCBCBD1a569629af5DC41c1622045
```

To opt-in

> Before you opt-in to GoPlus AVS, please ensure that you have the right infrastructure to keep the operator up and running. Non-performing AVS operators may be subjected to ejection out of GoPlus AVS.

> The ECDSA private key is used **only once** in `reg-with-avs` and `dereg-with-avs` steps, so for security reasons, we recommend removing the ECDSA keystore file after completing these operations.

1. Run `export BLS_KEY_PASSWORD=...` to export the password to BLS keystore file.
2. Run `make build-avs` to compile AVS.
3. Run `make reg-with-avs` to register. During execution, the user will be prompted to enter the path to the ECDSA keystore file and corresponding password.
4. **Wait at least 3 minutes** before proceeding with subsequent operations, as the GoPlus Gateway service needs time to synchronize the operator's information.

> It may take a few minutes for EigenLayer AVS and operator page to be updated This is an automatic process.

To opt-out

If you no longer want to run the AVS, you can opt out by running `make dereg-with-avs`.

Start GoPlus AVS

> GoPlus AVS runtime can be deployed in two ways: as a standalone process or via Docker Compose (recommended). When using Docker Compose, all services operate in **Host network** mode. Please ensure the following ports are available on your host machine:
>
> - API_PORT (configurable)
> - 9090 (metrics)
> - 3000 (monitoring)

Mainnet configuration

1. Start with Docker Compose:
   a. Run `export BLS_KEY_PASSWORD=...` to export the password to BLS keystore file.
   b. Run `make build-avs-docker-mainnet` to build the AVS Docker image.
   c. Run `make run-avs-docker` to start. This also starts Prometheus and Grafana.
2. Start as a standalone process:
   a. Run `export BLS_KEY_PASSWORD=...` to export the password to BLS keystore file.
   b. Run `sudo docker login -u goplusavs -p dckr_pat_wRhsTj4U7REe7IFnrgFkAOswjaM` to log in.
   c. Run `make run-avs` to start.

Testnet configuration

1. Start with Docker Compose:
   a. Run `export BLS_KEY_PASSWORD=...` to export the password to BLS keystore file.
   b. Run `make build-avs-docker-testnet` to build the AVS Docker image.
   c. Run `make run-avs-docker` to start. This also starts Prometheus and Grafana.
2. Start as a standalone process:
   a. Run `export BLS_KEY_PASSWORD=...` to export the password to BLS keystore file.
   b. Run `sudo docker login -u joker1034 -p dckr_pat_MH5qjNWvS3iahu8--rK4wW7NbEM` to log in.
   c. Run `make run-avs` to start.

Check AVS Running Status

1. Connectivity Check
   - Send a request to `{OPERATOR_URL}/avs/ping` to check the connectivity of the AVS web service.
2. Secware Running Status
   - AVS will periodically request Secware configuration from the Gateway and run Secware in Docker. It also regularly reports Secware's health status to the Gateway.
   - Run `sudo docker compose ls` to view Secware's running status.

If AVS is running in a Docker Compose environment, you can access Grafana at `http://{OPERATOR_URL}:3000` to view monitoring data. The default username and password are `goplus_avs/admin`.

FAQ

1. When attempting to deploy GoPlus AVS to my Kubernetes environment, why do I receive the error `secware config length: 4`?

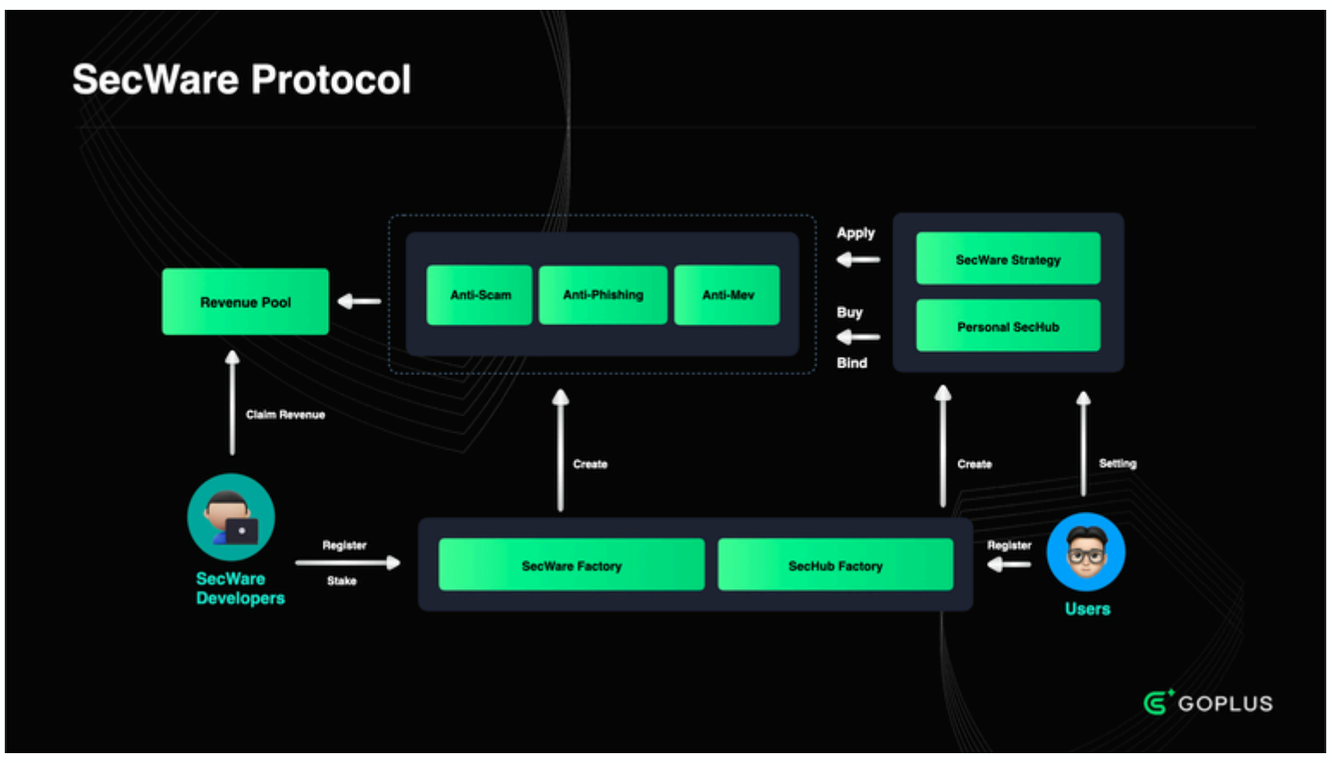
> Currently, our AVS relies on **Docker Compose** for container orchestration and is not natively compatible with Kubernetes environments. While adapting it would require significant architectural changes, we're considering adding Kubernetes support based on operator feedback.

Join us

Telegram: https://t.me/goplusoperators

Github: https://github.com/GoPlusSecurity/GoPlus-AVS

SecWare Protocol



SecWare Protocol

SecWare Protocol is a decentralized security service ecosystem built around users and security developers. The protocol consists of multiple contract entities that define the interactions and relationships between users, developers, and the security services they provide.

SecWare (Security Software)

In the SecWare Protocol, security services are defined as **SecWares**. Developers create SecWare instances on the blockchain, allowing users to access their security services through binding and purchasing mechanisms.

SecWare Creation

Developers create SecWare instances on the SecWare Protocol by registering their security services such as anti-scam, anti-phishing or anti-mev. The registration process involves providing essential information about the service, such as its description, pricing, and SLA (Service Level Agreement). These SecWare instances serve as on-chain representations of the actual security software services.

SecWare Binding and Purchase

Users can browse available SecWares and choose to bind or purchase the services they need with their **SecHub**. Binding a SecWare allows users to access its functions, while purchasing grants them additional privileges or premium services.

Personal SecHub

Users must create a personal security center instance, known as a SecHub, to manage their security strategies and interact with bound SecWares.

SecHub Creation

Users create their SecHub instance on the blockchain, which serves as a control panel for managing their security services and security strategy.

SecWare Management

Within their SecHub, users can bind, purchase, configure, and manage the SecWares they have access to. They can adjust parameters for each SecWare and set conditions under which the security services should take effect.

Revenue Pool and Profit Distribution

All profits generated by SecWare instances are collected in a Revenue Pool. Developers have the right to withdraw their share of the profits based on the usage and performance of their SecWares.

Profit Allocation

The Revenue Pool automatically allocates profits to developers based on predefined rules and the performance of their SecWares.

Profit Withdrawal

Developers can withdraw their earned profits from the Revenue Pool, providing a financial incentive for creating high-quality security services.

Stake and Slash Mechanism

To ensure the quality and reliability of SecWares and prevent malicious behavior, the SecWare Protocol incorporates a stake and slash mechanism.

Staking

Developers must stake a certain amount of tokens when creating a SecWare instance. This stake serves as a commitment to providing a reliable and effective security service.

Slashing

If a SecWare fails to meet the specified SLA or is found to be malicious, a portion of the staked tokens can be slashed as a penalty. This mechanism encourages developers to maintain high standards and adhere to the protocol's rules.

Conclusion

The SecWare Protocol is governed by a decentralized community of stakeholders, including users, developers, and token holders. Decisions regarding protocol upgrades, parameter adjustments, and dispute resolution are made through a decentralized governance process.

By leveraging the SecWare Protocol, we aim to create a robust, transparent, and decentralized security service ecosystem that empowers users and incentivizes developers to contribute innovative and effective security solutions.

GoPlus Intelligence

Introduction

GoPlus Security provides comprehensive security solutions that deliver real-time, automated security intelligence. These solutions are designed to provide developers and platforms with real-time, automated security insights to protect users from various potential risks associated with blockchain and cryptocurrency. This section will detail the various components of GoPlus Intelligence and their functions.

GoPlus Security Intelligence Overview

Key Features

- Real-time Data: Provides up-to-the-minute security information, ensuring users can make decisions based on the latest intelligence.
- Automated Analysis: Utilizes advanced algorithms and machine learning techniques to automatically identify and assess potential security risks.
- Scalability: Designed to handle large-scale queries and data analysis, suitable for applications of all sizes.
- Easy Integration: Provides clear documentation and SDKs, enabling developers to quickly incorporate security features into their projects.

GoPlus Intelligence Capabilities

Multi-chain Token Security

A decentralized, user-driven service that provides detailed security analysis of tokens.

- Features: Real-time risk assessment, including token contract security, liquidity analysis, holder distribution, etc.
- Applications: Helps exchanges, wallets, and DeFi platforms evaluate the security of newly listed tokens.

Malicious Address Detection

Provides a free, timely, and comprehensive malicious address library.

- Features: Identifies known malicious addresses, including those related to scams, phishing, and other illegal activities.
- Applications: Used for transaction screening, user warnings, and risk management systems.

NFT Security Assessment

Conducts comprehensive security assessments of NFTs, helping to detect scams or fraudulent activities in NFT transactions.

- Features: Analyzes the origin of NFTs, transaction history, and the security of related smart contracts.
- Applications: Security assurance for NFT marketplaces, collectibles platforms, and digital art exchanges.

Approval Security Analysis

Analyzes security risks associated with token approvals to prevent unauthorized or risky transactions.

- Features: Evaluates the security of approval requests, checks the credibility of approval recipients.
- Applications: Enhances the security of wallets and DeFi applications, protecting user assets.

dApp Security Information

Aggregates security information from various dApps, offering quick risk alerts and insights.

- Features: Real-time monitoring of dApp security status, including smart contract vulnerabilities, audit status, etc.
- Applications: Provides a secure dApp interaction environment for users, improving overall ecosystem security.

Signature Data Decoding

Decodes and analyzes ABI signature data for irregularities or signs of malicious activity.

- Features: In-depth parsing of transaction signatures, identifying potential malicious operations like many phishing activity using permit/permit2.
- Applications: Enhances transaction verification processes, preventing sophisticated attack methods.

Phishing Site Detection

Detects and blocks phishing sites before users fall victim to malicious attempts.

- Features: Real-time identification and flagging of suspicious phishing websites.
- Applications: URL verification in browser extensions, security tools, and wallet applications.

Multi-Chain Transaction Simulation

Simulates transactions across multiple blockchain networks to assess the results and detect any potential risks before they are executed on-chain.

- Features:
  - Previews transactions in a secure environment across various blockchain networks.
  - Identifies possible anomalies, risks, or unexpected outcomes.
  - Supports multiple popular blockchain networks including Ethereum, Binance Smart Chain, Solana, and others.
  - Provides detailed reports on gas estimation, token transfers, and contract interactions.
- Applications:
  - Enhances transaction security for multi-chain wallets and DeFi applications.
  - Helps developers test and debug smart contract interactions in a safe environment.
  - Allows users to understand the full impact of a transaction before committing it to the blockchain.

Conclusion

GoPlus Intelligence provides a robust security infrastructure for the blockchain and cryptocurrency ecosystem through its comprehensive suite of security capabilities. These tools not only help developers build more secure applications but also provide users with a safer interaction environment across multiple blockchain networks. GoPlus Intelligence will continue to innovate and expand its services to address emerging security challenges, driving the entire industry towards a more secure and trustworthy future.

GoPlus SafeToken Protocol

Overview

The GoPlus SafeToken Protocol represents a significant advancement in addressing fundamental security challenges in token creation and management within the DeFi ecosystem. Built upon GoPlus Security's extensive experience in token security analysis and risk detection, this protocol provides comprehensive solutions for secure token issuance and liquidity management.

Background

The DeFi ecosystem has witnessed numerous security incidents related to token contracts and liquidity management, including:

- Malicious token contracts with hidden features
- Unauthorized minting capabilities
- Liquidity removal scams
- Rugpulls due to poor liquidity management
- Token contract vulnerabilities

Drawing from its extensive database of token security incidents and deep understanding of contract vulnerabilities, GoPlus has developed the SafeToken Protocol to address these challenges at their source.

Key Components

1. GoPlus SafeToken Factory

A revolutionary platform for secure token creation and deployment:

Features:

- Free and open-source token contract templates
- Pre-audited, security-first contract designs
- Standardized security implementations
- Quick and efficient token deployment process
- Automated security checks during creation

Benefits for Developers:

- Reduced development time and costs
- Elimination of common security vulnerabilities
- Enhanced trust from the community
- Standardized security best practices
- Seamless deployment process

2. GoPlus SafeToken Locker

An advanced liquidity management solution designed to enhance project credibility and protect investor interests:

Key Features:

- Flexible lock-up period management
- Automated rewards distribution system
- Multi-signature security options
- Transparent lock tracking
- Cross-platform DEX compatibility

Advanced Capabilities:

- Customizable vesting schedules
- Emergency security measures
- Real-time monitoring and alerts
- Automated compliance checks
- Integration with major price websites

Technical Implementation

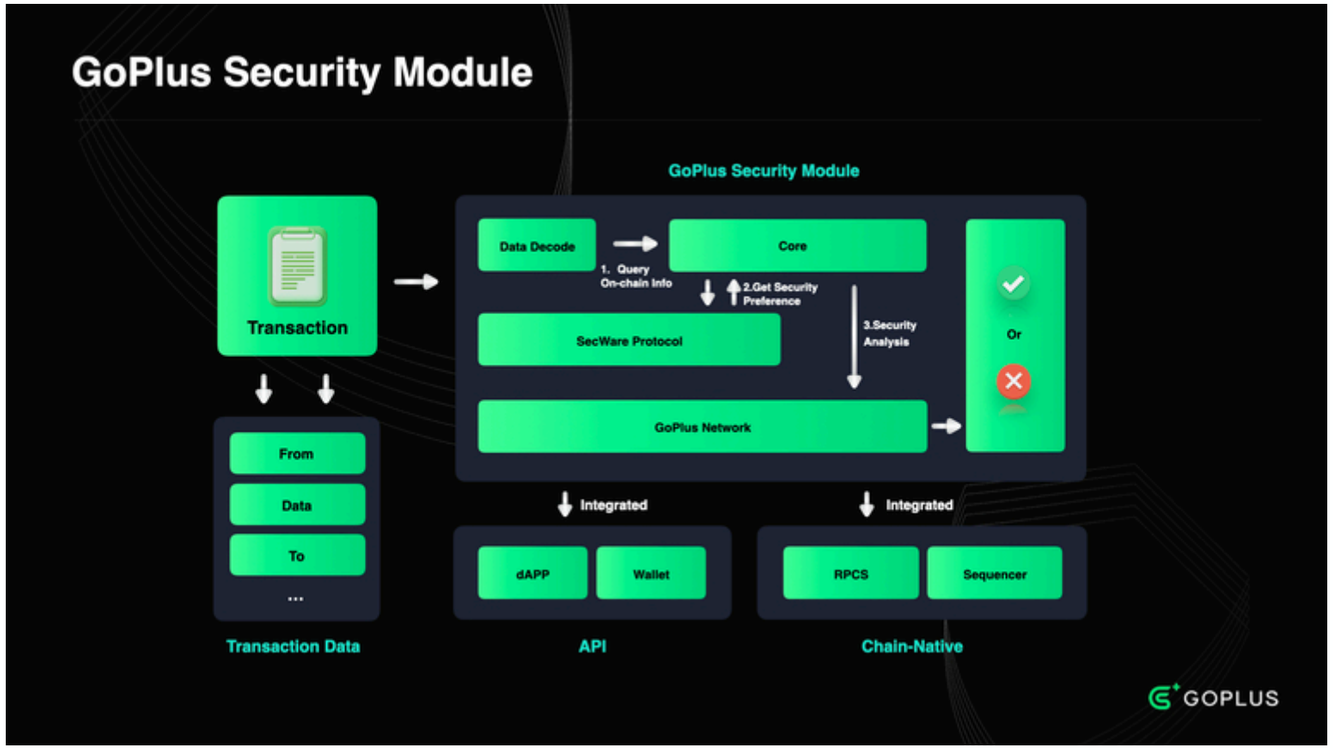
[SafeToken Factory Implementation]

[SafeToken Locker Implementation]

Conclusion

The GoPlus SafeToken Protocol represents a significant step forward in securing the token creation and management process in the DeFi ecosystem. By providing comprehensive, security-focused solutions through the SafeToken Factory and SafeToken Locker, the protocol addresses critical vulnerabilities at their source while promoting standardization and best practices in the industry.

Architecture Overview



GoPlus Security Module

GoPlus Security Module (GSM) is a critical component within the GoPlus Network's architecture. As illustrated in Image, the **GSM** is positioned at the core of the network, serving as a bridge between the chains and GoPlus Network.

The GSM is designed as a software development kit (SDK) that can be seamlessly integrated into various Web3 scenarios, such as wallets, dApps, RPCs, and even Layer 2 sequencers. This modular approach allows the GSM to be easily adapted to different blockchain environments, providing a consistent and robust security layer across multiple networks.

The primary function of the GSM is to facilitate the interaction between user-initiated transactions and GoPlus's SecWare services. When a transaction is triggered, the GSM intercepts the transaction data and forwards it to the SecWare. The SecWare, which leverages GoPlus's open security data and computing layers, performs real-time risk assessments on the transaction using advanced AI algorithms.

The results of the risk analysis are then relayed back to the GSM, which can take appropriate actions based on the security recommendations. This may include proceeding with the transaction if it is deemed safe, or reject the transaction which is malicious.

By serving as the interface between users and blockchains, the GSM enables GoPlus to provide a comprehensive, end-to-end security solution for Web3 users. The architecture of the GSM ensures that this security layer can be easily integrated and adapted to different blockchains and also can be a important module in modular blockchains and RaaS (Rollup as a Service).

Core Features

GoPlus Security Module (GSM) offers a range of core features designed to enhance the user security and usability of Web3 interactions. These features include:

Modular

The GSM's modular design makes it a crucial component in the rapidly evolving blockchain landscape, particularly in the context of emerging modular blockchain solutions and Rollup-as-a-Service (RaaS) providers. As the industry moves towards greater composability and interoperability, the GSM serves as a standalone user security and risk control module that can be seamlessly integrated into various modular blockchain architectures, enabling them to provide advanced security features without the need for in-house development.

GoPlus aims to establish the GSM as a standard component of the infrastructure for future Layer 2 and Layer 3 networks, ensuring that the majority of these solutions natively possess advanced user security and risk control capabilities. This integration will create a secure and trustworthy environment for users across the entire blockchain ecosystem, regardless of the specific scaling solution they are interacting with.

Multi-chain Support

The GSM is designed to be adaptable to multiple blockchain networks, providing a consistent security layer across different ecosystems. This multi-chain support ensures that users can benefit from GoPlus's advanced security features, regardless of the specific blockchain they are interacting with.

On-chain Risk Management

By leveraging GoPlus's advanced AI algorithms and security engine and comprehensive security data, the GSM enables real-time risk analysis and mitigation for user transactions. This on-chain risk management feature helps identify and prevent potential security threats, such as fraud or unauthorized access, before they can cause harm.

Customize Security Strategy

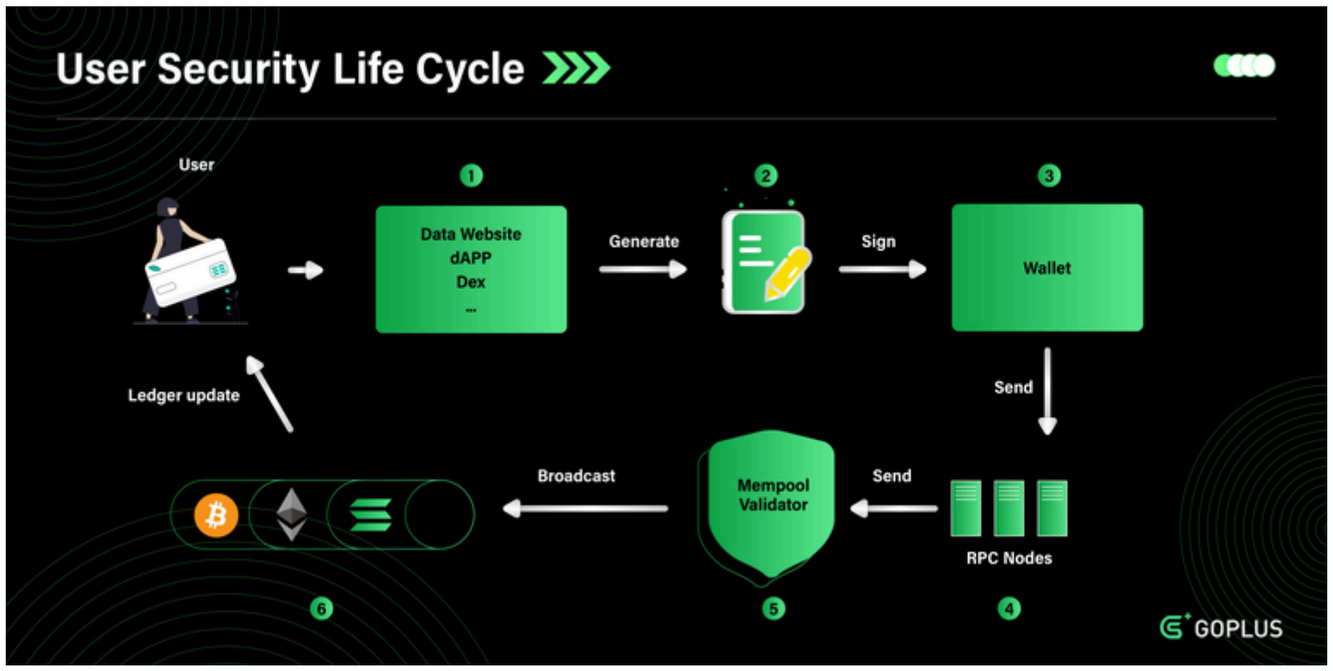
The GSM connects GoPlus Personal SecHub which allows users to customize their security preferences based on their individual risk tolerance and specific needs. This feature enables users to strike a balance between security and usability, ensuring that they can interact with Web3 applications in a way that aligns with their personal security requirements.

Seemlessly User Experience

The GSM's Seamless User Experience feature is a game-changer for Web3 adoption, as it provides users with a secure environment without requiring them to be security experts. By implementing automatic on-chain risk control, the GSM protects users even if they inadvertently sign dangerous transactions at the wallet level, intercepting them at the blockchain or RPC level.

This proactive approach to user security significantly lowers the barrier to entry for mainstream users, allowing them to engage with decentralized applications and services confidently. The GSM's ability to abstract away the technical complexities of Web3 user security and its seamless integration with various Web3 touchpoints make it a crucial component in driving mass adoption of Web3.

User Security Life Cycle



User Security Life Cycle

To address the pressing security concerns in the Web3 space and guide users towards adopting optimal security practices, we have introduced the Web3 User Security Life Cycle (USLC). This framework outlines the necessary steps users should take to protect themselves before, during, and after interacting with Web3 applications and services.

- **Pre-Event Phase:**

At this initial stage, the focus is on equipping users with the knowledge they need to navigate the Web3 space safely. Before engaging with any Web3 applications or services, such as a Data Website, dAPP, or Dex, users should be presented with up-to-date and accurate security information and risk assessments. Ensuring users are informed about potential threats and equipped with best security practices is crucial. This preventative measure lays the foundation for a secure interaction with Web3 platforms.

- **During-Event Phase**

As the user progresses to actively engage with Web3 services, they generate and sign transactions in their wallet. After the signing, the security measures are actively at play, with on-chain firewalls and rigorous security protocols operating in real-time to shield the user's assets. This protection extends as the transaction is sent through RPC nodes to the mempool (Step 4), where it's validated in mempool or by validators (Step 5) before ultimately being broadcasted and recorded on the ledger.

- **Post-Event Phase**

Once the transaction has been broadcast and the ledger is updated (Step 6), post-event security becomes paramount. This phase tackles the risks that linger after the interaction has taken place. It involves steps such as revoking unneeded permissions, conducting ongoing monitoring for anomalous activity, and ensuring that robust recovery solutions are in place to respond swiftly in the event of a security breach. This ensures that even after the transaction is completed, the user's security posture remains strong and resilient against latent threats.

The USLC emphasizes that securing a user’s journey in Web3 is not a one-off event but a cyclical and ongoing process. By diligently addressing the unique challenges inherent at each phase, the GoPlus ecosystem aims to cultivate a safer and more reliable Web3 experience for all users.

On-chain Firewall

Introduction to On-Chain Firewall

GoPlus introduces a groundbreaking security feature: the On-Chain Firewall. This innovative solution provides real-time transaction protection on the blockchain, automatically blocking malicious and dangerous transactions that could harm users. Unlike traditional pre-transaction warnings, our On-Chain Firewall implements seamless risk control and blocking capabilities, significantly enhancing the security of the blockchain environment.

Key Features of On-Chain Firewall

- Real-time Protection: Monitors and analyzes transactions in real-time as they occur on the blockchain.
- Automatic Blocking: Instantly stops malicious or high-risk transactions before they can cause harm.
- Seamless User Experience: Operates invisibly in the background, providing security without disrupting the user experience.

Implementation Strategy

To provide each blockchain with native risk control and security capabilities, GoPlus implements the security service through a three-step process:

1. GoPlus Intelligence Integration

- Provide GoPlus Intelligence support tailored for the specific blockchain.
- Offer APIs and SDKs to wallets and various projects on the chain.
- Enable developers to easily integrate robust security features into their applications.
- Deliver real-time security insights and risk assessments for transactions and smart contract interactions.

2. Secure RPC Service Provision

- Deploy secure, native RPC nodes for the blockchain.
- Allow users to directly utilize these secure RPC endpoints for enhanced protection.
- Implement additional security layers at the RPC level to filter and analyze transactions.
- Provide a trusted infrastructure layer for secure blockchain interactions.

3. Native Blockchain Integration

- Collaborate with blockchain projects to integrate a User Security Module.
- Implement an additional security layer directly into the blockchain's architecture.
- Enable native risk control capabilities at the chain level.
- Ensure that security checks are performed as an inherent part of the transaction validation process.

Conclusion

The On-Chain Firewall represents a significant leap forward in blockchain security. By providing real-time, seamless protection at multiple levels of the blockchain stack, GoPlus is setting a new standard for security in the decentralized world. As we continue to expand and refine this technology, we envision a blockchain ecosystem where users can transact and interact with confidence, knowing that advanced security measures are working tirelessly to protect their assets and interests.

Security RPC

We are excited to introduce GoPlus Security RPC, designed to provide users with a seamless and secure way to interact with the different blockchains. By connecting to Security RPC endpoints, users can access the robust security features of the GoPlus Security, ensuring that their transactions and interactions with these networks are protected against various security risks.

With GoPlus Security RPC, users can easily integrate our security solutions into their existing infrastructure, without the need for complex setup or configuration. Our RPC service is built on top of our network and different SecWares, leveraging the power of our network to deliver fast, reliable, and secure access to blockchains.

Whether you're a developer building decentralized applications or a user looking to securely interact with these networks, GoPlus Secuity RPC provides a simple and effective solution for enhancing the security of your on-chain activities. Stay tuned for more updates as we continue to expand our RPC service to support additional networks and features.

Currently, we have launched Security RPC services for various blockchain networks, including:

- Ethereum
- BNB Chain

GoPlus BNB RPC

Network name: BNB GoPlus SecNet

RPC URL: https://rpc.secwarex.io/bsc

Chain ID: 56

Currency symbol: BNB

Block explorer URL(Optional): https://bscscan.com

GoPlus ETH RPC

Network name: ETH GoPlus SecNet

RPC URL: https://rpc.secwarex.io/eth

Chain ID: 1

Currency symbol: ETH

Block explorer URL(Optional): https://etherscan.io

GoPlus Security Governance

GoPlus Network Governance empowers the community by fostering transparency, inclusivity, and collaboration. It establishes a decentralized framework where participants, including data providers, node providers, SecWare developers, and SecWare protocols, play vital roles in shaping the ecosystem.

Let's explore the relationships between GoPlus Network Governance and these key participants.

Ecosystem Contributor

Ecosystem Contributor including **Data Contributor**, **Compute Node Contributor** and **SecWare Developer**. Those contributors empower GoPlus ecosystem with security data, computing power and security service software. They will rewarded with token from GoPlus Network Governance. In return, they could also stake their token to GoPlus Network Governance to gain long-term rewards. Here are the brief introduction illustrating their contribution.

Data Contributor

- Data Contributors contribute high-quality security data, empowering developers and partners to create effective security products and services.
- Data Contributors are incentivized through a reward system that acknowledges their efforts and the value they bring to the network. This can include financial rewards in the form of token, as well as reputational benefits and increased visibility within the community.

Compute Node Contributor

- Compute Node Contributors are integral to the GoPlus Network, providing the essential computational power required for SecWare. By operating reliable nodes, these contributors maintain and strengthen the network's infrastructure, enabling it to deliver robust security services effectively.
- To recognize the vital role of Compute Node Contributors, GoPlus Network has implemented a remuneration system that rewards them for their computational contributions. They receive tokens as compensation, which correlates with the amount of computational resources they supply and the consistency of their service.

SecWare Developer

- SecWare Developers play a creative role in the GoPlus ecosystem, designing and developing the cutting-edge security services that populate the network. They transform raw security data into practical tools and services using GoPlus Security Engine, harnessing the collective computing power of the network to provide users with powerful defenses against a wide array of security threats.
- In appreciation of their innovation and technical skill, SecWare Developers are rewarded through a comprehensive incentive program within the GoPlus Network. This program provides monetary rewards in the form of tokens. Moreover, they gain recognition as pioneers in the field of blockchain security.

Ecosystem Revenue

SecWare Security Service Fees

In the GoPlus Network, users are required to pay a security gas fee in tokens every time they utilize SecWare security services to safeguard their transactions. This fee functions as the security fuel for transactions, with each protected transaction consuming a certain amount of token. This mechanism not only ensures the sustained operation of security services but also incentivizes the providers within the network through the circulation of tokens. A portion of the security fuel fees is allocated to the developers of the security services as a reward for their contribution to and maintenance of the security applications. Another portion flows into the GoPlus Foundation to support the foundation's operations, ongoing research and development, and other ecosystem-building initiatives.

Staking Requirements for Contributors

To maintain the security and credibility of the ecosystem, all Contributors, including data contributors, compute node contributors, and SecWare developers, are required to stake a certain amount of tokens to the GoPlus Foundation. These staked tokens serve multiple purposes: they provide an economic incentive to ensure participants are motivated to maintain the system's integrity and security; they also act as a security mechanism to prevent malicious activities and enhance the overall stability of the network.

Staking-Based Voting Weight Mechanism

In the governance of the GoPlus Network, all users wishing to partake in voting must stake tokens to acquire voting weight. The voting weight of a user is directly correlated with the number of tokens they stake, ensuring that voters are sufficiently committed to and responsible for the ecosystem. The greater the stake, the more confidence a user has in the future of the network, and the greater their influence in shaping its development. This staking-based voting mechanism encourages long-term holding and responsible governance participation.

Through these mechanisms, the GoPlus Network ensures the vibrancy of the ecosystem while also providing contributors and users with the motivation to participate in the growth of the ecosystem. Every security gas fee used, every stake made, and every vote cast is a testament to the trust and commitment of the users and contributors to the long-term success of the GoPlus ecosystem.

Conclusion

Through collaboration and collective decision-making, GoPlus Network Governance ensures that participants actively contribute to shaping the ecosystem. Community proposals, voting, and open discussions enable the governance framework to reflect the interests and values of the community.

In conclusion, GoPlus Network Governance empowers the community by fostering transparency, inclusivity, and collaboration. It establishes a decentralized framework where data providers, node providers, SecWare developers, and SecWare protocols play crucial roles in shaping the ecosystem. This collaborative approach strengthens the ecosystem's reliability, security, and sustainability, ensuring the community's active participation in its development and success.

Data Contributor

Data Contributors are pivotal members of the GoPlus Ecosystem, playing a vital role within the SecWare framework. They stand as the backbone of the SecWare's data layer, supplying the necessary on-chain and off-chain information that is critical for the SecWare computing layer and the overall system's operation.

Responsibilities of Data Contributors

Data Provision

Data Contributors are tasked with the crucial responsibility of sourcing and supplying high-quality, relevant security data to the SecWare data layer. They utilize a wide array of sources to ensure the data's accuracy and applicability.

Data Verification

Data Contributors also bear the essential duty of verifying the accuracy and legitimacy of the information they provide. Rigorous checks and validation processes are a standard part of their workflow, serving to maintain the security data's integrity.

Incentive Acquisition

When the data provided by Data Contributors is successfully submitted and verified by the data layer, the GoPlus Network Governance facilitates the distribution of incentives in the form of token. This reward system acknowledges the significant contributions Data Contributors make to the SecWare ecosystem.

Conclusion

Data Contributors are the linchpins of the SecWare service, with their commitment to accuracy and validation upholding the integrity and reliability of the system. Through the incentives offered by the GoPlus Network Governance, their critical input is rightly recognized and rewarded. Their work is not only appreciated but also fundamental to the SecWare ecosystem's success and functionality. For more specific security data type, see Types of Risk Data.

AVS Operator

The **AVS Operator** in our ecosystem plays an important role, leveraging the advanced framework of **EigenLayer's Actively Validated Services (AVS)**. This allows the network to harness decentralized, secure, and efficient computing resources. AVS Operators will act as the computing providers for security services. They will receive security inspection requests from users and run different **SecWares** based on the requests to deliver security services. AVS Operators can provide real-time support to users by conducting transaction simulations, risk analyses, and more, ensuring the overall stability of the security services.

Responsibilities of Compute Node Contributors

AVS Operators play a critical role in ensuring the integrity and efficiency of the network. These operators participate in validating the execution of off-chain security analysis tasks while being subject to on-chain slashing conditions if they fail to meet security or operational standards. Through EigenLayer, operators can reuse their staked Ethereum for multiple services, allowing them to earn additional rewards while reinforcing the security framework.

Provision of Computational Resources

The core duty of AVS Operators is to deploy and operate SecWare AVS, supplying the required computational power to support the continuous operation of SecWare AVS. This includes ensuring sufficient processing capacity and memory resources to perform security services and risk analyses, which are critical for handling large-scale data and executing real-time analytics.

Security Enforcement

Operators are responsible for maintaining the security of the services they validate. They are incentivized to perform their duties honestly due to slashing conditions imposed by EigenLayer’s smart contracts. Malicious or negligent behavior could result in the forfeiture of their staked assets.

Network and System Maintenance

AVS Operators are also responsible for the maintenance of their nodes, including monitoring system health, performing regular updates and upgrades, and addressing any technical issues. This ensures that the compute nodes can continuously provide efficient, reliable services supporting the stable operation of SecWare.

Incentives

AVS Operators are compensated through multiple revenue streams. To register as an operator, they are required to **restake a certain amount of ETH or $GPS**. By validating services that run on the EigenLayer framework, they can earn additional staking rewards. Alongside the base Ethereum staking rewards and extra fees from SecWare AVS reward, operators can also earn **$GPS token rewards** by running **SecWare**. **More details regarding the registration process for GoPlus SecWare AVS Operators will be announced at launch.** This provides operators with a robust incentive to participate and contribute to the GoPlus Network.

How to become an AVS operator?

See the guide: AVS Operator Guide

Conclusion

AVS Operators are vital for ensuring that SecWares in the GoPlus Ecosystem operate efficiently and continuously. By providing stable and reliable computational resources, they enable SecWare to offer in-depth security analysis and risk assessments to users in real time, thus enhancing the overall security and efficacy of the ecosystem.

SecWare Developer

SecWare Developer is a member of Ecosystem Contributor. In the ecosystem, the role of a SecWare Developer is to create and develop SecWare applications. These developers utilize the predefined SecWare Protocol to build applications that adhere to the established standards within the ecosystem. Once the applications are completed, SecWare developers could publish the SecWare to the marketplace for others to purchase and use.

Responsibilities of SecWare Developers

SecWare Development

SecWare Developers are responsible for the creation and continuous improvement of security applications. Their work involves designing innovative solutions that utilize the latest in blockchain technology and cybersecurity practices. This includes developing applications for anti-fraud, anti-phishing, transaction simulation, risk assessment, and more, which are critical in providing users with a secure Web3 experience.

SecWare Publication

After completing the development of a SecWare, SecWare developers have the opportunity to publish their creations through SecWare Protocol. By making their SecWare available to the ecosystem, they enable users to access and utilize their SecWares.

Continuous Monitoring and Upgrades

To ensure the efficacy and reliability of their SecWares, SecWare Developers are also tasked with monitoring their performance and making necessary updates and improvements. This includes patching vulnerabilities, enhancing features, and optimizing performance to keep up with evolving security threats and technological advancements.

Incentive Acquisition

Recognizing their vital role in the ecosystem, SecWare Developers are rewarded for their contributions through a structured incentive program. They receive tokens based on the adoption and performance of their SecWares.

Community Engagement and Reputation

SecWare Developers engage with the broader GoPlus community to gather feedback and comment. This collaborative approach fosters a vibrant developer community and drives the collective advancement of the network's security capabilities.

Conclusion

For SecWareX users, using SecWare typically involves consuming tokens as a means of payment. This allows users to utilize the functionality provided by the SecWares developed by SecWare Developers. In conclusion, the role of a SecWare developer in the SecWare ecosystem is to develop SecWare based on the SecWare protocol. By adhering to the established standards, SecWare developers contribute to the growth and diversity of the ecosystem. As an incentive, a portion of the profits generated by the SecWare protocol is distributed to the developers. This encourages innovation and rewards developers for their contributions, fostering a vibrant and thriving SecWare community.

Users

Users are at the heart of the GoPlus Ecosystem, playing a critical role in its functionality and growth. They interact with the various SecWares developed by SecWare Developers, providing the essential real-world use cases that drive the continuous evolution of the network.

Role and Impact of Users

Engagement with SecWare

Users are the primary consumers of Secware. They need to top up and cost tokens to subscribe or use SecWare. SecWare protects every transactions of theirs. During the protection, SecWare consume tokens as **Security Gas** and distribute the profit to all Ecosystem Contributor.

Feedback and Community Interaction

The feedback provided by users is invaluable in the continuous development cycle of the GoPlus Ecosystem. Through forums, social media, and direct interactions with developers, users voice their experiences, suggestions, and concerns. This feedback is crucial for identifying usability issues and improving user interface and experience across the different SecWares.

Participation in Security Campaigns on GoPlus APP

Users are encouraged to engage in diverse campaigns run on GoPlus APP. These campaigns are structured to simulate real-world threats in a controlled environment, allowing users to practice safe Web3 navigation. By actively participating, users not only improve their own security skills but also contribute to the collective security intelligence of the GoPlus Ecosystem.

Rewards and Incentives

Active participation in campaigns on GoPlus APP is incentivized with various rewards, including ecosystem points and tokens. These rewards serve not only as a motivation for users to engage more deeply with the security features of the network but also as a means of acknowledging their contributions to the ecosystem’s resilience.

Stake to Vote

Users can stake tokens to gain voting rights in the GoPlus Network’s governance. This staking not only signifies a user's commitment to the ecosystem but also grants them the power to influence decision-making processes. The amount of tokens staked directly correlates with the weight of their vote, emphasizing the principle that those who are more invested in the network have a greater say in its governance.

Conclusion

Users are not just participants but are fundamental drivers of the GoPlus Ecosystem. Their active involvement, feedback, participate in campaigns, vote and advocacy play a pivotal role in shaping the network's development, ensuring that it remains responsive to the needs of the Web3 community. By engaging with the ecosystem, users not only protect their assets but also contribute to a broader movement towards a more secure, transparent, and user-focused environment.

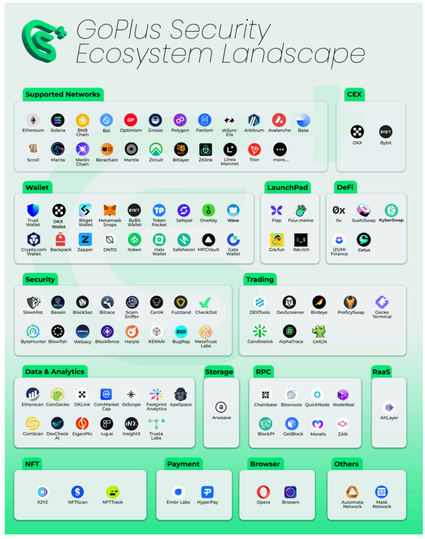
🍏 Tokenomics

The $GPS token will be the cornerstone of the GoPlus Security Network, aligning incentives across our decentralized security infrastructure and empowering the next generation of Web3 security solutions to “**Protect Your Every Transaction**”.

Our Vision: A Decentralized Security Network

The introduction of $GPS marks a new chapter in our journey — transforming GoPlus from a security provider to a decentralized network where everyone contributes to and benefits from Web3 security. We envision a future where security becomes open, permissionless, transparent, and user-driven. Through GoPlus Network, we aim to create a “Secure Universe” where protection is available for every transaction across all blockchain networks.

With a proven track record since 2020, GoPlus has established itself as a crucial security layer for Web3. Every day, we process over **30 million** security detection requests, actively combat emerging scams and threats across more than **30 blockchains**, and, in doing so, protect **billions** in user assets. Our security intelligence technology has been integrated by leading platforms including major price websites, DEXs, and wallets, with over **10,000+** projects and developers relying on our services.



GoPlus Ecosystem

GoPlus security services cover the entire transaction lifecycle, ensuring users are protected at every step of their Web3 journey. This extensive coverage, combined with a sustainable business model that has generated considerable revenue, demonstrates both our technical capabilities and operational skill.

As we move towards a more decentralized future, it’s the perfect time to align $GPS incentives across our ecosystem and take Web3 security to the next level.

Product Lines

Business Solutions:

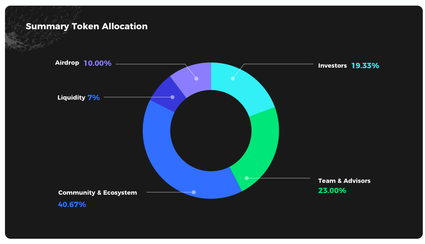
- Security Intelligence: Comprehensive security APIs powered by AI and advanced code analysis capabilities, helping projects and chains protect their users
- SafeToken Protocol: Standardizing secure token issuance and providing professional liquidity management tools to enhance token ecosystem security

Consumer Products:

- GoPlus APP: A one-stop security hub protecting users’ every transaction with features like wallet security scanning, transaction protection, and real-time risk alerts

Tokenomics Overview

With a total supply of **10,000,000,000 $GPS** tokens, we’ve designed our tokenomics to prioritize ecosystem growth and community participation while ensuring proper alignment with initial contributors.



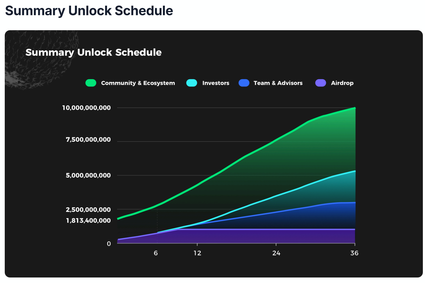
60.67% for Ecosystem & Community Growth

- Community & Development (24.67%) For incentivizing community participation, rewards
- Ecosystem Growth (10%) Allocated to grow network adoption and enhance security service
- Marketing & Growth (6%) Supporting ecosystem expansion and market development
- Airdrop (10%) Rewarding early adopters and active participants
- Liquidity (7%) Ensuring market stability and trading efficiency
- Advisors (3%) Supporting strategic guidance and ecosystem development

39.33% for Initial Contributors and Private Investors

- Team (20%) 20% of the supply has been allocated to the initial team of contributors, who have been building GoPlus since 2021 and will continue leading the development of Web3’s security layer for years to come. Initial contributors will have **6 months lockup** (cliff) after the TGE, followed by **2 years of monthly linear vesting**, demonstrating our long-term commitment to the project’s success. 🏋️ *For these early contributors, it will take **7 years** to receive all the tokens from the beginning.*
- Early backers from 2021–2024 (19.33%)
- GoPlus has allocated 19.33% of the supply to early backers across different investment rounds. These partners have supported our vision of making Web3 safer. While their initial lockup period is slightly shorter than the team’s, they share the same **2-year monthly linear vesting** after cliff. Complete cliff details will be available in our whitepaper soon.

Summary Unlock Schedule



$GPS Utility

> $GPS Token — Powering the Complete Web3 Security Lifecycle

GoPlus Network is building a comprehensive security infrastructure covering three critical aspects:

- Secure Asset Issuance: Standardizing token creation and deployment
- Secure Asset Management: Professional liquidity and token management
- Secure Trading: Protected transactions and risk prevention

The $GPS token is designed to power every aspect of this security lifecycle:

1. **Security Service Fees**

- End users pay security gas in $GPS when using transaction protection services
- Business users pay in $GPS to access security intelligence
- Projects pay in $GPS to utilize SafeToken Protocol for liquidity management

2. **StakingEcosystem**

Ecosystem contributors have to stake $GPS to become security service computing nodes or security data providers and earn rewards for their contributions.

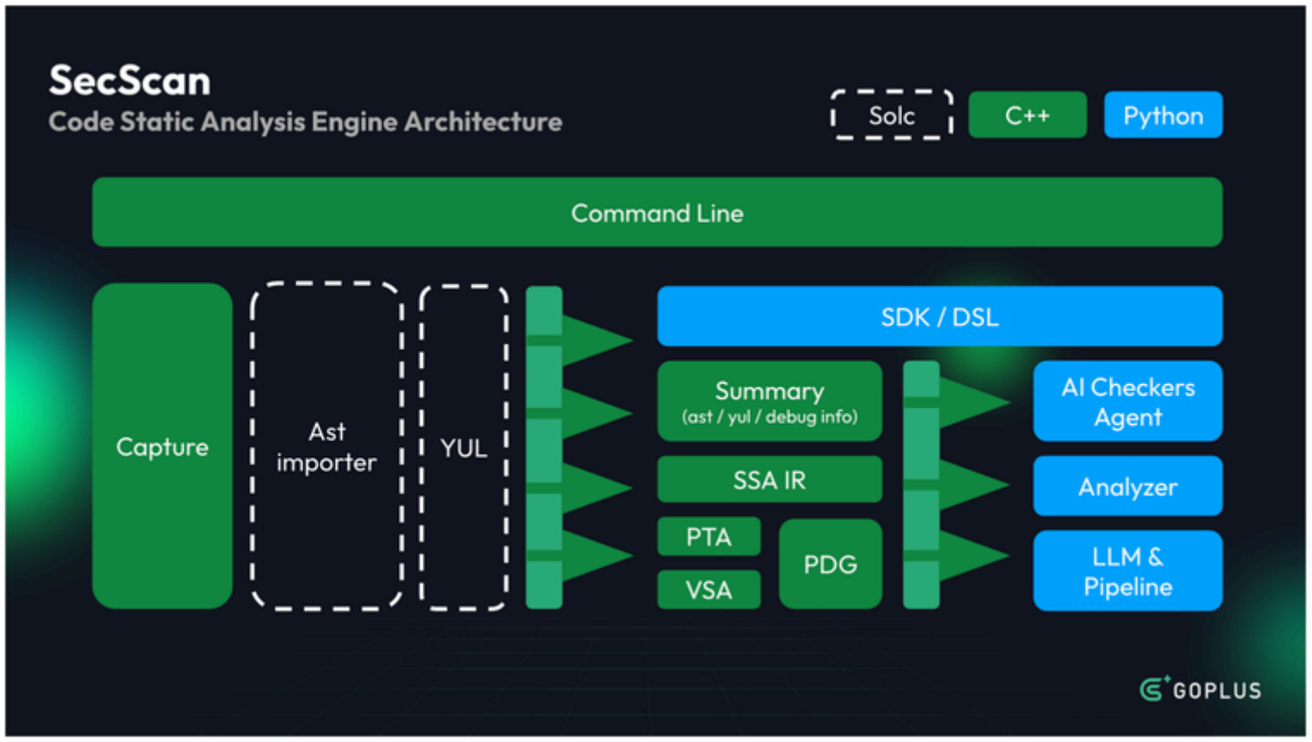
3. **Security Trading Fees**

Cooking 👨‍🍳. Stay tuned for upcoming details on how $GPS will power on-chain Defi trading across all chains.

Conclusion

The launch of the $GPS token marks a new era for GoPlus Security. We’ll share TGE details soon, along with exciting new features for our security products.

Automated Security Testing



SecScan

Introduction

In the world of blockchain, Token is a concept symbolizing different assets that drive decentralized finance. When tokens are implemented as smart contracts, they follow specific standards to ensure unified manipulation, such as making token transfer between addresses. The well-known established standards include ERC-20, ERC-721. However, adhering to a token standard never guarantees the "merit" of a token. Some token owner would insert some malicious behaviors in the token implementations, resulting in token holders being unable to enjoy their rightful asset benefits.

Worse, the malicious tokens deliberately disguise themselves by employing various tricks to conduct their maclious behaviors. Even a seasoned contract security engineer would require significant time to distinguish such issues. In reponse to the severe situation, we introduces **SecScan**, an automatic solution based on cutting-edge static analysis techniques. Overall, **SecScan** accepts ERC-20 tokens of any solidity versions and detects malicous behaviors. The detection process is both efficient, completing in under three seconds per token, and effective.

Next, we delve into the inner operations of **SecScan**. For each input program, we first construct our Intermediate Representation (GIR) from the solidity compiler. Our IR namely GIR, shares the same spirit as program dependence graph. It not only encompasses various dependency information such as data-, control-, and order dependencies, but also provides a comprehensive depiction of these dependencies. Specifically, our data dependence analysis is enhanced with static single assignment form and field sensitive alias analysis, And we have further captured the implicit control dependencies introduced by features such as revert statements, and function modifiers. Subsequently, we employ value flow analysis on the ERC-20 interfaces of the contract to identify different financial concepts, such as balance variables, to enhance GIR. Such financial concept identification enables us to comprehend token behaviors at a high level.

On this powerful GIR, we conduct various kinds of detection for malicious token behaviors, such as abnormal tax modification, minting, and blacklist. A portion of the malicious behaviors is modeled as value flow reachability problem. In this case, we have developed a highly productive value flow engine that abstracts many program details, such as deep function call chains, enabling users to describe malicious behavior at its core. To further enhance productivity, we leveraged Large Language Model (LLM) to help us in writing these checkers. As for the malicious behaviors involving numerical values, we address them strictly with constraint solving. Specifically, the behaviors of a token is encoded into symbolic constraints, while the malicious behaviors are encoded as predicates to check. This crucially guarantees our high precision.

SecScan has identified a significant number of malicious behaviors across various aspects and is continuously ensuring the security of users in investment contracts.

Fuzzing Testing

Paper Link: https://arxiv.org/abs/2312.04512

Background

Smart contracts have emerged as the cornerstone of blockchain technology, enabling the automation and execution of agreements without the need for intermediaries. It has revolutionized the way agreements are executed in decentralized environments, offering transparency, efficiency, and trustlessness.

However, smart contracts, like any software, are susceptible to vulnerabilities that can be exploited by malicious actors. Common vulnerabilities include reentrancy attacks, integer overflow, authorization flaws, and logic errors. These vulnerabilities can result in financial loss, privacy breaches, and disruption of services. While the potential benefits of smart contracts are vast, ensuring the security of users who interact with them is paramount. By focusing on user security, we aim to foster a safer and more resilient ecosystem for smart contract adoption and usage.

Example. Imagine a scenario where a token contract allows users to buy tokens but not sell them, resulting in the token price continuously rising due to the inability to sell. This setup attracts more users to invest, believing they will benefit from the increasing value of the tokens. However, if the contract owner withdraws all funds at this point, the users' funds are drained from the contract, causing significant financial losses to investors. This is a classic example of a honeypot contract, deployed with malicious intent to exploit the trust and naivety of users in the blockchain ecosystem. To mitigate the risks associated with honeypot contracts, users must exercise caution and conduct thorough due diligence before interacting with smart contracts, especially those offering high returns or incentives that seem too good to be true. In addition, developers have a responsibility to prioritize security and transparency when deploying smart contracts. By adhering to best practices and conducting thorough security audits, developers can help prevent the creation of honeypot contracts that pose a threat to unsuspecting users.

Towards smart contract user security, we propose a novel fuzzing paradigm MuFuzz, which can dynamically test smart contract and expose potential vulnerability in the contracts. Fuzzing has been proven to be a practical technique in the field of smart contract security for uncovering vulnerabilities. Users can utilize MuFuzz to preemptively detect smart contracts on the blockchain, allowing them to identify potential pitfalls in the contract beforehand and mitigate possible risks associated with the contract.

Methodology

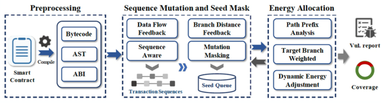


Figure1 A high-level architecture and analysis pipeline of MuFuzz

In the following, we present the specific technical details of MuFuzz, which consists of three key components, namely sequence-aware mutation, mask-guided seed mutation, and dynamic-adaptive energy adjustment. Figure 1 shows the high-level architecture and analysis pipeline of MuFuzz.

A. Sequence-Aware Mutation

MuFuzz begins by taking the contract source code as inputs, which is then compiled into three types of representations, i.e., bytecode, application binary interface (ABI), and abstract syntax tree (AST). Bytecode is disassembled into EVM instructions for fuzzing. Meanwhile, MuFuzz captures the data dependencies of all state variables in the contract. By analyzing the ABI and AST, MuFuzz is able to figure out which state variables are defined and which functions contain state variables. Since smart contracts are stateful programs, MuFuzz ignores functions that do not contain any state variables because they cannot affect the persistent state. MuFuzz then tracks each state variable and its read and write operations, such as assignments and comparisons.

Afterwards, MuFuzz derives a transaction sequence based on the information gathered from the data dependency analysis of the state variables. Put succinctly, MuFuzz approximately determines a sequence of transactions in which transaction T1 is executed before transaction T2 only if T1 writes a state variable V where T2 reads it. As a result, MuFuzz is able to estimate the invocation order of each transaction in the sequence.

B. Mask-Guided Seed Mutation

MuFuzz then determines the inputs of the transaction sequence. A trivial way is to randomly generate the test inputs of transactions. However, due to the randomness, it suffers from inherent difficulties in satisfying complicated branch conditions. To address these challenges, MuFuzz introduces a seed evolution paradigm that iteratively refines the test inputs of transactions. MuFuzz first adopts a branch-distance-feedback seed selection strategy, guiding the fuzzer to select high-quality seeds. Furthermore, MuFuzz employs a mask-guided seed mutation strategy, which allows the fuzzer to identify the certain parts of the test inputs that should not be mutated, thus guiding the seed mutation to hit target branches more efficiently. MuFuzz starts by creating an empty seed queue and a set of seeds as inputs, followed by performing seed selection and seed mutation, respectively.

Branch Distance Feedback. MuFuzz tracks the seed execution and records the branches that each test case covers. Whenever a test case covers a new branch, it is added to the seed queue. While this strategy has been shown to quickly traverse most of the branches, it still has difficulty in covering those branches that are guarded by strict conditions.

Mutation Masking. To further bias test input generation towards target branches, MuFuzz incorporates a mask-guided seed mutation into MuFuzz. The mutation masking strategy derives from the observations that: (1) certain parts of a test input that hits a deeply nested branch are critical to satisfying the necessary conditions for reaching that branch; (2) certain parts of a test input that makes the distance to cover a branch smaller play a key role in approaching that branch. Therefore, to generate more mutated inputs hitting target branches, the crucial parts of the test inputs should not be mutated. Inspired by this, MuFuzz first customizes the selection of test inputs to mutate from the seed queue. It selects the inputs that either hit the deeply nested branches or make the branch distance smaller. We say that a branch *br* is a nested branch if and only if *br* contains at least two nested conditional statements. Each nested branch is associated with a nested score, which is set to the number of nested conditional statements. After filtering out which seeds need to be mutated, MuFuzz introduces a mutation mask computation algorithm, aiming to approximate the critical parts of the test inputs that are not allowed to mutate. MuFuzz engages a set of mutation operators, including byte flipping, replacing bytes with interesting values, byte insertion, and byte deletion.

C. Dynamic Energy Adjustment

In practice, after reviewing a large number of real-world smart contracts, we empirically observe that the updating of state variables tends to be protected by strict branch conditions or hidden in deeply nested branches. Unfortunately, conventional fuzzers may waste massive resources in fuzzing common branches, while the allocated energy is insufficient for the deeply nested branches or branches that are likely to contain bugs. To address this problem, MuFuzz adopts a dynamic-adaptive energy adjustment mechanism, which enables the fuzzing resource allocation for each branch more balanced and flexible.

MuFuzz is equipped with a pre-fuzz phase that executes a test input on an instrumented EVM to collect the exercised path. Given the path *P*, MuFuzz initializes the fuzzing resources. After that, it analyzes all split points (i.e., branch instruction) in *P*. During the pre-fuzz phase, MuFuzz will set a weight value for each exercised branch. Note that the nested branches are assigned different weight values based on the value of nested score, and the branch covering a vulnerable instruction is assigned an additional weight value. It is worth mentioning that the pre-fuzz phase yields little impact on the overall runtime overhead of the fuzzer.

In subsequent fuzzing rounds, MuFuzz dynamically adjusts resource allocation according to the weight value of each branch. This suggests that the higher the weight value of a target branch, the more fuzzing resources will be allocated along the path to that branch. Moreover, MuFuzz also leverages the energy allocation feedback to guide seed mutation, namely the seeds that reach branches covering the vulnerable instructions are preferentially selected and fuzzed. With the assistance of the dynamic-adaptive energy allocation strategy, MuFuzz is able to take care of these target branches, making the fuzzing process more balanced for each branch.

Result Analysis

A. Effectiveness

Bug Type	True Positives
BLOCK DEPENDENCY	15
UNPROTECTED DELEGATECALL	17
ETHER FREEZING	14
INTEGER OVER-/UNDER- FLOW	62
REENTRANCY	16
UNPROTECTED SELF-DESTRUCT	23
STRICT ETHER EQUALITY	19
TRANSACTION ORIGIN USE	2
UNHANDLED EXCEPTION	27
Total	195

Table 1 The nine types of smart contract vulnerabilities can be detected by MuFuzz

MuFuzz now is able to detect nine types of smart contract vulnerabilities. On 155 vulnerable smart contracts, MuFuzz uncovers 195 true positives, which are summarized in Table 1.

B. Real-World Case Study

Bug Type	Reported Bugs	True Positives
BLOCK DEPENDENCY	21	20
UNPROTECTED DELEGATECALL	0	0
ETHER FREEZING	0	0
INTEGER OVER-/UNDER- FLOW	42	42
REENTRANCY	10	7
UNPROTECTED SELF-DESTRUCT	1	1
STRICT ETHER EQUALITY	2	2
TRANSACTION ORIGIN USE	0	0
UNHANDLED EXCEPTION	10	9
Total	86	81

Table 1 Real-World Case Studies of MuFuzz

We randomly select 100 real smart contracts from Etherscan, where each contract contains more than 30,000 transactions in Ethereum. We manually check the bug detection results and classify them into true positives. In addition, we present the overall branch coverage (i.e., the average of the 100 contract runs) of MuFuzz. Table 2 summarizes the experimental results. From the table, we can see that MuFuzz reports a total of 86 bug alarms. Out of the 100 contracts, 39 contracts are flagged as having at least one of these alarms. We manually verify the alarms and confirm that 94% of them are true positives.

Conclusion

Overall, user security is a fundamental consideration in the design and deployment of smart contracts. By actively addressing prevalent vulnerabilities, we can establish a more secure ecosystem conducive to the widespread adoption and utilization of smart contracts. As the smart contract landscape continues to evolve and expand, it is critical to place a strong emphasis on strengthening user security to foster trust in decentralized systems. MuFuzz, with its integration of advanced technologies such as sequence-aware mutation, mask-guided seed mutation, and dynamic adaptive energy adjustment, is a critical tool for dynamically testing smart contracts. By enabling users to proactively identify potential security risks in advance, MuFuzz serves as a critical safeguard for protecting their interests from potential violations.

Phishing Site Detection

Paper Link: https://arxiv.org/abs/2311.12372

Background

Phishing detection is an increasingly critical area in the realm of cybersecurity, addressing the pervasive threat that phishing attacks pose to users' privacy, data security, and trust in digital communications. Phishing, a form of social engineering attack, typically involves deceiving individuals into revealing sensitive information, clicking malicious links, or performing actions that compromise their security. The evolving sophistication of these attacks underscores the urgent need for robust detection mechanisms that can adapt to the changing tactics of adversaries.

The importance of phishing detection extends beyond protecting individual users; it is vital for maintaining the integrity and security of entire digital ecosystems. Effective detection tools help safeguard personal and financial information, preserve the reputation of businesses, and ensure the trustworthiness of online platforms. As we transition into the era of Web3.0, characterized by decentralized networks, blockchain technologies, and a greater emphasis on user sovereignty and data privacy, the landscape of phishing attacks and the strategies for their detection must evolve correspondingly.

Machine learning-based phishing detection technologies, with their robust data processing and learning capabilities, are increasingly supplanting traditional rule-based and signature-based detection methods. Conventional web features, such as page behavior, content, and HTML code, can be harnessed to construct efficient phishing detection models. However, phishing links often have a short lifespan, rendering a vast archive of phishing web page records inaccessible. This scenario limits researchers' ability to retrieve and utilize information related to web content, behavior, or HTML code. Consequently, utilizing URLs to train machine learning models has become a predominant method for phishing detection. Given that URLs serve as gateways to web pages and contain a wealth of information, machine learning models can effectively identify phishing sites by analyzing and learning from these details, even in the absence of additional supportive data.

Solution

We introduce a pre-trained model-guided phishing webpage detection framework utilizing a multi-layer attention mechanism. This framework starts by extracting subword and character-level URL information using a pre-trained network. It then incorporates three pivotal modules: hierarchical feature extraction, layer-aware attention, and spatial pyramid pooling. Hierarchical feature extraction leverages pyramid feature learning to derive multi-level URL embeddings from CharBERT's various Transformer layers. The layer-aware attention module discerns and weights interconnections across hierarchical feature levels. Spatial pyramid pooling further processes the weighted feature pyramid through multiscale downsampling, capturing both local and global feature nuances. Our approach achieves near-perfect detection accuracy in real-world tests.

Backbone Network

We utilize the pretrained CharBERT model as our backbone network, primarily for its ability to focus on both subword and character-level features simultaneously. CharBERT is an enhancement of the BERT model, incorporating the Transformer architecture with a novel dual-channel framework. This framework is specifically designed to capture information at both the subword and character levels. The key advancements in CharBERT consist of two main components: (1) the Character Embedding Module, which encodes character sequences derived from input tokens, and (2) the Heterogeneous Interaction Module, which facilitates the integration and encoding of these character sequences.

Hierarchical Feature Extraction

In deep pre-trained models, even though the output features of one layer serve as the input for the next, the intricate computations within each layer could lead to the degradation of low-to-mid level features, impeding the comprehensive feature learning process. This understanding underscores the necessity of integrating output information across all layers. In this module, we leverage the pretrained model to amalgamate aspect features from every layer during the large-scale, self-supervised URL information learning process. Contrasting with methods that solely rely on the final layer's classification features, our approach significantly enhances detection performance by utilizing the distinct features learned at each layer.

Layer-Aware Attention

To effectively discern and highlight the importance of specific features across various layers, we develop a Layer-Aware Attention mechanism, drawing inspiration from channel attention principles. This mechanism empowers the model to independently discern and assign differentiated weights to feature maps at different layers, thus boosting both processing efficiency and precision. In particular, we consolidate spatial data from pyramid feature maps, extracted via the Hierar-chical Feature Extraction Module, using both average and max pooling. This yields two unique spatial context descriptors.

Spatial Pyramid Pooling

We apply Spatial Pyramid Pooling (SPP) to the weighted feature results. Originally utilized in computer vision tasks and convolutional neural networks, SPP segments feature maps into locally spatial partitions from fine to coarse levels, aggregating local features and thus becoming a key component in classification and detection systems. We innovatively combine SPP with Transformer technology, applying it to the weighted features extracted by our Layer-Aware Attention module. In the final stage of our network, we perform mean pooling along the concatenated feature map and fixed sequence length dimension. This is followed by processing through a standard dropout layer and a fully connected layer, transforming the URL features into a binary class representation for prediction. This methodology enhances the representational capability of features and improves the model's adaptability to different scale features, thereby increasing overall predictive accuracy.

Competitive Advantage

Our approach outperforms the current best methods across a range of challenging real-world scenarios, including class imbalance, few-shot learning, multi-classification, non-independent and identically distributed (non-IIdD) settings, and adversarial attacks. It also achieves near-perfect detection accuracy in online tests.

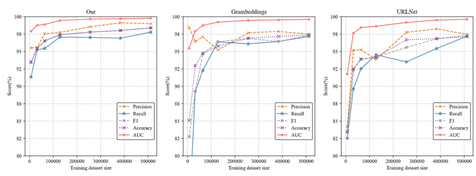


Fig.1 Comparison of small sample learning capabilities

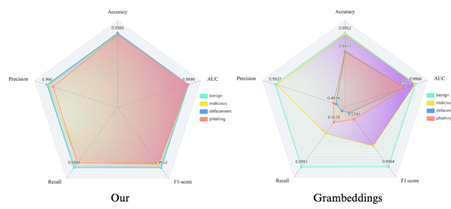


Fig.2 Comparison of multi-classification capabilities

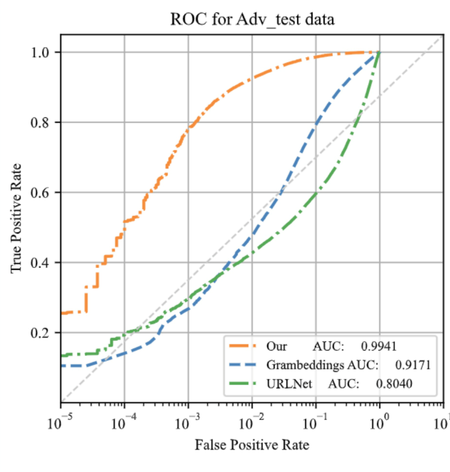


Fig.3 Performance comparison under adversarial sample attack

Phishing Address Detection

Paper Link: https://dl.acm.org/doi/abs/10.1145/3650400.3650499

Background & Motivation

The rapid growth and adoption of blockchain technology, particularly Ethereum, have paved the way for decentralized finance (DeFi) applications. These applications enable peer-to-peer transactions and financial services without the need for traditional intermediaries, offering users increased financial sovereignty and efficiency. However, this technological advancement has also attracted malicious actors who exploit the system for phishing scams, costing users substantial financial losses.

Phishing scams on the Ethereum platform are sophisticated and adaptive, often employing tactics such as giveaway scams and fraudulent investment schemes to deceive users into interacting with malicious accounts. These scams can involve complex smart contracts and transaction patterns that are difficult to detect using traditional anti-phishing methods. The transparent and immutable nature of Ethereum's transaction records, while a boon for security and trust, also presents challenges for scam detection, as attackers continually evolve their strategies to evade existing security measures.

Given the evolving threat landscape, there is an urgent need for advanced detection methods that can identify and mitigate phishing attacks on the Ethereum network. Such methods must be capable of analyzing the intricate patterns of transactions and user interactions within the blockchain ecosystem, taking into account both the spatial relationships between transactions and the temporal sequences in which they occur.

Methodology

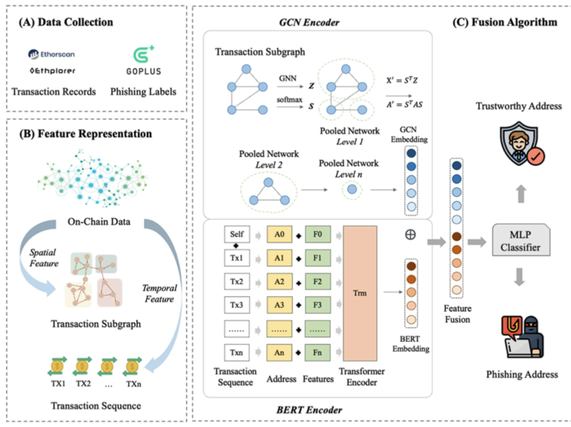


Figure 1. Overview of the STFN workflow

The Spatio-Temporal Fusion Network (STFN) is a sophisticated approach designed to detect phishing scams on the Ethereum network by analyzing both the spatial and temporal aspects of transactions. The methodology is divided into several key steps:

1. **Data Collection**: The initial step involves gathering comprehensive Ethereum transaction data from Etherscan and GoPlus Security. This data includes transaction details such as sender and recipient addresses, transaction amounts, timestamps, and other relevant attributes. This dataset forms the foundation for subsequent analysis.
2. **Transaction Subgraph Construction**: Each Ethereum address is treated as a node in a graph, and transactions between addresses are represented as edges. This constructs a transaction subgraph for each address, capturing the spatial relationships between transactions and the addresses involved. These subgraphs are dynamic and reflect the actual flow of funds within the Ethereum network.
3. **Temporal Sequence Formation**: Concurrently, transaction sequences are formed for each Externally Owned Account (EOA). Transactions are ordered chronologically based on their timestamps, creating a timeline that reflects the temporal progression of an account's activity.
4. **Feature Extraction**: The spatial features are extracted using a Graph Convolutional Network (GCN) encoder. The GCN processes the transaction subgraphs to identify patterns such as transaction direction, amount, and frequency. These features provide insights into the structure and behavior of the transactions around each address.

   Similarly, the temporal features are captured using a BERT encoder. The BERT model, pre-trained on Ethereum transaction sequences, is fine-tuned to generate representations that are sensitive to the order and timing of transactions. This allows the model to identify temporal patterns indicative of phishing activities.
5. **Feature Fusion**: The spatial and temporal features extracted by the GCN and BERT encoders are fused to create a comprehensive representation of each transaction and address. This fusion process is crucial as it allows the model to consider both the 'who' and 'how' of transactions (spatial) as well as the 'when' (temporal).
6. **Machine Learning Classification**: The fused features are then used as input for a machine learning algorithm, specifically a Multilayer Perceptron (MLP), to classify Ethereum addresses into phishing and non-phishing categories. The MLP learns to distinguish between benign and malicious transaction patterns based on the integrated spatial-temporal features.
7. **Evaluation and Optimization**: The performance of STFN is evaluated using standard metrics such as Area Under the Curve (AUC), Precision, Recall, and F1-Score. The model is optimized through techniques such as cross-validation to ensure its robustness and generalizability.

This methodology represents a holistic approach to phishing detection on the Ethereum network, combining the strengths of graph analysis and sequence modeling to effectively identify and mitigate phishing threats. The integration of spatial and temporal features within STFN is a novel contribution to the field of blockchain security, offering a robust solution to protect users from the evolving landscape of cyber threats.

Results

Results of STFN for Ethereum Phishing Detection

The Spatio-Temporal Fusion Network (STFN) has been thoroughly evaluated through a series of experiments to measure its effectiveness in detecting phishing scams on the Ethereum network. STFN's performance was assessed using a range of metrics, including Area Under the Curve (AUC), Precision, Recall, and F1-Score. These metrics provide a multi-faceted view of the model's accuracy, with AUC offering an overall measure of the model's ability to distinguish between phishing and legitimate transactions, and Precision, Recall, and F1-Score providing insights into the model's performance in terms of false positives, false negatives, and overall accuracy.

STFN was compared against several state-of-the-art baseline methods to demonstrate its effectiveness. These baselines included traditional machine learning approaches using handcrafted features, as well as advanced graph-based methods such as DeepWalk, Node2Vec, and Trans2Vec. Additionally, the performance of STFN was compared with more recent methods like Graph Attention Networks (GAT), GraphSAGE, and Temporal Transaction Aggregation Graph Network (TTAGN). STFN achieved an AUC score of 93.26%, indicating a high level of discrimination between phishing and non-phishing transactions. This score is significantly higher than the AUC scores of the baseline methods, showcasing STFN's superior ability to correctly classify transactions. The model also demonstrated excellent Precision, with a score of 91.08%, suggesting that it rarely misclassifies legitimate transactions as phishing attempts. STFN's Recall score of 94.53% indicates its strong capability to identify actual phishing transactions without missing many positive cases. The F1-Score, which harmonizes Precision and Recall, was 92.77%, further confirming the model's overall effectiveness in balancing the detection of phishing transactions while maintaining low error rates.

Conclusion

To conclude, the results of the experiments conducted on STFN indicate that it is a highly effective tool for detecting phishing scams on the Ethereum network. The model's integration of spatial and temporal features, combined with its ability to outperform several state-of-the-art baselines, positions it as a leading solution in the field of blockchain security and phishing scam detection.

💻 API & SDK Integration Guide

GoPlus Security offers a wide range of user security solutions, designed to provide real-time protection against a variety of Web3 threats such as phishing, malicious tokens, and fraudulent activities. These solutions have been widely adopted across the industry, trusted by leading platforms, wallets, and dApps to ensure a safer and more secure blockchain environment.

If you're interested in integrating GoPlus Security's advanced features into your own platform, we provide an easy-to-use suite of APIs & SDK that deliver automated security intelligence. These tools are designed for scalability and seamless integration, empowering developers to protect their users with reliable security insights.

For full documentation and step-by-step integration instructions, visit our API Overview.

🗺️ Roadmap

At GoPlus Network, we have a clear vision and a well-defined roadmap to guide our development and ensure the timely delivery of our key milestones. Our roadmap is focused on expanding our ecosystem, enhancing our security services, and driving the adoption of our token.

> ℹ️ Due to the current stage of development and testing of the entire network, the roadmap will be continuously updated and adjusted according to actual conditions. The currently published roadmap is still just a reference.

2024 Q2-Q3:

- [x] **Launch of Security RPC Services - GoPlus SecNet:** Covering Ethereum and BNB Chain, allowing GoPlus APP users to massively access and experience real-time on-chain risk control.
- [x] **SecHub Launch:** Introduction of the Personal Security Center module, offering users various levels of security risk preferences to allow for the personalized configuration of their security strategies.
- [x] **SecWare Protocol Opening:** Building the GoPlus Network developer ecosystem, enabling more services to serve users through GoPlus APP. Opening the ecosystem to security service companies and developers interested in joining.
- [x] **Release of GoPlus Security AVS (Test Version):** Based on Eigenlayer, this release will achieve the design of a decentralized security computing architecture.
- [x] **USM SDK Release:** Release the USM SDK, which will enable open integration into various RPCs, Wallets as well as Sequencers across different chains, expanding the scale and scope of security services. This will facilitate modular public chain and RaaS integration partnerships.
- [x] **GoPlus Stack Support for Solana:** Providing Solana network users and all applications and wallets with Security Intelligence via API service.

2024 Q4:

- [x] **Release of GoPlus SafeToken Protocol:** Launch of secure token issuance and liquidity management solutions to provide standardized token security implementations.
- [x] **Release of GoPlus Network Compute Layer:** Launch of the Compute Layer based on the Eigenlayer AVS architecture, with operators able to register and join the network.
- [x] **GoPlus Stack Support for SUI:** Providing SUI network users and all applications and wallets with Security Intelligence via API service.
- [x] **Release of Browser Extension:** Launch of the GoPlus Browser Extension with real-time smart risk alerts, wallet scanner and Security Assistant integration.

2025 Q1:

- [x] **Token Generation:** Launch of $GPS token, listed on Cex and Dex.
- [x] **Staking:** Enable GPS token staking mechanisms for network participants.
- [ ] **Browser Extension Open Source:** Release the source code of GoPlus Security browser extension to the community.
- [ ] **Cross-Chain Bridge:** Support token bridge to BSC and Solana networks.
- [ ] **Transaction Security:** Enhance transaction security risk control features.

2025 Q2:

- [ ] **Security Data Layer Test Version:** Introduce our Security Data Layer, allowing users to contribute security data to our network.
- [ ] **Official Release of Security Data Layer:** Official release Security Data Layer, allowing data contributors to become data contribution nodes by staking $GPS. The launch will also include the rollout of our data verification system, ensuring the integrity and reliability of the contributed security data.
- [ ] **Partial Open Sourcing of GoPlus Security Engine:** Opening up the developer platform and Playground for community engagement and development.
- [ ] **AVS GPS Support:** Enable GPS token support for AVS staking mechanisms.
- [ ] **AI Agent Audit:** Launch AI-powered smart contract audit functionality.

Next update we will announce more plans, stay tuned. 🛡️

🫶 Community & Support

At GoPlus Network, we are committed to fostering a vibrant and engaged community of users, developers, and ecosystem partners. We believe that building a strong community is essential to the success and growth of our project.

🫶 COMMUNITY & SUPPORT

Community

Join our community

- 🔗 Official Website
- 🐦 Official Twitter
- 🏠 Official Discord
- 🌍 SecWareX Website
- 🔧 API Documents

📘 Glossary

Glossary of Terms

Term	Definition
GoPlus Network	A decentralized user security infrastructure for the Web3, providing comprehensive security data and services.
SecWare	Security Software, decentralized security services or applications built on top of the GoPlus Network infrastructure, offering various security functionalities to end-users.
SecHub	Security Hub, a user's personalized security center within GoPlus APP, where they can manage their security preferences, interact with SecWare services, and monitor their overall security status.
GSM	GoPlus Security Module, positioned at the core of the network, serving as a bridge between the user-facing applications and the underlying SecWare Protocol and SecWare Service Network which is designed as a software development kit (SDK) that can be seamlessly integrated into various Web3 scenarios, such as wallets, dApps, RPCs, and even Layer 2 sequencers.
AVS	Actively Validated Service, a service that incorporates EigenLayer's verification mechanism, ensuring the integrity and reliability of the service through active validation by multiple parties.
$GPS	GoPlus Network Token
RaaS	Rollup as a service, enabling developers to deploy scalable blockchain with ease, RaaS leverages rollup technology to enhance transaction throughput and scalability on Layer 2 networks while finalizing transactions on the main blockchain
Sequencer	An off-chain component which provides transactions to the L2 execution engine. Transactions posted by the batch submitter to Ethereum are determined by the state of L2 derived by the sequencer node. As part of the main rollup node, it also watches Ethereum to derive the state of the L2 from the previously posted batches and state roots, and to observe deposit transactions. The L2 derivation includes checking that previous state roots and transactions were recorded on Ethereum, as well as tracking things like gas price on Ethereum to update relevant values on L2.
Energy Block / EB	Energy block is currently the points on the GoPlus APP platform, representing an important asset of GoPlus.