CERTIK

Code Security Assessment

Shibaswife

Jan 20th, 2022

Table of Contents

Summary

This report has been prepared for Shibaswife to discover issues and vulnerabilities in the source code of the Shibaswife project as well as any contract dependencies that were not part of an officially recognized library. A comprehensive examination has been performed, utilizing Static Analysis and Manual Review techniques.

The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts.

The security assessment resulted in findings that ranged from critical to informational. We recommend addressing these findings to ensure a high level of security standards and industry practices. We suggest recommendations that could better serve the project from the security perspective:

- Enhance general coding practices for better structures of source codes;
- Add enough unit tests to cover the possible use cases;
- Provide more comments per each function for readability, especially contracts that are verified in public;
- Provide more transparency on privileged activities once the protocol is live.

Overview

Project Summary

Project Name	Shibaswife
Platform	bsc
Language	Solidity
Codebase	https://github.com/shibaswife/contract
Commit	4a0ee7ced47a76b4f31b29531e8bafa42aa5c8ce 33da3203ad24f4953558736d0add2dc18840cb92

Audit Summary

Delivery Date	Jan 20, 2022
Audit Methodology	Static Analysis, Manual Review

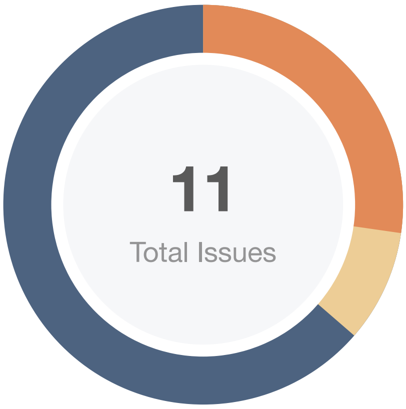
Vulnerability Summary

Vulnerability Level	Total	Pending	Declined	Acknowledged	Partially Resolved	Resolved
● Critical	0	0	0	0	0	0
● Major	3	0	0	2	0	1
● Medium	0	0	0	0	0	0
● Minor	1	0	0	1	0	0
● Informational	7	0	0	2	0	5
● Discussion	0	0	0	0	0	0

Audit Scope

ID	File	SHA256 Checksum
SCP	shibaswife.sol	58b1e0295e87bd0aa0d187b0b4a6506cce91db0bc13da4651de881105d254ccb

Findings



🟥 **Critical**	**0** (0.00%)
🟧 **Major**	**3** (27.27%)
🟨 **Medium**	**0** (0.00%)
🟨 **Minor**	**1** (9.09%)
🟦 **Informational**	**7** (63.64%)
🟩 **Discussion**	**0** (0.00%)

ID	Title	Category	Severity	Status
SCP-01	Centralization Related Risks	**Centralization / Privilege**	● **Major**	ⓘ Acknowledged
SCP-02	Variable Could Be Declared as `constant`	Gas Optimization	● Informational	✓ Resolved
SCP-03	External Dependencies Risk	Control Flow	● Minor	ⓘ Acknowledged
SCP-04	Comment Typo	Coding Style	● Informational	✓ Resolved
SCP-05	Ambitious Function `deliver`	Control Flow	● Informational	✓ Resolved
SCP-06	Function Visibility Optimization	Gas Optimization	● Informational	✓ Resolved
SCP-07	Comment Typo	Coding Style	● Informational	✓ Resolved
SCP-08	Contract Gains Non-Withdrawable ETH Via The `swapAndLiquify` Function	Logical Issue	● Major	✓ Resolved
SCP-09	Return Value Unhandled	Volatile Code	● Informational	ⓘ Acknowledged
SCP-10	Centralized Risk in `addLiquidity`	**Centralization / Privilege**	● **Major**	ⓘ Acknowledged
SCP-11	Redundant Code	Logical Issue	● Informational	ⓘ Acknowledged

SCP-01 | Centralization Related Risks

Category	Severity	Location	Status
Centralization / Privilege	● Major	shibaswife.sol: 558, 567, 939, 949, 973, 977, 981, 985, 989, 993, 997, 1003	ⓘ Acknowledged

Description

The owner of contract `ShibasWife` has authority below listed variables/behaviors/functions:

- `renounceOwnership()`
- `transferOwnership()`
- `excludeFromReward()`
- `includeInReward()`
- `excludeFromFee()`
- `includeInFee()`
- `setTaxFeePercent()`
- `setCauseFeePercent()`
- `setGrowthFeePercent()`
- `setLiquidityFeePercent()`
- `setMaxTxPercent()`
- `setSwapAndLiquifyEnabled()`

Any compromise to the `owner]` account may allow the hacker to take advantage of this and update the sensitive settings and executive sensitive functions of the project.

Recommendation

The risk describes the current project design and potentially makes iterations to improve in the security operation and level of decentralization, which in most cases can't be resolved entirely at the present stage. We advise the client to carefully manage the privileged account's private key to avoid any potential risks of being hacked. In general, we strongly recommend centralized privileges or roles in the protocol be improved via a decentralized mechanism or smart-contract-based accounts with enhanced security practices, e.g., Multisignature wallets.

Indicatively, here are some feasible suggestions that would also mitigate the potential risk at a different level in terms of short-term, long-term and permanent:

Short Term:

Timelock and Multi sign (⅔, ⅗) combination *mitigate* by delaying the sensitive operation and avoiding a single point of key management failure.

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
  AND
- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to the private key compromised;
  AND
- A medium/blog link for sharing the timelock contract and multi-signers addresses information with the public audience.

Long Term:

Timelock and DAO, the combination, were able to *mitigate* by applying decentralization and transparency.

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
  AND
- Introduction of a DAO/governance/voting module to increase transparency and user involvement.
  AND
- A medium/blog link for sharing the timelock contract, multi-signers addresses, and DAO information with the public audience.

Permanent:

Renouncing the ownership or removing the function can be considered *fully resolved*.

- Renounce the ownership and never claim back the privileged roles OR
- Remove the risky-functionalities

Alleviation

`[Shibaswife]`: Will be addressed through contract renouncement

SCP-02 | Variable Could Be Declared as `constant`

Category	Severity	Location	Status
Gas Optimization	● Informational	shibaswife.sol: 793, 799, 820	✓ Resolved

Description

Variables that never changed after assignment, can be declared as `constant`

- `_tTotal`
- `_decimals`
- `numTokensSellToAddToLiquidity`

Recommendation

We recommend declaring those variables as `constant`.

Alleviation

`[CertiK]`: The Shibaswife team heeded the advice and resolved the finding by adding `constant` to the highlighted variables in the commit 33da3203ad24f4953558736d0add2dc18840cb92

SCP-03 | External Dependencies Risk

Category	Severity	Location	Status
Control Flow	● Minor	shibaswife.sol: 813	ⓘ Acknowledged

Description

The contract is serving as the underlying entity to interact with external dependencies Uniswap protocols. The scope of the audit would treat those external dependencies entities as black boxes and assume functional correctness. In fact, any external dependencies might be compromised that led to assets being lost or stolen.

Recommendation

We understand that the business logic of the protocol requires the interaction Uniswap protocol for adding liquidity to Shibaswife-ETH protocol and swap tokens. We encourage the team to constantly monitor the statuses of those external dependencies to mitigate the side effects when unexpected activities are observed.

Alleviation

`[Shibaswife]`: Will be addressed through contract renouncement

SCP-04 | Comment Typo

Category	Severity	Location	Status
Coding Style	● Informational	shibaswife.sol: 827	✓ Resolved

Description

The event input variable contains a typo in its body, namely `tokensIntoLiqudity` should be `tokensIntoLiquidity`.

Recommendation

We advise to address the event input variable name.

Alleviation

`[CertiK]`: The Shibaswife team heeded the advice and resolved the finding by fixing the typo in the commit 33da3203ad24f4953558736d0add2dc18840cb92

SCP-05 | Ambitious Function `deliver`

Category	Severity	Location	Status
Control Flow	● Informational	shibaswife.sol: 913	✓ Resolved

Description

The function `deliver` can be called by anyone. It accepts an uint256 number parameter `tAmount`. The function reduces the Shibaswife token balance of the caller by `rAmount`, which is `tAmount` reduces the transaction fee. Then, the function adds `tAmount` to variable `_tFeeTotal`, which represents the contract's total transaction fee.

Recommendation

We wish the team could explain more on the purpose of having such functionality.

Alleviation

`[CertiK]`: The Shibaswife team removed the `deliver()` function in the commit 33da3203ad24f4953558736d0add2dc18840cb92

SCP-06 | Function Visibility Optimization

Category	Severity	Location	Status
Gas Optimization	● Informational	shibaswife.sol: 913, 939, 973, 977, 1003	✓ Resolved

Description

Public functions that are never called by the contract could be declared external. When the inputs are arrays external functions are more efficient than `public` functions. Public functions that are never called by the contract could be declared external. When the inputs are arrays external functions are more efficient than `public` functions.

Example functions :

- `deliver(uint)`
- `excludeFromReward(address)`
- `excludeFromFee(address)`
- `includeInFee(address)`
- `setSwapAndLiquifyEnabled(bool)`

Recommendation

Consider using the external attribute for functions never called from the contract.

Alleviation

`[CertiK]`: The Shibaswife team heeded the advice and updated the function visibilities to `external` in the commit 33da3203ad24f4953558736d0add2dc18840cb92

SCP-07 | Comment Typo

Category	Severity	Location	Status
Coding Style	● Informational	shibaswife.sol: 1008	✓ Resolved

Description

The linked comment contains a typo in its statement, namely `recieve` && `swaping` should be `receive` && `swapping`.

Recommendation

We advise to address the comment text.

```
//to receive ETH from uniswapV2Router when swapping`.
```

Alleviation

`[CertiK]`: The Shibaswife team heeded the advice and resolved the finding by fixing the typo in the commit 33da3203ad24f4953558736d0add2dc18840cb92

SCP-08 | Contract Gains Non-Withdrawable ETH Via The `swapAndLiquify` Function

Category	Severity	Location	Status
Logical Issue	● Major	shibaswife.sol: 1178	✓ Resolved

Description

The `swapAndLiquify` function converts half of the `contractTokenBalance` Shibaswife tokens to ETH. The other half of Shibaswife tokens and part of the converted ETH are deposited into the Shibaswife-ETH pool on uniswap as liquidity. For every `swapAndLiquify` function call, a small amount of ETH leftover in the contract. This is because the price of Shibaswife drops after swapping the first half of Shibaswife tokens into ETHs, and the other half of Shibaswife tokens require less than the converted ETH to be paired with it when adding liquidity. The contract doesn't appear to provide a way to withdraw those ETH, and they will be locked in the contract forever.

Recommendation

It's not ideal that more and more ETH are locked into the contract over time. The simplest solution is to add a `withdraw` function in the contract to withdraw ETH. Other approaches that benefit the Shibaswife token holders can be:

- Distribute ETH to Shibaswife token holders proportional to the amount of token they hold.
- Use leftover ETH to buy back Shibaswife tokens from the market to increase the price of Shibaswife.

Alleviation

`[CertiK]`: The Shibaswife team heeded the advice and resolve the finding by adding `leftOverBalanceAfterSwap` and `withdrawLockedBNB()` function in the commit 33da3203ad24f4953558736d0add2dc18840cb92

SCP-09 | Return Value Unhandled

Category	Severity	Location	Status
Volatile Code	● Informational	shibaswife.sol: 1224	ⓘ Acknowledged

Description

The return values of function `addLiquidityETH` is not properly handled.

```
1224 uniswapV2Router.addLiquidityETH{value: ethAmount}(
1225     address(this),
1226     tokenAmount,
1227     0, // slippage is unavoidable
1228     0, // slippage is unavoidable
1229     owner(),
1230     block.timestamp
1231 );
```

Recommendation

We recommend using variables to receive the return value of the functions mentioned above and handle both success and failure cases if needed by the business logic.

SCP-10 | Centralized Risk in `addLiquidity`

Category	Severity	Location	Status
Centralization / Privilege	● **Major**	shibaswife.sol: 1229	ⓘ Acknowledged

Description

```
1223 // add the liquidity
1224 uniswapV2Router.addLiquidityETH{value: ethAmount}(
1225     address(this),
1226     tokenAmount,
1227     0, // slippage is unavoidable
1228     0, // slippage is unavoidable
1229     owner(),
1230     block.timestamp
1231 );
```

The `addLiquidity` function calls the `uniswapV2Router.addLiquidityETH` function with the `to` address specified as `owner()` for acquiring the generated LP tokens from the `Shibaswife-ETH` pool. As a result, over time the `_owner` address will accumulate a significant portion of LP tokens. If the `_owner` is an EOA (Externally Owned Account), mishandling of its private key can have devastating consequences to the project as a whole.

Recommendation

We advise the `to` address of the `uniswapV2Router.addLiquidityETH` function call to be replaced by the contract itself, i.e. `address(this)`, and to restrict the management of the LP tokens within the scope of the contract's business logic. This will also protect the LP tokens from being stolen if the `_owner` account is compromised. In general, we strongly recommend centralized privileges or roles in the protocol be improved via a decentralized mechanism or smart-contract-based accounts with enhanced security practices, e.g., Multisignature wallets.

Indicatively, here are some feasible suggestions that would also mitigate the potential risk at a different level in terms of short-term, long-term and permanent:

Short Term:

Timelock and Multi sign (⅔, ⅗) combination *mitigate* by delaying the sensitive operation and avoiding a single point of key management failure.

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
  AND
- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to the private key compromised;
  AND
- A medium/blog link for sharing the timelock contract and multi-signers addresses information with the public audience.

Long Term:

Timelock and DAO, the combination, were able to *mitigate* by applying decentralization and transparency.

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;
  AND
- Introduction of a DAO/governance/voting module to increase transparency and user involvement.
  AND
- A medium/blog link for sharing the timelock contract, multi-signers addresses, and DAO information with the public audience.

Permanent:

Renouncing the ownership or removing the function can be considered *fully resolved*.

- Renounce the ownership and never claim back the privileged roles OR
- remove the risky-functionalities

Alleviation

`[Shibaswife]`: Will be addressed through contract renouncement

SCP-11 | Redundant Code

Category	Severity	Location	Status
Logical Issue	● Informational	shibaswife.sol: 1244	ⓘ Acknowledged

Description

The condition `!_isExcluded[sender] && !_isExcluded[recipient]` can be included in `else` .

Recommendation

The following code can be removed:

```
... else if (!_isExcluded[sender] && !_isExcluded[recipient]) {
    _transferStandard(sender, recipient, amount);
} ...
```

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.

Gas Optimization

Gas Optimization findings do not affect the functionality of the code but generate different, more optimal EVM opcodes resulting in a reduction on the total gas cost of a transaction.

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.

Control Flow

Control Flow findings concern the access control imposed on functions, such as owner-only functions being invoke-able by anyone under certain circumstances.

Volatile Code

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

Coding Style

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

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.

Disclaimer

This report is subject to the terms and conditions (including without limitation, description of services, confidentiality, disclaimer and limitation of liability) set forth in the Services Agreement, or the scope of services, and terms and conditions provided to you (“Customer” or the “Company”) in connection with the Agreement. This report provided in connection with the Services set forth in the Agreement shall be used by the Company only to the extent permitted under the terms and conditions set forth in the Agreement. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes, nor may copies be delivered to any other person other than the Company, without CertiK’s prior written consent in each instance.

This report is not, nor should be considered, an “endorsement” or “disapproval” of any particular project or team. This report is not, nor should be considered, an indication of the economics or value of any “product” or “asset” created by any team or project that contracts CertiK to perform a security assessment. This report does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance.

This report should not be used in any way to make decisions around investment or involvement with any particular project. This report in no way provides investment advice, nor should be leveraged as investment advice of any sort. This report represents an extensive assessing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk. CertiK’s position is that each company and individual are responsible for their own due diligence and continuous security. CertiK’s goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or functionality of the technology we agree to analyze.

The assessment services provided by CertiK is subject to dependencies and under continuing development. You agree that your access and/or use, including but not limited to any services, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis. Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives, false negatives, and other unpredictable results. The services may access, and depend upon, multiple layers of third-parties.

ALL SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF ARE PROVIDED “AS IS” AND “AS

AVAILABLE" AND WITH ALL FAULTS AND DEFECTS WITHOUT WARRANTY OF ANY KIND. TO THE MAXIMUM EXTENT PERMITTED UNDER APPLICABLE LAW, CERTIK HEREBY DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS. WITHOUT LIMITING THE FOREGOING, CERTIK SPECIFICALLY DISCLAIMS ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, AND ALL WARRANTIES ARISING FROM COURSE OF DEALING, USAGE, OR TRADE PRACTICE. WITHOUT LIMITING THE FOREGOING, CERTIK MAKES NO WARRANTY OF ANY KIND THAT THE SERVICES, THE LABELS, THE ASSESSMENT REPORT, WORK PRODUCT, OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF, WILL MEET CUSTOMER'S OR ANY OTHER PERSON'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULT, BE COMPATIBLE OR WORK WITH ANY SOFTWARE, SYSTEM, OR OTHER SERVICES, OR BE SECURE, ACCURATE, COMPLETE, FREE OF HARMFUL CODE, OR ERROR-FREE. WITHOUT LIMITATION TO THE FOREGOING, CERTIK PROVIDES NO WARRANTY OR UNDERTAKING, AND MAKES NO REPRESENTATION OF ANY KIND THAT THE SERVICE WILL MEET CUSTOMER'S REQUIREMENTS, ACHIEVE ANY INTENDED RESULTS, BE COMPATIBLE OR WORK WITH ANY OTHER SOFTWARE, APPLICATIONS, SYSTEMS OR SERVICES, OPERATE WITHOUT INTERRUPTION, MEET ANY PERFORMANCE OR RELIABILITY STANDARDS OR BE ERROR FREE OR THAT ANY ERRORS OR DEFECTS CAN OR WILL BE CORRECTED.

WITHOUT LIMITING THE FOREGOING, NEITHER CERTIK NOR ANY OF CERTIK'S AGENTS MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND, EXPRESS OR IMPLIED AS TO THE ACCURACY, RELIABILITY, OR CURRENCY OF ANY INFORMATION OR CONTENT PROVIDED THROUGH THE SERVICE. CERTIK WILL ASSUME NO LIABILITY OR RESPONSIBILITY FOR (I) ANY ERRORS, MISTAKES, OR INACCURACIES OF CONTENT AND MATERIALS OR FOR ANY LOSS OR DAMAGE OF ANY KIND INCURRED AS A RESULT OF THE USE OF ANY CONTENT, OR (II) ANY PERSONAL INJURY OR PROPERTY DAMAGE, OF ANY NATURE WHATSOEVER, RESULTING FROM CUSTOMER'S ACCESS TO OR USE OF THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS.

ALL THIRD-PARTY MATERIALS ARE PROVIDED "AS IS" AND ANY REPRESENTATION OR WARRANTY OF OR CONCERNING ANY THIRD-PARTY MATERIALS IS STRICTLY BETWEEN CUSTOMER AND THE THIRD-PARTY OWNER OR DISTRIBUTOR OF THE THIRD-PARTY MATERIALS.

THE SERVICES, ASSESSMENT REPORT, AND ANY OTHER MATERIALS HEREUNDER ARE SOLELY PROVIDED TO CUSTOMER AND MAY NOT BE RELIED ON BY ANY OTHER PERSON OR FOR ANY PURPOSE NOT SPECIFICALLY IDENTIFIED IN THIS AGREEMENT, NOR MAY COPIES BE DELIVERED TO, ANY OTHER PERSON WITHOUT CERTIK'S PRIOR WRITTEN CONSENT IN EACH INSTANCE.

NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING

MATERIALS AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS.

THE REPRESENTATIONS AND WARRANTIES OF CERTIK CONTAINED IN THIS AGREEMENT ARE SOLELY FOR THE BENEFIT OF CUSTOMER. ACCORDINGLY, NO THIRD PARTY OR ANYONE ACTING ON BEHALF OF ANY THEREOF, SHALL BE A THIRD PARTY OR OTHER BENEFICIARY OF SUCH REPRESENTATIONS AND WARRANTIES AND NO SUCH THIRD PARTY SHALL HAVE ANY RIGHTS OF CONTRIBUTION AGAINST CERTIK WITH RESPECT TO SUCH REPRESENTATIONS OR WARRANTIES OR ANY MATTER SUBJECT TO OR RESULTING IN INDEMNIFICATION UNDER THIS AGREEMENT OR OTHERWISE.

FOR AVOIDANCE OF DOUBT, THE SERVICES, INCLUDING ANY ASSOCIATED ASSESSMENT REPORTS OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.

About

Founded in 2017 by leading academics in the field of Computer Science from both Yale and Columbia University, CertiK is a leading blockchain security company that serves to verify the security and correctness of smart contracts and blockchain-based protocols. Through the utilization of our world-class technical expertise, alongside our proprietary, innovative tech, we're able to support the success of our clients with best-in-class security, all whilst realizing our overarching vision; provable trust for all throughout all facets of blockchain.

CERTIK