DeHacker

Code Security Assessment

MEGATON FINANCE

August 27th, 2024

Contents

CONTENTS 1
SUMMARY 7
ISSUE CATEGORIES 8
OVERVIEW 9
PROJECT SUMMARY 9
VULNERABILITY SUMMARY 9
AUDIT SCOPE 10
FINDINGS 11
CRITICAL 17
LPM-01 : All funds can be stolen via forged op::transfer_notification to lp-minter 17
DESCRIPTION 17
RECOMMENDATION 17
MAJOR 18
LPM-02 : lp-minter always rejects op::transfer 18
DESCRIPTION 18
RECOMMENDATION 18
MAJOR 19
LPM-03 : Argument order is incorrect in save_data() 19
DESCRIPTION 19
RECOMMENDATION 19
MAJOR 20
LPM-04 : handle_provide_wallet_address() returns incorrect address 20
DESCRIPTION 20
RECOMMENDATION 20
MAJOR 21
LPM-05 : min_amount storage field is shadowed and overwritten by incoming argument in lp-minter::handle_transfer_notification() 21
DESCRIPTION 21
RECOMMENDATION 21
MAJOR 22
LPW-01 : lp-wallet doesn't guarantee pending_balance consistency 22
DESCRIPTION 22
RECOMMENDATION 23
MAJOR 24
LPW-02 : Sending op::init_pending_balance to lp-wallet wipes the deposits 24
DESCRIPTION 24
RECOMMENDATION 24
MEDIUM 25
JET-01 : jetton-minter::op::mint allows to send invalid messages 25
DESCRIPTION 25
RECOMMENDATION 26
MEDIUM 27
LPM-06 : update_mining_index() can ignore next_mining_rate_cell 27
DESCRIPTION 27
RECOMMENDATION 27

Contents

MEDIUM 28
LPM-07 : Wrong response_address used for op::burn message in lp-minter::handle_transfer() 28
DESCRIPTION 28
RECOMMENDATION 28
MEDIUM 29
LPM-08 : msg_value is not controlled at lp-minter on op::claim 29
DESCRIPTION 29
RECOMMENDATION 29
MEDIUM 30
LPM-09 : msg_value is not controlled at lp-minter on op::check_mintable_notification 30
DESCRIPTION 30
RECOMMENDATION 31
MEDIUM 32
LPM-10 : Pending jettons can be returned if lp_minter is_stopped 32
DESCRIPTION 32
RECOMMENDATION 32
MEDIUM 33
LPW-03 : lp-wallet/lp-minter don't follow 33
DESCRIPTION 33
RECOMMENDATION 33
MEDIUM 34
ROT-01 : router::handle_change_lp_content() is never executed 34
DESCRIPTION 34
RECOMMENDATION 34
MEDIUM 35
ROU-01 : Wrong destination address used in case of rejected swap request 35
DESCRIPTION 35
RECOMMENDATION 35
MEDIUM 36
ROU-02 : router doesn't validate the sender_address on op::transfer_notification 36
DESCRIPTION 36
RECOMMENDATION 36
MEDIUM 37
ROU-03 : The swap payload from EOA is not properly validated in router::handle_transfer_notification() 37
DESCRIPTION 37
RECOMMENDATION 37

Contents

MINOR 38
ALL-01 : Bounced op::transfer message from governance_jetton_wallet_address is ignored in `allocator::handle_claim() 38
DESCRIPTION 38
RECOMMENDATION 38
MINOR 39
AMM-01 : end_parse() Is Missing 39
DESCRIPTION 39
RECOMMENDATION 39
MINOR **40**
CON-01 : Pull-Over-Push Pattern is not used in admin changing 40
DESCRIPTION 40
RECOMMENDATION 40
MINOR **41**
CON-02 : Token data is not following 41
DESCRIPTION 41
RECOMMENDATION 41
MINOR **42**
JEO-01 : msg_value is not controlled at jetton-minter on op::mint 42
DESCRIPTION 42
RECOMMENDATION 42
MINOR **43**
LPM-11 : parse_std_addr() can be used to parse address 43
DESCRIPTION 43
RECOMMENDATION 43
MINOR **44**
LPM-12 : msg_value is not controlled at router on op::create_pool 44
DESCRIPTION 44
RECOMMENDATION 44
MINOR **45**
LPM-13 : mined and current_index calculation can be simplified 45
DESCRIPTION 45
RECOMMENDATION **45**
MINOR 46
LPM-14 : lp-minter::handle_burn() doesn't call force_chain() 46
DESCRIPTION 46
RECOMMENDATION 46

Contents

MINOR 47
LPM-15 : msg_value is not controlled at lp-minter on op::burn 47
DESCRIPTION 47
RECOMMENDATION 47
MINOR 48
LPM-16 : lp-minter sends op::transfer to jettonA_wallet_address in non-bounceable mode 48
DESCRIPTION 48
RECOMMENDATION 48
MINOR 49
KHI-06 | Unsafe handle of op::transfer_notification 49
DESCRIPTION 49
RECOMMENDATION 49
MINOR 50
LPM-17 : Gas management in lp-minter::handle_transfer() is inconsistent 50
DESCRIPTION 50
RECOMMENDATION 50
MINOR 51
LPM-18 : to_jetton_address is not checked in lp-minter::handle_transfer_notification() 51
DESCRIPTION 51
RECOMMENDATION 51
MINOR 52
LPM-19 : lp-minter silently accepts incoming LP transfers 52
DESCRIPTION 52
RECOMMENDATION 52
MINOR 53
LPM-20 : op::claim event emitted in lp-minter::handle_change_lp_mining_rate() 53
DESCRIPTION 53
RECOMMENDATION 53
MINOR 54
LPM-21 : min_amount is not respected by lp-minter::handle_mintable_notification() 54
DESCRIPTION 54
RECOMMENDATION 54
MINOR 55
LPM-22 : lp-minter accepts incoming transfers of unrecognized jettons 55
DESCRIPTION 55
RECOMMENDATION 55

Contents

MINOR 55
LPW-04 : Wrong fwd_count calculation 55
DESCRIPTION 55
RECOMMENDATION 55
MINOR 56
LPW-05 : jetton_address is not validated in lp-wallet::check_mintable() 56
DESCRIPTION 56
RECOMMENDATION 56
MINOR 57
LPW-06 : lp-wallet::on_bounce() is redundant 57
DESCRIPTION 57
RECOMMENDATION 57
MINOR 58
ROU-04 : router allows op::pool_created from pool_creator_address 58
DESCRIPTION 58
RECOMMENDATION 58
MINOR 59
ROU-05 : router::handle_change_lp_mining_rate() gas consumption is inconsistent 59
DESCRIPTION 59
RECOMMENDATION 60
MINOR 61
ROU-06 : jettonA_address/jettonB_address can be arbitrary, irrelevant to real jettons 61
DESCRIPTION 61
RECOMMENDATION 61
MINOR 62
UTI-01 : mined() can be simplified 62
DESCRIPTION 62
RECOMMENDATION 62
INFORMATIONAL 64
CON-03 : Misleading comments 64
DESCRIPTION 64
RECOMMENDATION 65
INFORMATIONAL 66
IMP-01 : Unused code 66
DESCRIPTION 66
RECOMMENDATION 66

Contents

INFORMATIONAL 67
LPM-23 : update_mining_index() can be refactored 67
DESCRIPTION 67
RECOMMENDATION 67
INFORMATIONAL 68
LPM-24 : Usage of Magic Numbers 68
DESCRIPTION 68
RECOMMENDATION 68
INFORMATIONAL 69
LPM-25 : in_msg_body is unused in lp-minter::handle_claim() 69
DESCRIPTION 69
RECOMMENDATION 69
INFORMATIONAL 70
LPM-26 : op::change_router can't be handled properly by lp-minter 70
DESCRIPTION 70
RECOMMENDATION 70
INFORMATIONAL 71
OPC-01 : Response messages op don't have high-order bit set 71
DESCRIPTION 71
RECOMMENDATION 71
INFORMATIONAL 72
ROU-07 : Argument names of router::get_lp_address() are misleading 72
DESCRIPTION 72
RECOMMENDATION 72
INFORMATIONAL 73
ROU-08 : either_forward_payload variable is unused 73
DESCRIPTION 73
RECOMMENDATION 73
INFORMATIONAL 74
UTI-02 : calculate_jetton_wallet_address() can be replaced with calculate_contract_address() 74
DESCRIPTION 74
RECOMMENDATION 74
INFORMATIONAL 75
UTI-03 : Long and complicated message building statements can be formatted 75
DESCRIPTION 75
RECOMMENDATION 75
INFORMATIONAL 76
UTI-04 : calculate_jetton_minter_address() is unused and dangerous 76
DESCRIPTION 76
RECOMMENDATION 76
DISCLAIMER 77
APPENDIX 78
ABOUT 79

Summary

DeHacker's objective was to evaluate the repository for security-related issues, code quality, and adherence to specification and best prac tices. Possible issues we looked for included (but are not limited to):

- Transaction-ordering dependence
- Timestamp dependence
- Mishandled exceptions and call stack limits
- Unsafe external calls
- Integer overflow/underflow
- Number rounding errors
- Reentrancy and cross-function vulnerabilities
- Denial of service/logical oversights
- Access control
- Centralization of power
- Business logic contradicting the specification
- Code clones, functionality duplication
- Gas usage
- Arbitrary token minting

Issue Categories

Every issue in this report was assigned a severity level from the following:

Critical severity issues

A vulnerability that can disrupt the contract functioning in a number of scenarios or creates a risk that the contract may be broken.

Major severity issues

A vulnerability that affects the desired outcome when using a contract or provides the opportunity to use a contract in an unintended way.

Medium severity issues

A vulnerability that could affect the desired outcome of executing the contract in a specific scenario.

Minor severity issues

A vulnerability that does not have a significant impact on possible scenarios for the use of the contract and is probably subjective.

Informational

A vulnerability that has informational character but is not affecting any of the code.

Overview

Project Summary

Project Name	MEGATON FINANCE
Platform	TON
Website	megaton.fi/
Type	DeFi
Language	FunC
Codebase	update e0c69d1f942a5ec4c9190af572f5cbdcee9e5aec

Vulnerability Summary

Vulnerability Level	Total	Mitigated	Declined	Acknowledged	Partially Resolved	Resolved
Critical	1	0	0	0	0	1
Major	6	0	0	0	0	6
Medium	11	0	0	1	0	10
Minor	24	0	0	5	0	19
Informational	12	0	0	4	0	8
Discussion	0	0	0	0	0	0

Audit scope

ID	File	SHA256 Checksum
LPW	contracts/amm/lp-wallet.fc	189bedab2e0072e2f3694d7e731d7825323a56e9edf598be20259372be470a98
OPC	contracts/imports/op-codes.fc	2db5f4e6f0087b8c0ebb63faf4398cd07f78d83cf1685a8b4a50cbb788f0eaaa
JEO	contracts/jetton-wallet.fc	d0b14a28428efc117f389936d221c4e2cf6fe3547206ed494a8ecb931ee6a834

Findings

ID	Title	Severity	Status
LPM-01	All Funds Can Be Stolen Via Forgedop::transfer_notification To lp-minter	Critical	Resolved
LPM-02	lp-minter Always Rejects op::transfer	Major	Resolved
LPM-03	Argument Order Is Incorrect In save_data()	Major	Resolved
LPM-04	handle_provide_wallet_address() ReturnsIncorrect Address	Major	Resolved
LPM-05	min_amount Storage Field Is Shadowed AndOverwritten By Incoming Argument In lp-minter::handle_transfer_notification()	Major	Resolved
LPW-01	lp-wallet Doesn't Guaranteepending_balance Consistency	Major	Resolved
LPW-02	Sending op::init_pending_balance To lp-wallet Wipes The Deposits	Major	Resolved
JET-01	jetton-minter::op::mint Allows To SendInvalid Messages	Medium	Resolved
LPM-06	update_mining_index() Can Ignorenext_mining_rate_cell	Medium	Resolved
LPM-07	Wrong response_address Used For op::burnMessage In lp-minter::handle_transfer()	Medium	Resolved

Findings

ID	Title	Severity	Status
LPM-08	msg_value Is Not Controlled At lp-minter Onop::claim	Medium	Resolved
LPM-09	msg_value Is Not Controlled At lp-minter Onop::check_mintable_notification	Medium	Resolved
LPM-10	Pending Jettons Can Be Returned If lp_minteris_stopped	Medium	Resolved
LPW-03	lp-wallet / lp-minter Don't Follow TEP-74Standard	Medium	Resolved
ROT-01	router::handle_change_lp_content() Is NeverExecuted	Medium	Resolved
ROU-01	Wrong Destination Address Used In Case OfRejected Swap Request	Medium	Resolved
ROU-02	router Doesn't Validate The sender_addressOn op::transfer_notification	Medium	Resolved
ROU-03	The Swap Payload From EOA Is Not ProperlyValidated Inrouter::handle_transfer_notification()	Medium	Acknowledged
ALL-01	Bounced op::transfer Message Fromgovernance_jetton_wallet_address Is IgnoredIn allocator::handle_claim()	Minor	Acknowledged

Findings

ID	Title	Severity	Status
AMM-01	end_parse() Is Missing	Minor	Resolved
CON-01	Pull-Over-Push Pattern Is Not Used In AdminChanging	Minor	Resolved
CON-02	Token Data Is Not Following TEP-64 Standard	Minor	Acknowledged
JEO-01	msg_value Is Not Controlled At jetton-minterOn op::mint	Minor	Resolved
LPM-11	parse_std_addr() Can Be Used To ParseAddress	Minor	Resolved
LPM-12	msg_value Is Not Controlled At router Onop::create_pool	Minor	Resolved
LPM-13	mined And current_index Calculation Can BeSimplified	Minor	Resolved
LPM-14	lp-minter::handle_burn() Doesn't Callforce_chain()	Minor	Resolved
LPM-15	msg_value Is Not Controlled At lp-minter Onop::burn	Minor	Resolved
LPM-16	lp-minter Sends op::transfer TojettonA_wallet_address In Non-BounceableMode	Minor	Resolved
LPM-17	Gas Management In lp-minter::handle_transfer() Is Inconsistent	Minor	Resolved

Findings

ID	Title	Severity	Status
LPM-18	to_jetton_address Is Not Checked In lp-minter::handle_transfer_notification()	Minor	Acknowledged
LPM-19	lp-minter Silently Accepts Incoming LPTransfers	Minor	Resolved
LPM-20	op::claim Event Emitted In lp-minter::handle_change_lp_mining_rate()	Minor	Resolved
LPM-21	min_amount Is Not Respected By lp-minter::handle_mintable_notification()	Minor	Resolved
LPM-22	lp-minter Accepts Incoming Transfers OfUnrecognized Jettons	Minor	Resolved
LPW-04	Wrong fwd_count Calculation	Minor	Resolved
LPW-05	jetton_address Is Not Validated In lp-wallet::check_mintable()	Minor	Resolved
LPW-06	lp-wallet::on_bounce() Is Redundant	Minor	Resolved
ROU-04	router Allows op::pool_created Frompool_creator_address	Minor	Acknowledged
ROU-05	router::handle_change_lp_mining_rate() GasConsumption Is Inconsistent	Minor	Resolved

Findings

ID	Title	Severity	Status
ROU-06	jettonA_address / jettonB_address Can BeArbitrary, Irrelevant To Real Jettons	Minor	Acknowledged
UTI-01	mined() Can Be Simplified	Minor	Resolved
CON-03	Misleading Comments	Informational	Resolved
IMP-01	Unused Code	Informational	Resolved
LPM-23	update_mining_index() Can Be Refactored	Informational	Acknowledged
LPM-24	Usage Of Magic Numbers	Informational	Acknowledged
LPM-25	in_msg_body Is Unused In lp-minter::handle_claim()	Informational	Resolved
LPM-26	op::change_router Can't Be Handled ProperlyBy lp-minter	Informational	Acknowledged
OPC-01	Response Messages op Don't Have High-OrderBit Set	Informational	Resolved
ROU-07	Argument Names Ofrouter::get_lp_address() Are Misleading	Informational	Resolved
ROU-08	either_forward_payload Variable Is Unused	Informational	Resolved
UTI-02	calculate_jetton_wallet_address() Can BeReplaced With calculate_contract_address()	Informational	Resolved

Findings

ID	Title	Severity	Status
UTI-03	Long And Complicated Message BuildingStatements Can Be Formatted	Informational	Acknowledged
UTI-04	calculate_jetton_minter_address() Is UnusedAnd Dangerous	Informational	Resolved

CRITICAL

LPM-01|ALL FUNDS CAN BE STOLEN VIA FORGEDop::transfer_notification TO lp-minter

Issue	Severity	Location	Status
Control Flow	Critical	contracts/amm/lp-minter.fc (update6): 842~847	Resolved

Description

lp-minter::handle_transfer_notification() is supposed to handle op::transfer_notification messages from lp-minter wallets. Those messages carry the data required to perform adding liquidity or swapping operations. However, suchmessages can be sent by an externally owned account with forged arguments. This allows the extraction of all the fundsfrom the lp-minter wallets.
Also, the check

```
840 throw_unless(75, msg_value > const::jetton_transfer_gas_consumption + fwd_fee);
```

is performed, however, lp_forward_router_gas_consumption (0.1 TON) is forwarded to "router". op::transfer will notbe processed due to not enough gas.

Recommendation

We recommend sending messages only back to sender_address instead of real wallet address to avoid spoofing. Werecommend fixing the gas requirements.

MAJOR

LPM-02|lp-minter ALWAYS REJECTS op::transfer

Issue	Severity	Location	Status
Volatile Code	Major	contracts/amm/lp-minter.fc (base): 1150~1153	Resolved

Description

```
1150 if (op == op::transfer) {
1151 handle_transfer(query_id, in_msg_body, sender_address);
1152 }
```

There is no return () in this case, so throw(0xffff) will be executed discarding all the uncommitted changes.

Recommendation

We recommend adding return ();

MAJOR

LPM-03|ARGUMENT ORDER IS INCORRECT IN save_data()

Issue	Severity	Location	Status
Logical Issue	Major	contracts/amm/lp-minter.fc (base): 379	Resolved

Description

```
379 save_data(total_supply + lp_amount, min_amount, swap_fee, ...
```

save_data() accepts swap_fee as the second argument, min_amount as the third.

Recommendation

We recommend fixing the argument order.

MAJOR

LPM-04|handle_provide_wallet_address() RETURNS INCORRECTADDRESS

Issue	Severity	Location	Status
Logical Issue	Major	contracts/amm/lp-minter.fc (base): 379	Resolved

Description

```
583 msg = msg.store_slice(calculate_user_jetton_wallet_address(owner_address, my_address(), lp_wallet_code));
```

handle_provide_wallet_address() is supposed to provide lp-wallet address. However, an incorrect jetton-walletaddress is returned.

Recommendation

We recommend fixing the code this way:

```
583 msg = msg.store_slice(calculate_user_lp_wallet_address(sender_address, my_address(), lp_wallet_code, jettonA_address, jettonB_address));
```

MAJOR

LPM-05|min_amount STORAGE FIELD IS SHADOWED ANDOVERWRITTEN BY INCOMING ARGUMENT IN lp-minter::handle_transfer_notification()

Issue	Severity	Location	Status
Volatile Code	Major	contracts/amm/lp-minter.fc (base): 733~734, 832~833	Resolved

Description

lp-minter has min_amount storage field with a minimal allowed LP amount for each account. The functionhandle_transfer_notification() gets the min_amount argument from in_msg_body practically shadowing the storagefield. Moreover, the shadowing value is saved to the storage instead. This allows the end user to set any min_amount forany lp-minter at will.

Recommendation

We recommend renaming the argument variable to avoid shadowing.

MAJOR

LPW-01 | lp-wallet DOESN'T GUARANTEE pending_balanceCONSISTENCY

Issue	Severity	Location	Status
Logical Issue	Major	contracts/amm/lp-wallet.fc (base): 62~63	Resolved

Description

lp-wallet allows minting via op::check_mintable and canceling via op::check_pending_jetton at the same time.This leads to double-spending.

Minting LP is currently working this way:

1. Sender deposits jettonA via sending op::transfer to lp-minter walletA.
2. lp-minter walletA sends op::transfer_notification to lp-minter .
3. lp-minter sends op::check_mintable to sender lp-wallet .
4. sender lp-wallet sends op::check_mintable_notification to lp-minter if both pending amounts arepositive
5. lp-minter sends op::init_pending_balance to sender lp-wallet . Pending amounts are zeroed.
6. lp-minter sends op::internal_transfer to sender lp-wallet . Sender's LP balance is increased.

However, between steps 3 and 5, the sender lp-wallet can get and execute another op::check_pending_jetton andextract both pending jetton deposits.

According to Message delivery guarantees we can't be sure which message, 3 or 4 will be delivered first.

Description

The attack scenario:

1. Sender deposits jettonB. sender lp-wallet::jettonB_pending_balance is updated.
2. Sender deposits jettonA. sender lp-wallet op::check_mintable is executed. Since both pending balances arepositive, op::check_mintable_notification is sent to lp-minter .
3. Sender sends op::check_pending_jetton to sender lp-wallet .
4. sender lp-wallet sends 2 op::check_pending_jetton_notification to lp-minter . Pending balances arezeroed.
5. lp-minter returns Sender's deposits to their wallets.
6. op::check_mintable_notification is delivered to lp-minter . A new LP is minted on Sender's wallet, their zeropending balances are zeroed again.

As a result, Sender extracted deposited jettons in step 5 and minted the corresponding LP in step 6.

Recommendation

We recommend dropping of lp-wallet support and managing pending balances in lp-minter directly. We recommenddecreasing the balances before transaction action phase.

MAJOR

LPW-02| SENDING op::init_pending_balance TO lp-walletWIPES THE DEPOSITS

Issue	Severity	Location	Status
Control Flow	Major	contracts/amm/lp-wallet.fc (base): 299~300	Resolved

Description

op::init_pending_balance in lp-wallet zeroes the user's jetton pending balances. It can be sent directly by the user oras part of op::check_mintable flow.

Sending it directly wipes users' jetton deposits and makes lp-minter pending balances inconsistent.

in_msg_body argument is not used by init_pending_balance()

Recommendation

We recommend allowing op::init_pending_balance to be processed only if received from lp-minter . We recommenddropping of unused arguments. We recommend merging the handler with lp-wallet::receive_tokens() .

MEDIUM

JET-01| jetton-minter::op::mint ALLOWS TO SEND INVALIDMESSAGES

Issue	Severity	Location	Status
Logical Issue	Medium	contracts/jetton-minter.fc (base): 76~77	Resolved

Description

```
71 if (op == op::mint) {
72 throw_unless(73, equal_slices(sender_address, minter_address));
73 slice to_address = in_msg_body~load_msg_addr();
74 cell master_msg = in_msg_body~load_ref();
75 slice master_msg_cs = master_msg.begin_parse();
76 master_msg_cs~skip_bits(32 + 64); ;; op + query_id
77 int jetton_amount = master_msg_cs~load_coins();
78
79 mint_tokens(msg_value, to_address, jetton_wallet_code, master_msg);
```

jetton-minter::op::mint is supposed to allow minter_address to mint new jettons to to_address via sending ofop::internal_transfer message. However, master_msg is not validated:

- any op can be used
- the op::internal_transfer message format is not validated
- the forward_ton_amount argument is not respected, min_tons_for_storage is not provided
- msg_value is not controlled, CARRY_REMAINING_GAS mode is not used
- the bounced message is not handled, total_supply is not decreased back in case of failure

Recommendation

We recommend checking all the required arguments of op::internal_transfer message, we recommend handling ofbounced message.

MEDIUM

LPM-06| update_mining_index() CAN IGNOREnext_mining_rate_cell

Issue	Severity	Location	Status
Logical Issue	Medium	contracts/amm/lp-minter.fc (base): 151~153	Resolved

Description

If next_mining_rate_cell was set, the function update_mining_index() is supposed to calculate first_mined for thefirst period with the old mining rate and second_mined for the second period with the updated mining rate.

However, if first_mined <= last_mined , the second_mined will not even be checked, despite the fact it can be biggerthan last_mined . This can lead to loss of the reward.

Recommedation

We recommend checking if second_mined is bigger than last_mined even if first_mined is not

MEDIUM

LPM-07| WRONG response_address USED FOR op::burnMESSAGE IN lp-minter::handle_transfer()

Issue	Severity	Location	Status
Inconsistency	Medium	contracts/amm/lp-minter.fc (base): 683~684	Resolved

Description

lp-minter::handle_transfer() sends a op::burn message to ex-lp-owner-wallet. The sender_address is specified asa response_address argument, however, the sender_address is ex-lp-owner-wallet, not the ex-lp-owner. It is reasonableto send op::excesses to the originator of the transaction

Recommedation

We recommend using of from_address as a response_address to return unused fees.

MEDIUM

LPM-08| msg_value IS NOT CONTROLLED AT lp-minter ONop::claim

Issue	Severity	Location	Status
Inconsistency	Medium	contracts/amm/lp-minter.fc (base): 254~255	Resolved

Description

lp-minter::handle_claim() doesn't check that msg_value is enough.

Claiming reward works this way:

1. user_info_member sends op::claim to lp-minter . msg_value is not checked.
2. lp-minter sends op::claim to router , forwards 0.04 TON, and pays for processing and forwarding.
3. router sends op::transfer to governance_jetton_wallet_address , forwards 0.04 TON, and pays forprocessing and forwarding.
4. governance_jetton_wallet_address returns excesses to user_info_member .

As a result, if router has zero balance, the op::transfer will not be sent due to out-of-gas, and the reward will be lost. Ifrouter and lp-minter have enough gas, up to 0.04 TON can be stolen from lp-minter per each op::claim .

Recommedation

We recommend explicitly checking in minter::handle_claim() that msg_value is at least 2 * const::gas_consumption+ 2 * fwd_fee + 0.04 and forwarding to router const::gas_consumption + fwd_fee + 0.04 . We recommend toCARRY_REMAINING_GAS in router::handle_claim() .

MINOR

LPM-09| msg_value IS NOT CONTROLLED AT lp-minter ONop::check_mintable_notification

Issue	Severity	Location	Status
Inconsistency	Medium	contracts/amm/lp-minter.fc (base): 399~400	Resolved

Description

lp-minter::handle_mintable_notification() doesn't check that msg_value is enough. It can lead to funds draining orincomplete execution.

Handling of op::check_mintable_notification at lp-minter works this way:

1. sender deposits jettons to lp-minter and uses some forward_ton_amount
2. lp-minter gets op::transfer_notification . msg_value is not checked.
3. lp-minter sends op::check_mintable to sender lp-wallet , and forwards all remaining gas.
4. sender lp-wallet sends op::check_mintable_notification to lp-minter , and forwards all remaining gas.
5. lp-minter during the processing of op::check_mintable_notification sends 0.03 TON withop::init_pending_balance , 0.05 TON during the return of unused jettons (maybe twice), 0.2 TON to lp-walletduring minting.

As a result, if not enough forward_ton_amount , the deposit will not be handled properly leading to inconsistent pendingbalances. If lp-minter has enough balance, up to 0.25 TON can be stolen per each deposit.

Recommendation

We recommend:

1. explicitly checking in lp-minter op::transfer_notification handler, that msg_value is enough to finish theworkflow
2. trying to return jettons if msg_value is not enough or the payload is invalid
3. avoiding failure in the action phase
4. returning excesses in op::init_pending_balance handler or removing this message completely.

MEDIUM

LPM-10|PENDING JETTONS CAN BE RETURNED IF lp_minteris_stopped

Issue	Severity	Location	Status
Inconsistency	Medium	contracts/amm/lp-minter.fc (update6): 1227~1228	Resolved

Description

lp-minter::handle_pending_jetton() allows the user to return jettons deposited to add liquidity. The operation will fail ifis_stopped is set. However, handle_pending_jetton_notification() can be triggered via a directop::check_pending_jetton message to lp-wallet . This essentially allows skipping the check.

Also, router processes op::claim requests even if is_stopped . It is unclear if that behavior is intended

Recommendation

We recommend disallowing processing of op::check_pending_jetton messages in lp-wallet , if they are receiveddirectly from the wallet owner, or ignoring of is_stopped flag in lp-minter::handle_pending_jetton() . We recommendclarifying the intended behavior of router::handle_claim() in the case of is_stopped via code comments.

MEDIUM

LPW-03|lp-wallet / lp-minter DON'T FOLLOW TEP-74STANDARD

Issue	Severity	Location	Status
Inconsistency	Medium	contracts/amm/lp-wallet.fc (base): 92~93	Resolved

Description

According to TEP-74:

- op::transfer uses the destination address argument after amount . But lp-wallet::send_tokens() usesowner_address ("from") instead. At the same time, transfer#0f8a7ea5 tag is preserved.
- op::burn is rejected if received not from the owner. But lp-wallet::burn_tokens() accepts the message onlyfrom lp_minter_address .

lp-wallet interaction diagram:

lp-wallet doesn't allow direct transfers between wallets, all the state changes are controlled by lp-minter . The infinitesharding paradigm (when the transactions are processed independently on different accounts) can't be used in this case.The balance is mirrored between lp-wallet and lp-minter .The possible bounced messages between lp-wallet and lp-minter are not handled by both contracts.

Recommendation

We recommend dropping the lp-wallet and using lp-minter only. We recommend changing the op::transfer tag orthe arguments layout.

MEDIUM

ROT-01 | router::handle_change_lp_content() IS NEVEREXECUTED

Issue	Severity	Location	Status
Inconsistency	Medium	contracts/amm/router.fc (update6): 508~509	Resolved

Description

router::handle_change_lp_content() is supposed to change jetton content for the specific lp_address . However, thefunction is inaccessible, recv_internal() doesn't handle the corresponding message.

Recommendation

We recommend updating the router::recv_internal() .

MEDIUM

ROU-01 | WRONG DESTINATION ADDRESS USED IN CASE OFREJECTED SWAP REQUEST

Issue	Severity	Location	Status
Logical Issue	Medium	contracts/amm/router.fc (base): 301~302	Resolved

Description

The swapper deposits from_jettons and provides the payload (from_jetton, to_jetton, destination,min_amount) . In case the payload is invalid (too short), the jettons are returned to swapper from_wallet address.However, in case minter finds min_amount criteria is not satisfied, the jettons are "returned" to destination_walletaddress. It is unexpected by the swapper .

Recommendation

We recommend always returning from_wallet jettons to swapper from_wallet .

MEDIUM

ROU-02|router DOESN'T VALIDATE THE sender_address ONop::transfer_notification

Issue	Severity	Location	Status
Control Flow	Medium	contracts/amm/router.fc (base): 287~289, 320~321, 343~345, 349~350, 378~379	Resolved

Description

router::handle_transfer_notification() gets the from_address from the payload and treats it as trustworthy.

The attacker can send to the router the message { op::transfer_notification, query_id: any, jetton_amount:any, from_address: real lp-minter address, 0, (destination: self address) } . The router checks the lp-minter address is known and sends the op::transfer message back with 0.05 TON in non-bounceable mode. This candrain the router balance.

The same problem is reproduced if from_address is not lp-minter or the payload is incorrect. The router sends 0.3TONs back to attacker if the payload is valid.

Recommendation

We recommend sending op::transfer in CARRY_REMAINING_GAS mode with 0 TONs attached and bounceable flag set.

MEDIUM

ROU-03|THE SWAP PAYLOAD FROM EOA IS NOT PROPERLY VALIDATED IN router::handle_transfer_notification()

Issue	Severity	Location	Status
Volatile Code	Medium	contracts/amm/router.fc (base): 313~314, 337~341	Acknowledged

Description

When EOA sends the swap request to router via jettons depositing, the expected payload format is {from_jetton_address, to_jetton_address, destination, min_amount } . However, if the payload can't be parsed, theexecution terminates, and the jettons and TONs are not returned.

The function checks if slice_bits(swap_slice) <= 267 * 3 , but that doesn't guarantee the success of parsing.min_amount doesn't fit 267 * 3 bits payload.

Recommendation

We recommend using of TRY primitive and returning jettons/TONs in case of failure.

MINOR

ALL-01| BOUNCED op::transfer MESSAGE FROMgovernance_jetton_wallet_address IS IGNORED INallocator::handle_claim()

Issue	Severity	Location	Status
Volatile Code	Minor	contracts/amm/router.fc (base): 313~314, 337~341	Acknowledged

Description

allocator::handle_claim() sends an internal op::transfer message to governance_jetton_wallet_address inbounceable mode. In case this message can't be processed, for example, if transferring is currently paused, it will bebounced back and ignored by allocator . last_mined state field will not be decreased back.

Recommendation

We recommend catching the bounced messages and reverting the corresponding changes.

MINOR

AMM-01| end_parse() IS MISSING

Issue	Severity	Location	Status
Volatile Code	Minor	contracts/amm/allocator.fc (base): 17~18; contracts/amm/lp-wallet.fc (base): 29~30, 33~34	Resolved

Description

end_parse(slice s) ensures that no more data is available in s . This allows checking of message format correctness.

Recommendation

We recommend using end_parse() wherever possible to ensure the correct message format.

MINOR

CON-01| PULL-OVER-PUSH PATTERN IS NOT USED IN ADMINCHANGING

Issue	Severity	Location	Status
Volatile Code	Minor	contracts/amm/allocator.fc (base): 105~106, 114~115; contracts/amm/lp-minter.fc (base): 1084~1085; contracts/amm/router.fc (base): 409~410; contracts/jetton-minter.fc (base): 137~138, 144~145	Resolved

Description

The functions handle_change_claim_admin() / handle_change_admin() override the previously setclaim_admin_address / admin_address with the new value without guaranteeing they are able to actuate transactions on-chain.

Recommendation

We recommend using of the pull-over-push pattern whereby a new admin is first proposed and consequently needs toaccept the admin status ensuring that the account can actuate transactions on-chain.

MINOR

CON-02| TOKEN DATA IS NOT FOLLOWING TEP-64 STANDARD

Issue	Severity	Location	Status
Volatile Code	Minor	contracts/amm/lp-minter.fc (base): 1104~1105; contracts/amm/router.fc (base): 445~446; contracts/jetton-minter.fc (base): 151~152	Resolved

Description

TEP-64 standard describes the Token Data Standard. However, jetton-minter , lp-minter contracts don't validate thedata in op::change_content . router doesn't validate the data in handle_change_lp_default_content() .

Changing the Token Data (decimals, name, symbol) is not recommended

Recommendation

We recommend verifying that new token data follows the standard.

MINOR

JEO-01| msg_value IS NOT CONTROLLED AT jetton-minter ONop::mint

Issue	Severity	Location	Status
Volatile Code	Minor	contracts/amm/lp-minter.fc (base): 1104~1105; contracts/amm/router.fc (base): 445~446; contracts/jetton-minter.fc (base): 151~152	Resolved

Description

jetton-minter::mint_tokens() doesn't check, that msg_value is enough. As a result, op::internal_transfer can besuccessfully sent but not properly processed by jetton-minter due to out-of-gas exception. The bounced messaged willnot be created in this case, leaving jetton_minter::total_supply in inconsistent state.

Recommendation

We recommend explicitly checking that enough gas provided by the caller.

MINOR

LPM-11|parse_std_addr() CAN BE USED TO PARSE ADDRESS

Issue	Severity	Location	Status
Volatile Code	Minor	contracts/amm/lp-minter.fc (base): 406~409	Resolved

Description

```
406 slice tmp_addr = to_address;
407 tmp_addr~skip_bits(11);
408 int addr_hash = tmp_addr~load_uint(256);
```

The way the address is parsed heavily relies on internal address representations. This makes the code volatile. Not alllocations are mentioned.

Recommedation

We recommend using (int wc, int hash) = parse_std_addr(addr) .

MINOR

LPM-12|msg_value IS NOT CONTROLLED AT router ONop::create_pool

Issue	Severity	Location	Status
Inconsistency	Minor	contracts/amm/lp-minter.fc (base): 111~112	Resolved

Description

router::handle_create_pool() doesn't check that msg_value is enough.

Pool creation works this way:

1. pool_creator_address sends op::create_pool to router . msg_value is not checked.
2. router sends op::create_pool to lp-minter , forwards 0.1 TON, and pays for processing, forwarding, anddeploying.
3. lp-minter sends op::pool_created to router , keeps 0.03 TON for storage, pays for processing andforwarding, and sends all the rest.
4. router pays for processing and keeps the change.

As a result, it is unclear to the caller, what is the expected msg_value

Recommendation

We recommend explicitly check.ing in handle_create_pool() that the contract balance is bigger thanconst::min_tons_for_storage + const::gas_consumption + fwd_fee + 0.1 , or checking the msg_value andCARRY_REMAINING_GAS in router::create_pool()

MINOR

LPM-13|mined AND current_index CALCULATION CAN BESIMPLIFIED

Issue	Severity	Location	Status
Coding Style	Minor	contracts/amm/lp-minter.fc (base): 144~150, 183~189	Resolved

Description

```
183 if ((current_mining_rate != 0) & (const::total_mining_rate != 0)) {
184 this_mined = current_mining_rate * (current_mined - last_mined) /
const::total_mining_rate;
185 }
186 if ((this_mined != 0) & (total_supply != 0)) {
187 current_index = current_index + (this_mined * 1000000000000000000) /
total_supply; ;; 10^18
188 }
```

The check (current_mining_rate != 0) is redundant, since in this case this_mined will still be zero.The check (const::total_mining_rate != 0) is redundant, since the constant is not zero. If the constant can be zero, werecommend adding this check to lines 144, 154, or leaving the function immediately.The check (this_mined != 0) is redundant, since current_index is not changed in this case.muldiv() can be used to prevent potential overflows.

Recommendation

We recommend removing of redundant checks to simplify the code.

MINOR

LPM-14|lp-minter::handle_burn() DOESN'T CALL force_chain()

Issue	Severity	Location	Status
Volatile Code	Minor	contracts/amm/lp-minter.fc (base): 452~453	Resolved

Description

lp-minter::handle_burn() doesn't enforce the sender_address chain to be basechain . But user_info_dict isindexed by addr_hash only. Calling the function from another chain can lead to unexpected results.calculate_contract_address() enforces the address to be in workchain() . But calculate_*_state_init() functionsdo not.

Recommendation

We recommend enforcing the chain in recv_internal() , get_wallet_address() , handle_change_router() ,handle_change_admin() , and in other functions accepting addresses.

MINOR

LPM-15|msg_value IS NOT CONTROLLED AT lp-minter ONop::burn

Issue	Severity	Location	Status
Inconsistency	Minor	contracts/amm/lp-minter.fc (base): 559~560	Resolved

Description

lp-minter::handle_burn() doesn't check that msg_value is enough.
Burning works this way:
1. sender sends op::burn to lp-minter . msg_value is not checked, handle_burn() argument is unused.
2. lp-minter sends op::claim to router with 0.04 TON.
3. lp-minter sends op::burn to sender lp-wallet with 0.03 TON.
4. lp-minter sends 2 op::transfer to jetton wallets with 0.04 TON.5. All messages send excesses to the sender.

As a result, the sender can steal up to 0.15 TON from lp-minter per each op::burn message.

Recommendation

We recommend explicitly checking that enough gas provided by the caller. We recommend using CARRY_REMAINING_GASmode in the last send_raw_message() .

MINOR

LPM-16|lp-minter SENDS op::transfer TO jettonA_wallet_address IN NON-BOUNCEABLE MODE

Issue	Severity	Location	Status
Volatile Code	Minor	contracts/amm/lp-minter.fc (base): 330~331, 341~342, 361~362, 498~499, 517~518, 993, 1017	Resolved

Description

According to Guidelines, almost all internal messages sent between smart contracts should be bounceable. Then, if thedestination smart contract throws an unhandled exception while processing this message, the message will be "bounced"back carrying the remainder of the original value (minus all message transfer and gas fees).

lp-minter::handle_burn() , handle_pending_jetton_notification() , handle_mintable_notification() send non-bounceable messages to own wallets. Forwarded TONs will not be returned in case of exception.

Recommendation

We recommend sending all the messages in bounceable mode unless the destination is expected to keep the TONs.

MINOR

LPM-17|GAS MANAGEMENT IN lp-minter::handle_transfer() ISINCONSISTENT

Issue	Severity	Location	Status
Inconsistency	Minor	contracts/amm/lp-minter.fc (base): 709~710	Resolved

Description

lp-minter processes op::transfer this way:
1. sender lp-wallet checks that msg_value > forward_ton_amount + fwd_count * fwd_fee + 2 * 0.01 + 0.01+ 0.2 and sends op::transfer to lp-minter carrying all the value.
2. lp-minter::handle_transfer() sends op::burn to sender lp-wallet with 0.03 TONs attached, and paysforwarding fees.
3. lp-minter::handle_transfer() sends op::internal_transfer to destination lp-wallet with 0.03 +forward_ton_amount attached, pays forwarding fees.
4. lp-minter::handle_transfer() sends up to 2 op::claim to router with 0.04 TONs attached, and paysforwarding fees.

As a result, const::lp_transfer_gas_consumption (0.2 TON) is bigger than actually used. The excess is not returned toresponse_address . lp-minter will accumulate the value

Recommendation

We recommend carrying all the remaining gas to op::internal_transfer.

MINOR

LPM-18|to_jetton_address IS NOT CHECKED IN lp-minter::handle_transfer_notification()

Issue	Severity	Location	Status
Volatile Code	Minor	contracts/amm/lp-minter.fc (base): 736~737, 835~836	Acknowledged

Description

lp-minter::handle_transfer_notification() gets the payload (from_jetton_address, to_jetton_address,destination, min_amount) prepared by router . But to_jetton_address is not checked and is passed toemit_log_cell_ref() as is

Recommendation

We recommend checking that to_jetton_address == jettonB_address (or jettonA_address depending on the branch).

MINOR

LPM-19|lp-minter SILENTLY ACCEPTS INCOMING LP TRANSFERS

Issue	Severity	Location	Status
Volatile Code	Minor	contracts/amm/lp-minter.fc (base): 920~921835~836	Resolved

Description

lp-minter::handle_transfer_notification() silently accepts incoming LP transfers. The funds become locked

Recommendation

We recommend sending jettons back if they are not processed properly.

MINOR

LPM-20|op::claim EVENT EMITTED IN lp-minter::handle_change_lp_mining_rate()

Issue	Severity	Location	Status
Inconsistency	Minor	contracts/amm/lp-minter.fc (base): 1051	Resolved

Description

lp-minter::handle_change_lp_mining_rate() emits event with op::claim argument

Recommendation

We recommend using op::change_lp_mining_rate argument.

MINOR

LPM-21|min_amount IS NOT RESPECTED BY lp-minter::handle_mintable_notification()

Issue	Severity	Location	Status
Inconsistency	Minor	contracts/amm/lp-minter.fc (base): 320~321	Resolved

Description

min_amount is supposed to disallow the user to have a too small LP balance. However, the minted amount is not checkedin lp-minter::handle_mintable_notification()

Recommendation

We recommend not minting LP if the resulting user LP balance is less, than min_amount.

MINOR

LPM-22|lp-minter ACCEPTS INCOMING TRANSFERS OFUNRECOGNIZED JETTONS

Issue	Severity	Location	Status
Volatile Code	Minor	contracts/amm/lp-minter.fc (base): 923	Resolved

Description

lp-minter::handle_transfer_notification() accepts incoming transfers of unrecognized jettons. The funds becomelocked.Reverting the op::transfer_notification transaction will not return the funds.The transfer is treated as unrecognized if valid { from_jetton, to_jetton } payload was provided of known existing lp-minter , but the wrong jetton was actually sent to the router

Recommendation

We recommend sending the jettons back.

MINOR

LPW-04|WRONG fwd_count CALCULATION

Issue	Severity	Location	Status
Inconsistency	Minor	contracts/amm/lp-wallet.fc (base): 69~70	Resolved

Description

```
69 int fwd_count = forward_ton_amount ? 3 : 1;
70 throw_unless(709, msg_value >
71 forward_ton_amount +
72 ;; 5 messages: wal1->minter, minter->wal1, minter->wal2,wal2->owner, wal2-
>response
73 ;; but last one is optional (it is ok if it fails)
74 fwd_count * fwd_fee +
75 (2 * const::gas_consumption + const::min_tons_for_storage+
const::lp_transfer_gas_consumption));
```

As a result of lp-wallet::send_tokens() , 5 messages are generated: "wal1->minter, minter->wal1, minter->wal2, wal2->owner, wal2->response". The last one is optional. The message "wal2->owner" is not sent if forward_ton_amount == 0 .The expected fwd_count = forward_ton_amount ? 4 : 3 .

It is also expected that 4 message processing will be done. So, const::lp_transfer_gas_consumption is expected to beat least 2 * const::gas_consumption

Recommendation

We recommend updating the calculation of fwd_count .

MINOR

LPW-05|jetton_address IS NOT VALIDATED IN lp-wallet::check_mintable()

Issue	Severity	Location	Status
Volatile Code	Minor	jetton_address IS NOT VALIDATED IN lp-wallet::check_mintable()	Resolved

Description

lp-wallet::check_mintable() expects jetton_address argument to be either jettonA_address , orjettonB_address . However, that is not enforced.

Recommendation

We recommend ensuring the address is one of the expected.

MINOR

LPW-06|ip-wallet::on_bounce() IS REDUNDANT

Issue	Severity	Location	Status
Inconsistency	Minor	contracts/amm/lp-wallet.fc (base): 304~315	Resolved

Description

lp-wallet::on_bounce() processes op::internal_transfer bounced message. However, it is never sent by lp-wallet.

Recommendation

We recommend removing of unused code.

MINOR

ROU-04|router ALLOWS op::pool_created FROMpool_creator_address

Issue	Severity	Location	Status
Control Flow	Minor	contracts/amm/router.fc (base): 197~198	Acknowledged

Description

router handles op::pool_created not only from lp-minter but also from pool_creator_address . This allows forskipping several important checks:

1. pool_creator_address can provide jettonA_address / jettonB_address arguments in the wrong order.jetton_pair_to_lp will still be updated with the wrong address.
2. pool_creator_address can forget to deploy the lp-minter .
3. pool_count will be incremented after each message.
4. swap_fee can be fake, it will still be emitted

Recommendation

We recommend forbidding direct op::pool_created from pool_creator_address.

MINOR

ROU-05 | router::handle_change_lp_mining_rate() GASCONSUMPTION IS INCONSISTENT

Issue	Severity	Location	Status
Volatile Code	Minor	contracts/amm/router.fc (base): 454~455	Resolved

Description

router::handle_change_lp_mining_rate() checks that msg_value is at least const::change_mining_rate_router_gas_consumption + pool_count * const::change_mining_rate_lp_gas_consumption . That means, that each pool will have at least 0.1 TON forop::change_lp_mining_rate processing and router will have at least 1 TON for it.
However, the gas consumption of the router significantly depends on the pool_count :
1. The size of new_lp_mining_rate_dict depends on the pool_count .
2. The number of messages sent by the function also depends on the pool_count . The transfer fees are paid byrouter .

With a big enough pool_count the const::change_mining_rate_router_gas_consumption can be not enough to paytransfer fees.

handle_change_mining_amount() uses constants with the same names and is also affected.

Recommendation

We recommend:

1. Checking that msg_value > pool_count * (const::change_mining_rate_router_gas_consumption +const::change_mining_rate_lp_gas_consumption) .
2. Setting const::change_mining_rate_router_gas_consumption = const::gas_consumption .
3. Sending op::change_lp_mining_rate to lp-minter without PAY_FEES_SEPARATELY mode flag.
4. Renaming the constants to be more generic.

MINOR

ROU-06| jettonA_address / jettonB_address CAN BE ARBITRARY,IRRELEVANT TO REAL JETTONS

Issue	Severity	Location	Status
Volatile Code	Minor	contracts/amm/router.fc (base): 153~156	Acknowledged

Description

jettonA_address / jettonB_address are provided by pool_creator_address with op::create_pool message torouter . These addresses can be arbitrary, valid in basechain , and can be considered as "tags". They don't have to berelated to jettonA_wallet_address / jettonB_wallet_address . They are used to generate the lp-minter address andthe corresponding lp-wallet addresses. Swap operations must specify the same "tags" to get redirected to the same lp-minter .

Unlike jettonA_address / jettonB_address , wallets jettonA_wallet_address / jettonB_wallet_address aresignificant. lp-minter must be their owner for some unspecified jettons.

Recommendation

We recommend:
1. Providing the wallet code with op::create_pool
2. Validating the wallet addresses (and their real owner)

Or taking into account, and commenting the code correspondingly, that jetton addresses can be arbitrary.

MINOR

UTI-01| mined() CAN BE SIMPLIFIED

Issue	Severity	Location	Status
Coding Style	Minor	contracts/imports/utils.fc (base): 237~238	Resolved

Description

i and level variables are redundant in mined() . i < level condition can be replaced with half_life > 0 .

245 res = res + datetime_amount * (current_time - start_time + 1);

It is unclear, why one more second is added. For example, if current_time = minable_time = start_time = 0 , the resultis non-zero. We recommend clarifying the intended behavior and commenting the code.

Assignment operations (+= , /=) can be used to simplify the statements.

The function can be simplified to avoid redundant cycles and save gas

Recommendation

We recommend rewriting the function this way.

```
225 int mined(int mining_amount, int minable_time, int datetime_amount, int
half_life, int current_time) {
226 int elapsed = current_time - minable_time;
227 if (elapsed <= 0) {
228 return 0;
229 }
230
231 int res = 0;
232 if (half_life == 0) {
233 ;; constant mining speed
234 res = datetime_amount * elapsed;
235 } else {
236 ;; mining speed for the current period
237 int datetime_amount_now = datetime_amount >> (elapsed / half_life);
238 ;; mined for all full periods
239 res = (datetime_amount - datetime_amount_now) * 2 * half_life;
240 ;; mined in current period
241 res += datetime_amount_now * (elapsed % half_life);
242 }
243
244 ;; limit the result by mining_amount
245 if ((mining_amount > 0) & (res > mining_amount)) {
246 res = mining_amount;
247 }
248
249 return res;
250 }
```

INFORMATIONAL

CON-03|MISLEADING COMMENTS

Issue	Severity	Location	Status
Inconsistency	Informatinal	contracts/amm/lp-minter.fc (base): 85~86; contracts/amm/lp-wallet.fc (base): 180; contracts/amm/router.fc (base): 573, 621, 626;contracts/jetton-minter.fc (base): 92~93	Resolved

Description

```
180 .store_uint(0x10, 6) ;; nobounce - int_msg_info$0 ihr_disabled:Bool
bounce:Bool bounced:Bool src:MsgAddress -> 011000
```

The comment states 011000 , however, 010000 flags are used.

```
92 ;; NOTE : bridge minter  jetton custom_payload
```

The comments should be in English.

```
85 ;; sender_address can be admin or router
```

In fact, only messages from the router are accepted by handle_create_pool() .

```
573 send_raw_message(msg.end_cell(), 64); ;; pay transfer fees separately, revert
on errors
```

In fact, the mode is CARRY_REMAINING_GAS | REVERT_ON_ERRORS

```
626 if (op == op::change_lp_policy_admin) { ;; NOTE : swap_fee, min_amount admin
```

Recommendation

We recommend updating the comments.

INFORMATIONAL

IMP-01| UNUSED CODE

Issue	Severity	Location	Status
Inconsistency	Informational	contracts/imports/op-codes.fc (base): 4~5; contracts/imports/utils.fc (base): 5~14	Resolved

Description

These functions and variables are unused:

- utils::send_grams()
- message_utils::send_receipt_message()
- op::change_next_admin
- message_utils::send_text_receipt_message()
- message_utils::emit_log_simple()
- const::claim_gas_consumption

Recommendation

We recommend removing of unused code.

INFORMATIONAL

LPM-23|update_mining_index() CAN BE REFACTORED

Issue	Severity	Location	Status
Coding Style	Informational	contracts/amm/lp-minter.fc (base): 124, 712	Acknowledged

Description

lp-minter::update_mining_index() contains code repetitions. This decreases code readability and maintainability.Subroutine update_mining_index_for_mining_rate() can be created and used 3 times.

lp-minter::handle_transfer_notification() contains code repetitions, it can be significantly refactored to increasecode readability and maintainability

Recommendation

We recommend refactoring the functions. We recommend adding helper functions that prepare and send common messages.

INFORMATIONAL

LPM-24|USAGE OF MAGIC NUMBERS

Issue	Severity	Location	Status
Coding Style	Informational	contracts/amm/lp-minter.fc (base): 591~593	Acknowledged

Description

Different magic numbers are used as-is in code.

Recommendation

We recommend declaring constants to improve code maintainability and readability.

- SWAP_FEE_SCALE_FACTOR = 10000
- MINING_INDEX_SCALE_FACTOR = 1000000000000000000
- mode::REVERT_ON_ERRORS = 0
- mode::PAY_FEES_SEPARATELY = 1
- mode::IGNORE_ERRORS = 2
- mode::CARRY_REMAINING_GAS = 64
- etc.

INFORMATIONAL

LPM-25|in_msg_body IS UNUSED IN lp-minter::handle_claim()

Issue	Severity	Location	Status
Inconsistency	Informational	contracts/amm/lp-minter.fc (base): 929, 962	Resolved

Description

in_msg_body argument is unused in lp-minter::handle_claim() and lp-minter::handle_pending_jetton() .

Recommendation

We recommend removing of unused arguments.

INFORMATIONAL

LPM-26|op::change_router CAN'T BE HANDLED PROPERLY BYlp-minter

Issue	Severity	Location	Status
Coding Style	Informational	contracts/imports/utils.fc (base): 237~238	Resolved

Description

lp-minter allows to op::change_router . The router address is an argument of calculate_lp_minter_state_init() ,so, defines the lp-minter address. lp-minter address is used by router::create_pool() androuter::pool_created() .

lp-minter with a changed router address can't be added to another router , because it will have an address based onthe old router value. The router is allowed to change swap_fee and mining configuration, so, one can change it to EOA,change the configuration, change the router back, and op::claim more, than expected.

Recommendation

We recommend removing of op::change_router message handling.

INFORMATIONAL

OPC-01 | RESPONSE MESSAGES op DON'T HAVE HIGH-ORDERBIT SET

Issue	Severity	Location	Status
Coding Style	Informational	contracts/imports/op-codes.fc (base): 22, 32	Resolved

Description

Section 5 of the Internal Messages Guidelines recommends the "response" messages to have an op with the high-order bitset, i.e., in the range 2^31 .. 2^32-1 . This allows the contracts to ignore all the unhandled response messages easily.

op::pool_created is the response for op::create_pool .

op::check_mintable_notification is the response for op::check_mintable .

These op-codes have high-order bit unset

Recommendation

We recommend updating the op-codes in accordance with the Guidelines.

INFORMATIONAL

ROU-07|ARGUMENT NAMES OF router::get_lp_address() AREMISLEADING

Issue	Severity	Location	Status
Coding Style	Informational	contracts/amm/router.fc (base): 92~93	Resolved

Description

```
92 (slice, int) get_lp_address(slice jettonA_address, slice jettonB_address, cell
jetton_pair_to_lp) {
```

jettonA_address and jettonB_address argument names are misleading. The addresses can be in another order.

Recommendation

We recommend renaming the arguments to jetton1_address , jetton2_address for better code maintainability.

INFORMATIONAL

ROU-08|either_forward_payload VARIABLE IS UNUSED

Issue	Severity	Location	Status
Coding Style	Informational	contracts/amm/router.fc (base): 373~374	Resolved

Description

either_forward_payload local variable is never used.

Recommendation

We recommend removing of unused variables

INFORMATIONAL

UTI-02| calculate_jetton_wallet_address() CAN BE REPLACEDWITH calculate_contract_address().

Issue	Severity	Location	Status
Inconsistency	Informational	contracts/imports/utils.fc (base): 40~4874	Resolved

Description

calculate_jetton_wallet_address() can be removed. Universal calculate_contract_address() can be used instead.

Recommendation

We recommend removing of redundant code.

INFORMATIONAL

UTI-03| LONG AND COMPLICATED MESSAGE BUILDINGSTATEMENTS CAN BE FORMATTED

Issue	Severity	Location	Status
Coding Style	Informational	contracts/imports/utils.fc (base): 132~133	Acknowledged

Description

```
132 .store_dict(pack_lp_minter_data(0, 0, 0, admin_address,
router_address,jettonA_address, jettonA_address, 0, 0, jettonB_address,
jettonB_address, 0, 0, 0,0, 0, 0, 0, 0, begin_cell().store_uint(0, 32 + 64).end_cell(),
new_dict(),lp_default_content, lp_wallet_code))
```

Some statements are huge and complicated. That decreases readability and maintainability.

Recommendation

We recommend formatting of long statements using new lines and indentation

INFORMATIONAL

UTI-04| calculate_jetton_minter_address() IS UNUSED ANDDANGEROUS

Issue	Severity	Location	Status
Volatile Code	Informational	contracts/imports/utils.fc (base): 63~72, 82~85	Resolved

Description

calculate_jetton_minter_state_init() and calculate_jetton_minter_address() are unused.

calculate_jetton_minter_address() should not be used to discover the jetton-minter address. It usesadmin_address , minter_address , and content as arguments, which can be updated by the jetton-minter contract.

As a result, only providing original values will give the same jetton-minter address

Recommendation

We recommend removing of unused functions.

Disclaimer

This report is based on the scope of materials and documentation provided for a limited review at the time provided. Results may not be complete nor inclusive of all vulnerabilities. The review and this report are provided on an as-is, where-is, and as-available basis. You agree that your access and/or use, including but not limited to any associated services, products, protocols, platforms, content, and materials, will be at your sole risk. Blockchain technology remains under development and is subject to unknown risks and flaws. The review does not extend to the compiler layer, or any other areas beyond the programming language, or other programming aspects that could present security risks. A report does not indicate the endorsement of any particular project or team, nor guarantee its security. No third party should rely on the reports in any way, including for the purpose of making any decisions to buy or sell a product, service or any other asset. To the fullest extent permitted by law, we disclaim all warranties, expressed or implied, in connection with this report, its content, and the related services and products and your use thereof, including, without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement. We do not warrant, endorse, guarantee, or assume responsibility for any product or service advertised or offered by a third party through the product, any open source or third-party software, code, libraries, materials, or information linked to, called by, referenced by or accessible through the report, its content, and the related services and products, any hyperlinked websites, any websites or mobile applications appearing on any advertising, and we will not be a party to or in any way be responsible for monitoring any transaction between you and any third-party providers of products or services. As with the purchase or use of a product or service through any medium or in any environment, you should use your best judgment and exercise caution where appropriate.

FOR AVOIDANCE OF DOUBT, THE REPORT, ITS CONTENT, ACCESS, AND/OR USAGE THEREOF, INCLUDING ANY ASSOCIATED SERVICES OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, INVESTMENT, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.

Appendix

Finding Categories

Centralization / Privilege
Centralization / Privilege findings refer to either feature logic or implementation of components that act against the nature of decentralization, such as explicit ownership or specialized access roles in combination with a mechanism to relocate funds.

Coding Style
Coding Style findings usually do not affect the generated bytecode but rather comment on how to make the codebase more legible and, as a result, easily maintainable.

Volatile Code
Volatile Code findings refer to segments of code that behave unexpectedly on certain edge cases that may result in a vulnerability.

Logical Issue
Logical Issue findings detail a fault in the logic of the linked code, such as an incorrect notion on how block. timestamp works.

Checksum Calculation Method

The "Checksum" field in the "Audit Scope" section is calculated as the SHA-256 (Secure Hash Algorithm 2 with digest size of 256 bits) digest of the content of each file hosted in the listed source repository under the specified commit.

The result is hexadecimal encoded and is the same as the output of the Linux “sha256sum" command against the target file.

About

DeHacker is a team of auditors and white hat hackers who perform security audits and assessments. With decades of experience in security and distributed systems, our experts focus on the ins and outs of system security. Our services follow clear and prudent industry standards. Whether it's reviewing the smallest modifications or a new platform, we'll provide an in-depth security survey at every stage of your company's project. We provide comprehensive vulnerability reports and identify structural inefficiencies in smart contract code, combining high-end security research with a real-world attacker mindset to reduce risk and harden code.

BLOCKCHAIINS

Ethereum
Cosmos
Eos
Substrate

TECH STACK

Python
Solidity
Rust
c++

CONTACTS

https://dehacker.io
https://twitter.com/dehackerio
https://github.com/dehacker/audits_public
https://t.me/dehackerio
https://blog.dehacker.io/

August 27th 2024