EVMPatch: Timely Patching of Ethereum Smart Contracts with EVM Bytecode Rewriting

Michael Rodler, Wenting Li, Ghassan Karame, Lucas Davi

University of Duisburg-Essen, NEC Labs Europe
How to protect smart contracts after deployment?

Why don’t you deploy a patch?

Ethereum Smart Contracts are immutable
Patching Smart Contracts

• A lot of prior work on vulnerability detection
  • Symbolic execution (e.g., Oyente, teEther, EthBMC, …)
  • Static analysis (e.g., Securify, eThor, …)
  • Dynamic analysis (e.g., Sereum, TXSpector, …)
  • ...

• We regularly observe incidents on the blockchain.

• We need to enable smart contract developers to patch new issues!
Existing Patching Strategies

Migration to a New Contract
Deprecate old contract, deploy new contract, **manually migrate** state to new contract.

Upgradable Contract using a Proxy Contract
Contract is split into two:
• proxy contract
• logic contract
Requires **manual conversion**;
must ensure **storage layout compatibility**
Are Upgradable Contracts Practical?

Study with 6 Developers (4 with “production-grade” smart contract experience)

<table>
<thead>
<tr>
<th>Task</th>
<th>Median Minutes</th>
<th>Median Reported Confidence (1-7)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manual Patching</td>
<td>47.5</td>
<td>6</td>
</tr>
<tr>
<td>Manual Upgradable Contract (Proxy Pattern)</td>
<td>62.5</td>
<td>2.5</td>
</tr>
</tbody>
</table>

None of the manually created upgradable contracts were fully functional!
Upgrading smart contracts is cumbersome, time-consuming, and error-prone.
Introducing EVMPatch

- Fully **automated patching** framework
- Automates the delegatecall-**proxy pattern**
  - Automatic conversion to proxy pattern
  - Deployment of contract and upgrades
- Patching with **bytecode rewriting**
  - Template-based patching with custom DSL
  - Naturally preserves storage-layout
- **Differential patch testing**
  - Ensure equivalent behavior: original vs patched contract
EVMPatch Architecture

Developer

Vulnerability Detection
- Automatic Analysis Tools
- Vulnerability Disclosure

Vulnerable Contract Bytecode

EVMPatch
- Vulnerability Report
- Bytecode Rewriter
- Patch Templates
- Patched Contract Bytecode
- Contract Deployment

Patch Tester

Transaction History

Ethereum Blockchain
- Upgradable Contract

Attack Transactions
EVMPatch Integer Overflow Check

Original Contract

0xAB

EVM_INS1
EVM_INS2
MUL
POP
...

0xCD

JUMPDEST
...

Rewritten Contract

0xAB

PUSH2 0xFFF
JUMP
[INVALID]
[INVALID]
...
[INVALID]

0xCD

JUMPDEST
...

0xFFF

JUMPDEST
EVM_INS1
EVM_INS2
[CHECKED_MUL]
POP
PUSH1 0xCD
JUMP
Evaluation Results: Attacks

Evaluation on 5 known exploited ERC-20 Token Contracts

<table>
<thead>
<tr>
<th>Contract</th>
<th>CVE</th>
<th># Transactions</th>
<th># Attacks</th>
</tr>
</thead>
<tbody>
<tr>
<td>BEC</td>
<td>2018-10299</td>
<td>424,229</td>
<td>1</td>
</tr>
<tr>
<td>SMT</td>
<td>2018-10376</td>
<td>56,555</td>
<td>1</td>
</tr>
<tr>
<td>UET</td>
<td>2018-10468</td>
<td>24,034</td>
<td>55</td>
</tr>
<tr>
<td>SCA</td>
<td>2018-10706</td>
<td>292</td>
<td>1</td>
</tr>
<tr>
<td>HXG</td>
<td>2018-11239</td>
<td>1,497</td>
<td>9</td>
</tr>
</tbody>
</table>

Comparison with manual patches (SafeMath)

EVMPatch’ed contracts...

- Prevent same attacks as SafeMath
- Same behavior as original on non-attacks
- Comparable overhead to source-level patches
Evaluation Results: Practicality

- Additional Cost due to Gas Overhead
  - Per Transaction: < 0.01$
  - Per Upgrade: < 0.20$

- Developer Study

<table>
<thead>
<tr>
<th>Task</th>
<th>Median Minutes</th>
<th>Median Reported Confidence (1-7)</th>
</tr>
</thead>
<tbody>
<tr>
<td>EVMPatch Patch+Deploy</td>
<td>1.5</td>
<td>-</td>
</tr>
<tr>
<td>New EVMPatch Template</td>
<td>4.0</td>
<td>7</td>
</tr>
</tbody>
</table>

About 5 minutes to patch and deploy a new type of vulnerability with EVMPatch!
EVMPatch: Timely Patching of Ethereum Smart Contracts with EVM Bytecode Rewriting

Michael Rodler, Wenting Li, Ghassan Karame, Lucas Davi

- Practical Post-Deployment Protection
- Efficient EVM Bytecode Patching
- Timely Patching of Vulnerabilities
- Automated Upgradable Contracts

michael.rodler@uni-due.de
https://udue.de/evmpatch