

Making Web3 Space Safer for Everyone



BlockWallet Extension Wallet

Security Assessment

Published on : 27 Sep. 2023
Version v1.1



Security Report Published by KALOS

v1.1 27 Sep. 2023

Auditor : Andy Koo

Found issues

Severity of Issues	Findings	Resolved	Acknowledged	Comment
Critical	-	-	-	-
High	-	-	-	-
Medium	1	1	-	-
Low	4	4	-	-
Tips	1	1	-	-

TABLE OF CONTENTS

TABLE OF CONTENTS

ABOUT US

Executive Summary

OVERVIEW

FINDINGS

Enhance Port Management to Prevent Potential Denial of Service (DoS) Attacks

Strengthen Chain Addition Validation to Prevent Phishing Attacks

Strengthen URI Validation to Prevent Malicious Bypass

Implement Origin Checks to Mitigate Potential Phishing or XSS Attacks

If the name of a service is very long, it can be misleading to deceive users.

The data.id input need to be used as restricted format

DISCLAIMER

Appendix. A

Severity Level

Difficulty Level

ABOUT US

Making Web3 Space Safer for Everyone

KALOS is a flagship service of HAECHI LABS, the leader of the global blockchain industry. We bring together the best Web2 and Web3 experts. Security Researchers with expertise in cryptography, leaders of the global best hacker team, and blockchain/smart contract experts are responsible for securing your Web3 service.

Having secured \$60B crypto assets on over 400 main-nets, Defi protocols, NFT services, P2E, and Bridges, KALOS is the only blockchain technology company selected for the Samsung Electronics Startup Incubation Program in recognition of our expertise. We have also received technology grants from the Ethereum Foundation and Ethereum Community Fund.

Inquiries: audit@kalos.xyz

Website: <https://kalos.xyz>

Executive Summary

Purpose of this report

This report was prepared to audit the security of the Extension Wallet developed by the BlockWallet team. KALOS conducted the audit focusing on whether the system created by the BlockWallet team is soundly implemented and designed as specified in the published materials, in addition to the safety and security of the Extension Wallet.

In detail, we have focused on the following

- Connection on untrusted page without any interaction.
- Security of encrypted seed phrase and private key.
- Signature generation on the evil origin page.
- XSS/CSRF/Prototype Pollution vulnerability from user and page input.
- Proper implementation of EIP specs.

Codebase Submitted for the Audit

The codes used in this Audit can be found on GitHub
“<https://github.com/block-wallet/extension>”.

The last commit of the code used for this Audit is
“6a003cb3ba42e3f24d17227c0b7083a3c6344bec”.

Audit Timeline

Date	Event
2023/08/10	Audit Initiation
2023/08/31	Delivery of v1.0 report.
2023/09/27	Delivery of v1.1 report.

Findings

KALOS found 1 medium, 4 Low severity issues. There are 1 Tips issues explained that would improve the code's usability or efficiency upon modification. KALOS team has confirmed that all the identified issues have been properly fixed.

Severity	Issue	Status
Low	Enhance Port Management to Prevent Potential Denial of Service (DoS) Attacks	(Resolved - v1.1)
Low	Strengthen Chain Addition Validation to Prevent Phishing Attacks	(Resolved - v1.1)
Low	Strengthen URI Validation to Prevent Malicious Bypass	(Resolved - v1.1)
Medium	Implement Origin Checks to Mitigate Potential Phishing or XSS Attacks	(Resolved - v1.1)
Low	If the name of a service is very long, it can be misleading to deceive users.	(Resolved - v1.1)
TIPS	The data.id input need to be used as restricted format	(Resolved - v1.1)

OVERVIEW

Functional overview

BlockWallet provides tools for web3 and cryptocurrency transactions. It integrates with the bridge aggregator for token swaps, aiming to present users with optimal exchange rates. A gas tracking feature on the home screen offers real-time gas price metrics. The extension facilitates cryptocurrency purchases using various methods which don't store user KYC data.

For security, BlockWallet supports integration with hardware wallets such as Ledger and Trezor. Additional features like Phishing Protection artwork during data input and token allowance management tools are included to address potential risks. On the infrastructure side, BlockWallet works with private nodes for Ethereum and L2 networks. Its Chainlist integration aids users in finding and managing EVM-compatible chains.

Web3 Wallet Checklist

Category	Description
Access Control	<ul style="list-style-type: none">• Authentication of the user is required when accessing the seed phrase/private key.<ul style="list-style-type: none">▫ Even if the device is stolen, the wallet should be secure if the wallet password is unknown.▫ Re-authentication is necessary when exporting seed/private key.• Only requests from connected accounts should be allowed on connected sites.<ul style="list-style-type: none">▫ Even when multiple accounts are created, requests cannot be sent to unconnected accounts.• Websites should not be able to request read/write access to the wallet without the user's consent.• Transaction transfers and signature requests should only process requests that match the account's address.• Verify if it is possible to bypass the existing authentication logic and execute functions.
Auditor's Comment	<ul style="list-style-type: none">• The password is required when the user first opens the wallet.

- Users can set the lock timeout so that password protection is maintained.
- The user's approval is required for the transaction and sign request. The audit did not find any method of bypassing user action.
- Proper user authentication is required for exporting the seed phrase and private key.
- The permission structure maintains the relationship between each account and host.

Category	Description
Phishing Protection	<ul style="list-style-type: none"> ▪ Check the origin of Post Message ▪ Display signature and transaction information in a way that users can understand ▪ Show all information used for transaction transmission and signature to the user ▪ Prevent unauthorized changes to RPC Endpoint and Network ▪ Verify that the Transaction information displayed to the user may differ from the actual Transaction information ▪ Prevent phishing through HTML TAG/CSS Injection
Auditor's Comment	<ul style="list-style-type: none"> ▪ The transaction data is properly displayed to the user, including the address, value, method, and parameters. However, it is recommended to display the full host URL instead of the truncated one. (BLK-EXTWLT-05) ▪ The wallet_addEthereumChain RPC call can apply the chain with duplicated name(BLK-EXTWLT-02)

Category	Description
Key Management & Cryptography	<ul style="list-style-type: none"> ▪ Safety of the random number generator used for seed generation ▪ Encryption method for seed phrase/private key ▪ Storage location for seed phrase/private key ▪ Use of secure password rules ▪ Implementation of a timeout feature that deletes seed phrase/private key information after a certain period of time and requires re-authentication before wallet can be used again
Auditor's Comment	<ul style="list-style-type: none"> ▪ The proper validation of the password rule and the wallet's other properties was tested by the auditor. ▪ The hashed key is used to decrypt the vault when the wallet is unlocked. PBKDF is utilized for the decryption and encryption of the vault. ▪ The encrypted vault is stored in Chrome storage, and the hashed password remains in memory while the wallet is unlocked.
Category	Description
Web	<ul style="list-style-type: none"> ▪ XSS / CSRF ▪ ClickJacking ▪ Presence of Content Security Policy ▪ Presence of Cross Origin Restriction ▪ Security and communication encryption of servers used in wallet operation process
Auditor's Comment	<ul style="list-style-type: none"> ▪ No injection vulnerability in the HTML/CSS was found during this audit. However, the URI validation logic needs to be strengthened (BLK-EXTWLT-03). ▪ The web-accessible resources are restricted. Direct access to the extension's resources from the web page is not viable because of the CORS.

- The communication between the RPC endpoint and Node API server is protected by SSL/TLS.

Category	Description
Miscellaneous	<ul style="list-style-type: none"> • Use of safe 3rd party libraries • Whether important information is exposed in Debug Logs • Indication of security warnings • Whether important information is exposed when using a Monitoring Solution • Private keys/seeds should not be exposed in plain text in the browser's storage files during the wallet's operation process • Measures to prevent DoS attacks
Auditor's Comment	<ul style="list-style-type: none"> • The 3rd party issue of the wallet has not been found in the audit. • Information leakage to the 3rd party or log was not found in the audit.

FINDINGS

Enhance Port Management to Prevent Potential Denial of Service (DoS) Attacks

ID: BLK-EXTWLT-01

Severity: Low

Type: Miscellaneous

Difficulty: High

File: packages/background/src/controllers/BlankController.ts

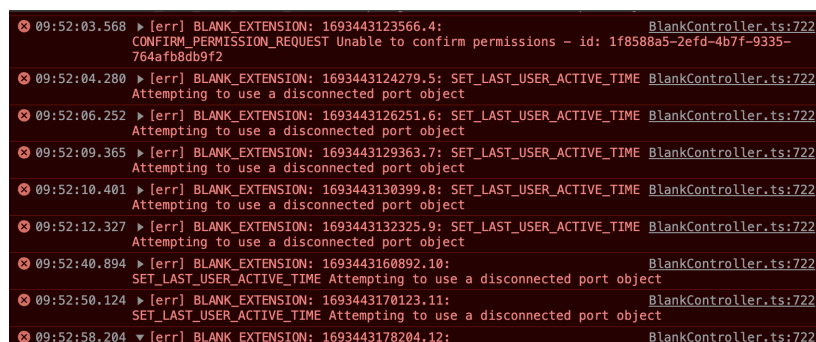
Issue

The extension and the content script communicate through *chrome.runtime.Port*. If the port becomes disconnected, the event subscription using the port should also be unsubscribed. It is possible for an abnormal web page to open the extension's popup, which uses the port and subscription, and disconnect the port to execute a DoS attack.

```
public handler<TMessageType extends MessageTypes>(  
  { id, message, request }: TransportRequestMessage<TMessageType>,  
  port: chrome.runtime.Port,  
  portId: string  
) : void {  
  ...  
  port.onDisconnect.addListener(() => {  
    const error = chrome.runtime.lastError;  
    isPortConnected = false;  
    if (error) {  
      log.error(error);  
    }  
  });  
}
```

[<https://github.com/block-wallet/extension/blob/b66583bd3f3b3c4333ea338feac1b8a103d95535/packages/background/src/controllers/BlankController.ts#L696>]

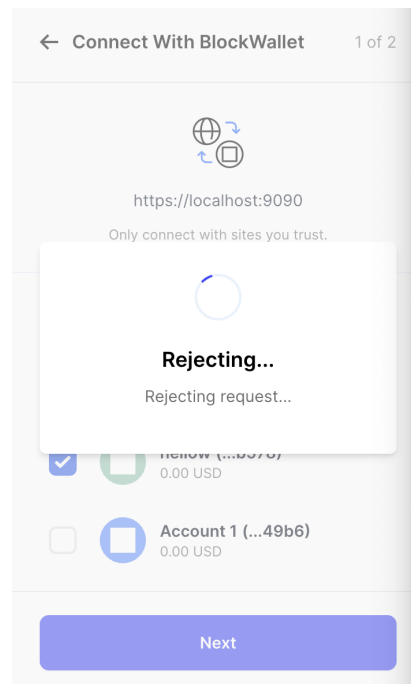
The event subscription that maintains the port is attempting to *postMessage* to a disconnected port repeatedly.



```
09:52:03.568 ▶ [err] BLANK_EXTENSION: 1693443123566.4: BlankController.ts:722  
CONFIRM_PERMISSION_REQUEST Unable to confirm permissions - id: 1f8588a5-2efd-4b7f-9335-  
764afb8db9f2  
09:52:04.280 ▶ [err] BLANK_EXTENSION: 1693443124279.5: SET_LAST_USER_ACTIVE_TIME BlankController.ts:722  
Attempting to use a disconnected port object  
09:52:06.252 ▶ [err] BLANK_EXTENSION: 1693443126251.6: SET_LAST_USER_ACTIVE_TIME BlankController.ts:722  
Attempting to use a disconnected port object  
09:52:09.365 ▶ [err] BLANK_EXTENSION: 1693443129363.7: SET_LAST_USER_ACTIVE_TIME BlankController.ts:722  
Attempting to use a disconnected port object  
09:52:10.401 ▶ [err] BLANK_EXTENSION: 1693443130399.8: SET_LAST_USER_ACTIVE_TIME BlankController.ts:722  
Attempting to use a disconnected port object  
09:52:12.327 ▶ [err] BLANK_EXTENSION: 1693443132325.9: SET_LAST_USER_ACTIVE_TIME BlankController.ts:722  
Attempting to use a disconnected port object  
09:52:40.894 ▶ [err] BLANK_EXTENSION: 1693443160892.10: SET_LAST_USER_ACTIVE_TIME BlankController.ts:722  
Attempting to use a disconnected port object  
09:52:50.124 ▶ [err] BLANK_EXTENSION: 1693443170123.11: SET_LAST_USER_ACTIVE_TIME BlankController.ts:722  
Attempting to use a disconnected port object  
09:52:58.204 ▼ [err] BLANK_EXTENSION: 1693443178204.12: BlankController.ts:722
```

[Disconnected port error log]

The user is unable to confirm or reject the popup, and the BlockWallet's other functionalities become unavailable through the UI.



[User's popup]

Recommendation

Upon port disconnection, ensure that all event listeners and subscriptions associated with that port are removed.

Fix Comment

[a5e1dd6] The patch resolves the issue of using disconnected ports by applying code that catches the exception, clears pending requests, and handles disconnected ports.

```
} catch (err) {  
  const safeError = toError(err);  
  Log.error('[err]', safeError.message);  
  if (  
    safeError.message  
      .toLowerCase()  
      .includes(  
        'attempting to use a disconnected port object'  
      )  
  ) {  
    port.disconnect();  
    this.unsubscribe(id);  
  }  
}
```

[Patch code snippet]

Strengthen Chain Addition Validation to Prevent Phishing Attacks

ID: BLK-EXTWLT-02

Severity: Low

Type: Phishing Protection

Difficulty: Medium

File: packages/background/src/controllers/NetworkController.ts

Issue

The `wallet_addEthereumChain` JSON RPC allows users or web pages to customize their network provider. When the popup UI displays the name of the user's chain, the user selects the chain by its name. Therefore, the chain name should not be duplicated, as this could make the user vulnerable to a phishing attack. However, the `wallet_addEthereumChain` RPC request only checks the chain ID, not the name, which could lead to potential security issues.

```
public async addNetwork(network: AddNetworkType): Promise<void> {
  if (!network.chainId || Number.isNaN(network.chainId)) {
    throw new Error('ChainId is required and must be numeric.');
```

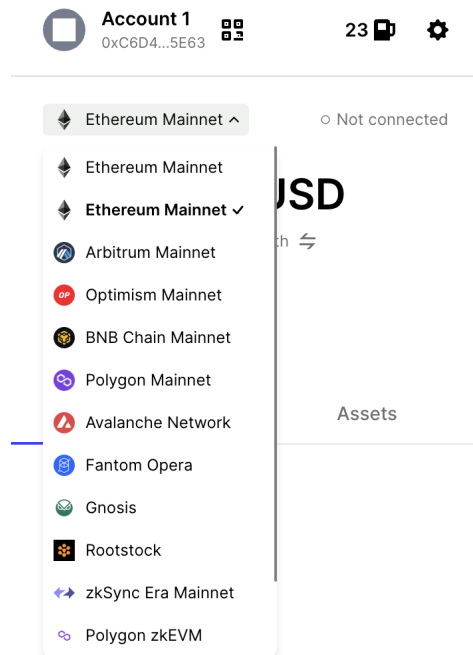
[<https://github.com/block-wallet/extension/blob/b66583bd3f3b3c4333ea338feac1b8a103d95535/packages/background/src/controllers/NetworkController.ts>]

The malicious page can request adding a custom chain with the same name as the default chain.

```
await ethereum.request({
  method: 'wallet_addEthereumChain',
  params: [{
    chainId: '0xAB0D',
    chainName: 'Ethereum Mainnet',
    rpcUrls: ['https://localhost:8080'],
    iconUrls:
[ 'https://raw.githubusercontent.com/block-wallet/assets/master/blockchains/ethereum/info/Logo.png'],
  ] }]);
```

[Request to add custom chain]

The user cannot distinguish the default chain and custom chain.



[User's popup]

Recommendation

In addition to validating the chainId, the method should also check the uniqueness of the chainName. If a chain with the same name already exists, the addition should be denied.

Fix Comment

[5bc8d32] The network name is checked to prevent users from adding potentially malicious networks to the wallet.

```
const networkNameExists = Object.values(availableNetworks).some(  
  (network) => {  
    return network.desc.toLowerCase() === chainName.toLowerCase()  
  }  
)
```

[Patch code snippet]

Add Network


https.andy

Allow this site to add a network?
This will allow this network to be used within BlockWallet

Network Name ⓘ
Ethereum Mainnet

Chain ID ⓘ
43789

Network URL
https://localhost:9545

Network Icon URL ⚠
https://raw.githubusercontent.com/block-wall...
 ☒ Save image URL

[View all details](#)

ⓘ BlockWallet does not verify custom networks. Make sure you understand [the potential risks adding a custom network may pose](#).

Reject

Add Network

[Network Name duplication handling]

Strengthen URI Validation to Prevent Malicious Bypass

ID: BLK-EXTWLT-03

Severity: Low

Type: Miscellaneous

Difficulty: Medium

File: packages/background/src/controllers/NetworkController.ts

Issue

The URI validation logic checks for the index of the string "https://" and fails validation if it is not included, indicated by an index of -1. This requirement can be bypassed by including the "https://" string within any other string, which may result in unexpected behavior. For instance, the explorerUrl, which is the URL of the custom chain, can be an arbitrary payload that could potentially be used in malicious behavior.

```
const rpcUrl = formatAndValidateRpcURL(
  network.rpcUrls?.[0] || chainDataFromList!.rpc[0]
);

// Check block explorer url
const explorerUrl =
  getUrlWithoutTrailingSlash(network.blockExplorerUrls) ||
  getUrlWithoutTrailingSlash(
    chainDataFromList?.explorers?.map(
      ({ url }: { url: string }) => url
    )
  ) ||
  '';

if (explorerUrl && explorerUrl.indexOf('https://') === -1) {
  throw new Error('Block explorer endpoint must be https');
}
```

[https://github.com/block-wallet/extension/blob/b66583bd3f3b3c4333ea338feac1b8a103d95535/packages/background/src/controllers/NetworkController.ts#L452]

Although the payload cannot be executed due to the Content-Security-Policy on the manifest.json, the validation logic can be easily bypassed.

```
const data = {
  id: '1692188734268.105',
  message: 'EXTERNAL_REQUEST',
  origin: 'BLANK_PROVIDER',
  request: {
    method: 'wallet_addEthereumChain',
    params: [
      {
        chainId: '0xAB0D',
        chainName: 'Ethereum Mainnet',
        rpcUrls: ['https://localhost:8080'],
        iconUrls:
          ['https://raw.githubusercontent.com/block-wallet/assets/master/blockchains/ethereum/info/Logo.png'],
        blockExplorerUrls: ['javascript:alert()//https://asdadsadasdasd']
      }
    ]
  }
};
```



```
    },  
  },  
  
  // Send the message  
  window.postMessage(data);  
}
```

[Example request that contains malicious explorerURL]

The state storage shows the blockExplorerUrls value set to be payload.

```
    "CHAIN-43789": {  
      "actionsTimeIntervals": {  
        "balanceFetch": 80000,  
        "blockNumberPull": 45000,  
        "exchangeRatesFetch": 60000,  
        "gasPricesUpdate": 30000,  
        "providerSubscriptionsUpdate": 15000,  
        "transactionWatcherUpdate": 90000,  
        "transactionsStatusesUpdate": 15000  
      },  
      "blockExplorerName": "Explorer",  
      "blockExplorerUrls": [  
        "javascript:alert();//https://asdadsadasdasdasd"  
      ],  
      "chainId": 43789,  
      "currentRpcUrl": "https://localhost:8080",  
      "desc": "Ethereum Mainnet",  
      "enable": true,  
      "ens": false,  
      "features": [  
        "sends"  
      ],  
      "hasFixedGasCost": false,  
      "name": "chain-43789",  
      "nativeCurrency": {  
        "decimals": 18,  
        "Logo":  
        "https://raw.githubusercontent.com/block-wallet/assets/master/blockchains/ethereum/info/Logo.png",  
        "name": "ETH",  
        "symbol": "ETH"  
      },  
      "nativelySupported": false,  
      "networkVersion": "43789",  
      "order": 12,  
      "showGasLevels": true,  
      "test": false  
    },  
  },  
}
```

[Example request that contains malicious explorerURL]

Recommendation

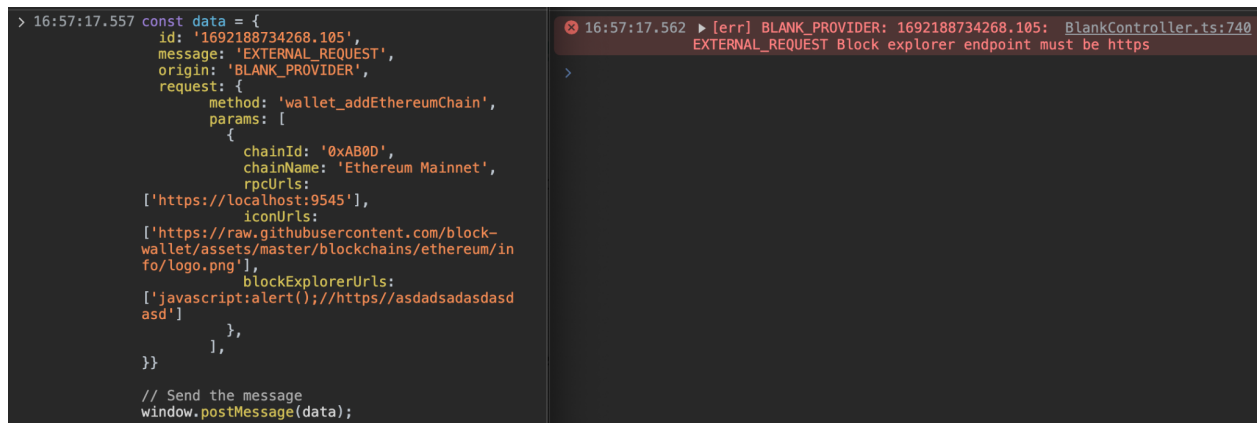
Use a more robust method to validate the URI. Ensure that the URI starts with "https://".

Fix Comment

[[5bc8d32](#)] The RPC, Explorer, Icon URIs are checked to prevent users from adding potentially malicious URIs to the wallet.

```
export function isHttpsURL(url: string): boolean {  
  return url.startsWith('https://');  
}
```

[Patch code snippet]



The screenshot shows a code editor with a JavaScript object and a console error. The object is a message from a 'BLANK_PROVIDER' to a 'BlankController'. It contains a request to 'wallet_addEthereumChain' with parameters including 'chainId', 'chainName', 'rpcUrls', 'iconUrls', and 'blockExplorerUrls'. The 'rpcUrls' array contains a local URL and a GitHub URL. The 'iconUrls' array contains a GitHub URL. The 'blockExplorerUrls' array contains a JavaScript alert call and a URL. The console error shows a message from the 'BLANK_PROVIDER' to the 'BlankController' with the message 'EXTERNAL_REQUEST Block explorer endpoint must be https'.

```
> 16:57:17.557 const data = {  
  id: '1692188734268.105',  
  message: 'EXTERNAL_REQUEST',  
  origin: 'BLANK_PROVIDER',  
  request: {  
    method: 'wallet_addEthereumChain',  
    params: [  
      {  
        chainId: '0xAB0D',  
        chainName: 'Ethereum Mainnet',  
        rpcUrls:  
          ['https://localhost:9545'],  
        iconUrls:  
          ['https://raw.githubusercontent.com/block-wallet/assets/master/blockchains/ethereum/info/logo.png'],  
        blockExplorerUrls:  
          ['javascript:alert();//https://asdadsadasdasd  
asd']  
      },  
    ],  
  },  
}  
  
// Send the message  
window.postMessage(data);
```

```
✖ 16:57:17.562 [err] BLANK_PROVIDER: 1692188734268.105: BlankController.ts:740  
EXTERNAL_REQUEST Block explorer endpoint must be https  
>
```

[Malicious URI handling]

Implement Origin Checks to Mitigate Potential Phishing or XSS Attacks

ID: BLK-EXTWLT-04

Severity: Medium

Type: Access Control

Difficulty: Medium

File: packages/provider/src/content.ts

Issue

An opaque origin cannot be serialized to a tuple of (scheme, host, port) like typical origins. The opaque origin (null) of a web page can indicate a phishing or XSS attack.

The following are typical use cases:

- The response from a data: URL
- A document created by about:blank
- A sandboxed iframe without the allow-same-origin flag

Because these pages barely need to interact with web3 wallets, we recommend checking the window's origin on the content script side.

```
const windowListener = async ({
  data,
  source,
}: MessageEvent<WindowTransportRequestMessage>): Promise<void> => {
  // Only allow messages from our window, by the inject
  if (
    source !== window ||
    data.origin !== Origin.PROVIDER ||
    !Object.values(EXTERNAL).includes(data.message)
  ) {
    return;
  }
}
```

[<https://github.com/block-wallet/extension/blob/b66583bd3f3b3c4333ea338feac1b8a103d95535/packages/provider/src/content.ts#L108>]

Recommendation

Before processing any messages in the content script, check the origin of the sender. If the origin is an opaque origin ("null")

Fix Comment

[\[13b8c2e\]](#) The content script checks the opaque origin to prevent potential malicious hosts from making calls to the extension.

```
// Only allow messages from our window, by the inject
if (
  source !== window ||
  source.origin === null ||
  source.origin === 'null' ||
  data.origin !== Origin.PROVIDER ||
  !Object.values(EXTERNAL).includes(data.message) ||
  // data.id should match the format indicated on BlankProvider.js because it could be set
  // maliciously by a web page
  // Regex validates the following format
  // `${Date.now()}.${++this._requestId}` --> 1694708163916.8
  !/^(\\d+)\\.\\d+$/.test(data.id)
) {
  return;
}
```

[Patch code snippet]

If the name of a service is very long, it can be misleading to deceive users.

ID: BLK-EXTWLT-05

Severity: Low

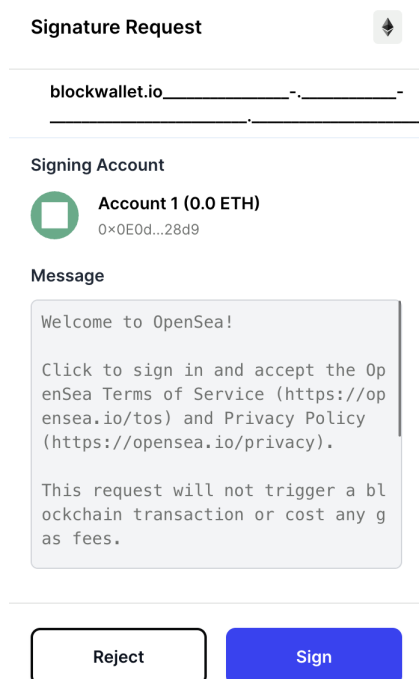
Type: Phishing Protection

Difficulty: Low

File: packages/ui/src/components/dApp/DAppOrigin.tsx

Issue

Service names (hosts) can be made longer to impersonate trusted service names that users rely on. If the service name requesting the signature is longer than the length that can be displayed in the window, the trailing string will be omitted. Users may be misled into accepting signature requests from malicious hosts.



[[The user is unable to view at once the full domain of the requesting service.]]

Recommendation

For enhanced clarity and user awareness, it's crucial to ensure the complete visibility of lengthy origins. We recommend specifying the style attribute so that it automatically wraps and shows the entire content.

The data.id input need to be used as restricted format

ID: BLK-EXTWLT-06

Severity: Tips

Type: Miscellaneous

Difficulty: Informational

File: packages/provider/src/content.ts

Issue

The input for `postMessage`, `data.id`, can be set by the web page and is used as the property selector for another variable. If an unexpected value, such as `__proto__`, is set for `data.id`, there is a possibility that the wallet's validation logic could be bypassed. While no direct abuse has been identified yet, it is recommended to validate the format of the data on the content script as the value is expected to follow the format `${Date.now()}.${++this._requestId}`.

```
const windowListener = async ({
  data,
  source,
}: MessageEvent<WindowTransportRequestMessage>): Promise<void> => {
  // Only allow messages from our window, by the inject
  if (
    source !== window ||
    data.origin !== Origin.PROVIDER ||
    !Object.values(EXTERNAL).includes(data.message)
  ) {
    return;
  }
}
```

[<https://github.com/block-wallet/extension/blob/b66583bd3f3b3c4333ea338feac1b8a103d95535/packages/provider/src/content.ts#L108>]

Recommendation

Implement rigorous validation checks for `data.id` to ensure it adheres to the expected format `${Date.now()}.${++this._requestId}`. Refrain from using external inputs, like `data.id`, directly for property selection.

Fix Comment

[\[5f5c552\]](#) The value of data.id is checked to ensure it conforms to the valid format.

```
const windowListener = async ({
  data,
  source,
}: MessageEvent<WindowTransportRequestMessage>): Promise<void> => {
  // Only allow messages from our window, by the inject
  if (
    source !== window ||
    source.origin === null ||
    data.origin !== Origin.PROVIDER ||
    !Object.values(EXTERNAL).includes(data.message) ||
    // data.id should match the format indicated on BlankProvider.js because it could be set
    // maliciously by a web page
    // Regex validates the following format
    // `${Date.now()}.${++this._requestId}` --> 1694708163916.8
    !/^(\\d+)\\.\\d+$/i.test(data.id)
  ) {
    return;
  }
}
```

[Patch code snippet]

DISCLAIMER

This report does not guarantee investment advice, the suitability of the business models, and codes that are secure without bugs. This report shall only be used to discuss known technical issues. Other than the issues described in this report, undiscovered issues may exist such as defects on the main network. In order to write secure codes, correction of discovered problems and sufficient testing thereof are required.

Appendix. A

Severity Level

CRITICAL	Must be addressed as a vulnerability that has the potential to seize or freeze substantial sums of money.
HIGH	Has to be fixed since it has the potential to deny users compensation or momentarily freeze assets.
MEDIUM	Vulnerabilities that could halt services, such as DoS and Out-of-Gas, need to be addressed.
LOW	Issues that do not comply with standards or return incorrect values
TIPS	Tips that makes the code more usable or efficient when modified

Difficulty Level

	Low	Medium	High
Privilege	anyone	Miner/Block Proposer	Admin/Owner
Capital needed	Small or none	Gas fee or volatile as price change	More than exploited amount
Probability	100%	Depend on environment	Hard as mining difficulty

End of Document