Bottract
Abusing Blockchains and Smart Contracts for Botnet Command and Control

Majid A. Malaika, Dr.Eng, CISSP
mmaliaka@omprotect.com
“The views expressed herein are those of the speaker and should not be attributed to the IMF, its executive board, or its management.”
Who we are?

- Group of research scholars and industry experts in cybersecurity.
- Work in various areas: malware analysis, application & mobile security, architecture risk analysis, threat modeling, crypto-currency & blockchain security, and more.
Agenda

- Blockchains
- Smart contracts
- Botnet
- Botnet architecture & evolution
- Botnet countermeasures
- Related work in smart contract security
- Putting things together
- Advantages for Botract
- Botract
  - Dataflow
  - Contribution
  - Solidity Snippets
  - Demo
- Limitations
- Future Work
- Questions
We won't cover...

• Infecting and recruiting bots
• Hacking the blockchain itself
• Zero-day in smart contracts
• Cryptocurrencies or mining
“The blockchain is an incorruptible digital ledger of economic transactions that can be programmed to record not just financial transactions but virtually everything of value.” Don & Alex Tapscott
“Smart contract” is a program that implements the conditional logic of a contract, which is verified and enforced by the blockchain.

There is no simple or agreed-upon definition.
Solidity Language

- Object-oriented, high level similar to JavaScript and C
- Designed to run on the EVM
- Statically typed
- Supports
  - Inheritance
  - libraries
  - User defined types
- Solidity file (.sol) => bytecode => blockchain
contract token {
    mapping (address => uint) public coinBalanceOf;
    event CoinTransfer(address sender, address receiver, uint amount);

    /* Initializes contract with initial supply tokens to the creator of the contract */
    function token(uint supply) {
        if (supply == 0) supply = 10000;
        coinBalanceOf[msg.sender] = supply;
    }

    /* Very simple trade function */
    function sendCoin(address receiver, uint amount) returns (bool sufficient) {
        if (coinBalanceOf[msg.sender] < amount) return false;
        coinBalanceOf[msg.sender] -= amount;
        coinBalanceOf[receiver] += amount;
        CoinTransfer(msg.sender, receiver, amount);
        return true;
    }
}
Related Work: Smart Contract Security

- Smart contract vulnerabilities
  - Race conditions
  - Reentrancy
  - Default public
  - Buffer overflow
  - Callstack exploits

- Smart contract vulnerability analysis tools
  - Oyente
  - Porosity
  - Securify
  - Dr. Y's Ethereum Contract Analyzer
  - Solidified

Botnets in a Nutshell

- Collection of internet-connected compromised devices (zombies) controlled by a single attacker (botmaster) through common malware.
  - Spam & Phishing
  - DDoS
  - Proxy
  - Harvesting
  - Hosting Servers
  - Password cracking

- Billions of dollars in losses globally.
- Approximately 500 million computers are infected globally each year.
Botnets Architecture & Evolution

• Traditional centralized command and control (C2)
  – IRC
  – Social media

• Emerging decentralized / distributed
  – Peer to Peer (P2P)
  – Multi-layered botnets

• Methods for botnets to avoid detection and disablement
  – Anonymity networks
  – Proxy redirection e.g. Fast flux
  – Encryption
  – Steganography combined with social media networks
Botnet Countermeasures

- Monitor network signatures and DNS queries
- Sinkholing: Global vs. local
- Network blacklisting and filtering NIDS
- Antivirus and executable binary signatures
- Kill switch or deactivation logic
- P2P detection of C2 communication
Putting them Together => Botract

- Blockchains and smart contracts can be a deadly combination to fuel Botnets
Advantages for Botract

- ‘Unstoppable apps’ in the blockchain
- Distributed platform
- Consistent ledger
- Persistent C2
- Difficult to takedown
- Fast bootstrap
- Programmability and extensibility
Bottract Dataflow
Bottract Dataflow

Blockchain

Publish the C2 Smart Contract

Bot Commander
Bottract Dataflow
Bottract Dataflow

BlockChain

setCommand()

Bot Commander
Botract Dataflow
Contributions

• Abusing the blockchain and smart contracts for deploying malicious code
  – Messaging platform for botnets
  – Spam and data harvesting logic

• Pointing out design flaws in blockchain
  – Implicit end user trust
  – Lack of code scrutiny
  – Lack of robust governance structure
Solidity Botract Snippet

BotnetCnC.sol

- Bottract ownership
- Sending and receiving commands
  - Get commands fields (Receive, Bots)
  - Set command fields (Send, Commander)
- Authentication and authorization
- Function invocation
Demo
Limitations

- Storage of the blockchain on the bots
- Initialization costs
- Continuous funding of the operation of Botract
Bot Minimization (Storage)

Issue
- Client storage footprint
  - ~27GB blockchain size Geth w/FAST Sync
    https://etherscan.io/chart2/chaintotalsizefast

Proposed Solutions
- Light Ethereum Client
  - ~10MB
- Remote Geth
Economics of Botract

- **Deployment costs**
  - Smart contract
  - Bot initialization
- **Operational costs**
  - Commands
    - Unicast vs Broadcast
    - Bot response

<table>
<thead>
<tr>
<th>Transaction fee</th>
<th>Deploy contract</th>
<th>Set command (1000 commands)</th>
<th>Get command</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Private chain</strong></td>
<td>0.00848 ETH</td>
<td>0.45837 ETH</td>
<td>0 ETH</td>
</tr>
<tr>
<td><strong>Ethereum chain</strong></td>
<td>0.00047 ETH (~$0.14)*</td>
<td>0.0025465 ETH (~$0.76)</td>
<td>0 ETH ($0)*</td>
</tr>
<tr>
<td><strong>(Safe low)</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Conversion rate 1 ETH = $300
Future Work

• How to abuse the future developments of the blockchain and smart contracts
  – Swarm: decentralized storage
  – Whisper: decentralized messaging
• Minimizing the footprint of the client bots
• Reducing the economic costs for the deployment and operations of Bottract
• Develop a governance model for the blockchain and smart contracts that meets future demands