Haven Protocol Security Audit Report

1. Executive Summary

On August 14, 2020, the SlowMist security team received the Haven Protocol team's security audit application for Haven Protocol, developed the audit plan according to the agreement of both parties and the characteristics of the project, and finally issued the security audit report.

The SlowMist security team adopts the strategy of "black, grey box lead, white box assists" to conduct a complete security test on the project in the way closest to the real attack.

SlowMist blockchain system test method:

Black box testing	Conduct security tests from an attacker's perspective externally.
Grey box testing	Conduct security testing on code module through the scripting tool, observing the internal running status, mining weaknesses.
White box testing	Based on the open source code, non-open source code, to detect wether there are vulnerabilities in programs suck as nodes, SDK, etc.

SlowMist blockchain risk level:

Critical vulnerabilities	Critical vulnerabilities will have a significant impact on the security of the blockchain, and it is strongly recommended to fix the critical vulnerabilities.
High-risk vulnerabilities	High-risk vulnerabilities will affect the normal operation of blockchain. It is strongly recommended to fix high-risk vulnerabilities.
Medium-risk	Medium vulnerability will affect the operation of blockchain. It is recommended

vulnerablities	to fix medium-risk vulnerabilities.
Low-risk vulnerabilities	Low-risk vulnerabilities may affect the operation of blockchain in certain scenarios. It is suggested that the project party should evaluate and consider whether these vulnerabilities need to be fixed.
Weaknesses	There are safety risks theoretically, but it is extremely difficult to reproduce in engineering.
Enhancement Suggestions	There are better practices for coding or architecture.

2. Project Background (Context)

2.1 Project Introduction

Project website: https://www.havenprotocol.org/

Project source code: https://github.com/haven-protocol-org/haven-offshore

Coin symbol: XHV

Audit version: Commit: 3c7439fd0e142870b48728f920bc41c517e2d029

2.2 Scope of Audit

The main types of security audit include:

No.	Audit Category	Audit Result
1	Code Compliance Audit	PASS

2	Random Number Generation Algorithm Audit	PASS
3	Keystore Audit	PASS
4	Cryptographic Component Call Audit	PASS
5	Encryption Strength Audit	PASS
6	Length Extension Attack Audit	PASS
7	Replay Attack Audit	PASS
8	Top-up Program Audit	PASS
9	RPC Permission Audit	PASS

(other unknown security vulnerabilities are not included in the scope of responsibility of this audit)

3. Code Overview

3.1 Infrastructure

Haven-offshore developed based on the open source monero release-v0.14, using PoW consensus algorithm, The main function changes are as follows:

(1). Generate stable currency xUSD based on price oracle machine;

(2). The ring signature is upgraded to CLSAG signature, which has better performance;

(3). Optimize address format, add Haven logo, etc.

3.2 Code Compliance Audit

Fork open source blockchain source code will cause problems such as replay attacks and node address pool pollution. We conduct relevant security compliance assessments for this.

(1) Node address pool pollution assessment

- patches/src/cryptonote_config.h.patch

```
boost::uuids::uuid const NETWORK_ID = { {
```

```
- 0x12 ,0x30, 0xF1, 0x71 , 0x61, 0x04 , 0x41, 0x61, 0x17, 0x31, 0x00, 0x82, 0x16, 0xA1,
0xA1, 0x10
- } }; // Bender's nightmare
- std::string const GENESIS_TX =
"013c01ff0001ffffffffffff03029b2e4c0281c0b02e7c53291a94d1d0cbff8883f8024f5142ee494ffbb
d08807121017767aafcde9be00dcfd098715ebcf7f410daebc582fda69d24a28e9d0bc890d1";
+ 0x05 ,0x39, 0xF1, 0x70 , 0x61, 0x04 , 0x41, 0x60, 0x17, 0x32, 0x00, 0x81, 0x16, 0xA1,
0xA1, 0x10
+ } };
```

- src/p2p/net_node.inl

```
template<class t_payload_net_handler>
bool node_server<t_payload_net_handler>::do_handshake_with_peer(peerid_type& pi,
p2p_connection_context& context_, bool just_take_peerlist)
{
network_zone& zone = m_network_zones.at(context_.m_remote_address.get_zone());

typename COMMAND_HANDSHAKE::request arg;
typename COMMAND_HANDSHAKE::response rsp;
get_local_node_data(arg.node_data, zone);
m_payload_handler.get_payload_sync_data(arg.payload_data);

epee::simple_event ev;
std::atomic<bool> hsh_result(false);

bool r = epee::net_utils::async_invoke_remote_command2<typename
COMMAND_HANDSHAKE::response>(context_.m_connection_id, COMMAND_HANDSHAKE::ID, arg,
zone.m_net_server.get_config_object(),
[this, &pi, &ev, &hsh_result, &just_take_peerlist, &context_](int code, const typename
COMMAND_HANDSHAKE::response& rsp, p2p_connection_context& context)
{
epee::misc_utils::auto_scope_leave_caller scope_exit_handler =
epee::misc_utils::create_scope_leave_handler([&](){ev.raise();});

if(code < 0)
{
LOG_WARNING_CC(context, "COMMAND_HANDSHAKE invoke failed. (" << code << ", " <<
epee::levin::get_err_descr(code) << ")");
return;
```

```
}

if(rsp.node_data.network_id != m_network_id)
{
LOG_WARNING_CC(context, "COMMAND_HANDSHAKE Failed, wrong network! (" <<
rsp.node_data.network_id << "), closing connection.");
return;
}
```

The network_id of Haven Protocol and Monero nodes are different, which will not cause the problem of pollution of address pools between nodes.

Vulnerability reference: https://mp.weixin.qq.com/s/UmricgYGUakAlZTb0ihqdw

(2) Security evaluation of price oracle machine

- patches/src/cryptonote_core/blockchain.cpp.patch

```
+bool Blockchain::get_pricing_record(offshore::pricing_record& pr, uint64_t timestamp)
const
+{
+ LOG_PRINT_L1("Requesting pricing record from Oracle - time : " << timestamp);
+
+ epee::net_utils::http::http_simple_client http_client;
+ COMMAND_RPC_GET_PRICING_RECORD::request req = AUTO_VAL_INIT(req);
+ COMMAND_RPC_GET_PRICING_RECORD::response res = AUTO_VAL_INIT(res);
+
+ std::array<std::string, 3> oracle_urls = {{"oracle.havenprotocol.org:443",
"oracle2.havenprotocol.org:443", "oracle3.havenprotocol.org:443"}};
+ std::shuffle(oracle_urls.begin(), oracle_urls.end(),
std::default_random_engine(crypto::rand<unsigned>()));
```

The price oracle machine is composed of 3 official nodes, with large centralized authority

Oracle code location: https://github.com/dweab/oracle-lite

(3) Prevention of oracle price manipulation

The exchange between xUSD and XHV does not require a counterparty. Due to the lack of initial liquidity, there is a price control problem. The exchange rate control of xUSD and XHV realized by short-term pull or smashing is used to issue a large number of xUSD or XHV.

In response to this problem, Haven Protocol has taken corresponding precautions: The

first point is that the exchange price is not the exchange price of a certain node, and the exchange price currently used is the average price recorded in the previous 720 blocks. Then there is a redemption and a fee rate. The current rate is determined according to the unlocking time of the redemption. The fastest unlocking period is 6 hours. If you want to unlock for 6 hours, you need to pay 20 of the total amount of your exchange. % As the exchange rate, 10% for 24 hours unlocking, 5% for 48 hours, and 0.2% for 168 hours. The later plan is to design a more reasonable fee rate mechanism, which will increase similar exchange slippage based on the size of the current liquidity and the impact of the user's exchange on the total amount at that time.

3.3 Random Number Generation Algorithm Audit

Generating random numbers based on /dev/urandom is security enough.

- src/crypto/crypto.cpp

```
secret_key crypto_ops::generate_keys(public_key &pub, secret_key &sec, const secret_key&
recovery_key, bool recover) {
ge_p3 point;

secret_key rng;

if (recover)
{
rng = recovery_key;
}
else
{
random_scalar(rng);
}
sec = rng;
sc_reduce32(&unwrap(sec)); // reduce in case second round of keys (sendkeys)

ge_scalarmult_base(&point, &unwrap(sec));
ge_p3_tobytes(&pub, &point);

return rng;
```

```
}
```

3.4 Keystore Audit

The keystore password strength has not been verified. Weak passwords such as `123456` can be used in the test, which can be easily crack.

3.5 Cryptographic Component Call Audit

Haven Protocol import the CLSAG signature. Compared with the current ring signature structure used in the Monero protocol, the CLSAG signature is smaller, faster, and has strict security.

Algorithm paper: https://eprint.iacr.org/2019/654.pdf

Related code: patches/src/ringct/rctSigs.cpp.patch

Since ring signatures are a key component of the Monero protocol, the Monero community commissioned a formal security audit of the CLSAG password (algorithm, security model and proof) and the implementation code to be deployed.

Audit results reference:

https://web.getmonero.org/resources/research-lab/audits/clsag.pdf

The SlowMist security team conducted a security assessment on the application of CLSAG and bulletproof algorithms based on known attack methods, and found no code defects that led to the issuance and destruction of tokens, and no security problems were found.

3.6 Encryption Strength Audit

Weak hash functions such as md5 and sha1 are not used.

3.7 Length Extension Attack Audit

All fields of the transaction are encoded and hashed, and there is no transaction malleability vulnerability.

- src/cryptonote_basic/cryptonote_basic.h

```
class transaction_prefix
{

public:
// tx information
size_t version;
uint64_t unlock_time; //number of block (or time), used as a limitation like: spend this tx not early then block/time

std::vector<txin_v> vin;
std::vector<tx_out> vout;
//extra
std::vector<uint8_t> extra;

BEGIN_SERIALIZE()
VARINT_FIELD(version)
if(version == 0 || CURRENT_TRANSACTION_VERSION < version) return false;
VARINT_FIELD(unlock_time)
FIELD(vin)
FIELD(vout)
FIELD(extra)
END_SERIALIZE()
```

3.8 Replay Attack Audit

Based on the UTXO model, there is no replay problem on the chain, the same UTXO does not exist on different chains, and transactions cannot be replayed.

3.9 Top-up Program Audit

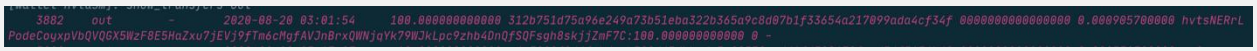
1. transfer XHV

```
transfer

hvtaPHpEsPFGkL7MVMoFKkeHkeo65hdJEgWdD4veGodmS1B58EZruaVKxXn5PDJD

Cuc6Pm99paTN13gxcydjsrFh17c9wGxC8U 100
```

Transfer record



received

2. offshore_transfer xUSD

```
offshore_transfer

hvtaPHpEsPFGkL7MVMoFKkeHkeo65hdJEgWdD4veGodmS1B58EZruaVKxXn5PDJD

Cuc6Pm99paTN13gxcydjsrFh17c9wGxC8U 100
```

Supports multiple currencies, including {"XHV", "xAG", "xAU", "xAUD", "xBTC", "xCAD", "xCHF", "xCNY", "xEUR", "xGBP", "xJPY", "xNOK", "xNZD", "xUSD"}

You need to view the funds with the view key, support multiple currencies, and pay attention to distinguishing the token symbols when entering the account.

3.10 RPC Permission Audit

RPC has the sign_transfer function, but the node only allows the RPC port to be opened locally by default, avoiding the remote transfer vulnerability similar to the "Black Valentine's Day" of Ethereum.

Vulnerability reference: https://mp.weixin.qq.com/s/Kk2lsoQ1679Gda56Ec-zJg

4. Audit Result

4.1 Enhancement Suggestions

- Price oracle have high centralized authority for machine storage, and it is recommended to decentralize authority to improve distributed security.

4.2 Exchange Security Summary

- Recharge entry needs to detect all relevant fields in the transaction and receipt structure, and reconcile it with the total account balance in real time. If an abnormality occurs, it needs to be manually checked before processing the entry to prevent "fake recharge attacks".
- There are multiple token assets in the main chain, and you need to pay attention to distinguishing token symbols when entering the account.

4.3 Conclusion

Audit result: PASS

Audit No. : BCA002008250001

Audit date: August 25, 2020

Audit team: SlowMist security team

Summary conclusion: After correction, all problems found have been fixed and the above risks have been eliminated by Haven Protocol. Comprehensive assessed, Haven Protocolhas no risks above already.

5. Statement

SlowMist issues this report with reference to the facts that have occurred or existed before the issuance of this report, and only assumes corresponding responsibility base on these.

For the facts that occurred or existed after the issuance, SlowMist is not able to judge the security status of this project, and is not responsible for them. The security audit analysis and other contents of this report are based on the documents and materials provided to SlowMist by the information provider till the date of the insurance this report (referred to as "provided information"). SlowMist assumes: The information provided is not missing, tampered with, deleted or concealed. If the information provided is missing, tampered with, deleted, concealed, or inconsistent with the actual situation, the SlowMist shall not be liable for any loss or adverse effect resulting therefrom. SlowMist only conducts the agreed security audit on the security situation of the project and issues this report. SlowMist is not responsible for the background and other conditions of the project.

Official Website
www.slowmist.com

E-mail
team@slowmist.com

Twitter
@SlowMist_Team

Github
https://github.com/slowmist