Snap Audit for Ducat

Our Clients

Description of service

The service provided is a **Whitbox-Box Penetration Test** following the OWASP guidelines for **MetaMask Snap**.

As part of the research, Sayfer's technical team will look for, investigate and alert on technical, organizational, and logical hazards that could harm one of the customer's products. The team will then write a detailed report specifying the security risks.

During the project, Sayfer will review the code of the MetaMask Snap to ensure user funds are safe. We will verify there are no JS vulnerabilities and that the Snap's code is following the security best practices.

We will perform over 100 different tests on the Snap, all based on the OWASP WSTG V4.2 standard. You can find the list of tests in the appendix here.

After the audit, we also provide 5 hours of consulting and an additional 200 lines of code auditing to ensure the latest version of your deployed contract is completely covered.

The following code will be tested:

Repository
https://github.com/DUCAT-UNIT/ducat-snap

Marketing & PR Package(Optional)

Building Trust in Web3

The Web3 world depends on community and trust. We want to earn users' trust by being transparent about how we keep them secure.

Strengthening Security

After Sayfer submits the final report and the fixes are fixed, we will ping you in our chat and share a short form that will allow us to share useful insights on the product or service you have and about the company.

Sharing Our Security Efforts

With your OK, we will write an article talking about the steps we take together to better safeguard your product and company. This will show users some of the behind-the-scenes work to build trust.

Getting the Word Out

We commit to getting your story on 7 popular crypto sites seen by over 200,000 people. Sites include Benzinga, CoinMarketCap, Crypto Daily, and more. We'll promote it on our social channels too for more visibility.

See Past Examples

Check out one media coverage we recently secured for Bolide Finance on CoinMarketCap

Pricing and Payment Terms

The full payment for the services in this price proposal will be divided as such:

1. **Metamask Snap Audit** - 2.5K USD
2. **Marketing Package(optional) -** 2K USD

Payment will be made before we start working.

The Team

The most important aspect of a quality penetration test is the team. This is why we share with our clients the team members who will conduct the research in every project.

Or D., CTO - With over ten years of experience in server development and IT and more than ten years experience of in the cyber security industry, Or will lead the project and make sure every aspect of the platform is tested.

One of his more interesting findings (which we can publicly disclose) was the famous Badreveal exploit, which affected 10% of all NFT projects. The vulnerability enables attackers to know what is the rarest NFT before the reveal of the project. This allows an attacker an uneven advantage amongst investors to buy the rarest and most expensive piece.

Avigdor Sason Cohen, Web3 Senior Security Researcher - Avigdor is a dedicated security researcher at Sayfer. With a fervent passion for cybersecurity and blockchain, his primary mission is to fortify web3 protocols, making them accessible and secure for broader adoption. Drawing upon his engineering background, Avigdor thrives when faced with intricate systems and challenges, taking pleasure in deconstructing and resolving them.

In his 5 years of experience, Avigdor has already made a substantial mark, conducting dozen of audits as part of his esteemed work at Sayfer. Furthermore, he has delved deep into multiple distinct long-lasting research projects on DeFi security, a testament to his commitment and expertise in the field.

Not just limited to the technical realm, Avigdor's academic accomplishments are commendable. He holds a BSC in Mathematical and Physical Engineering. Taking his passion a notch higher, he pursued a MSC in Cybersecurity from the renowned ESILV Paris engineering school.

Jakub Heba, Lead Security Auditor - Jakub is a cybersecurity expert with over six years of experience in the industry. For two years associated with blockchain technology as a Senior Smart Contract and Blockchain security auditor. He has conducted over 30 audits of various protocols, mostly related to Decentralized Finances (DeFi).

He specializes in the security of contracts written in TypeScript & Rust, in ecosystems such as Cosmos (CosmWasm), Polkadot (Substrate) or MultiversX (Elrond), as well as has a deep technical understanding of EVM and Solidity language.

He participated in assessments testing low-level aspects of blockchain technology, such as finality proof verifications (GRANDPA, BEEFY), serialization libraries (SSZ) and bridges between multiple ecosystems. He has experience in auditing L0/1 Blockchains written in Rust and MOVE.

As an expert, he performed more than fiveteen assessments related to layer between Web2 and Web3, such as off-chain components, wallets and Metamask snaps.

Before moving to Web3, he was a Lead Security Researcher and Penetration Tester managing a team up to 11 engineers, specializing in Web Applications, API's and Red Teaming. He also specialized in low-level binary exploitation in both UNIX and Windows environments. Holder of OSCP and OSCE as well as Lead ISO27001 Auditor certificates.

Project Schedule

The project will start on an agreed-upon date after the following price proposal is signed. Subsequently, the project will follow the following phases:

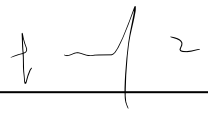
1. One week before the agreed-upon starting date of the project, we will schedule a kickoff meeting. The meeting will provide us with knowledge about the platform with a live demo. The meeting will also include a short business risk analysis.
2. We will work for 1 week on the project from the agreed-upon starting date.
1. We will then open a DM communication channel like slack to ensure you'll be updated in real-time on all important information.
3. Then we will share with the client a full detailed report with all the vulnerabilities we found, including an explanation of all the adjustments that need to be done to eliminate or mitigate the found vulnerabilities.
4. At this point, we enter the revision phase where the client's developers will fix the security findings, and we will make sure these are fixed correctly. This step usually takes one to two revisions. The time for each revision mostly depends on the client.
   We will be available to answer any questions before, during, and after the tests are done.

After finishing the revision phase, your system will have a competent level of security and receive the following **Sayfer Badge** (Available in several design options to suit your design materials).

General Terms

1. To the extent that Sayfer has received payment, IP rights developed by Sayfer for the customer in the performance of the service will be owned by the customer.
2. The customer and Sayfer are independent contractors and this proposal does not create an employment relationship between the customer and Sayfer
3. The service and any report or advice are provided “AS-IS” without warranty of any kind. The customer understands and agrees that there is no guarantee that every vulnerability and possible security issue in the platform will be identified during the service, i.e. the service does nor guarantee the nonexistence of any further findings of security issues. Sayfer does not assume any responsibility or liability in relation to the platform.
4. The customer guarantees that the performance of the service does not violate any laws or rights of third parties and that it has obtained the necessary rights and permissions to permit Sayfer to perform the service.

   If there will be a need to deviate from said guidelines, Sayfer will provide notice to the customer prior to such deviation.

BVI COMPANY NUMBER: 2146148		**Ducat Protocol Inc**
Company Tax/Bn/ID Number	Client Signature	registered company name

Sayfer - Who are we

Sayfer is a leading Web3 cybersecurity company based in Israel. We specialize in working with Web3 SaaS companies and startups, providing high-quality cyber security services and making sure our clients' applications are secure.

By employing highly skilled security researchers who are passionate about hacking, and with the combination of our unique *business cybersecurity risk model* we are able to understand our clients' significant potential security breaches and make sure these are protected.

In our smart contract audits, we follow the latest and extensive Smart Contract Security Verification Standard (SCSVSv2), which is accepted by all major regulations such as SOC2, ISO27001, HIPPA, and many more. The standard is also embraced by tech industry leaders such as the Ethereum Foundation and much more Web3 businesses.

At your service at any time,
Sayfer Company

Our Core Values

We aspire to understand our clients' businesses and operate with the understanding that security is as important as other aspects of their businesses for our clients. Therefore, we start every project with a kickoff meeting and our "Business Cyber Risk Questionnaire" to make sure we build a comprehensive picture of the client's valuable assets and understand the security posture of the client's systems. The information gathered at the meeting will help us to identify and prioritize our efforts when looking for security breaches.

We specialize in performing Web3 security audits. We believe that by focusing our expertise on cyber security and our understanding of Web3, we are able to provide first-rate technical research for our clients.

We understand our client's need for a fast-paced environment. Thus we provide fast results without compromising the quality of the tests. We ensure efficient time management while performing the tests, and we are available to our clients for any questions or assistance both during and after the project.

Appendix 1 - All security tests we will perform on your platform

*Only relevant tests will be performed

Information Gathering	Test Name
WSTG-INFO-01	Conduct Search Engine Discovery Reconnaissance for Information Leakage
WSTG-INFO-02	Fingerprint Web Server
WSTG-INFO-03	Review Webserver Metafiles for Information Leakage
WSTG-INFO-04	Enumerate Applications on Webserver
WSTG-INFO-05	Review Webpage Content for Information Leakage
WSTG-INFO-06	Identify application entry points
WSTG-INFO-07	Map execution paths through application
WSTG-INFO-08	Fingerprint Web Application Framework
WSTG-INFO-09	Fingerprint Web Application
WSTG-INFO-10	Map Application Architecture

Configuration and Deploy Management Testing	Test Name
WSTG-CONF-01	Test Network Infrastructure Configuration
WSTG-CONF-02	Test Application Platform Configuration
WSTG-CONF-03	Test File Extensions Handling for Sensitive Information
WSTG-CONF-04	Review Old Backup and Unreferenced Files for Sensitive Information
WSTG-CONF-05	Enumerate Infrastructure and Application Admin Interfaces
WSTG-CONF-06	Test HTTP Methods

WSTG-CONF-07	Test HTTP Strict Transport Security
WSTG-CONF-08	Test RIA cross domain policy
WSTG-CONF-09	Test File Permission
WSTG-CONF-10	Test for Subdomain Takeover
WSTG-CONF-11	Test Cloud Storage

Identity Management Testing	Test Name
WSTG-IDNT-01	Test Role Definitions
WSTG-IDNT-02	Test User Registration Process
WSTG-IDNT-03	Test Account Provisioning Process
WSTG-IDNT-04	Testing for Account Enumeration and Guessable User Account
WSTG-IDNT-05	Testing for Weak or unenforced username policy

Authentication Testing	Test Name
WSTG-ATHN-01	Testing for Credentials Transported over an Encrypted Channel
WSTG-ATHN-02	Testing for Default Credentials
WSTG-ATHN-03	Testing for Weak Lock Out Mechanism
WSTG-ATHN-04	Testing for Bypassing Authentication Schema
WSTG-ATHN-05	Testing for Vulnerable Remember Password
WSTG-ATHN-06	Testing for Browser Cache Weaknesses
WSTG-ATHN-07	Testing for Weak Password Policy
WSTG-ATHN-08	Testing for Weak Security Question Answer
WSTG-ATHN-09	Testing for Weak Password Change or Reset Functionalities
WSTG-ATHN-10	Testing for Weaker Authentication in Alternative Channel

Authorization	Test Name

Testing	
WSTG-ATHZ-01	Testing Directory Traversal File Include
WSTG-ATHZ-02	Testing for Bypassing Authorization Schema
WSTG-ATHZ-03	Testing for Privilege Escalation
WSTG-ATHZ-04	Testing for Insecure Direct Object References

Session Management Testing	Test Name
WSTG-SESS-01	Testing for Session Management Schema
WSTG-SESS-02	Testing for Cookies Attributes
WSTG-SESS-03	Testing for Session Fixation
WSTG-SESS-04	Testing for Exposed Session Variables
WSTG-SESS-05	Testing for Cross Site Request Forgery
WSTG-SESS-06	Testing for Logout Functionality
WSTG-SESS-07	Testing Session Timeout
WSTG-SESS-08	Testing for Session Puzzling
WSTG-SESS-09	Testing for Session Hijacking

Data Validation Testing	Test Name
WSTG-INPV-01	Testing for Reflected Cross Site Scripting
WSTG-INPV-02	Testing for Stored Cross Site Scripting
WSTG-INPV-03	Testing for HTTP Verb Tampering
WSTG-INPV-04	Testing for HTTP Parameter Pollution
WSTG-INPV-05	Testing for SQL Injection
WSTG-INPV-06	Testing for LDAP Injection
WSTG-INPV-07	Testing for XML Injection
WSTG-INPV-08	Testing for SSI Injection

WSTG-INPV-09	Testing for XPath Injection
WSTG-INPV-10	Testing for IMAP SMTP Injection
WSTG-INPV-11	Testing for Code Injection
WSTG-INPV-12	Testing for Command Injection
WSTG-INPV-13	Testing for Format String Injection
WSTG-INPV-14	Testing for Incubated Vulnerability
WSTG-INPV-15	Testing for HTTP Splitting Smuggling
WSTG-INPV-16	Testing for HTTP Incoming Requests
WSTG-INPV-17	Testing for Host Header Injection
WSTG-INPV-18	Testing for Server-side Template Injection
WSTG-INPV-19	Testing for Server-Side Request Forgery

Error Handling	Test Name
WSTG-ERRH-01	Testing for Improper Error Handling
WSTG-ERRH-02	Testing for Stack Traces

Cryptography	Test Name
WSTG-CRYP-01	Testing for Weak Transport Layer Security
WSTG-CRYP-02	Testing for Padding Oracle
WSTG-CRYP-03	Testing for Sensitive Information Sent via Unencrypted Channels
WSTG-CRYP-04	Testing for Weak Encryption

Business logic Testing	Test Name
WSTG-BUSL-01	Test Business Logic Data Validation
WSTG-BUSL-02	Test Ability to Forge Requests
WSTG-BUSL-03	Test Integrity Checks
WSTG-BUSL-04	Test for Process Timing

WSTG-BUSL-05	Test Number of Times a Function Can be Used Limits
WSTG-BUSL-06	Testing for the Circumvention of Work Flows
WSTG-BUSL-07	Test Defenses Against Application Mis-use
WSTG-BUSL-08	Test Upload of Unexpected File Types
WSTG-BUSL-09	Test Upload of Malicious Files

Client Side Testing	Test Name
WSTG-CLNT-01	Testing for DOM-Based Cross Site Scripting
WSTG-CLNT-02	Testing for JavaScript Execution
WSTG-CLNT-03	Testing for HTML Injection
WSTG-CLNT-04	Testing for Client Side URL Redirect
WSTG-CLNT-05	Testing for CSS Injection
WSTG-CLNT-06	Testing for Client Side Resource Manipulation
WSTG-CLNT-07	Test Cross Origin Resource Sharing
WSTG-CLNT-08	Testing for Cross Site Flashing
WSTG-CLNT-09	Testing for Clickjacking
WSTG-CLNT-10	Testing WebSockets
WSTG-CLNT-11	Test Web Messaging
WSTG-CLNT-12	Testing Browser Storage
WSTG-CLNT-13	Testing for Cross Site Script Inclusion