SALUS SECURITY

May 2023

PENETRATION TESTING REPORT

INK FINANCE

Overview

Project Name	Ink Finance
File Name	Penetration Testing Report for Ink Finance
Creator	Salus Security
Create Date	12 May 2023
Total days	7 Days
Receive Date	22 May 2023

Modify History

Version	Modify Date	Modifier	Modify Type	Modified Chapter	Modified Content

*Modified Type are categorized by **A** - ADDED **M** - MODIFIED **D** – DELETED

Copyright Statement

All content appearing in this document, unless otherwise specified, is copyrighted by Salus. No individual or institution may copy, decipher or quote any fragment of the document in any way without the written authorisation of Salus Security.

Table of Contents

Introduction **4**
1 About Salus Security 4
2 Assessment Scope 4
3 Risk Summary Description 5
4 Disclaimer 5
5 Web Risk details 6
5.1 Redis Server Unauthenticated Access 6
5.2 Debug Endpoint pprof Exposure 8
5.3 Wordpress Username Enumeration 10
5.4 Unauthenticated Full Path Disclosure 10
5.5 Webpack front-end source code disclosure vulnerability 11
5.6 Kubernetes System Metrics Leakage Vulnerability 13
5.7 Weak Cipher Suites 14
5.8 TLS certificate 15
5.9 GraphQL Field Suggestion Information Disclosure 16
5.10 Brute force cracking 17
5.11 Cross-origin resource sharing 19
Security Summary **21**
Security Recommendations 21

SALUS

Introduction

1 About Salus Security

At Salus Security, we are in the business of trust. We are dedicated to tackling the toughest security challenges facing the industry today. By building foundational trust in technology and infrastructure through security, we help clients to lead their respective industries and unlock their full Web3 potential.

Our team of security experts employ industry-leading proof-of-concept (PoC) methodology for demonstrating smart contract vulnerabilities, coupled with advanced red teaming capabilities and a stereoscopic vulnerability detection service, to deliver comprehensive security assessments that allow clients to stay ahead of the curve. In addition to smart contract audits and red teaming, our Rapid Detection Service for smart contracts aims to make security accessible to all. This high calibre, yet cost-efficient, security tool has been designed to support a wide range of business needs including investment due diligence, security and code quality assessments, and code optimisation.

2 Assessment Scope

Penetration testing coverage is assessed based on the scope of the asset that needs to be tested. Because the scope of this penetration test only includes assets under the main domain name of inkfinance.xyz. Therefore we have the high-level importance assets under the main domain name of inkfinance.xyz as the main assessment penetration object from the asset list. Therefore, This report reflects a detailed security summary report related to business security.

SALUS

3 Risk Summary Description

Overall Risk Level: **High**

Description: Through the security penetration test of real environment, it is found that Ink Finance has dangerous vulnerabilities such as unauthorized access to redis server, exposure of debugging endpoint pprof, disclosure of sensitive information, invalid certificate expiration and so on. Based on these vulnerabilities, it can be judged that they may be used by hackers or criminals, which will bear huge security risks.

4 Disclaimer

This report is considered by Salus Security to be private information; it is licensed to Ink Finance under the terms of the project statement of work. Material within this report may not be reproduced or distributed in part or in whole without the express written permission of Salus Security.

Test Coverage Disclaimer

All activities undertaken by Salus Security in association with this project were performed in accordance with a statement of work and mutually agreed upon project plan. Security Penetration Testing projects are time-boxed and often reliant on the information that may be provided by a client, its affiliates, or its partners. As a result, the findings documented in this report should not be considered a comprehensive list of security issues, flaws, or defects in the target system or codebase.

Salus Security uses automated testing techniques to test the controls and security properties of software rapidly. These techniques augment our penetration testing work, but each has its limitations. Their use is also limited by the time and resource constraints of a project.

Salus Security makes all effort but holds no responsibility for the findings of this penetration testing. Salus Security makes no judgments on the underlying business model or the individuals involved in the project.

SALUS

5 Web Risk details

5.1 Redis Server Unauthenticated Access

Vulnerability Name:	Redis Server Unauthenticated Access
Risk Level:	High
Vulnerability URL:	activity.inkfinance.xyz:6379
Vulnerability Description:	Unauthenticated Redis servers may be targeted by attackers to get a foothold on your internal network by compromising the server. Different attacks may be used against a Redis server to get arbitrary code execution on its underlying operating system. For example, an attacker may use Redis commands to write a web shell into the web root. When used to store objects, an attacker may also store deserialization payloads for different languages such as Java, Python, Ruby, PHP, etc. in order to get remote code execution on the deserializating endpoint. Please note that even if the Redis server is not exposed externally, an external attacker may still reach it via a Server-Side Request Forgery vulnerability in any application on the same network. Redis Servers can be attacker using HTTP or Gopher protocols for example.
Vulnerability Detail:	Redis provides the info command, which returns various information and statistics about the Redis server. Request: info quit Response: $3341 # Server redis_version:5.0.7 redis_git_sha1:00000000 redis_git_dirty:0 redis_build_id:66bd629f924ac924 redis_mode:standalone os:Linux 5.11.0-1022-aws x86_64 arch_bits:64 multiplexing_api:epoll atomicvar_api:atomic-builtin

SALUS

```
gcc_version:9.3.0
process_id:2129616
run_id:e9ee31bf3846eb1a6093599846915948092442 20
tcp_port:6379
uptime_in_seconds:7973
uptime_in_days:0
hz:10
configured_hz:10
lru_clock:5978855
executable:/usr/bin/redis-server
config_file:/etc/redis/redis.conf

# Clients
connected_clients:5
client_recent_max_input_buffer:2
client_recent_max_output_buffer:0
blocked_clients:0

# Memory
used_memory:948864
used_memory_human:926.62K
used_memory_rss:6008832
used_memory_rss_human:5.73M
used_memory_peak:1173520
used_memory_peak_human:1.12M
used_memory_peak_perc:80.86%
used_memory_overhead:917302
used_memory_startup:796208
used_memory_dataset:31562
used_memory_dataset_perc:20.68%
allocator_allocated:1432640
allocator_active:1781760
allocator_resident:7651328
total_system_memory:16779427840
total_system_memory_human:15.63G
used_memory_lua:56320
used_memory_lua_human:55.00K
used_memory_scripts:3176
used_memory_scripts_human:3.10K
number_of_cached_scripts:1
maxmemory:0
maxmemory_human:0B
maxmemory_policy:noeviction
allocator_frag_ratio:1.24
allocator_frag_bytes:349120
allocator_rss_ratio:4.29
allocator_rss_bytes:5869568
```

SALUS

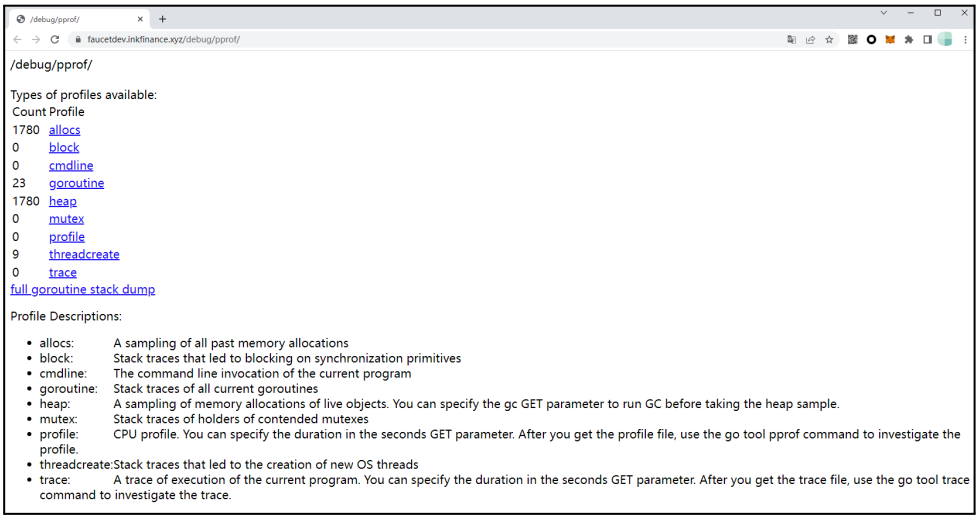
	rss_overhead_ratio:0.79 rss_overhead_bytes:-1642496 mem_fragmentation_ratio:6.78 mem_fragmentation_bytes:5122880 mem_not_counted_for_evict:0 mem_replication_backlog:0 mem_clients_slaves:0 mem_clients_normal:117382 mem_aof_buffer:0 mem_allocator:jemalloc-5.2.1 active_defrag_running:0 lazyfree_pending_objects:0 # Persistence loading:0 rdb_changes_since_last_save:41 rdb_bgsave_in_progress:0 rdb_last_save_time:1683696633 rdb_last_bgsave_status:err rdb_last_bgsave_time_sec:0 rdb_current_bgsave_time_sec:-1 rdb_last_cow_size:0 aof_enabled:0 aof_rewrite_in_progress:0 aof_rewrite_scheduled:0 aof_last_rewrite_time_sec:-1
Fix Suggestion:	Repair reference: https://redis.io/docs/management/security/

5.2 Debug Endpoint pprof Exposure

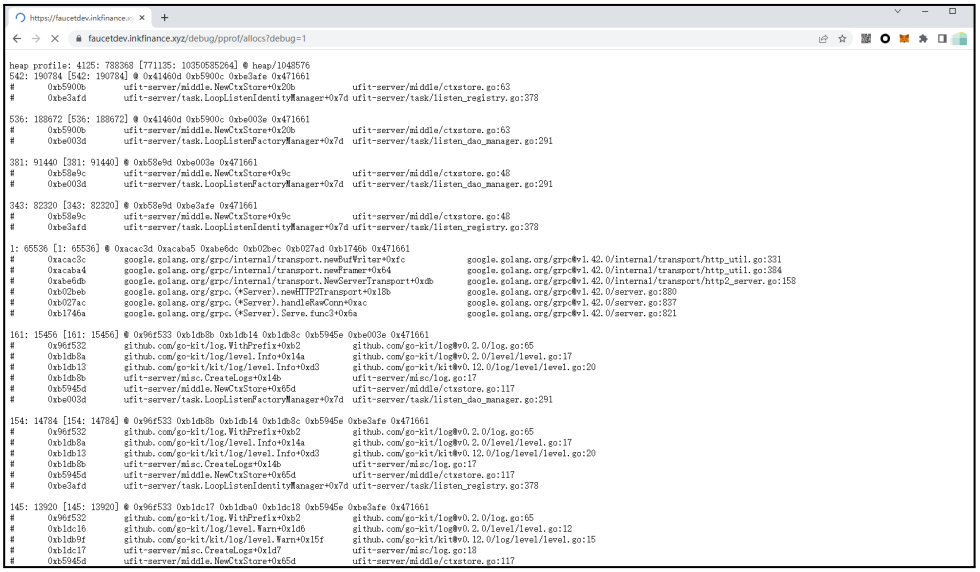
Vulnerability Name:	Debug Endpoint pprof Exposure
Risk Level:	High
Vulnerability URL:	https://faucetdev.inkfinance.xyz/debug/pprof/ https://faucet-sol-dev.inkfinance.xyz/debug/pprof/goroutine?debug=1 https://faucet-beta-goerli.inkfinance.xyz/debug/pprof/goroutine?debug=1 https://faucet-beta-polygon.inkfinance.xyz/debug/pprof/

SALUS

	https://faucet-avax-beta.inkfinance.xyz/debug/pprof/
Vulnerability Description:	The debugging endpoint /debug/pprof is exposed over the unauthenticated Kubelet healthz port. This debugging endpoint can potentially leak sensitive information such as internal Kubelet memory addresses and configuration, or for limited denial of service. Versions prior to 1.15.0, 1.14.4, 1.13.8, and 1.12.10 are affected.
Vulnerability Detail:	To access the url: https://faucetdev.inkfinance.xyz/debug/pprof/



Click any directory file:



A large amount of sensitive server operating indicators and configuration data is leaked.

Fix Suggestion:	1. Upgrade to the latest patch version of 1.15, 1.14, or 1.13. 2. Update the node configuration to set healthzBindAddress to 127.0.0.1. Reference: https://github.com/kubernetes/kubernetes/issues/81023

SALUS

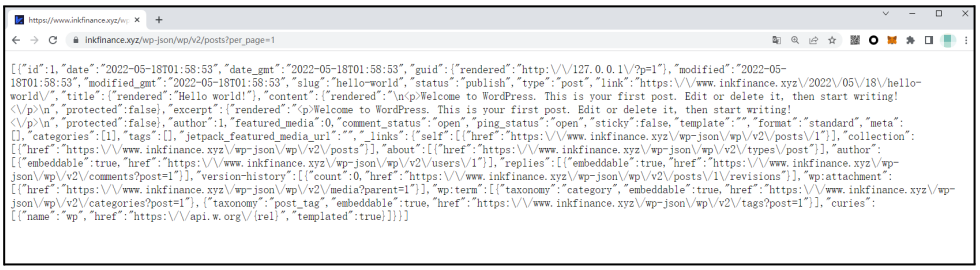
5.3 Wordpress Username Enumeration

Vulnerability Name:	Wordpress Username Enumeration
Risk Level:	Medium
Vulnerability URL:	https://www.inkfinance.xyz/?rest_route=/wp/v2/users/
Vulnerability Description:	WordPress Core before 4.7.1 is susceptible to user enumeration because it does not properly restrict listings of post authors via wp-includes/rest-api/endpoints/class-wp-rest-users-controller.php in the REST API, which allows a remote attacker to obtain sensitive information via a wp-json/wp/v2/users request.
Vulnerability Detail:	Visit Website: https://www.inkfinance.xyz/?rest_route=/wp/v2/users/ you can see the associated username.There is currently only one username, but in the future, as the data increases, there may be a risk of leakage.
Fix Suggestion:	Use the official patch given by WordPress to fix the vulnerability. https://wordpress.org/news/2017/01/wordpress-4-7-1-security-and-maintenance-release/

5.4 Unauthenticated Full Path Disclosure

Vulnerability Name:	Unauthenticated Full Path Disclosure
Risk Level:	Medium
Vulnerability URL:	https://www.inkfinance.xyz/wp-json/wp/v2/posts?per_page=1 https://betaapp.inkfinance.xyz/wp-json/wp/v2/posts?per_page=1

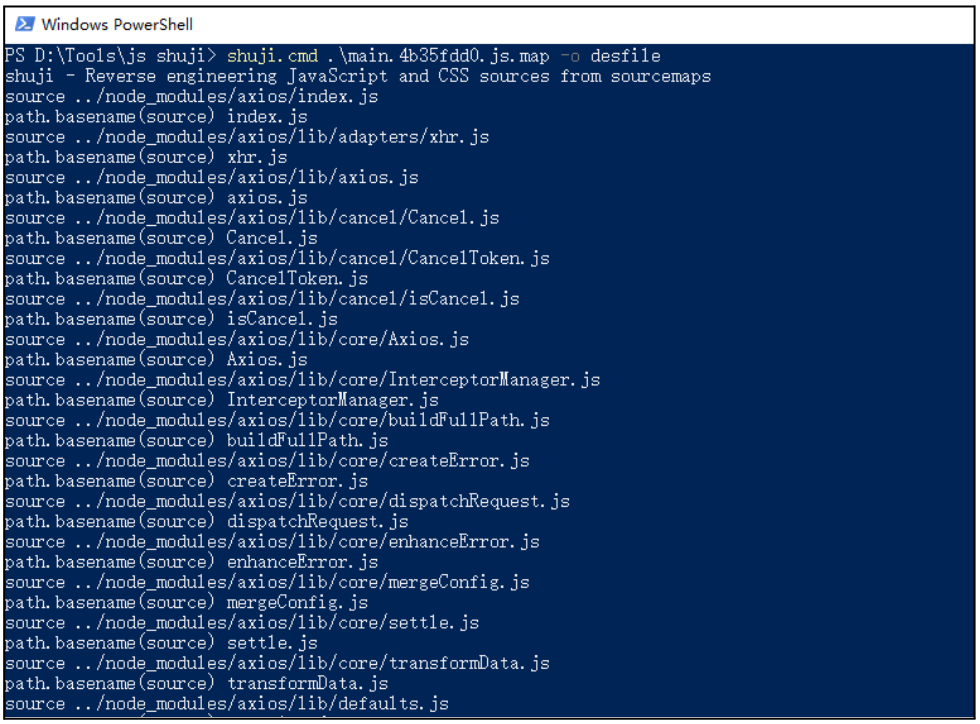
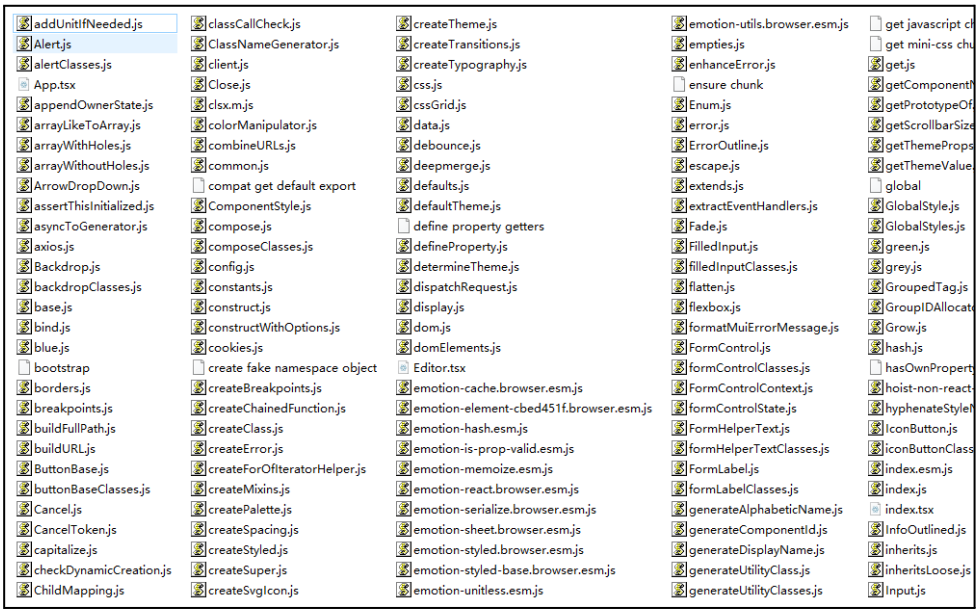
SALUS

	https://gov.inkfinance.xyz/wp-json/wp/v2/posts?per_page=1 https://govern.inkfinance.xyz/wp-json/wp/v2/posts?per_page=1
Vulnerability Description:	The Yoast SEO WordPress plugin (from versions 16.7 until 17.2) discloses the full internal path of featured images in posts via the wp/v2/posts REST endpoints which could help an attacker identify other vulnerabilities or help during the exploitation of other identified vulnerabilities.
Vulnerability Detail:	Add /wp-json/wp/v2/posts? per_page=1 Visit Website:https://www.inkfinance.xyz/wp-json/wp/v2/posts?per_page=1  [{"id":1,"date":"2022-05-18T01:58:53","date_gmt":"2022-05-18T01:58:53","guid":{"rendered":"http:\/\/127.0.0.1\/?p=1"},"modified":"2022-05-18T01:58:53","modified_gmt":"2022-05-18T01:58:53","slug":"hello-world","status":"publish","type":"post","link":"https:\/\/www.inkfinance.xyz\/2022\/05\/18\/hello-world\/","title":{"rendered":"Hello world!"},"content":{"rendered":"\n<p>Welcome to WordPress. This is your first post. Edit or delete it, then start writing!<\/p>\n","protected":false},"excerpt":{"rendered":"<p>Welcome to WordPress. This is your first post. Edit or delete it, then start writing!<\/p>\n","protected":false},"author":1,"featured_media":0,"comment_status":"open","ping_status":"open","sticky":false,"template":"","format":"standard","meta":[],"categories":[1],"tags":[],"jetpack_featured_media_url":"","_links":{"self":[{"href":"https:\/\/www.inkfinance.xyz\/wp-json\/wp\/v2\/posts\/1"}],"collection":[{"href":"https:\/\/www.inkfinance.xyz\/wp-json\/wp\/v2\/posts"}],"about":[{"href":"https:\/\/www.inkfinance.xyz\/wp-json\/wp\/v2\/types\/post"}],"author":[{"embeddable":true,"href":"https:\/\/www.inkfinance.xyz\/wp-json\/wp\/v2\/users\/1"}],"replies":[{"embeddable":true,"href":"https:\/\/www.inkfinance.xyz\/wp-json\/wp\/v2\/comments?post=1"}],"version-history":[{"count":0,"href":"https:\/\/www.inkfinance.xyz\/wp-json\/wp\/v2\/posts\/1\/revisions"}],"wp:attachment":[{"href":"https:\/\/www.inkfinance.xyz\/wp-json\/wp\/v2\/media?parent=1"}],"wp:term":[{"taxonomy":"category","embeddable":true,"href":"https:\/\/www.inkfinance.xyz\/wp-json\/wp\/v2\/categories?post=1"},{"taxonomy":"post_tag","embeddable":true,"href":"https:\/\/www.inkfinance.xyz\/wp-json\/wp\/v2\/tags?post=1"}],"curies":[{"name":"wp","href":"https:\/\/api.w.org\/{rel}","templated":true}]}}] A large number of internal full directory paths have been leaked.
Fix Suggestion:	Upgrade to version 17.3.

5.5 Webpack front-end source code disclosure vulnerability

Vulnerability Name:	Webpack front-end source code disclosure vulnerability
Risk Level:	Medium
Vulnerability URL:	https://marketing-mail.inkfinance.xyz/static/js/main.4b35fdd0.js.map https://serviceavax.inkfinance.xyz/static/js/main.4b35fdd0.js.map https://service-dev-sol.inkfinance.xyz/static/js/main.4b35fdd0.js.map https://service-beta-avax.inkfinance.xyz/static/js/main.4b35fdd0.js.map https://service-beta-ropsten.inkfinance.xyz/static/js/main.4b35fdd0.js.map

SALUS

Vulnerability Description:	If you can get the js code of the program, then you can target the source code for various information in the code such as hidden interfaces, APIs, encryption algorithms, administrator emails, internal functions, etc., or interface APIs can try unauthorized vulnerabilities, splicing interface override vulnerabilities, find the keywords in the source code to go to GitHub to find the source code of the program for code auditing.
Vulnerability Detail:	Browsers visiting the above URL can directly download the js.map file To restore the front-end code using the shuji tool.  After restoration, the original file was obtained as follows.  Get partial source code.

```
Windows PowerShell
PS D:\Tools\js shuji> shuji.cmd .\main.4b35fdd0.js.map -o desfile
shuji - Reverse engineering JavaScript and CSS sources from sourcemaps
source ../node_modules/axios/index.js
path.basename(source) index.js
source ../node_modules/axios/lib/adapters/xhr.js
path.basename(source) xhr.js
source ../node_modules/axios/lib/axios.js
path.basename(source) axios.js
source ../node_modules/axios/lib/cancel/Cancel.js
path.basename(source) Cancel.js
source ../node_modules/axios/lib/cancel/CancelToken.js
path.basename(source) CancelToken.js
source ../node_modules/axios/lib/cancel/isCancel.js
path.basename(source) isCancel.js
source ../node_modules/axios/lib/core/Axios.js
path.basename(source) Axios.js
source ../node_modules/axios/lib/core/InterceptorManager.js
path.basename(source) InterceptorManager.js
source ../node_modules/axios/lib/core/buildFullPath.js
path.basename(source) buildFullPath.js
source ../node_modules/axios/lib/core/createError.js
path.basename(source) createError.js
source ../node_modules/axios/lib/core/dispatchRequest.js
path.basename(source) dispatchRequest.js
source ../node_modules/axios/lib/core/enhanceError.js
path.basename(source) enhanceError.js
source ../node_modules/axios/lib/core/mergeConfig.js
path.basename(source) mergeConfig.js
source ../node_modules/axios/lib/core/settle.js
path.basename(source) settle.js
source ../node_modules/axios/lib/core/transformData.js
path.basename(source) transformData.js
source ../node_modules/axios/lib/defaults.js
```

```
addUnitIfNeeded.js      classCallCheck.js       createTheme.js          emotion-utils.browser.esm.js   get javascript ch
Alert.js                ClassNameGenerator.js   createTransitions.js    empties.js                     get mini-css chu
alertClasses.js         client.js               createTypography.js     enhanceError.js                get.js
App.tsx                 Close.js                css.js                  ensure chunk                   getComponentN
appendOwnerState.js     clsx.m.js               cssGrid.js              Enum.js                        getPrototypeOf
arrayLikeToArray.js     colorManipulator.js     data.js                 error.js                       getScrollbarSize
arrayWithHoles.js       combineURLs.js          debounce.js             ErrorOutline.js                getThemeProps
arrayWithoutHoles.js    common.js               deepmerge.js            escape.js                      getThemeValue
ArrowDropDown.js        compat get default export defaults.js           extends.js                     global
assertThisInitialized.js ComponentStyle.js      defaultTheme.js         extractEventHandlers.js        GlobalStyle.js
asyncToGenerator.js     compose.js              define property getters  Fade.js                       GlobalStyles.js
axios.js                composeClasses.js       defineProperty.js       FilledInput.js                 green.js
Backdrop.js             config.js               determineTheme.js       filledInputClasses.js          grey.js
backdropClasses.js      constants.js            dispatchRequest.js      flatten.js                     GroupedTag.js
base.js                 construct.js            display.js              flexbox.js                     GroupIDAllocat
bind.js                 constructWithOptions.js dom.js                  formatMuiErrorMessage.js       Grow.js
blue.js                 cookies.js              domElements.js          FormControl.js                 hash.js
bootstrap               create fake namespace object Editor.tsx         formControlClasses.js          hasOwnPropert
borders.js              createBreakpoints.js    emotion-cache.browser.esm.js FormControlContext.js     hoist-non-react
breakpoints.js          createChainedFunction.js emotion-element-cbed451f.browser.esm.js formControlState.js hyphenateStyleN
buildFullPath.js        createClass.js          emotion-hash.esm.js     FormHelperText.js              IconButton.js
buildURL.js             createError.js          emotion-is-prop-valid.esm.js formHelperTextClasses.js  iconButtonClass
ButtonBase.js           createForOfIteratorHelper.js emotion-memoize.esm.js FormLabel.js               index.esm.js
buttonBaseClasses.js    createMixins.js         emotion-react.browser.esm.js formLabelClasses.js       index.js
Cancel.js               createPalette.js        emotion-serialize.browser.esm.js generateAlphabeticName.js index.tsx
CancelToken.js          createSpacing.js        emotion-sheet.browser.esm.js generateComponentId.js    InfoOutlined.js
capitalize.js           createStyled.js         emotion-styled.browser.esm.js generateDisplayName.js   inherits.js
checkDynamicCreation.js createSuper.js          emotion-styled-base.browser.esm.js generateUtilityClass.js inheritsLoose.js
ChildMapping.js         createSvgIcon.js        emotion-unitless.esm.js generateUtilityClasses.js      Input.js
```

SALUS

	``
	// @flow
	import unitless from '@emotion/unitless';
	
	// Taken from https://github.com/facebook/react/blob/b87aabdfe1b7461e7331abb3601d9e6bb27544bc/packages/react-dom/sr
	export default function addUnitIfNeeded(name: string, value: any): any {
	  // https://github.com/amilajack/eslint-plugin-flowtype-errors/issues/133
	  // $FlowFixMe
	  if (value == null || typeof value === 'boolean' || value === '') {
	  return '';
	  }
	
	  if (typeof value === 'number' && value !== 0 && !(name in unitless)) {
	  return `${value}px`; // Presumes implicit 'px' suffix for unitless numbers
	  }
	
	  return String(value).trim();
	}
	
	``

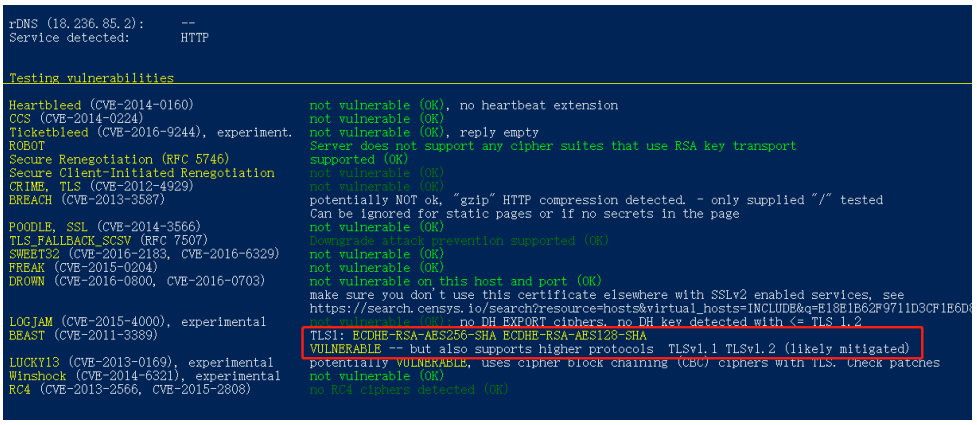
Fix Suggestion:	1.Modify the build object productionSourceMap: false in config/index.js under the project path. 2.Recommend removing or disabling access to the js.map file in the official environment.

5.6 Kubernetes System Metrics Leakage Vulnerability

Vulnerability Name:	Kubernetes System Metrics Leakage Vulnerability
Risk Level:	Medium
Vulnerability URL:	https://exploredev.inkfinance.xyz/metrics
Vulnerability Description:	Due to improper developer configuration, exposing the interface to the public web or not configuring restricted access, hackers can use the following Actuator monitoring native endpoints to gain access to some sensitive data about the website, such as the metrics interface, leaking various application metrics, such as memory usage and HTTP request counts.
Vulnerability Detail:	Access to specific vulnerable URL and following relevant data can be seen.

SALUS

	https://exploredev.inkfinance × +
	exploredev.inkfinance.xyz/metrics
	
	``# TYPE pending_block_operations_count gauge
	# HELP pending_block_operations_count Number of records in pending_block_operations table
	pending_block_operations_count 1
	# TYPE import_errors_count counter
	# HELP import_errors_count Number of database import errors
	import_errors_count 0
	# TYPE missing_block_count gauge
	# HELP missing_block_count Number of missing blocks in the database
	missing_block_count 0
	# TYPE delay_from_last_node_block gauge
	# HELP delay_from_last_node_block Delay from the last block on the node in seconds
	delay_from_last_node_block 0
	# TYPE block_import_stage_runner_duration_microseconds histogram
	# HELP block_import_stage_runner_duration_microseconds Block import stage, runner and step in runner processing time
	block_import_stage_runner_duration_microseconds_bucket{stage="addresses",runner="addresses",step="created_address_code_indexed_at_transactions",le="1000"} 442615
	block_import_stage_runner_duration_microseconds_bucket{stage="addresses",runner="addresses",step="created_address_code_indexed_at_transactions",le="5000"} 442615
	block_import_stage_runner_duration_microseconds_bucket{stage="addresses",runner="addresses",step="created_address_code_indexed_at_transactions",le="10000"} 442615
	block_import_stage_runner_duration_microseconds_bucket{stage="addresses",runner="addresses",step="created_address_code_indexed_at_transactions",le="100000"} 442615
	block_import_stage_runner_duration_microseconds_bucket{stage="addresses",runner="addresses",step="created_address_code_indexed_at_transactions",le="+Inf"} 442615
	block_import_stage_runner_duration_microseconds_count{stage="addresses",runner="addresses",step="created_address_code_indexed_at_transactions"} 442615
	block_import_stage_runner_duration_microseconds_sum{stage="addresses",runner="addresses",step="created_address_code_indexed_at_transactions"} 13271.43
	block_import_stage_runner_duration_microseconds_bucket{stage="block_pending",runner="internal_transactions",step="acquire_transactions",le="1000"} 859199
	block_import_stage_runner_duration_microseconds_bucket{stage="block_pending",runner="internal_transactions",step="acquire_transactions",le="5000"} 859199
	block_import_stage_runner_duration_microseconds_bucket{stage="block_pending",runner="internal_transactions",step="acquire_transactions",le="10000"} 859199
	block_import_stage_runner_duration_microseconds_bucket{stage="block_pending",runner="internal_transactions",step="acquire_transactions",le="100000"} 859199
	block_import_stage_runner_duration_microseconds_bucket{stage="block_pending",runner="internal_transactions",step="acquire_transactions",le="+Inf"} 859199
	block_import_stage_runner_duration_microseconds_count{stage="block_pending",runner="internal_transactions",step="acquire_transactions"} 859199
	block_import_stage_runner_duration_microseconds_sum{stage="block_pending",runner="internal_transactions",step="acquire_transactions"} 757484.81
	block_import_stage_runner_duration_microseconds_bucket{stage="block_following",runner="current_token_balances",step="acquire_contract_address_tokens",le="1000"} 789
	block_import_stage_runner_duration_microseconds_bucket{stage="block_following",runner="current_token_balances",step="acquire_contract_address_tokens",le="5000"} 789
	block_import_stage_runner_duration_microseconds_bucket{stage="block_following",runner="current_token_balances",step="acquire_contract_address_tokens",le="10000"} 789
	block_import_stage_runner_duration_microseconds_bucket{stage="block_following",runner="current_token_balances",step="acquire_contract_address_tokens",le="100000"} 789
	block_import_stage_runner_duration_microseconds_bucket{stage="block_following",runner="current_token_balances",step="acquire_contract_address_tokens",le="+Inf"} 789
	block_import_stage_runner_duration_microseconds_count{stage="block_following",runner="current_token_balances",step="acquire_contract_address_tokens"} 789
	block_import_stage_runner_duration_microseconds_sum{stage="block_following",runner="current_token_balances",step="acquire_contract_address_tokens"} 1025.624```
	``

	Using an insufficient length for a key in an encryption/decryption algorithm opens up the possibility (or probability) that the encryption scheme could be broken.
Vulnerability Detail:	Use testssl to detect the corresponding domain name, and you can find that there are SSL related vulnerabilities.  rDNS (18.236.85.2): -- Service detected: HTTP Testing vulnerabilities Heartbleed (CVE-2014-0160) not vulnerable (OK), no heartbeat extension CCS (CVE-2014-0224) not vulnerable (OK) Ticketbleed (CVE-2016-9244), experiment. not vulnerable (OK), reply empty ROBOT Server does not support any cipher suites that use RSA key transport Secure Renegotiation (RFC 5746) supported (OK) Secure Client-Initiated Renegotiation not vulnerable (OK) CRIME, TLS (CVE-2012-4929) not vulnerable (OK) BREACH (CVE-2013-3587) potentially NOT ok, "gzip" HTTP compression detected. - only supplied "/" tested Can be ignored for static pages or if no secrets in the page POODLE, SSL (CVE-2014-3566) not vulnerable (OK) TLS_FALLBACK_SCSV (RFC 7507) Downgrade attack prevention supported (OK) SWEET32 (CVE-2016-2183, CVE-2016-6329) not vulnerable (OK) FREAK (CVE-2015-0204) not vulnerable (OK) DROWN (CVE-2016-0800, CVE-2016-0703) not vulnerable on this host and port (OK) make sure you don't use this certificate elsewhere with SSLv2 enabled services, see https://search.censys.io/search?resource=hosts&virtual_hosts=INCLUDE&q=E18B1B62F9711D3CF1E6D6 LOGJAM (CVE-2015-4000), experimental not vulnerable (OK): no DH EXPORT ciphers, no DH key detected with <= TLS 1.2 BEAST (CVE-2011-3389) TLS1: ECDHE-RSA-AES256-SHA ECDHE-RSA-AES128-SHA VULNERABLE -- but also supports higher protocols TLSv1.1 TLSv1.2 (likely mitigated) LUCKY13 (CVE-2013-0169), experimental potentially VULNERABLE, uses cipher block chaining (CBC) ciphers with TLS. Check patches Winshock (CVE-2014-6321), experimental not vulnerable (OK) RC4 (CVE-2013-2566, CVE-2015-2808) no RC4 ciphers detected (OK)
Fix Suggestion:	Reconfigure the affected application to avoid use of weak cipher suites. Repair reference: https://github.com/ssllabs/research/wiki/SSL-and-TLS-Deployment-Best-Practices

5.8 TLS certificate

Vulnerability Name:	TLS certificate
Risk Level:	Medium
Vulnerability URL:	https://origin.inkfinance.xyz https://alpha.inkfinance.xyz
Vulnerability Description:	TLS (or SSL) helps to protect the confidentiality and integrity of information in transit between the browser and server, and to provide authentication of the server's identity. To serve this purpose, the server must present an TLS certificate that is valid for the server's hostname, is issued by a trusted authority and is valid for the current date. If any one of these requirements is not met, TLS connections to the server will not provide the full protection for which TLS is designed. It should be noted that various attacks exist against TLS in general, and in the context of HTTPS web connections in particular. It may be possible for a determined and suitably-positioned attacker to compromise TLS

SALUS

	connections without user detection even when a valid TLS certificate is used.
Vulnerability Detail:	The following problems were identified with the server's TLS certificate: The server's certificate is not valid for the server's hostname. The server's certificate is not trusted. The server's certificate has expired. The server presented the following certificate: Issued to: *.ufit.live, ufit.live Issued by: Sectigo RSA Domain Validation Secure Server CA Valid from: Tue Mar 09 08:00:00 CST 2021 Valid to: Sun Apr 10 07:59:59 CST 2022
Fix Suggestion:	Repair reference: https://github.com/ssllabs/research/wiki/SSL-and-TLS-Deployment-Best-Practices

5.9 GraphQL Field Suggestion Information Disclosure

Vulnerability Name:	GraphQL Field Suggestion Information Disclosure
Risk Level:	Low
Vulnerability URL:	https://exploredev.inkfinance.xyz/graphql
Vulnerability Description:	If introspection is disabled on your target, Field Suggestion can allow users to still earn information on the GraphQL schema. By default, GraphQL backends have a feature for fields and operations suggestions. If you try to query a field but you have made a typo, GraphQL will attempt to suggest fields that are similar to the initial attempt.
Vulnerability Detail:	Request:

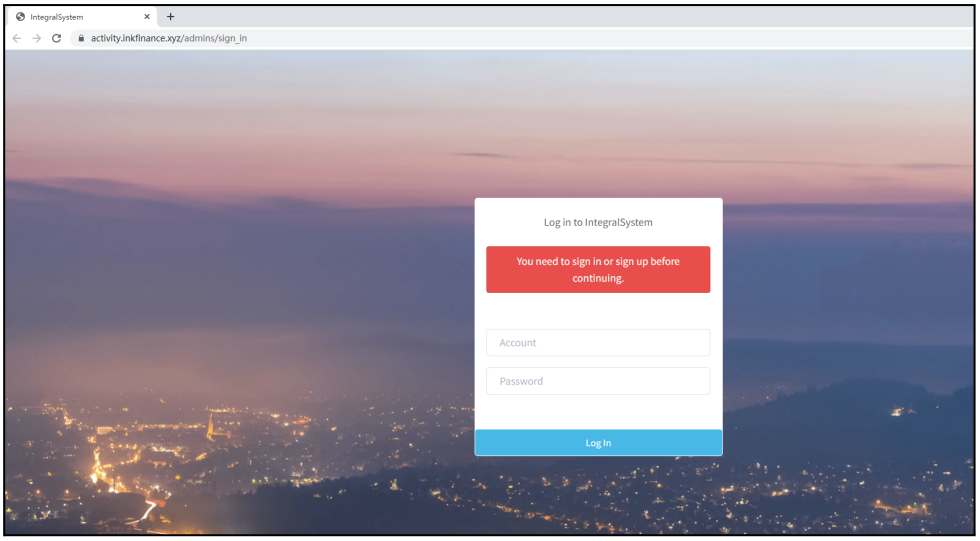
SALUS

	POST /graphql HTTP/1.1 Host: exploredev.inkfinance.xyz User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/37.0.2062.124 Safari/537.36 Connection: close Content-Length: 69 Content-Type: application/json Accept-Encoding: gzip {"query":"query {\n __schema {\n directive\n }\n}","variables":null} Response: HTTP/1.1 200 OK Connection: close Content-Length: 165 Access-Control-Allow-Credentials: true Access-Control-Allow-Origin: * Access-Control-Expose-Headers: Cache-Control: max-age=0, private, must-revalidate Content-Type: application/json; charset=utf-8 Date: Wed, 10 May 2023 09:21:31 GMT Server: nginx/1.18.0 (Ubuntu) X-Request-Id: F129xW4G9WwAseIEUsxB {"errors":[{"locations":[{"column":2,"line":3}],"message":"Cannot query field \"directive\" on type \"__Schema\". Did you mean \"description\" or \"directives\"?"}]}
Fix Suggestion:	Repair reference: https://github.com/webonyx/graphql-php/issues/454

5.10 Brute force cracking

Vulnerability Name:	Brute force cracking
Risk Level:	Low
Vulnerability URL:	https://activity.inkfinance.xyz/admins/sign_in https://activity.inkfinance.xyz/home/login
Vulnerability Description:	No anti-brute-force cracking mechanism is implemented on the login page. For example, there is no verification code, but the verification code is not verified on the server, and there is no limit on the number of login errors. As a result, an attacker may obtain user login accounts and passwords through brute force cracking and gain access to the website login.
Vulnerability Detail:	Visit the website: https://activity.inkfinance.xyz/admins/sign_in

SALUS



Fetch packet:

```
1 POST /admins/sign_in HTTP/1.1
2 Host: activity.inkfinance.xyz
3 Connection: close
4 Content-Length: 154
5 Cache-Control: max-age=0
6 sec-ch-ua: "Chromium";v="112", "Google Chrome";v="112", "Not:A-Brand";v="99"
7 sec-ch-ua-mobile: ?0
8 sec-ch-ua-platform: "Windows"
9 Upgrade-Insecure-Requests: 1
10 Origin: https://activity.inkfinance.xyz
11 Content-Type: application/x-www-form-urlencoded
12 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36
13 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
14 Sec-Fetch-Site: same-origin
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-User: ?1
17 Sec-Fetch-Dest: document
18 Referer: https://activity.inkfinance.xyz/admins/sign_in
19 Accept-Encoding: gzip, deflate
20 Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,und;q=0.7
21 Cookie: _integral_system_session=
   G%2F1QqbMjOAtVOYMxDWX4wOtKXoIpeNZ9blm%2FxeBFdydX7Gw5jTIbB74cMqtqW8UwOHO74u24sCTYcRFtSy2KvXTbujSatOmxoBASLE7%2B7okKCxLu9KpxihOr6NJbfHqRkbO4ursoe414i%2BnxPXCe%2Bcm0%2Fh6DAImthqeW4zUOGLFiCO
   XvB6vTgzcbzCOSi2w4Gc5jnyggiWCnZZPbeDXCXrXOS71UnHa%2BmdPjAn%2BpVmDEEVMEgwZmRP5jemKccoSja1y2G0zY5UDDBPaMTtxNIZYreTL2zydOZ%2BIjlIzIGZodMOsfjM9I6y9LC4cmzGaDF%2F1gcBsQgXoRwpYeO3wvfyqGOws%3D--
   %2FK5GDZ4ZijSazkb7--BJKAg16%2FpBBsjayDihy5XQ%3D%3D
22
23 authenticity_token=PNAx2JWoTbNO1gl253jEze5-u7grmjswgOQ2Hw7uWm-K4FSzlXhwYqpeahzEkXn3WSb3yMONyUCrwivDtlSh_Q&admin%5Bemail%5D=admin&admin%5Bpassword%5D=admin
```

User names and passwords can be brute-force cracked without restriction protection.

0		200			3747
1	admin	200			3753
2	admin123	200			3747
3	123456	200			3753
4	1234567	200			3751
5	12345678	200			3749
6	password	200			3751
7	Aa123456.	200			3755
8	password123	200			3753
9	Password1234	200			3769
10	admin@123456	200			3761
11	P@ssw0rd	200			3739
12	Passw0rd	200			3751
13	passw0rd	200			3755
14	password1	200			3745
15	Password1	200			3759
16	Aa1234	200			3757
17	Aa12345	200			3755
18	Aa123456	200			3761

Fix Suggestion:	Add the verification code mechanism to add pictures (the verification code is generated dynamically and meets randomness) or SMS verification code (the timeout period of the verification code is generally 1 minute, and if the number of errors exceeds 3 times within the time limit, the verification code will be locked for 1 minute before it can be obtained

SALUS

	again, and the verification code will automatically become invalid after the timeout).

5.11 Cross-origin resource sharing

Vulnerability Name:	Cross-origin resource sharing
Risk Level:	Low
Vulnerability URL:	https://exploredev.inkfinance.xyz/
Vulnerability Description:	An HTML5 cross-origin resource sharing (CORS) policy controls whether and how content running on other domains can perform two-way interaction with the domain that publishes the policy. The policy is fine-grained and can apply access controls per-request based on the URL and other features of the request. Trusting arbitrary origins effectively disables the same-origin policy, allowing two-way interaction by third-party web sites. Unless the response consists only of unprotected public content, this policy is likely to present a security risk. If the site specifies the header Access-Control-Allow-Credentials: true, third-party sites may be able to carry out privileged actions and retrieve sensitive information. Even if it does not, attackers may be able to bypass any IP-based access controls by proxying through users' browsers.
Vulnerability Detail:	The part marked in red is the problem part. Request: GET / HTTP/1.1 Host: exploredev.inkfinance.xyz Connection: close Pragma: no-cache Cache-Control: no-cache sec-ch-ua: "Chromium";v="112", "Google Chrome";v="112", "Not:A-Brand";v="99" sec-ch-ua-mobile: ?0 sec-ch-ua-platform: "Windows" Upgrade-Insecure-Requests: 1 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7 Sec-Fetch-Site: none Sec-Fetch-Mode: navigate Sec-Fetch-User: ?1 Sec-Fetch-Dest: document Accept-Encoding: gzip, deflate Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,und;q=0.7 Origin: https://nlzfwwwffhca.com Response:

SALUS

```
HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Thu, 11 May 2023 07:29:07 GMT
Content-Type: text/html; charset=utf-8
Connection: close
access-control-allow-credentials: true
access-control-allow-origin: *
access-control-expose-headers:
cache-control: max-age=0, private, must-revalidate
content-security-policy: connect-src 'self'  https://api-js.mixpanel.com https://api2.am
cross-origin-window-policy: deny
x-content-type-options: nosniff
x-download-options: noopen
x-permitted-cross-domain-policies: none
x-request-id: F14GN-lyObEO4kAEphIC
x-xss-protection: 1; mode=block
Content-Length: 44016

<!DOCTYPE html>
<html lang="en-US">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link rel="stylesheet" href="/css/main-page-d16bda2570bf6f2c16f82ad6c6e7bda2.css?vsn
    <link rel="preload" href="/js/chain-1cec941d81f5e55046cee7c561ca5ecc.js?vsn=d" as="s
```

The application implements an HTML5 cross-origin resource sharing (CORS) policy for this request that allows access from any domain. The application allowed access from the requested origin https://exppavqtqumo.com The response uses a wildcard in the Access-Control-Allow-Origin header and also specifies that credentials are allowed. Note that browsers do not allow this combination, and the Access-Control-Allow-Credentials header will be ignored. Since the Vary: Origin header was not present in the response, reverse proxies and intermediate servers may cache it. This may enable an attacker to carry out cache poisoning attacks.

Fix Suggestion:	Rather than using a wildcard or programmatically verifying supplied origins, use a whitelist of trusted domains.

SALUS

Security Summary

Security Recommendations

1) Strictly control the access rights of relevant pages and check the rights of access roles.
2) Fuzzify relevant sensitive information, do it on the server side, and strictly check the data returned by the server side. the query data and page display data has to be consistent and never return redundant data.
3) Delete pages that are not related to the business. If it is a middleware management page that must be used, it is recommended to control the access rights of the page.

SALUS