



Bipart Wallet Security Audit



revision 2.2

Final Report

Prepared for
Bifrost

Prepared by
Theori

November 10th, 2021
(rev 2.1) November 8th, 2021
(rev 2.0) October 22th, 2021
(rev 1.0) July 23th, 2021

Table of Contents

Table of Contents	2
Revision	6
'Bipart Wallet Security Audit' 이행점검	7
rev1.0 - 6. 인증이 필요하지 않은 ROUTE를 통한 인증 우회 (Failed to patch → Fixed)	9
rev1.0 - 7. Private Key, Seed Phrase 조회 시 비밀번호 재입력 권고 (Failed to patch → Fixed)	9
rev1.0 - 8. Private Key, Seed Phrase 조회 시 경고 문구 강화 권고 (Not Fixed → Fixed)	9
rev1.0 - 9. 트랜잭션 전송 시 상세 정보 표기 권고 (Failed to patch → Fixed)	9
rev1.0 - 10. 길이 제한 없는 Origin 렌더링으로 인한 요청 Origin UI 스푸핑 (Not Fixed → Fixed)	9
rev2.0 - 1. 컨트랙트 기반의 토큰이 아닌 기본적으로 등록된 Native Asset의 정보 변경 가능 (Fixed)	10
rev2.0 - 2. RPC 메서드 구현 미흡으로 인한 Denial of Service (Fixed)	10
rev2.0 - 3. 길이 제한 없는 Origin 렌더링으로 인한 요청 Origin UI 스푸핑 (Fixed)	10
rev2.0 - 4. 트랜잭션 전송 시 요청 Origin이 표기되지 않음 (Fixed)	10
rev2.0 - 5. 다수의 트랜잭션 전송 시 상세 정보 표기 권고(Fixed)	10
rev2.0 - 6. 디지털 자산 추가 시 상세 정보 표기 권고 (Fixed)	10
rev2.0 - 7. 추가적인 인증 없이 접근 가능한 개인키, 시드구문 (Fixed)	11
rev2.0 - 8. 개인키, 시드구문 조회 시 경고 문구 강화 권고 (Fixed)	11
rev2.1 - 9. 트랜잭션 전송 시 transfer data 가 있을 경우 Native Asset 전송 수량이 표기되지 않음 (Fixed)	11
Scope of Work	12
Executive Summary	12
Project Overview	13
Application Summary	13
Engagement Summary	13

Issue Breakdown by Severity	14
Issue Breakdown by Category	14
Biport Wallet 검수 개요	15
공격 표면 (Attack Surface)	15
위협 시나리오	15
1. 컨트랙트 기반의 토큰이 아닌 기본적으로 등록된 Native Asset의 정보 변경 가능	17
Description	17
Impact	20
Remediation	21
Short Term	21
Long Term	21
2. RPC 메서드 구현 미흡으로 인한 Denial of Service	22
Description	22
Impact	23
Remediation	25
Short Term	25
Long Term	25
3. 길이 제한 없는 Origin 렌더링으로 인한 요청 Origin UI 스푸핑	26
Description	26
Impact	26
Remediation	28
Short Term	28
Long Term	28
4. 트랜잭션 전송 시 요청 Origin이 표기되지 않음	29
Description	29
Impact	29
Remediation	30
Short Term	30
5. 다수의 트랜잭션 전송 시 상세 정보 표기 권고	31
Description	31

Impact	31
Remediation	31
Short Term	31
Long Term	32
6. 디지털 자산 추가 시 상세 정보 표기 권고	34
Description	34
Impact	34
Remediation	34
Short Term	34
Long Term	34
7. 추가적인 인증 없이 접근 가능한 개인키, 시드구문	36
Description	36
Impact	36
Remediation	37
Short Term	38
Long Term	38
8. 개인키, 시드구문 조회 시 경고 문구 강화 권고	39
Description	39
Impact	40
Remediation	40
Short Term	40
Long Term	41
9. 트랜잭션 전송 시 transfer data 가 있을 경우 Native Asset 전송 수량이 표기되지 않음	42
Description	42
Impact	42
Remediation	44
Short Term	44
Long Term	45
상세 대응 방안 및 추가 권고 사항	46
결론	47

Appendix	48
‘Bifrost Wallet Security Audit - 2021.07.23’ 이행점검	49
1. 연결되지 않은 서비스에서의 TX 요청 (Fixed)	51
2. ETH_SENDTRANSACTION RPC 메서드 파라미터 검증 미흡으로 인한 Denial of Service (Fixed)	51
3. background RPC 핸들링 미흡으로 인한 Denial of Service (Fixed)	52
4. Local Storage에 평문으로 저장되는 지갑 개인키 정보 (Fixed)	52
5. Local Storage에 해시로 저장된 비밀번호 (Fixed)	52
6. 인증이 필요하지 않은 ROUTE를 통한 인증 우회 (Failed to patch)	52
7. Private Key, Seed Phrase 조회 시 비밀번호 재입력 권고 (Failed to patch)	52
8. Private Key, Seed Phrase 조회 시 경고 문구 강화 권고 (Not Fixed)	53
9. 트랜잭션 전송 시 상세 정보 표기 권고 (Failed to patch)	53
10. 길이 제한 없는 Origin 렌더링으로 인한 요청 Origin UI 스푸핑 (Not Fixed)	53
기능 구현 버그	53
1. Private Key를 통해 불러온 지갑이 존재할 경우 BTC 관련 기능 일부 사용 불가	54
2. 피싱 사이트 접속 시 경고 페이지를 보여주는 기능이 작동하지 않음	54
3. 트랜잭션 창에서 발생하는 메모리 누수	55
4. BIFI 서비스 버튼이 올바르지 않은 네트워크 체인에서도 보여짐	55
5. 구현되지 않은 RPC 기능	57

Revision

- 1.0 (July 23th, 2021)

Bifrost Wallet Security Audit - 1차 점검

- 2.0 (October 22th, 2021)

Biport Wallet Security Audit - 2차 점검

- 2.1 (November 8th, 2021)

Biport Wallet Security Audit - 2차 점검에 대한 이행점검

문제점 9 추가

- 2.2 (November 10th, 2021)

Biport Wallet Security Audit - 문제점 9에 대한 이행점검

'Biport Wallet Security Audit' 이행점검

2021년 7월(rev1.0), 2021년 10월(rev2.0) 진행한 Bifrost Wallet Security Audit에서 발견한 사항들이 올바르게 수정되었는지 확인하였으며, 상세 이행점검 내용은 아래와 같다.

Version	2021. 11. 04 • https://github.com/bifrost-platform/Biport Commit Hash: 946e3e11966a6e2436047323e87a3a6fe6ab9573
---------	---

Fixed: 조치가 완료된 문제점

Failed to patch: 당장의 문제점은 수정되었지만 더 나은 보안을 위해 추가적인 조치가 필요한 문제점, 혹은 새로운 문제점이 발견된 경우

Not Fixed: 조치되지 않은 문제점

● 2021년 7월 (rev1.0)

#	문제점	상태 (rev2.1)
1	연결되지 않은 서비스에서의 TX 요청	Fixed
2	ETH_SENDTRANSACTION RPC 메서드 파라미터 검증 미흡으로 인한 Denial of Service	Fixed
3	background RPC 핸들링 미흡으로 인한 Denial of Service	Fixed
4	Local Storage에 평문으로 저장되는 지갑 개인키 정보	Fixed
5	Local Storage에 해시로 저장된 비밀번호	Fixed
6	인증이 필요하지 않은 ROUTE를 통한 인증 우회	Fixed (Update)
7	Private Key, Seed Phrase 조회 시 비밀번호 재입력 권고	Fixed (Update)
8	Private Key, Seed Phrase 조회 시 경고 문구 강화 권고	Fixed (Update)

9	트랜잭션 전송 시 상세 정보 표기 권고	Fixed (Update)
10	길이 제한 없는 Origin 렌더링으로 인한 요청 Origin UI 스푸핑	Fixed (Update)

- 2021년 10월 (rev2.0)

#	문제점	상태 (rev2.1)
1	컨트랙트 기반의 토큰이 아닌 기본적으로 등록된 Native Asset의 정보 변경 가능	Fixed
2	RPC 메서드 구현 미흡으로 인한 Denial of Service	Fixed
3	길이 제한 없는 Origin 렌더링으로 인한 요청 Origin UI 스푸핑	Fixed
4	트랜잭션 전송 시 요청 Origin이 표기되지 않음	Fixed
5	다수의 트랜잭션 전송 시 상세 정보 표기 권고	Fixed
6	디지털 자산 추가 시 상세 정보 표기 권고	Fixed
7	추가적인 인증 없이 접근 가능한 개인키, 시드구문	Fixed
8	개인키, 시드구문 조회 시 경고 문구 강화 권고	Fixed

- 2021년 11월 (rev2.1)

#	문제점	상태 (rev2.2)
9	트랜잭션 전송 시 transfer data 가 있을 경우 Native Asset 전송 수량이 표기되지 않음	Fixed

rev1.0 - 6. 인증이 필요하지 않은 ROUTE를 통한 인증 우회
(Failed to patch → Fixed)

- allowInitializePages / allowUnlockPages 두 가지 상태를 구분지어 검증.

rev1.0 - 7. Private Key, Seed Phrase 조회 시 비밀번호 재입력 권고
(Failed to patch → Fixed)

- 추가적인 인증 없이 접근이 가능했던 Route를 삭제함
다만 Background에서 주요 정보(개인키, 시드구문)를 반환할 때 이용자의 비밀번호를 사용하지 않고 있어 [THE-BIOPORTWALLET-007](#) Remediation에 작성된 Long Term 내용에 따라 추가적인 조치를 할 것을 권고함.

rev1.0 - 8. Private Key, Seed Phrase 조회 시 경고 문구 강화 권고
(Not Fixed → Fixed)

- 개인키, 시드구문 조회 시 경고문구가 강화됨.

rev1.0 - 9. 트랜잭션 전송 시 상세 정보 표기 권고
(Failed to patch → Fixed)

- 트랜잭션 전송 시 요청 Origin을 표기함.

rev1.0 - 10. 길이 제한 없는 Origin 렌더링으로 인한 요청 Origin UI 스푸핑
(Not Fixed → Fixed)

- Origin이 출력되는 페이지 별로 Style 옵션(lineBreak) 추가를 통해 모든 내용을 출력.

rev2.0 - 1. 컨트랙트 기반의 토큰이 아닌 기본적으로 등록된 Native Asset의 정보 변경 가능 (Fixed)

- Web3.getCode 함수를 사용해서 컨트랙트인지 검증 후 사용함. 이 외에도 상세 정보 표기 및 블록체인상의 데이터와 일치 여부를 출력하게 수정됨.
다만 특정 네트워크의 경우 genesis block에서 특정 주소(0x0000...0000)에 컨트랙트 코드를 할당했기 때문에 예외가 발생할 수 있다.(e.g. Klaytn ^{1,2})

rev2.0 - 2. RPC 메서드 구현 미흡으로 인한 Denial of Service (Fixed)

- 구현체 추가 완료 및 버그 수정.

rev2.0 - 3. 길이 제한 없는 Origin 렌더링으로 인한 요청 Origin UI 스푸핑 (Fixed)

- Origin이 출력되는 페이지 별로 Style 옵션(lineBreak) 추가를 통해 모든 내용을 출력.

rev2.0 - 4. 트랜잭션 전송 시 요청 Origin이 표기되지 않음 (Fixed)

- 트랜잭션 전송 시 요청 Origin을 표기함.

rev2.0 - 5. 다수의 트랜잭션 전송 시 상세 정보 표기 권고(Fixed)

- 다수의 트랜잭션 전송이 요청될 경우 몇개의 트랜잭션이 요청되었는지, 몇번째 트랜잭션 전송 요청을 처리중인지 표기함.

rev2.0 - 6. 디지털 자산 추가 시 상세 정보 표기 권고 (Fixed)

- 디지털 자산 추가 시 상세 정보(address, decimals, token name/symbol) 표기 및 블록체인상의 데이터와 일치 여부를 출력하게 수정됨.

¹ https://raw.githubusercontent.com/klaytn/klaytn/9bde9/blockchain/genesis_alloc.go

² <https://github.com/klaytn/klaytn/blob/9bde9/contracts/cypress/credit.so>

rev2.0 - 7. 추가적인 인증 없이 접근 가능한 개인키, 시드구문 (Fixed)

- 추가적인 인증 없이 접근이 가능했던 Route를 삭제함
다만 Background에서 주요 정보(개인키, 시드구문)를 반환할 때 이용자의 비밀번호를 사용하지 않고 있어 [THE-BIOPORTWALLET-007](#) Remediation에 작성된 Long Term 내용에 따라 추가적인 조치를 할 것을 권고함.

rev2.0 - 8. 개인키, 시드구문 조회 시 경고 문구 강화 권고 (Fixed)

- 개인키, 시드구문 조회 시 경고문구가 강화됨.

rev2.1 - 9. 트랜잭션 전송 시 transfer data 가 있을 경우 Native Asset 전송 수량이 표기되지 않음 (Fixed)

- 전송되는 Native Asset의 수량이 트랜잭션 창 하단에 표기됨.

Scope of Work

Biport Wallet 웹 브라우저 익스텐션 코드 레벨 화이트박스 보안 검수

- Biport Wallet Extension

- <https://github.com/bifrost-platform/Biport>

1차 전달 Commit Hash: 35453928422206fc3ff88b89d94d62464cc2a522

2차 전달 Commit Hash: 0a846ae39da8557b0e3e03f976083a9352199cd6

Executive Summary

본 보안 컨설팅 수행은 Bifrost의 Wallet Browser Extension 서비스에 대한 보안 문제점 개선 사항을 파악하여 이에 대한 적절한 대책을 수립하기 위한 것으로, 소스코드를 인계받아 문제점 및 권고 사항을 도출하는 화이트박스 테스팅으로 진행하였다.

본 프로젝트 보고서는 2021년 10월 11일부터 2021년 10월 22일까지 2주일간 수행하는 코드 보안 검수를 바탕으로 작성 되었다.

- 소스 코드 인계: 10/11(월), 10/15(금)
- 정적 코드 분석: 10/11(월) ~ 10/22(금)
- 프로젝트 보고: 10/22(금)

Biport Wallet 서비스에 대한 이해와 위협 모델링을 통해 공격 표면을 도출하고 소스코드 보안 검수를 진행해 8건의 문제점을 발견하였으며, Native Asset 정보 수정, 공격자의 사이트 방문 후 더 이상 Biport Wallet을 사용할 수 없도록 하는 DoS 취약점 등 이용자 대상으로 가용성을 해치거나, 금전적 손실을 발생시킬 수 있는 위협이 발견되었다.

Project Overview

Application Summary

Name	Biport Wallet
Version	2021. 10. 11 • https://github.com/bifrost-platform/Biport 1차 전달 Commit Hash: 35453928422206fc3ff88b89d94d62464cc2a522 2차 전달 Commit Hash: 0a846ae39da8557b0e3e03f976083a9352199cd6
Application Type	Web Browser Extension
Platforms	React, WebExtensions API, Typescript

Engagement Summary

Dates	2021. 10. 11. - 2021. 10. 22. (2 Weeks)
Methodology	Source code auditing

Issue Breakdown by Severity

분류	Count	발견된 문제점
Medium	3	<ul style="list-style-type: none">• THE-BIOPORTWALLET-001• THE-BIOPORTWALLET-002• THE-BIOPORTWALLET-009
Low	1	<ul style="list-style-type: none">• THE-BIOPORTWALLET-003
Informational	5	<ul style="list-style-type: none">• THE-BIOPORTWALLET-004• THE-BIOPORTWALLET-005• THE-BIOPORTWALLET-006• THE-BIOPORTWALLET-007• THE-BIOPORTWALLET-008

Issue Breakdown by Category

분류	Count	발견된 문제점
Modification of Assumed-Immutable Data	1	<ul style="list-style-type: none">• THE-BIOPORTWALLET-001
Improper Check for Exceptional Conditions	1	<ul style="list-style-type: none">• THE-BIOPORTWALLET-002
UI Spoofing	2	<ul style="list-style-type: none">• THE-BIOPORTWALLET-003• THE-BIOPORTWALLET-009
UI Improvement	3	<ul style="list-style-type: none">• THE-BIOPORTWALLET-004• THE-BIOPORTWALLET-005• THE-BIOPORTWALLET-006
Insufficient UI Warning of Dangerous Operations	1	<ul style="list-style-type: none">• THE-BIOPORTWALLET-008
Improper Authentication	1	<ul style="list-style-type: none">• THE-BIOPORTWALLET-007

Biport Wallet 검수 개요

공격 표면 (Attack Surface)

Biport Wallet 보안 검수 전 이용자 혹은 공격자의 값이 입력되는 표면을 도출하였다. Browser Extension 서비스에 존재하는 표면을 분리하였고 공통적으로 입력될 수 있는 공격 표면을 나열하였다.

ContentScript: Extension content script

Inpage: Extension content script가 삽입한 inpage script

Background: Extension background script

External: 외부 서비스

- 변조된 네트워크 RPC 혹은 API 응답 (Background ↔ External)
 - 사전 정의된 네트워크 RPC(bitcore, infura)
 - UTIL API: ETH GAS STATION
 - 커스텀 네트워크 RPC: 지원하지 않음
- 네트워크 Man in the Middle (MITM) 공격 (Background ↔ External)
- 블록체인 데이터 (Background ↔ External)
 - 트랜잭션, 컨트랙트 등
- Wallet RPC Method (Web Service ↔ Inpage ↔ ContentScript ↔ Background)
- Wallet Route
- Biport Wallet에서 사용하는 오픈소스 라이브러리 (npm package)
- 이용자의 기기 (Web Service / Inpage / Background / Local Storage)

위협 시나리오

나열된 각 공격 표면에서 발생할 수 있는 위협 시나리오는 아래와 같다.

- 멀티체인지갑
 - 이용자가 선택한 네트워크 (RPC)가 아닌 변조된 네트워크 사용
 - 타 네트워크에서 사용 가능한 서명 정보 획득
- 연결되지 않은 서비스 인증 우회
 - 연결되지 않은 서비스 (Origin)에서 제한된 RPC 호출
- Cross Site Scripting (XSS)
 - 악의적인 RPC 호출로 Browser Extension 내 background.js (Browser-Extension Origin)에 Cross Site Scripting (XSS) 공격
 - Browser Local Storage에 저장되어 있는 값 탈취
- UI-Spoofing: [THE-BIPORTWALLET-001](#), [THE-BIPORTWALLET-003](#),
[THE-BIPORTWALLET-004](#), [THE-BIPORTWALLET-005](#), [THE-BIPORTWALLET-006](#)
 - 이용자에게 보여지는 UI를 실제 내용과 다르게 표시하는 공격
- Denial of Service (DoS): [THE-BIPORTWALLET-002](#)
 - 악의적인 요청으로 이용자의 서비스 이용을 거부하는 공격
 - 블록체인 상의 데이터가 로드될 때 앱 내 계정 정보나 데이터가 삭제되는 경우
 - 앱을 사용할 수 없게 만드는 경우
- 기기 탈취: [THE-BIPORTWALLET-007](#)
 - 물리적으로 이용자의 기기가 탈취되는 공격 / 타 프로그램 등의 취약점으로 이용자의 기기가 탈취되는 공격
 - 기기 탈취 후 Biport Wallet에 저장된 지갑 정보를 얼마나 얻어낼 수 있는지의 여부
- Network Layer에서 발생하는 Man in the middle 공격, DNS Spoofing 등
 - Network Layer에서 통신하는 패킷을 변조해 발생하는 공격
 - MITM 공격을 통해 통신하는 데이터 변조
 - DNS Spoofing을 통해 원하는 도메인의 정보 조작
- 오픈소스 라이브러리 1-day 를 통한 기타 공격
- 기타 권고 사항: [THE-BIPORTWALLET-008](#)

1. 컨트랙트 기반의 토큰이 아닌 기본적으로 등록된 Native Asset의 정보 변경 가능

Severity: Medium

Finding ID: THE-BIPORTWALLET-001

Type: Modification of Assumed-Immutable Data

Target: Browser Extension

Description

Biport Wallet은 설치시 자산 리스트에 기본적으로 추가되어 있는 Native Asset³이 존재한다. 이 외의 자산들은 Biport Wallet 내 UI를 사용하거나, wallet_watchAsset 메소드를 사용하여 추가가 가능하다. 이러한 기능을 사용하여 추가되는 자산은 컨트랙트 기반의 토큰이다. 하지만 wallet_watchAsset 메소드의 인자로 컨트랙트의 주소를 0x0000...0000으로 지정하면 컨트랙트 기반의 토큰이 아닌 Native Asset의 심볼과 decimals를 변경할 수 있다.

```
ethereum.request({
  method: 'wallet_watchAsset',
  params: {
    type: 'ERC20',
    options: {
      address: '0x0000000000000000000000000000000000000000000000000000000000000000',
      symbol: 'EXAMPLE', // 심볼 변경
      decimals: 19, // decimals 변경
      image: 'https://s3.amazonaws.com/airswap-token-images/ETH.png',
    },
  },
})
```

wallet_watchAsset을 통해 Native Asset의 심볼과 decimals값 변경 PoC

위 PoC를 Javascript Console에서 실행하면 이용자에게 토큰을 추가할지 묻는 팝업창이 나타난다. 이용자가 수락을 누르면 'EXAMPLE' 토큰이 추가되는 것이 아닌, 이용자 지갑에 선택되어 있는 체인의 Native Asset 심볼이 'EXAMPLE'로 변경되며, decimals 값이 19로 변경된다.

다음은 디지털 자산 추가시 호출되는 함수이다.

³ 이더리움, 이더리움 - Kovan, BNB 코인, BNB 코인 - 테스트넷, 비트코인

```
// src/scripts/controllers/preferences.js
async addToken(rawAddress, symbol, decimals, image) {
    const address = normalizeAddress(rawAddress);
    const newEntry = { address, symbol, decimals: Number(decimals) };
    // tokens에는 컨트랙트 기반의 디지털 자산뿐만 아니라 이더리움, 비트코인같은 디지털
    // 자산도 포함
    const { tokens, hiddenTokens } = this.store.getState();
    // ...
    const previousEntry = tokens.find((token) => {
        return token.address === address; // address 값을 기준으로 기존에 있는
        // 코인인지 판단
    });
    const previousIndex = tokens.indexOf(previousEntry);
    // ...
    tokens[previousIndex] = newEntry;
}
```

디지털 자산 추가 시 호출되는 addToken 함수

해당 함수에서 사용되는 tokens 변수에는 컨트랙트 기반의 토큰 뿐만 아니라 이더리움, 비트코인같은 Native Asset도 포함되어 있다. 그리고 기본적으로 추가되어 있는 Native Asset의 정보는 다음과 같다.

```
// src/constants/asset.ts
export const NATIVE_TOKEN_ADDRESS =
'0x000000000000000000000000000000000000000000000000000000000000000';

export const DEFAULT_TOKEN = [
{
    address: NATIVE_TOKEN_ADDRESS,
    decimals: 18,
    image: 'https://s3.amazonaws.com/airswap-token-images/ETH.png',
    symbol: 'ETH',
    chainId: NETWORK[MAINNET].chainId,
},
// ...
];
```

기본적으로 추가되어 있는 Native Asset 정보

예시에 나타나 있는 디지털 자산의 정보는 이더리움이며, 컨트랙트 기반의 토큰이 아니므로 컨트랙트 주소가 0x0000...0000로 초기화되어 있다.

따라서 컨트랙트 주소를 0x0000...0000 으로 설정하여 addToken 함수를 호출할 경우 기본적으로 추가되어 있는 Native Asset 정보를 변경하게 된다.

The image displays two side-by-side screenshots of a mobile wallet application interface, showing the state of native assets on the Ethereum Mainnet before and after a Proof of Concept (PoC) execution.

Left Screenshot (PoC Execution Before):

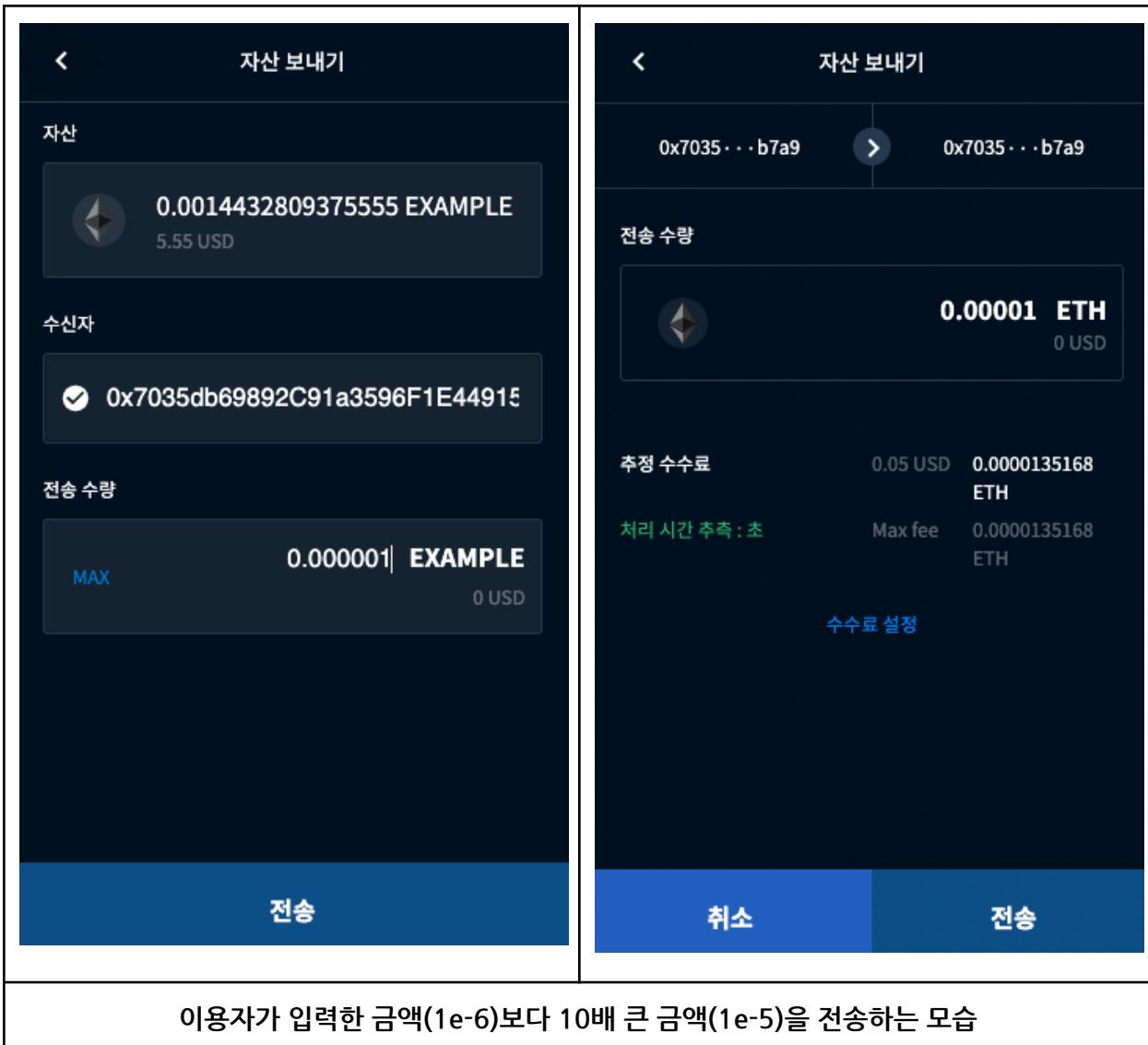
- Top bar: Ethereum Mainnet, 멀티체인 지갑, Settings icon.
- Balance: \$ 0
- Buttons: BiFi 서비스, 보내기, 받기, 토큰 관리.
- Table:
 - Asset: Ethereum (Ethereum Mainnet), Value: 0 ETH, 0 USD.
 - Asset: Ethereum (Kovan), Value: 0.0014 ETH.
 - Asset: Binance Coin (Binance Smart Chain), Value: 0 BNB, 0 USD.
 - Asset: Binance Coin (Binance Smart Chain Test), Value: 0 BNB.

Right Screenshot (PoC Execution After):

- Top bar: Ethereum Mainnet, 멀티체인 지갑, Settings icon.
- Balance: \$ 0
- Buttons: BiFi 서비스, 보내기, 받기, 토큰 관리.
- Table:
 - Asset: Ethereum (Ethereum Mainnet), Value: 0 EXAMPLE, 0 USD.
 - Asset: Ethereum (Kovan), Value: 0.0014 ETH.
 - Asset: Binance Coin (Binance Smart Chain), Value: 0 BNB, 0 USD.
 - Asset: Binance Coin (Binance Smart Chain Test), Value: 0 BNB.

Text at the bottom: PoC로 인해 변경된 Native Asset의 심볼 (좌: PoC 실행 전, 우: PoC 실행 후)

이를 통해, Native Asset의 decimals 값을 변경하여 이용자가 입력한 전송 금액과 실제 전송 금액을 다르게 만들 수 있다. 자산 전송 시, 위 PoC에 의해 이더리움의 decimals 값이 19로 바뀐 경우 다음과 같이 작동한다.



Impact

디지털 자산의 decimals, 심볼과 같은 정보를 변경하여 이용자에게 혼란을 발생시킬 수 있다.
공격자는 decimals 값을 변경하는 것을 통해 이체 금액이 이용자가 의도한 금액과 다르게 설정하여 이를 피싱에 악용 할 가능성이 있다.

Remediation

Short Term

- 디지털 자산 추가 시, 현재 추가/변경 하고자 하는 디지털 자산이 컨트랙트 기반 토큰인지 확인해야 한다. 만약 컨트랙트 기반 토큰이 아니며, 추가/변경 하고자 하는 디지털 자산이 기본으로 추가되어 있는 자산이면 요청을 거절해야 한다.
- 현재 Biport Wallet은 디지털 자산 추가 시, 상세한 정보 대신 디지털 자산의 이미지, 심볼, 현재 잔고만을 표시한다. 그로 인해 RPC-API를 통해 추가하려는 자산의 정보를 정확하게 이용자가 인식하는 것이 불가능하다. 따라서 위 정보와 함께 decimals나 컨트랙트 주소와 같은 상세한 정보를 보여줘야 한다.

Long Term

- RPC 메소드를 통해 추가하려는 디지털 자산이 이미 Biport 토큰 관리에 존재하는 Native Asset과 일치할 경우 RPC 메소드를 통해 전달된 인자(이미지, 심볼, decimals)를 직접 등록하는 것이 아닌, 신뢰할 수 있는 경로로부터 가져온 정보를 사용하여 디지털 자산을 추가해야 한다.

Reference

- [외부 입력 데이터에 대한 상세 표기 권고](#)
- [THE-BIPORTWALLET-006](#)

2. RPC 메서드 구현 미흡으로 인한 Denial of Service

Severity: Medium

Finding ID: THE-BIPORTWALLET-002

Type: Improper Check for Unusual or
Exceptional Conditions

Target: Browser Extension

Description

RPC 메서드중 일부 메서드는 이용자의 수락(User Interaction)이 필요하다. 이러한 RPC 메서드가 호출되면 Biport Wallet에서는 이용자의 수락을 받기 위해 notification.html 을 팝업창으로 띄운 다음, 호출한 RPC 메서드에 해당하는 화면을 보여준다. 이 때 몇몇 메서드는 팝업창을 띄우는 함수를 잘못 호출하여 예외가 발생하게 된다. 다음은 예외가 발생하는 함수이다.

```
// src/scripts/bifrost-controller.js

// eth_sign
newUnsignedMessage(msgParams, req) {
    const promise = this.messageManager.addUnapprovedMessageAsync(msgParams,
req);
    this.sendUpdate();
    this.opts.showUserConfirmation();
    return promise;
}

// personal_sign
async newUnsignedPersonalMessage(msgParams, req) {
    const promise =
this.personalMessageManager.addUnapprovedMessageAsync(msgParams, req);
    this.sendUpdate();
    this.opts.showUserConfirmation();
    return promise;
}

// eth_signTypedData
newUnsignedTypedMessage(msgParams, req, version) {
    const promise =
this.typedMessageManager.addUnapprovedMessageAsync(msgParams, req, version);
    this.sendUpdate();
    this.opts.showUserConfirmation();
    return promise;
}
```

```

}

// eth_decrypt
async newRequestDecryptMessage(msgParams, req) {
    const promise =
this.decryptMessageManager.addUnapprovedMessageAsync(msgParams, req);
    this.sendUpdate();
    this.opts.showUserConfirmation();
    return promise;
}

// eth_getEncryptionPublicKey
async newRequestEncryptionPublicKey(msgParams, req) {
    // ...
    const promise =
this.encryptionPublicKeyManager.addUnapprovedMessageAsync(msgParams, req);
    this.sendUpdate();
    this.opts.showUserConfirmation();
    return promise;
}

```

존재하지 않는 객체 참조로 인해 에러가 발생하는 함수

BifrostController 클래스는 constructor에서 opts 객체를 인자로 전달 받지만, 저장하지 않기 때문에 클래스 내 opts 객체는 존재하지 않는 값이다. 따라서 opts 객체의 showUserConfirmation을 호출하게 되면 ‘Cannot read properties of undefined’ 에러가 발생하며, 그로 인해 페이지를 렌더하는 중에 에러가 발생하여 페이지가 정상적으로 출력되지 않는다.

Impact

사이트에서 예외가 발생하는 RPC 요청을 보내 DoS(Denial of Service)를 발생시키고, 이용자가 브라우저를 재시작하거나 확장 프로그램을 재로드하기 전까지 Biport Wallet을 정상적으로 사용할 수 없게 만들 수 있다.

DoS를 유발시키는 PoC는 다음과 같다.

```
await ethereum.request({
  method: 'eth_decrypt',
  params: ["xxx", ethereum.selectedAddress],
})

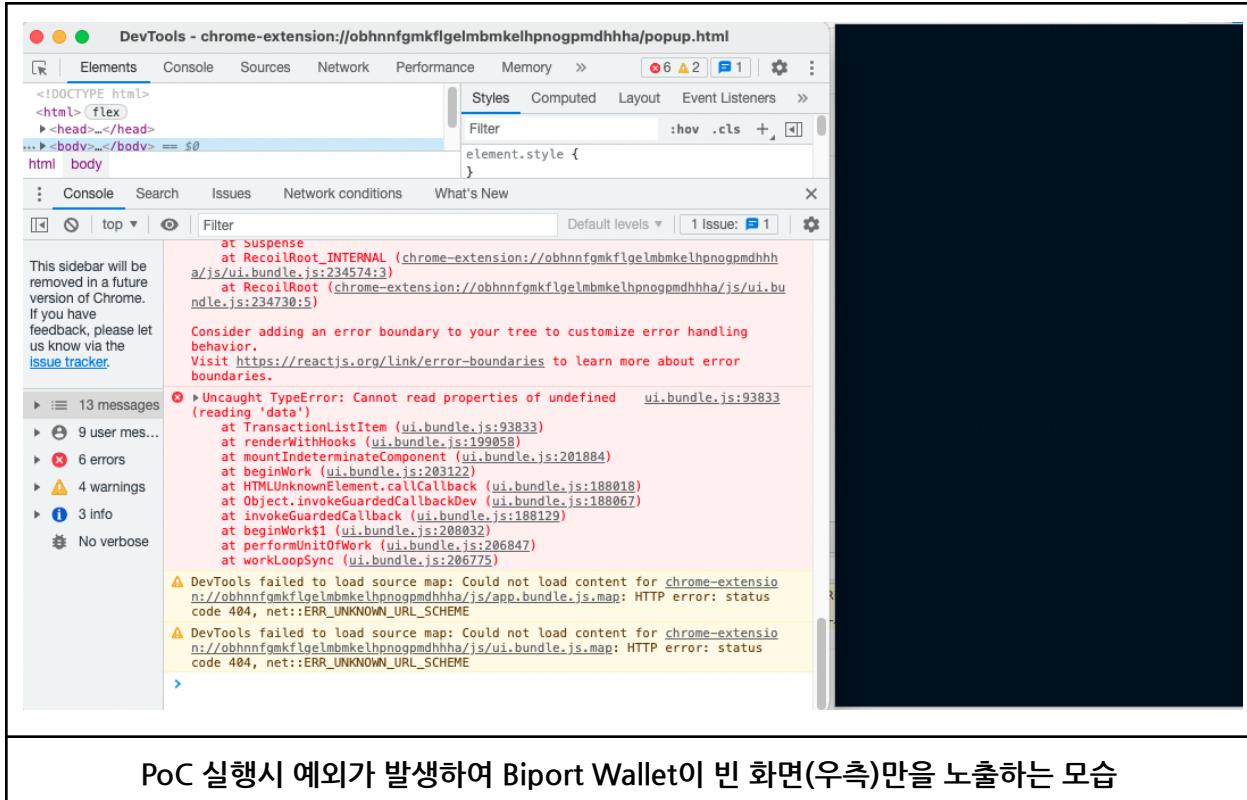
await ethereum.request({
  method: 'personal_sign',
  params: ["xxx", ethereum.selectedAddress],
})

await ethereum.request({
  method: 'eth_signTypedData',
  params: ["xxx", ethereum.selectedAddress],
})

await ethereum.request({
  method: 'eth_decrypt',
  params: ["xxx", ethereum.selectedAddress],
})

await ethereum.request({
  method: 'eth_getEncryptionPublicKey',
  params: [ethereum.selectedAddress],
})
```

DoS를 유발시키는 PoC들



PoC 실행시 예외가 발생하여 Biport Wallet이 빈 화면(우측)만을 노출하는 모습

Remediation

Short Term

- 기능에서 예외가 발생하더라도 프로그램이 서비스 거부 상태가 되지 않도록 각 기능 별 예외 처리를 구현한다.
- 현재 구현이 완벽하게 되어있지 않아 예외가 발생하는 RPC에 대해 구현체를 작성한다.

Long Term

- N/A

3. 길이 제한 없는 Origin 렌더링으로 인한 요청 Origin UI

스푸핑

Severity: Low

Finding ID: THE-BIPORTWALLET-003

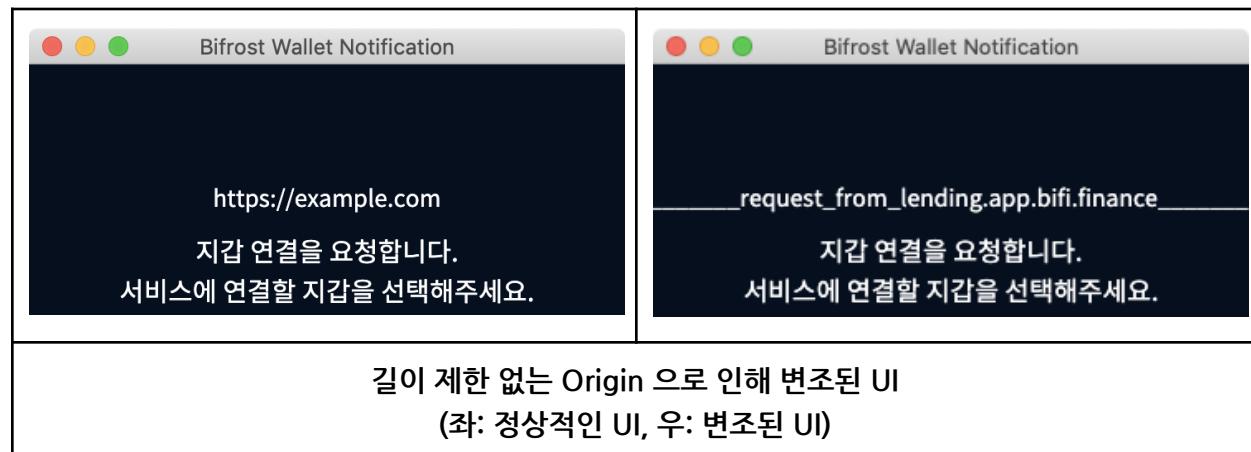
Type: UI Spoofing

Target: Browser Extension

Description

Bipart Wallet은 웹 페이지와 Browser Extension 간의 RPC를 지원하기 위해 웹 페이지에 Inpage Provider를 제공한다. 웹 페이지는 Inpage Provider를 통해 Bipart Wallet의 background에 요청을 보낼 수 있다.

Bipart Wallet은 웹 페이지의 RPC 요청을 처리할 때 웹 페이지의 권한을 제어하기 위해 RpcCap⁴ 라이브러리를 사용한다. 만약 웹 페이지가 Inpage Provider를 통해 RPC 요청을 보냈을 때 권한이 없는 경우 지갑 연결을 요청하는 화면이 표시된다.

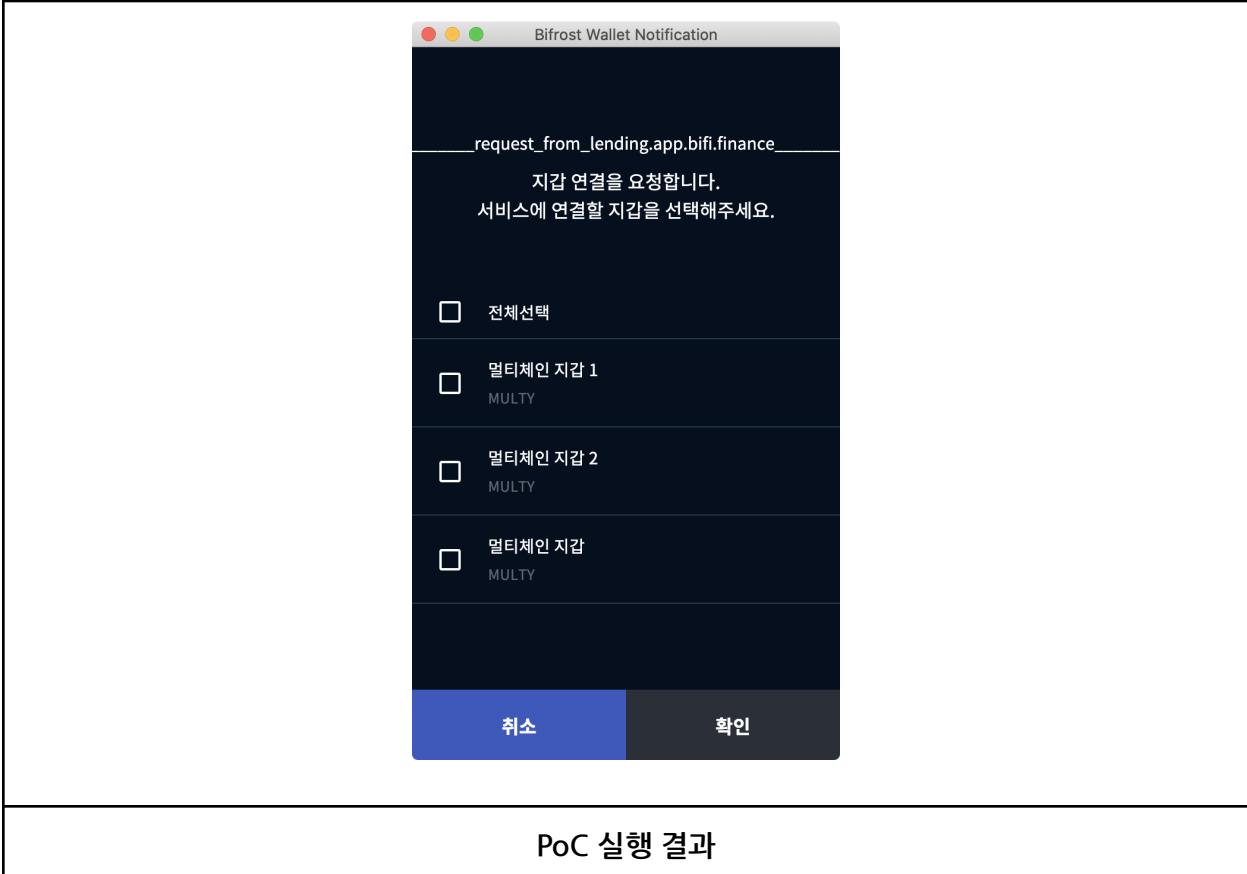


요청 Origin 을 렌더링 하는 과정에서 Origin의 길이 제한을 하지 않아 요청한 사이트의 Origin이 아닌 임의의 Origin이 표기되게 하는 UI Spoofing 공격에 취약하다.

⁴ <https://github.com/MetaMask/rpc-cap>

Impact

아래는 악성주소⁵에서 Biport Wallet에 요청을 보낸 PoC이다. UI-Spoofing이 발생해 이용자는 타 Origin("lending.app.bifi.finance")에서 요청을 보낸 것으로 혼동할 수 있다.



5

http://aaaaaaaaaaaa.aaaaaaaaaaaaaa.a.request_from_lending.app.bifi.finance.bifi.finance.bifi.finance.others.domain/

Remediation

Short Term

- 현재는 Origin이 화면에 한번에 표시할 수 있는 길이보다 길게 보내면 앞, 뒤 내용이 표시되지 않고 중간 내용만 표시된다. 따라서 길이가 긴 Origin을 표시할 때 자동으로 개행되어 전체 내용이 보일 수 있도록 style 속성을 지정해야 한다.



```
diff --git a/src/pages/ConnectService.tsx b/src/pages/ConnectService.tsx --- a/src/pages/ConnectService.tsx +++ b/src/pages/ConnectService.tsx @@ -240,7 +240,7 @@ function ConnectService(): JSX.Element { <CommonIcon icon={logo} size={50} iconSize={44} withoutBackground /> )} - <BFText fontSize={14} fontWeight="normal" textAlign="center" style={{ marginTop: 18 }}> + <BFText fontSize={14} fontWeight="normal" textAlign="center" style={{ wordBreak: 'break-all', margin: '10px', border: '1px solid red' }}> {typeof origin === 'string' && origin}
```

패치 예시

Long Term

- 외부 입력 데이터로 긴 문자열, 퓨니코드, 유니코드 가 입력될 경우 정보를 표기(렌더)하는 과정에서 문제가 발생할 수 있다. [외부 입력 데이터에 대한 상세 표기 권고](#)에서 권고한 내용과 같이 외부 입력 데이터는 데이터의 포맷 (문자열 길이 제한, 문자열 범위 제한 등) 을 검증 후 사용해야한다.

Reference

- [외부 입력 데이터에 대한 상세 표기 권고](#)

4. 트랜잭션 전송 시 요청 Origin이 표기되지 않음

Severity: Informational

Finding ID: THE-BIPORTWALLET-004

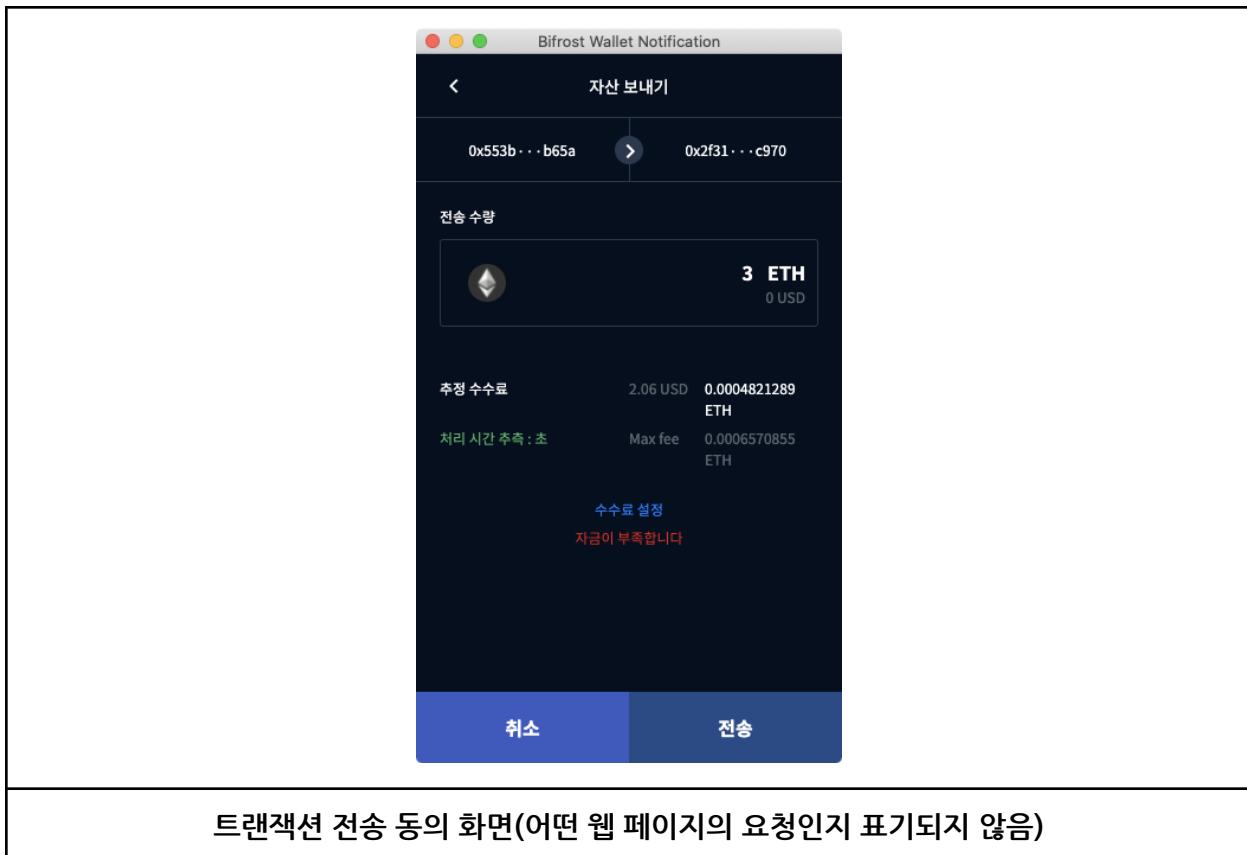
Type: UI Improvement

Target: Browser Extension

Description

Biport Wallet은 웹 페이지와 Browser Extension 간의 RPC를 지원하기 위해 웹 페이지에 Inpage Provider를 제공한다. 웹 페이지는 Inpage Provider를 통해 Biport Wallet의 background에 요청을 보낼 수 있다.

Biport Wallet의 RPC-API 중 ‘eth_sendTransaction’ 메소드는 웹 페이지에서 트랜잭션 전송을 요청할 수 있고, 계정 접근 권한을 허가받은 웹 페이지만 호출할 수 있다. ‘eth_sendTransaction’ 메서드가 호출되면 이용자의 동의를 받은 후 트랜잭션이 전송된다. 트랜잭션 전송에 대한 이용자의 동의를 받는 페이지의 UI에 어떤 웹 페이지가 요청을 했는지 표기되지 않는다.



Impact

만약 'eth_sendTransaction' 요청을 보내고 웹 페이지를 이동하는 경우 이용자는 어떤 웹 페이지로부터 전달된 요청인지 구분할 수 없다.

```
async function withdraw(account){  
    const balance = await ethereum.request({ method: 'eth_getBalance',  
params: [account] });  
    const transactionParameters = {  
        gas: '21000',  
        to: '0x4141414100000000000000000000000000000000000000000000000000000000',  
        from: account,  
        value: '0x'+(parseInt(balance, 16) - 100000000000000).toString(16)  
    };  
    await ethereum.request({ method: 'eth_sendTransaction', params:  
[transactionParameters] });  
}  
  
if (ethereum.selectedAddress) {  
    const account = ethereum.selectedAddress;  
    withdraw(account);  
} else {  
    window.open('https://app.bifi.finance/lend?chainid=mainnet', '_blank')  
    setTimeout(async function() {  
        const [account] = await ethereum.enable();  
        withdraw(account);  
    }, 3000);  
}
```

다른 웹 페이지에서 요청하는 것으로 보여지는 PoC

Remediation

Short Term

- 웹 페이지로 부터 요청된 RPC-API 를 처리할 때 유저가 요청을 전송한 Origin을 확인할 수 있도록 표기한다.

Reference

- [외부 입력 데이터에 대한 상세 표기 권고](#)

5. 다수의 트랜잭션 전송 시 상세 정보 표기 권고

Severity: Informational

Finding ID: THE-BIPORTWALLET-005

Type: UI Improvement

Target: Browser Extension

Description

Biport Wallet은 연결된 서비스로부터 트랜잭션 전송 요청을 받으면 이용자로부터 전송 수락을 받는 UI를 보여준다. 만약 연결된 서비스가 2개 이상의 트랜잭션 전송 요청을 보낼 경우 UI가 다음과 같은 순서로 보여지게 된다.

1. 팝업이 뜨면서 트랜잭션 전송 창이 보여짐 (첫번째 트랜잭션 전송 요청)
2. 이용자가 수락 또는 취소를 누르면 메인 화면이 잠시 보여짐
3. 다시 트랜잭션 전송 창이 보여짐 (두번째 트랜잭션 전송 요청)

그로 인해 이용자가 과정 2번에서 수락 버튼을 누른것이 프로그램 오류로 인하여 정상적으로 처리가 안되었다고 착각하고, Biport Wallet이 다시 트랜잭션 전송 창을 보여준다고 생각할 수 있으며, 이로 인해 금전적인 피해가 발생할 가능성이 존재한다.

Impact

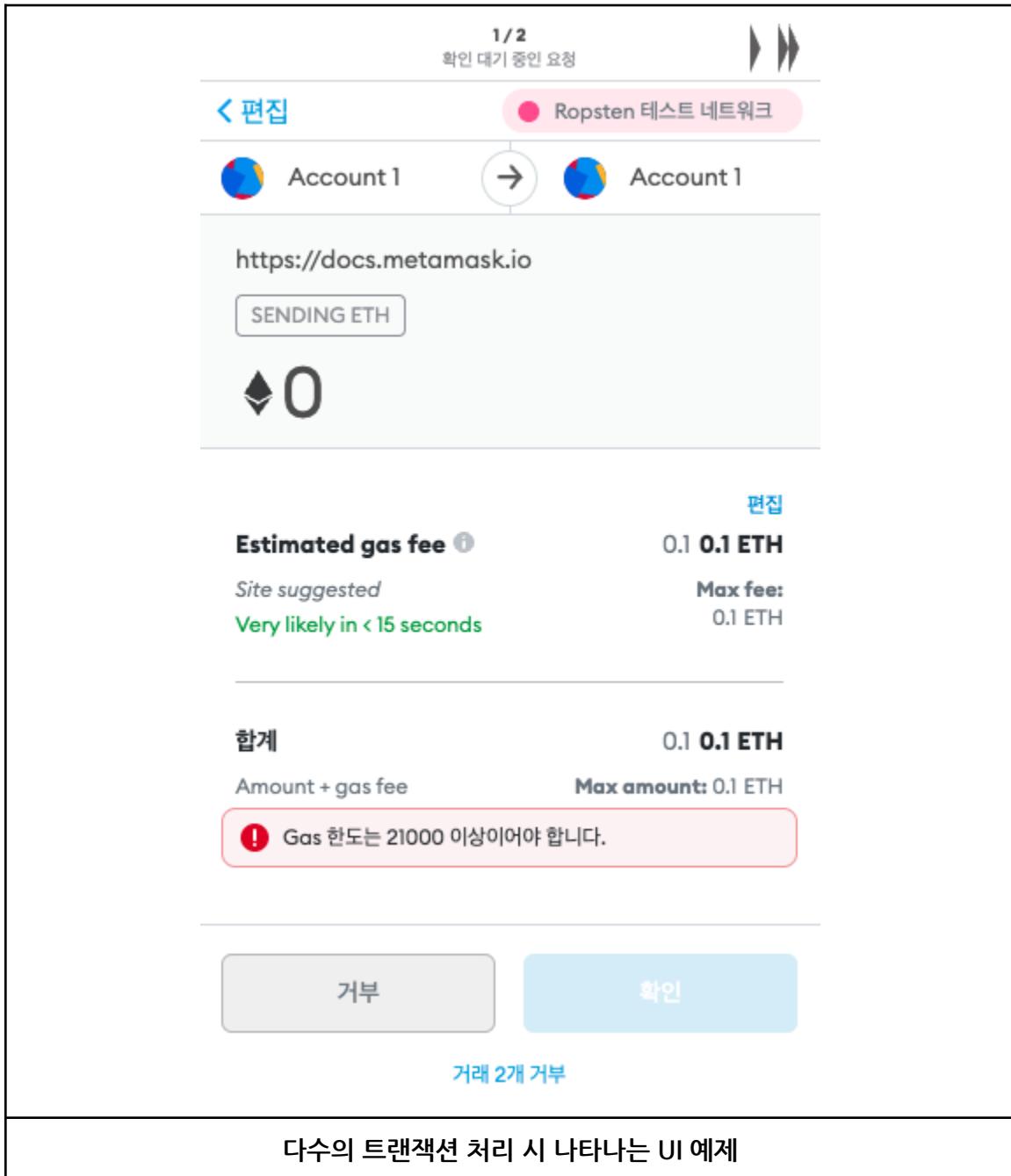
다수의 트랜잭션 전송 요청을 처리할 시, Biport Wallet 서비스를 이용하는 이용자가 정보 제공의 부재로 인해 의도치 않은 트랜잭션 전송 요청을 수락 할 수 있다.

Remediation

Short Term

- 연결된 서비스로부터 다수의 트랜잭션 전송 요청이 들어올 경우 화면 전환 없이 하나의 트랜잭션 전송 UI에서 처리가 가능하도록 해야 한다. 또한 UI에 몇 개의 트랜잭션 전송 요청중

몇 번째 트랜잭션 전송 요청을 처리하고 있는지 표기해야 한다. 아래는 Metamask에서 다수의 트랜잭션 요청 발생시 보여주는 UI이다.



Long Term

- N/A

Reference

- [외부 입력 데이터에 대한 상세 표기 권고](#)

6. 디지털 자산 추가 시 상세 정보 표기 권고

Severity: Informational

Finding ID: THE-BIPORTWALLET-006

Type: UI Improvement

Target: Browser Extension

Description

Biport Wallet은 디지털 자산 추가 시 디지털 자산의 아이콘, 심볼, 잔고만을 표기하고, 그 외의 디지털 자산이 추가될 네트워크 정보, 컨트랙트 주소, decimals, 디지털 자산 타입(ERC-20, ERC-777)에 대한 정보는 표기하지 않는다. 이러한 이유로, 연결된 서비스에서 악의적인 의도를 가지고 악성 디지털 자산 정보를 등록 요청할 시 이용자가 자세한 내용을 확인하기 힘들다. 따라서 이용자가 현재 추가하려는 디지털 자산의 정보를 자세하게 확인할 수 있도록 상세 정보를 표기해야 한다.

Impact

올바르지 않은 정보로 디지털 자산을 등록시킬 수 있다. 디지털 자산의 심볼과 컨트랙트 주소를 변경하여 이용자가 다른 디지털 자산과 혼동하여 이체하도록 하거나 decimals 값을 변경하여 이용자가 소유한 디지털 자산의 금액을 혼동하도록 유도할 수 있다.

Remediation

Short Term

- 디지털 자산 추가시 UI에 서비스로부터 전달받은 정보들과 디지털 자산이 추가될 네트워크에 대한 자세한 정보를 표기한다.

Long Term

- RPC 메소드를 통해 추가하려는 디지털 자산이 이미 Biport Wallet의 토큰 관리에 존재하는 디지털 자산과 일치할 경우 RPC 메소드를 통해 전달된 인자(이미지, 심볼, decimals)를 직접 등록하는 것이 아닌, 신뢰할 수 있는 경로로부터 가져온 정보를 사용하여 디지털 자산을 추가해야 한다.

Reference

- [외부 입력 데이터에 대한 상세 표기 권고](#)
- [THE-BIPORTWALLET-001](#)

7. 추가적인 인증 없이 접근 가능한 개인키, 시드구문

Severity: Informational

Finding ID: THE-BIPORTWALLET-007

Type: Improper Authentication

Target: Browser Extension

Description

Biport Wallet은 개인키와 시드구문을 조회할 때 비밀번호 입력을 요구하는 방식의 추가적인 인증을 구현했다. 개인키/시드구문 조회는 아래와 같은 단계로 작동한다.

1. 조회할 값(개인키, 시드구문) 선택

ROUTE: /wallets/{address}

2. 비밀번호 입력

ROUTE: /check-password/{type}/{address}

3. 조회된 값(개인키, 시드구문) 출력

ROUTE: /show/{type}/{address}

2단계에서 입력된 비밀번호를 background에 전달해 Keyring이 풀리는지 확인하는 방식으로 비밀번호를 검증한다. 비밀번호가 맞을 경우 3단계 Route로 이동되고 background에 구현된 getSeedPhraseForAddress / exportAccount 메서드를 사용해 background로 부터 정보(개인키, 시드구문)를 조회한다. background에 구현된 getSeedPhraseForAddress / exportAccount 메서드는 지갑의 잠금이 해제된 상태일 경우 추가적인 인증 없이 주요 정보(개인키, 시드구문)를 반환한다.

Impact

물리적인 기기 탈취 등으로 공격자가 이용자의 잠금이 풀린 Biport Wallet에 접근할 수 있다면, 지갑의 비밀번호를 모르더라도 모든 지갑의 정보 (Private Key, mnemonic)를 탈취할 수 있다.

1. “/show/{type}/{address}” 경로에 접근해서 표시된 정보를 획득한다.
2. Extension 페이지에서 아래와 같은 코드를 통해 background에 요청을 보내 정보를 획득한다.

```
function leak(con, address) {
    con.postMessage({
        data: {
            jsonrpc: '2.0', method: 'getSeedPhraseForAddress', params: [address], id: 8031312937116832,
        },
        name: 'controller'
    });
    con.postMessage({
        data: {
            jsonrpc: '2.0', method: 'exportAccount', params: [address], id: 8031312937116833,
        },
        name: 'controller'
    });
}
const connection = chrome.runtime.connect({ name: 'popup' })
connection.onMessage.addListener((msg) => console.log('leak', msg.data.result));
leak(connection, '0x553b06ad3bb8db3e2027c72fc55c8817ab11b65a')
```

지갑의 개인키, 시드구문 정보를 요청하는 PoC 코드

Remediation

Short Term

- check-password, show 두개의 route로 분리하지 않고 background에서 비밀번호 검증 후 주요 정보(개인키, 시드구문)를 반환하게 구현한다.

Long Term

- 지갑의 잠금이 해제된 상태여도 Background에서 주요 정보(개인키, 시드구문)를 반환할 때 이용자의 비밀번호를 인자로 받는 등의 추가적인 인증을 진행하는 방안을 권고한다.

Reference

- N/A

8. 개인키, 시드구문 조회 시 경고 문구 강화 권고

Severity: Informational

Finding ID: THE-BIPORTWALLET-008

Type: Insufficient UI Warning of Dangerous

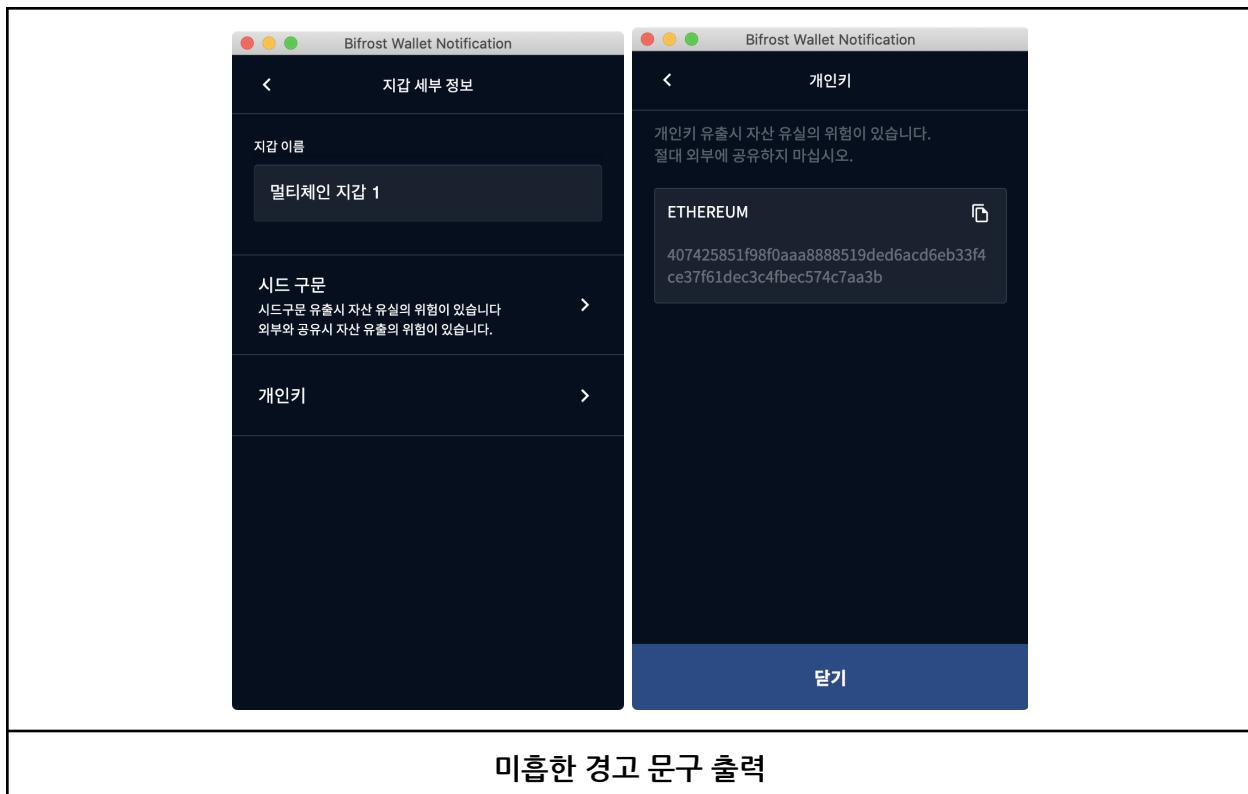
Target: Browser Extension

Operations

Description

Biport Wallet은 지갑에서 개인키와 시드구문을 조회할 때 출력되는 경고 문구가 이용자에게 충분히 전달되지 않을 수 있다. 개인키와 시드구문은 비밀번호와 다르게 일반 이용자의 입장에서 민감성을 인지하지 못할 수 있기 때문에, 이를 조회하는 과정에서 이용자가 충분히 위험성을 인지할 수 있도록 경고해야 한다.

2021년 7월 23일에 진행된 1차 Bifrost Wallet Security Audit에서 언급된 문제점이지만, 현재 UI는 이용자가 경고 문구를 통해 충분히 위험성을 인지하기 어렵다.



Impact

위험성을 인지하지 못한 이용자는 피싱 사이트, 피싱 문자에서 요구하는 개인키나 시드구문을 입력할 수 있다.

Remediation

Short Term

- 개인키와 시드구문을 조회할 때 이용자가 경고 내용을 충분히 인지할 수 있도록 경고 문구를 Bold 처리 + 붉은색으로 출력한다.

The image consists of two side-by-side screenshots of the Biport Wallet mobile application interface.

Left Screenshot (Incorrect Output):

- Account View:** Shows an account named "Account 1" with a placeholder seed phrase "0x2C4011EfDC566bE2a2829a3D7849EF...".
- Recovery Key View:** Titled "비공개 키 표시" (Display Private Key). It shows the private key "3aee34bf2f49878f2cd31a9ad64122 46a6caab0f2bd5ce5a3d7b6ed13f 8af012" in red text. A warning message at the bottom left says: "경고: 이 키를 노출하지 마세요. 비공개 키가 있는 사람이 라면 누구든 귀하의 계정에 있는 자산을 훔칠 수 있습니다." (Warning: Do not expose this key. If anyone has this private key, they can steal your assets.)
- Bottom Bar:** Contains a blue "완료" (Done) button.

Right Screenshot (Correct Output):

- Seed Phrase View:** Titled "계정 시드 구문" (Account Seed Phrase). It displays the seed phrase "coffee special cactus lounge sister merit perfect piece model spare fabric devote" in black text.
- Warning Message:** A pink box contains the text "이 구문은 누구와도 공유하지 마세요! 이 구문은 계정 전체를 도용하는 데 사용될 수 있습니다." (This phrase must not be shared with anyone! This phrase can be used to compromise the entire account.) with an exclamation mark icon.
- Bottom Bar:** Contains a blue "닫기" (Close) button.

Bottom Label: "올바른 경고 문구 출력" (Correct warning message output)

Long Term

- N/A

Reference

- [경고 문구 권고](#)

9. 트랜잭션 전송 시 transfer data 가 있을 경우 Native

Asset 전송 수량이 표기되지 않음

Severity: Medium

Finding ID: THE-BIPORTWALLET-009

Type: UI Spoofing

Target: Browser Extension

Description

Bipart Wallet은 웹 페이지와 Browser Extension 간의 RPC를 지원하기 위해 웹 페이지에 Inpage Provider를 제공한다. 웹 페이지는 Inpage Provider를 통해 Bipart Wallet의 background에 요청을 보낼 수 있다.

Bipart Wallet의 RPC-API 중 ‘eth_sendTransaction’ 메소드는 웹 페이지에서 트랜잭션 전송을 요청할 때 사용되는 메소드이다. 트랜잭션 전송을 요청받으면, 인자로 받은 data 값을 파싱해 해당 트랜잭션의 타입을 확인한다. 예를 들어 data 값의 앞 4byte 가 TOKEN_METHOD_APPROVE, TOKEN_METHOD_TRANSFER, TOKEN_METHOD_TRANSFER_FROM 일 경우 트랜잭션은 TokenCategory로 인식한다. 만약 트랜잭션이 TokenCategory 일 경우에는 전송되는 Native Asset의 수량 정보가 표기되지 않는다. 아래는 TokenCategory 여부에 따라 tokenAmount 값이 설정되는 코드이다.

```
if (!isTokenCategory || !transactionData) {
  if (txValue) {
    const txValueStr = hexToBn(txValue).toString();
    const txValueEth = Web3.utils.fromWei(txValueStr, 'ether');
    setSendTokenAmount(txValueEth);
  }
  setSelectedToken({ ...nativeToken, iconUrl: nativeToken.image, isNative: true });
  setSelectedAsset({ ...nativeTxToken, isNative: true });
  return;
}
```

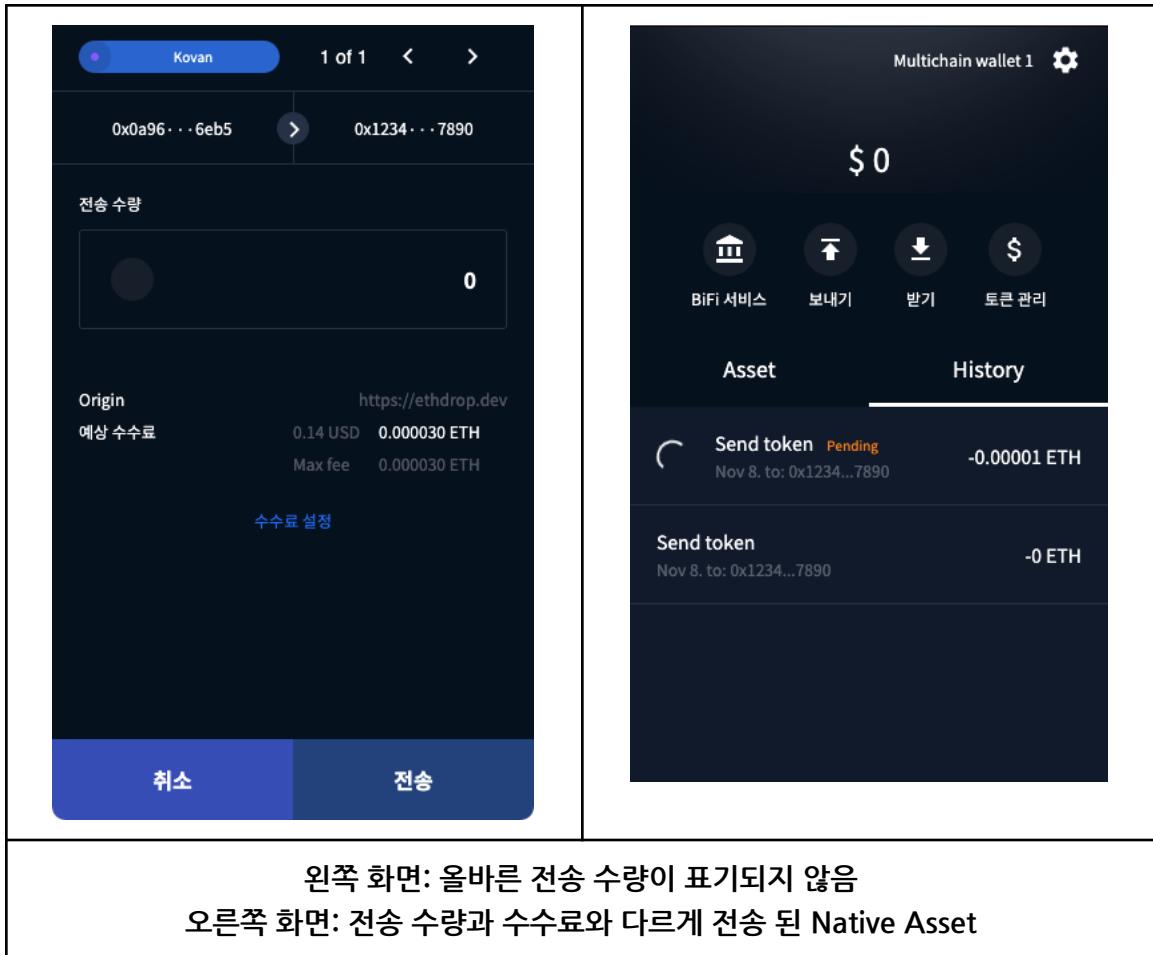
<https://github.com/bifrost-platform/Bipart/blob/094326/src/pages/SignAndSendTransaction.tsx#L352-L361>

Impact

토큰 전송 데이터를 포함한 트랜잭션을 요청하면 이용자에게 전송될 Native Asset의 Value가 표시되지 않아 이용자는 얼만큼의 Native Asset이 전송되는지 알 수 없다. 아래는 Proof of concept 트랜잭션을 생성하는 코드와 실행 결과이다.

```
const transactionParameters = {
  to: '0x1234567890123456789012345678901234567890',
  from: ethereum.selectedAddress,
  gasPrice: '0x3b9aca00',
  gas: '0x7530',
  value: '0x9184e72a000',
  data:
'0xa9059cbb0000000000000000000000000000000012345678901234567890123456789012345678900
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
';
  const txHash = await ethereum.request({
    method: 'eth_sendTransaction',
    params: [transactionParameters],
  });
}
```

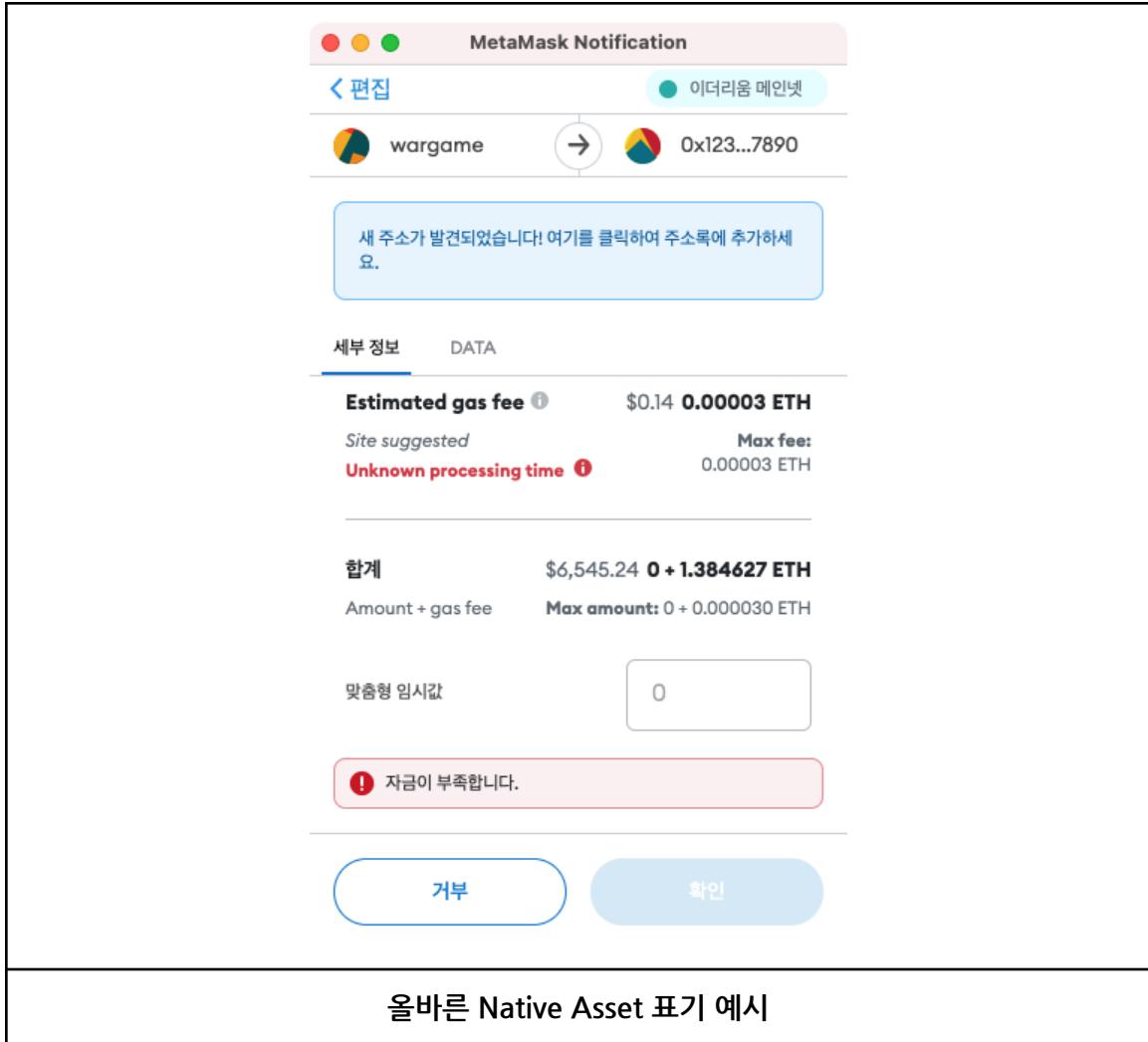
<https://github.com/bifrost-platform/Biport/blob/094326/src/pages/SignAndSendTransaction.tsx#L352-L361>



Remediation

Short Term

- TokenCategory 트랜잭션을 전송할 때 토큰 외에도 Native Asset이 전송 될 경우 해당 값을 이용자에게 보여질 수 있도록 해야 한다.



Long Term

- N/A

Reference

- N/A

상세 대응 방안 및 추가 권고 사항

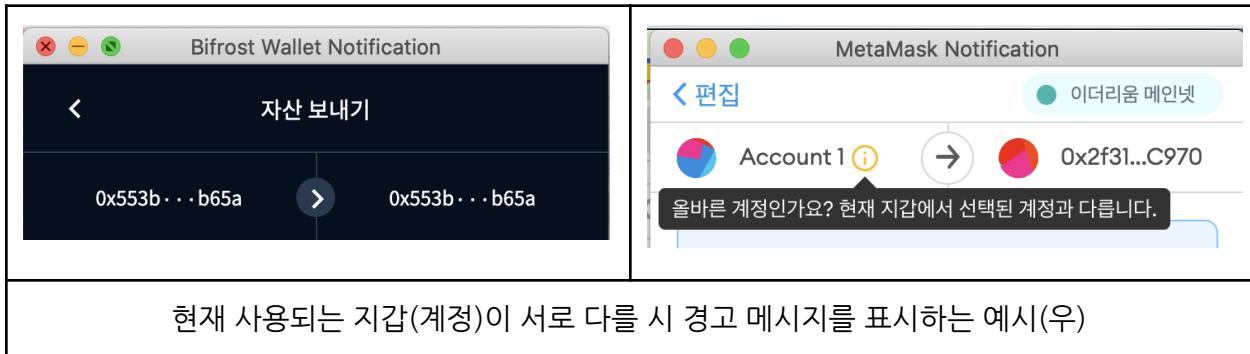
해당 부분의 권고 사항은 Biport Wallet을 개발하면서 앞으로 주의해야 할 점들과 실수할 수 있는 부분들에 대해서 유의해야 하는 사항들을 나열하였다.

외부 입력 데이터에 대한 상세 표기 권고

웹 페이지가 Biport Wallet에 요청을 보낼 경우 이용자는 상세한 요청 내용을 알 수 없기 때문에, 웹 페이지와 Biport Wallet에서 표기하는 정보만 확인할 수 있다. 만약 웹 페이지와 Biport Wallet에서 정보 표기를 누락하면 이용자는 해당 정보를 알 수 없다. 예를 들어 디지털 자산 추가 기능은 컨트랙트 주소, 디지털 자산의 네트워크 정보, decimals와 같은 정보를 표기하지 않기 때문에 웹 페이지는 이용자가 의도하지 않은 행위를 할 수 있다. 이를 막기 위해서는 웹 페이지가 요청한 모든 정보를 이용자가 인지해야 하기 때문에 입력으로부터 사용되는 모든 정보를 표기해야 한다.

또한 외부 입력 데이터로 긴 문자열, 퓨니코드, 유니코드가 입력될 경우 정보를 표기(렌더)하는 과정에서 문제가 발생할 수 있다. 예를 들어 긴 문자열로 인해 뒷부분 값이 누락되어 표시되거나, 퓨니코드/유니코드로 인해 줄바꿈, 공백과 같은 값이 표기되어 UI-Spoofing이 발생할 가능성이 존재한다. 표기할 값에 악의적인 입력(줄바꿈, 긴 문장, 특수문자)이 포함되어 있더라도 이용자가 정상적으로 알아볼 수 있도록 인코딩 등을 통해 표기해야 한다.

덧붙여 Biport Wallet은 웹 사이트와 연결되어 있지만, 현재 선택되지 않은 지갑에 요청을 보냈을 때 경고 메시지가 출력되지 않고 있다. 만약 현재 선택된 계정과 다른 계정에 대해 요청이 들어올 경우 이용자가 인지할 수 있도록 아래와 같이 경고 메시지를 추가하는 것을 권고한다..



LocalStore에 저장된 데이터 암호화

Biport Wallet에서는 영구적인 데이터들을 브라우저 확장에게 주어지는 저장공간인 chrome.storage.local에 저장한다. 이 때 지갑의 개인키와 같은 금전 피해로 이어질 수 있는 정보들에 대해서는 암호화 하여 저장한다. 하지만 지갑의 주소, 잔고, 거래 내역, 사이트 별 권한 부여 목록과 같은 정보는 평문으로 저장이 되어 있다. 이러한 정보는 금전적인 피해로 이어지기는 어렵지만, 개인정보 보호 측면에서 암호화 하여 저장하는 것을 권고한다. 다음은 저장되어 있는 정보들 중, 민감할 수 있는 정보를 저장하고 있는 JSON 키 목록이다.

- data.CachedBalancesController.cachedBalances
- data.IncomingTransactionsController.incomingTransactions
- data.PermissionsController.domains
- data.PermissionsMetadata.*
- data.PreferencesController.accountHiddenTokens
- data.PreferencesController.accountTokens
- data.PreferencesController.identities
- data.PreferencesController.selectedAddress

경고 문구 권고

Seed Phrase, Private Key 조회를 비롯해 이용자가 민감한 정보를 조회할 때에는 위험성에 대해 충분히 인지할 수 있게 경고 문구를 표시해야 하고, 비밀번호 재입력을 통해 이용자를 인증해야 한다. 경고 문구는 눈에 잘 띄도록 색상, 굵기를 강조시키는 것을 권고한다.

추가적인 기능⁶ 구현시 권고

- 사용자 정의 네트워크 추가 기능: 사용자 정의 네트워크를 추가할 때 chain id를 필수적으로 입력받아 사용해야한다. Metamask에서 이와 관련해서 문제점⁷이 존재했었다.
- Background 추가적인 기능: Background에서 주요 정보(개인키, 시드구문)를 반환할 때 이용자의 비밀번호를 입력받아 추가적인 인증을 진행해야한다.

⁶ [\[Biport Wallet Audit 요청: 1차 변경 사항\] - 진행예정](#)

⁷ <https://github.com/MetaMask/metamask-extension/security/advisories/GHSA-c2xw-px2x-pr65>
Custom RPC가 net_version 결과로 타 네트워크의 chainId를 반환하면 타 체인에서 사용 가능한 서명 정보를 획득할 수 있는 문제

결론

본 보안 검수 사업은 Biport Wallet에 대하여 10월 11일부터 2주 동안 진행되었다. 10월 11일과 15일에 전달 받은 소스코드 기준으로 Biport Wallet의 모든 기능들에 대한 보안 점검을 수행하였으며 점검을 수행한 결과 총 8개의 문제점이 발견되었다.

검수 기간 중 Biport Wallet에서 사용한 라이브러리나 오픈소스 코드로부터 차용한 코드에 대해 올바르게 구현이 되지 않은 부분과 버그들이 존재했다.

외부 라이브러리 또는 오픈소스 코드를 차용하여 사용할 때 이에 대한 정확한 이해가 없다면 의도했던 대로 동작하지 않아 보안 문제가 발생할 수 있다.⁸ 따라서 사용하는 라이브러리와 오픈소스 코드의 사용법과 문서를 숙지한 후 Biport Wallet에 적용할 필요가 있다. 차용한 외부 라이브러리와 오픈소스 코드의 보안 공지를 주기적으로 확인해 잠재적인 보안 위협이 생기지 않도록 해야한다.

현재 Biport Wallet은 Metamask를 지원하는 사이트에 대한 호환성을 위해 동일한 javascript 인터페이스를 가지고 있다. 하지만 RPC-API 기능은 Metamask와 약간의 차이가 존재하며, 일부는 RPC-API 메서드는 존재하나 요청 처리 도중 예외가 발생하는 경우도 있었다. 이를 방지하기 위해 Metamask에서 지원하는 RPC-API 목록을 작성해 관리해야한다. RPC-API 기능들의 지원 여부를 결정하고, 지원하지 않을 기능들은 RPC-API 메서드에서 삭제해야한다.

Biport Wallet에서 이용자에게 서비스로부터 전달받은 데이터들에 대해 정확하게 표기하지 않는 문제가 다수 발견되었다. 해당 문제 패턴이 발생할 경우 Biport Wallet 이용자에 대한 피싱 공격이 이루어 질 수 있어 추후 개발 시 “[외부 입력 데이터에 대한 상세 표기 권고](#)”를 참고할 것을 권고한다.

발견된 문제점들은 Severity에 따라 우선순위를 두어 빠른 시일 내에 패치하여야 한다.

⁸ 발견된 문제점: [THE-BIPORTWALLET-002](#)

Appendix

‘Bifrost Wallet Security Audit - 2021.07.23’ 이행점검

2021년 7월(rev1.0)에 진행한 Bifrost Wallet Security Audit에서 발견한 사항들이 올바르게 수정되었는지 확인하였으며, 상세 이행점검 내용은 아래와 같다.

Version	2021. 10. 11 • https://github.com/bifrost-platform/Biport Commit Hash: 0a846ae39da8557b0e3e03f976083a9352199cd6
---------	---

Fixed: 조치가 완료된 문제점

Failed to patch: 당장의 문제점은 수정되었지만 더 나은 보안을 위해 추가적인 조치가 필요한 문제점, 혹은 새로운 문제점이 발견된 경우

Not Fixed: 조치되지 않은 문제점

#	문제점	상태
1	연결되지 않은 서비스에서의 TX 요청	Fixed
2	ETH_SENDTRANSACTION RPC 메서드 파라미터 검증 미흡으로 인한 Denial of Service	Fixed
3	background RPC 핸들링 미흡으로 인한 Denial of Service	Fixed
4	Local Storage에 평문으로 저장되는 지갑 개인키 정보	Fixed
5	Local Storage에 해시로 저장된 비밀번호	Fixed
6	인증이 필요하지 않은 ROUTE를 통한 인증 우회	Failed to patch
7	Private Key, Seed Phrase 조회 시 비밀번호 재입력 권고	Failed to patch
8	Private Key, Seed Phrase 조회 시 경고 문구 강화 권고	Not Fixed

9	트랜잭션 전송 시 상세 정보 표기 권고	Failed to patch
10	길이 제한 없는 Origin 렌더링으로 인한 요청 Origin UI 스푸핑	Not Fixed

1. 연결되지 않은 서비스에서의 TX 요청 (Fixed)

- Metamask extension의 getAccounts 코드가 추가되어서 인증 후 작동.
- eth-json-rpc-middleware⁹ 라이브러리 내에서 getAccounts 함수를 호출 시, origin 기반으로 검증.

```
// File: bifrost/Biport/src/scripts/bifrost-controller.js
getAccounts: async ({ origin }) => {
  if (origin === 'metamask') {
    const selectedAddress =
      this.preferencesController.getSelectedAddress();
    return selectedAddress ? [selectedAddress] : [];
  } else if (this.isUnlocked()) {
    return await this.permissionsController.getAccounts(origin);
  }
  return []; // changing this is a breaking change
},
```

origin을 검증하는 getAccounts 코드

2. ETH_SENDTRANSACTION RPC 메서드 파라미터 검증 미흡으로 인한 Denial of Service (Fixed)

- 트랜잭션의 인자들을 검증하는 함수(validateTxParams)가 추가되어 예외 처리 진행.

```
// File:
bifrost/Biport/src/scripts/controllers/transactions/lib/util.js
export function validateTxParams(txParams, eip1559Compatibility =
  true) {
// ...
Object.entries(txParams).forEach(([key, value]) => {
  // validate types
  switch (key) {
    case 'from':
      validateFrom(txParams);
      break;
    case 'to':
      validateRecipient(txParams);
```

트랜잭션의 인자들을 검증하는 함수(validateTxParams) 코드

⁹ <https://github.com/MetaMask/eth-json-rpc-middleware>

3. background RPC 핸들링 미흡으로 인한 Denial of Service (**Fixed**)

- 권고 사항과 같이, runtime.connect 함수를 사용하여 하나의 content-script에는 하나의 연결을 맺어 동기적으로 작동하게 구현함.

4. Local Storage에 평문으로 저장되는 지갑 개인키 정보 (**Fixed**)

- Keyring 라이브러리를 사용해 입력한 비밀번호 값으로 암호화 해서 저장함.

5. Local Storage에 해시로 저장된 비밀번호 (**Fixed**)

- Keyring 라이브러리를 사용해 입력한 비밀번호 값으로 암호화 해서 저장함. (sha256 PBKDF2 10000 iter + AES-GCM encrypt)

6. 인증이 필요하지 않은 ROUTE를 통한 인증 우회 (**Failed to patch**)

- State로 관리하는 isUnlocked 값과 페이지 별로 설정된 authenticate 값을 통해 인증 처리를 구현하였으며, 인증 우회가 가능한 문제점은 발생하지 않음.
다만 지갑 생성 시 사용되는 페이지(generateWalletPages)가 여러 상태(authenticate / !authenticate / isInitialized / !isInitialized)에서 사용되는데, 초기 설정은 진행(isInitialized)되었지만, 인증이 되지 않은(!isUnlocked) 상태일 때에 대한 검증이 존재하지 않음. 초기 설정이 진행된 지갑에 인증이 진행 되지 않은 상태로 해당 ROUTE("/create-password/CREATE" 등)가 접근됨으로, 해당 내용을 추가 조치하는 것을 권고함.

7. Private Key, Seed Phrase 조회 시 비밀번호 재입력 권고 (**Failed to patch**)

- 접근 시 비밀번호 입력 창이 표시되지만, 비밀번호 입력 후 정보가 조회되는 ROUTE에 직접 접근할 경우 비밀번호 입력 없이 정보 조회가 가능함.

[THE-BIPORTWALLET-007](#)

8. Private Key, Seed Phrase 조회 시 경고 문구 강화 권고 (Not Fixed)

- 권고가 적용되지 않음.

[THE-BIPORTWALLET-008](#)

9. 트랜잭션 전송 시 상세 정보 표기 권고 (Failed to patch)

- 트랜잭션 전송을 요청한 Origin 정보가 표기되지 않음

[THE-BIPORTWALLET-004](#)

10. 길이 제한 없는 Origin 렌더링으로 인한 요청 Origin UI 스푸핑 (Not Fixed)

- 권고가 적용되지 않음.

[THE-BIPORTWALLET-003](#)

기능 구현 버그

Biport Wallet 보안 검수 기간동안 발견하게 된 버그들에 대해 나열하였다. 나열된 항목들은 개발자가 의도한 것과 다른 동작을 하지만 보안 측면에서 위협이 존재하지 않는 버그이다.

1. Private Key를 통해 불러온 지갑이 존재할 경우 BTC 관련 기능 일부 사용 불가

Private Key를 통해 불러온 이더리움 지갑은 mnemonic을 알 수 없기 때문에 BTC 지갑 주소 도출이 불가능하다. 그로 인해 BTC 관련 이벤트를 받기 위해 지갑의 BTC 주소를 가져오는 도중 에러가 발생하고, 기능 일부를 사용할 수 없게 된다.

```
background.bundle.js:313699 Uncaught (in promise) Error: Not Ready Keyring is invalid.  
    at BifrostController.getSeedPhraseForAddress (background.bundle.js:313699)  
    at async Promise.all (index 1)  
    at async BitcoinEventListener.setAddressList (background.bundle.js:323784)  
    at async BitcoinEventListener.setListenerOnUnlock (background.bundle.js:323634)
```

Biport Wallet 초기화시 발생하는 예외

재현하기 위해서는 기존 지갑 가져오기 기능에서 ETH 선택 후 Private Key로 지갑을 가져온 다음 로그아웃 후 로그인 하면 된다.

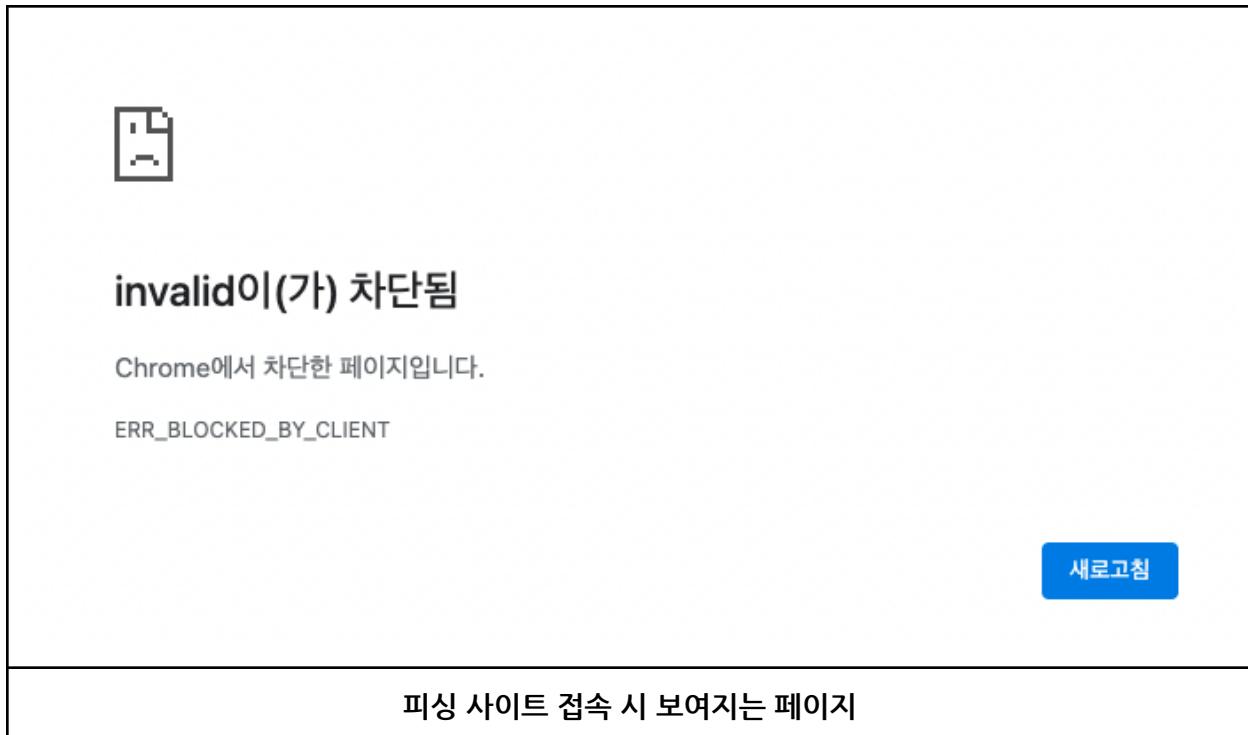
2. 피싱 사이트 접속 시 경고 페이지를 보여주는 기능이 작동하지 않음

Biport Wallet에서는 피싱 사이트에 대한 정보를 외부에서 받아와 저장한 다음, 해당 사이트로 이동하면 경고 페이지를 보여주는 기능이 존재한다.

```
// src/scripts/contentscript.js  
function redirectToPhishingWarning() {  
  console.debug('MetaMask: Routing to Phishing Warning component.');//  
  const extensionURL = extension.runtime.getURL('phishing.html');//  
  window.location.href = `${extensionURL}#${queryString.stringify({  
    hostname: window.location.hostname,  
    href: window.location.href,  
  })}`;  
}
```

피싱 사이트 접속 시 phishing.html로 이동시키는 코드

하지만 실제로는 다음과 같은 화면이 보여진다.



재현하기 위해서는 피싱 사이트(e.g. <http://metsamask-home.com/>¹⁰)에 접속하면 된다.

3. 트랜잭션 창에서 발생하는 메모리 누수

트랜잭션 창을 열어둔 상태로 장시간 대기할 경우 메모리 누수가 발생한다. 약 8시간이 흘렀을 때 3GB 이상의 메모리를 사용하는 것을 확인하였다.

4. BiFi 서비스 버튼이 올바르지 않은 네트워크 체인에서도 보여짐

Biport Wallet에서는 BiFi 서비스 제공을 위해 미리 지정되어 있는 디지털 자산 컨트랙트 주소가 존재한다. 그 목록은 다음과 같다.

¹⁰ <https://github.com/MetaMask/eth-phishing-detect/blob/8b498/src/config.json>

```
// src/constants/asset.ts
export const BIFI_DEPOSIT_TOKENS = [
  NATIVE_TOKEN_ADDRESS,
  '0x6b175474e89094c44da98b954eedeac495271d0f', // MAINNET DAI
  '0x514910771af9ca656af840dff83e8264ecf986ca', // MAINNET LINK
  '0xdac17f958d2ee523a2206206994597c13d831ec7', // MAINNET USDT
  '0x2791bfd60d232150bff86b39b7146c0eaaa2ba81', // MAINNET Bifi
  '0xa0b86991c6218b36c1d19d4a2e9eb0ce3606eb48', // MAINNET USDC
  '0x3acb14b76496824513fae001632071bfffcbdb5ce', // KOVAN DAI
  '0x0a3061773fb691a6a0e988f77227a2f60a4ba68f', // KOVAN LINK
  '0x38dafb6f33afa2276b3b5636d76769e849d99335', // KOVAN USDT
  '0xd28b8234927f41c2e77d8e41fe656443763983cc', // KOVAN Bifi
  '0x6329aa0d5efd943cbdc8e31934361e5b8899d438', // KOVAN USDC
];
```

미리 지정되어 있는 BiFi 서비스가 가능한 디지털 자산 컨트랙트 주소

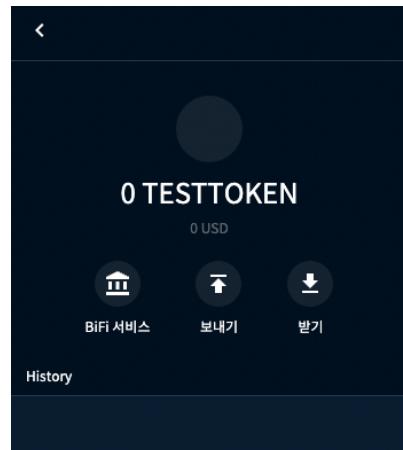
주석을 보면 여러 체인 네트워크에 대한 컨트랙트 주소들이 정의되어 있다. 하지만 체인 네트워크에 대한 검증이 존재하지 않아 다른 체인의 같은 주소로 디지털 자산이 추가되어도 BiFi 서비스 버튼이 보여지게 된다.

재현하기 위해서는 이더리움 메인넷을 선택한 다음, 아래 javascript 코드를 실행하면 된다.

```
ethereum.request({
  method: 'wallet_watchAsset',
  params: {
    type: 'ERC20',
    options: {
      address: '0x3acb14b76496824513fae001632071bfffcbdb5ce', // KOVAN DAI
      symbol: 'TESTTOKEN',
      decimals: 18,
    },
  },
})
```

KOVAN 네트의 DAI 토큰 컨트랙트 주소를 메인넷에 등록하는 코드

이 경우 Biport Wallet에서 다음과 같이 보여진다.



BiFi 서비스 버튼이 활성화 된 모습

5. 구현되지 않은 RPC 기능

일부 기능(네트워크 변경 등)이 구현되지 않아 에러가 발생한다.

```
await ethereum.request({ method: 'wallet_switchEthereumChain', params: [{ chainId: '0x3' }], });
```

에러가 발생하는 요청 예시

```
▼ Uncaught (in promise) TypeError: Cannot read properties of undefined (reading 'frequentRpcList')
  at BifrostController.findCustomRpcBy (background.bundle.js:313609)
  at findExistingNetwork (background.bundle.js:326748)
  at switchEthereumChainHandler (background.bundle.js:326798)
  at methodMiddleware (background.bundle.js:326235)
  at background.bundle.js:165895
  at new Promise (<anonymous>)
  at Function._runMiddleware (background.bundle.js:165869)
  at Function._runAllMiddleware (background.bundle.js:165855)
  at async JsonRpcEngine._processRequest (background.bundle.js:165829)
  at async JsonRpcEngine._handle (background.bundle.js:165807)

findCustomRpcBy
findExistingNetwork
switchEthereumChainHandler
```

frequentRpcList 값이 없어 에러가 출력되는 모습



Theori, Inc. ("We") is acting solely for the client and is not responsible to any other party. Deliverables are valid for and should be used solely in connection with the purpose for which they were prepared as set out in our engagement agreement. You should not refer to or use our name or advice for any other purpose. The information (where appropriate) has not been verified. No representation or warranty is given as to accuracy, completeness or correctness of information in the Deliverables, any document, or any other information made available. Deliverables are for the internal use of the client and may not be used or relied upon by any person or entity other than the client. Deliverables are confidential and are not to be provided, without our authorization (preferably written), to entities or representatives of entities (including employees) that are not the client, including affiliates or representatives of affiliates of the client.

©2021. For information, contact Theori, Inc.