BRSCPP
Security Audit Report

Crypto Payment Gateway Infrastructure

Status	✅ PASSED - Production Ready

Version	1.0
Audit Date	December 26, 2025
Author	Slavcho Ivanov
Document	Public

1. Executive Summary

This security audit evaluates the BRSCPP Crypto Payment Gateway infrastructure, including smart contracts deployed on Ethereum Sepolia, BSC Testnet, and Polygon Amoy, as well as the backend API server.

Key Findings

Category	Tests	Passed	Failed	Status
Smart Contract Security	12	12	0	■ SECURE
Malicious Contract Attacks	6	6	0	■ SECURE
API Security	20	20	0	■ SECURE
Business Logic Attacks	15	15	0	■ SECURE
Static Analysis (Slither)	100 det.	0 real	2 FP	■ CLEAN
Static Analysis (Mythril)	Symbolic	0 real	17 FP	■ CLEAN

Overall Security Score: 100% (53/53 tests passed)

Risk Assessment

Risk Level	Count	Description
■ Critical	0	No critical vulnerabilities found
■ High	0	No high-risk issues found
■ Medium	0	No medium-risk issues found
■ Low	0	No low-risk issues found
■ Informational	2	Design decisions documented

2. Audit Scope

Audit Scope

The audit employed multiple testing methodologies: Static Analysis (Slither + Mythril), Dynamic Testing (API fuzzing, injection attacks), and On-Chain Attacks using a deployed MaliciousAttacker contract.

Smart Contracts Audited

Network	Contract	Address	Version
Sepolia	CryptoPaymentGateway	0x31b8...41Dd	Solidity 0.8.27
BSC Testnet	CryptoPaymentGateway	0xee61...bAA	Solidity 0.8.27
Polygon Amoy	CryptoPaymentGateway	0xC4De...E49	Solidity 0.8.27

Backend Services Audited

Service	URL	Technology
Payment API	api.brscpp.slavy.space	Node.js + Express
Authentication	JWT + API Keys	bcrypt + HMAC
Database	PostgreSQL	Prisma ORM

Attack Contract

Contract	Address	Purpose
MaliciousAttacker	0xc418...5dDA	Reentrancy & attack simulation

3. Smart Contract Security

Security Features Implemented

Feature	Implementation	Status
Reentrancy Protection	OpenZeppelin ReentrancyGuard	■ Active
Access Control	Ownable pattern	■ Active
Emergency Stop	Pausable modifier	■ Active
Quote Expiration	Block-based validity	■ Active
Quote Single-Use	isUsed flag	■ Active
Quote Ownership	Creator binding	■ Active
Oracle Validation	Chainlink + staleness check	■ Active
Amount Validation	Exact match required	■ Active
Token Whitelist	supportedTokens mapping	■ Active
Safe Transfers	OpenZeppelin SafeERC20	■ Active

Smart Contract Test Results

Test	Attack Vector	Result
Reentrancy Protection	receive() callback attack	■ BLOCKED
Quote Reuse Prevention	Double-spend attempt	■ BLOCKED
Quote Theft	Front-running attack	■ BLOCKED
Zero Address Merchant	Payment to 0x0	■ BLOCKED
Fake Quote ID	Random bytes32	■ BLOCKED
Expired Quote	Old validUntilBlock	■ BLOCKED
Value Mismatch	Underpayment attempt	■ BLOCKED

Unsupported Token	Non-whitelisted token	■ BLOCKED
Access Control (Fee)	Non-owner setFee	■ BLOCKED
Access Control (Pause)	Non-owner pause	■ BLOCKED
Concurrent Race	3 parallel payments	■ BLOCKED
Pause Enforcement	Operations while paused	■ BLOCKED

4. API Security

HTTP Security Headers

Header	Status	Value
X-Frame-Options	■ Present	DENY
X-Content-Type-Options	■ Present	nosniff
X-XSS-Protection	■ Present	1; mode=block
Strict-Transport-Security	■ Present	max-age=31536000
Content-Security-Policy	■ Present	Configured
X-Powered-By	■ Hidden	Not exposed

Authentication Security

Test	Result	Details
Invalid Credentials	■ PASS	Properly rejected
Empty Credentials	■ PASS	Validation error returned
SQL Injection	■ PASS	Parameterized queries
NoSQL Injection	■ PASS	Input sanitization
JWT Manipulation	■ PASS	Invalid tokens rejected
Rate Limiting (Login)	■ PASS	5 attempts / 15 min
Rate Limiting (General)	■ PASS	15/50 blocked

TLS Configuration

Protocol	Status
TLS 1.0	■ Disabled

TLS 1.1	■ Disabled
TLS 1.2	■ Enabled
TLS 1.3	■ Enabled

5. Static Analysis

Slither Analysis (Pattern Matching)

Slither is a static analysis framework that detects vulnerabilities through pattern matching across 100+ detectors. Analysis completed on 24 contracts.

Severity	Findings	Status
High	0	■ None found
Medium	0	■ None found
Low	2	■■ False Positives
Informational	0	■ None found

Low Severity Findings (False Positives):

Finding: Timestamp used for comparisons in _tryGetPrimaryPrice() and _tryGetSecondaryPrice()

Assessment: FALSE POSITIVE - Chainlink oracles return timestamps, not block numbers. Using block.timestamp is the correct and industry-standard approach for oracle staleness checks. The 3600-second threshold makes miner manipulation (±15 sec) negligible.

Mythril Analysis (Symbolic Execution)

Mythril performs symbolic execution to find deep logic bugs and edge cases. Analysis ran with 300-second execution timeout on deployed bytecode.

SWC ID	Vulnerability	Count	Status
SWC-101	Integer Overflow/Underflow	17	■■ False Positive
SWC-107	Reentrancy	0	■ None found
SWC-106	Unprotected Selfdestruct	0	■ None found
SWC-104	Unchecked Call Return	0	■ None found
SWC-105	Unprotected Ether Withdrawal	0	■ None found
SWC-115	tx.origin Authorization	0	■ None found

SWC-101 Findings (False Positives):

Assessment: All 17 findings are FALSE POSITIVES because Solidity 0.8.27 has built-in overflow/underflow protection. All arithmetic operations automatically revert with Panic(0x11) if overflow occurs. Mythril analyzes raw bytecode and cannot determine the Solidity version, thus reports all arithmetic as potentially vulnerable.

Combined Static Analysis Summary

Tool	Method	Real Issues	False Positives	Status
Slither	Pattern Matching	0	2	■ CLEAN
Mythril	Symbolic Execution	0	17	■ CLEAN
Total	**-**	**0**	**19**	**■ SECURE**

6. Attack Simulation Results

Malicious Contract Attack Suite

Real on-chain attacks executed using deployed MaliciousAttacker contract:

Attack	Method	Result	Defense
Reentrancy	attackReentrancy()	■ BLOCKED	ReentrancyGuard
Front-Running	attackFrontRun()	■ BLOCKED	Creator validation
Replay Attack	Double payment	■ BLOCKED	QuoteAlreadyUsed
No Approval	attackERC20WithoutApproval()	■ BLOCKED	SafeERC20
Zero Value	msg.value = 0	■ BLOCKED	AmountMismatch
Force-Send ETH	selfdestruct	■ IMMUNE	No balance deps

API Attack Suite

Attack	Payload	Result
SQL Injection	' OR '1'='1	■ BLOCKED
NoSQL Injection	{"$gt":""}	■ BLOCKED
XSS Payloads	<script>alert(1)</script>	■ SANITIZED
Path Traversal	../../../etc/passwd	■ BLOCKED
Command Injection	; ls -la	■ BLOCKED
IDOR	Access other merchant data	■ BLOCKED
Mass Assignment	{"role":"admin"}	■ BLOCKED
Timing Attack	User enumeration	■ MITIGATED

7. Security Features

Smart Contract Protections

Layer	Protection	Implementation
Reentrancy	ReentrancyGuard	OpenZeppelin modifier
Access	Ownable	Owner-only admin functions
Emergency	Pausable	pause() / unpause()
State	CEI Pattern	Checks-Effects-Interactions
Quotes	Expiration	Block-based validity
Quotes	Single-use	isUsed flag enforcement
Quotes	Ownership	Creator binding validation
Oracle	Staleness	Timestamp validation
Tokens	SafeERC20	Safe transfer wrappers

API Protections

Layer	Protection	Implementation
Transport	TLS 1.2+	Apache2 SSL
Headers	Security headers	Helmet middleware
Auth	JWT + API Keys	bcrypt + HMAC-SHA256
Rate Limit	Request throttling	express-rate-limit
Input	Validation	Joi + Prisma
Database	SQL Injection prevention	Parameterized queries
CORS	Origin validation	Whitelist domains

8. Recommendations

Implemented ■

#	Recommendation	Status
1	Use ReentrancyGuard on all state-changing functions	■ Done
2	Implement CEI pattern	■ Done
3	Add rate limiting to auth endpoints	■ Done
4	Disable TLS 1.0/1.1	■ Done
5	Hide X-Powered-By header	■ Done
6	Implement quote expiration	■ Done
7	Add oracle staleness checks	■ Done
8	Use SafeERC20 for token transfers	■ Done

Future Considerations

#	Recommendation	Priority
1	Professional third-party audit before mainnet	High
2	Bug bounty program	Medium
3	Automated monitoring and alerting	Medium
4	Multi-sig for admin functions	Low

9. Conclusion

The BRSCPP Crypto Payment Gateway has successfully passed all security tests across smart contracts, API endpoints, and business logic. The implementation demonstrates:

- **Strong smart contract security** with OpenZeppelin libraries and CEI pattern
- **Robust API protection** with rate limiting, input validation, and proper authentication
- **Resistance to common attacks** including reentrancy, replay, and injection attacks
- **Proper access control** with Ownable pattern and JWT/API key authentication

Security Audit Certification

Project	BRSCPP Crypto Payment Gateway
Version	2.1
Audit Date	December 26, 2025
Status	■ **PASSED**
Total Tests	53
Passed	53
Failed	0
Critical Vulnerabilities	0
Recommendation	Ready for Production Deployment

Appendix

A. Test Scripts Location

Script	Path	Purpose
API Security	Tests/api-security-test.sh	HTTP/Auth testing
Brutal API	Tests/brutal-attack-test.sh	Business logic attacks
Smart Contract	blockchain/scripts/brutal-attacks.js	On-chain security
Malicious Contract	blockchain/scripts/malicious-attacks.js	Contract attacks

B. Contract Verification

All contracts verified on respective block explorers:

Network	Explorer Link
Sepolia	sepolia.etherscan.io/address/0x31b8...41Dd
BSC Testnet	testnet.bscscan.com/address/0xee61...4074
Polygon Amoy	amoy.polygonscan.com/address/0xC4De...E49

C. Security Contact

For security-related inquiries or to report vulnerabilities:

• Website: https://me.slavy.space

• Project: https://brscpp.slavy.space

Report Generated: December 26, 2025
Document Version: 1.0
Author: Slavcho Ivanov
Classification: Public