NSA Codebreaker Challenge

Fall 2018

Scenario

- A new strain of ransomware has managed to penetrate several critical government networks and NSA has been called upon to assist in remediating the infection
- An encrypted copy of the key needed to decrypt the ransomed files has been stored in a smart contract on the Ethereum blockchain*

* - a private Ethereum blockchain has been created with no real monetary value associated with the Ether

Mission

Your mission is to ultimately:

1. Unlock the ransomware without giving in to the attacker's demands
2. Recover the funds already paid by other victims

Tasks (1)

- Task 0: Discover IP address of attacker's listening post
- Task 1: Analyze ransomware client-registration protocol and find the victim ID, encrypted ransom key, authentication code, and Escrow contract address
- Task 2: Reverse-engineer ransomware and recover key used for authentication
- Task 3: Reverse-engineer victim ID generation process

Tasks (2)

- Task 4: Analyze internal state of Escrow contract on the blockchain and enumerate victims that have / haven't paid the ransom
- Task 5: Develop a method for determining whether any victims are on a particular network segment based on their victim ID
- Task 6: Analyze Ransom & Escrow contracts to figure out a way of getting decryption key without actually paying any Ether for the ransom
- Task 7: Find and exploit vulnerabilities in the smart contracts to recover ALL funds paid by the victims and transfer the Ether back into their accounts

Skills Learned

- Network traffic analysis
- Binary Reverse-Engineering
- Cryptanalysis
- Blockchain Analysis
- Ethereum "smart" contract development
- Vulnerability Analysis
- Exploit Development

New Features

- Updated scoring system:
  - Point-based
  - Balanced rewards for progression vs. participation
- All tasks available at launch
  - No requirement to complete in order (though still recommended to do so)
- Many behind-the-scenes updates (e.g., new website)

How to Participate

- Visit: https://codebreaker.ltsnet.net
- Register for an account with your .edu email address
- Learn and have fun!

Technical Background

Network Traffic Analysis

- Recommended tool: Wireshark
- Cross platform, parsers for many protocols
- Available features/functionality:
  - Display filters to focus in on traffic
  - TCP stream following
  - Extract files from packet payloads
  - Dissecting custom protocols (Lua script interface)
  - Traffic statistics/characterization
  - See https://www.wireshark.org/ for more details

Binary Reverse Engineering

- Recommended Tools: IDA Pro, Binary Ninja
- Start by examining strings in the binary
  - Look for clues that relate to the functionality you are trying to find / reverse
  - Utilize xrefs to find code that references the string(s) of interest
- Utilize symbols (e.g., function names) to help determine what a section of code does
- Leverage online resources, e.g., Intel manuals, RE lectures, etc. for help on reverse-engineering

Ethereum (1)

- Open and programmable blockchain platform
- Enables development and use of decentralized applications that run on blockchain technology
- Ethereum Virtual machine (EVM)
  - Can execute code of arbitrary complexity
  - Every node runs the EVM and executes same instructions

Ethereum (2)

- Two types of accounts
  - Externally Owned Accounts (EOAs), which are controlled by private keys
  - Contract Accounts, which are controlled by contract code and can only be "activated" by an EOA
- Term "smart contracts" refers to code in a Contract Account
- Users pay small transaction fees to the network
  - Sender of transaction pays for each step of the "program" they activated, including computation and memory storage
  - Fees are paid in Ethereum's native value token: "Ether"

Ethereum (3)

- Base unit of Ether is called Wei (1e18 Wei = 1 ether)
- "Gas" is the constant cost of network resources/utilization
  - Intent is for real value of Gas to never change
  - Keeps cost of sending a transaction to always be the same
- "Gas Price" is expressed in terms of Ether
  - If price of ether goes up, the Gas Price should go down to keep real cost of Gas the same
- "Gas Limit" is the max amount of Gas that can be used per block
  - Considered the max computational load, transaction volume, or block size of a block
- "Gas Fee" is amount of Gas needed to be paid to run a particular transaction

Ethereum Resources

- Ethereum Docs: http://www.ethdocs.org/en/latest/
- Solidity Docs: https://solidity.readthedocs.io/en/latest/
- Web3.js 1.0 API: https://web3js.readthedocs.io/en/1.0/web3-eth.html
- Web3.py: https://web3py.readthedocs.io/en/stable/
- Metamask: https://metamask.io/
- Remix IDE: https://github.com/ethereum/remix-ide

Getting Started

Codebreaker and Ethereum (1)

- For the Codebreaker Challenge, NSA is running a private chain and controls all the nodes
- Students will transact with the blockchain over an RPC interface
  - Remote connections will go through an RPC proxy that's been developed to provide unique per-student URLs for authenticating to the blockchain
  - RPC proxy should not interfere with normal Web3 commands / blockchain transactions

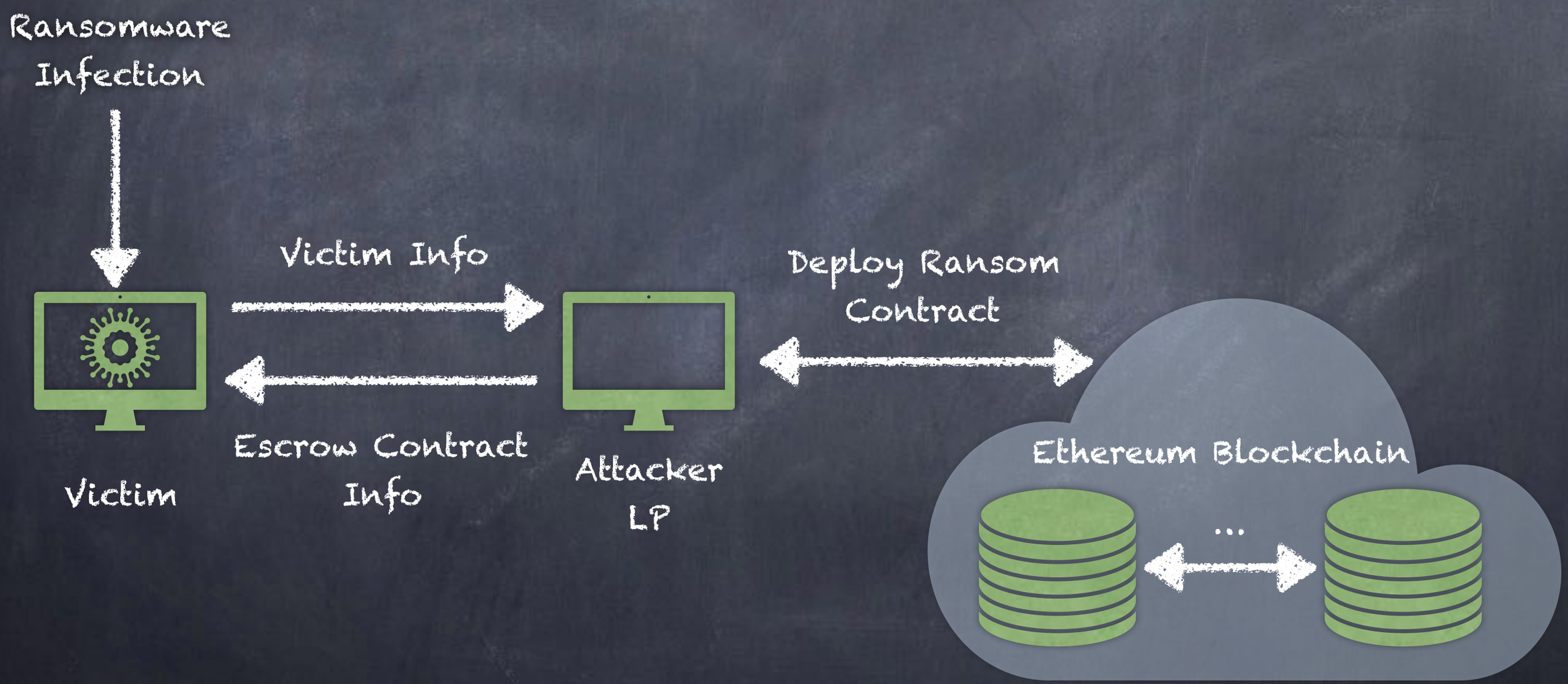
Codebreaker and Ethereum (2)

- Each student will be given the private key to an account (pre-funded with Ether) on the NSA blockchain
- Recommended setup is to use Metamask and Remix IDE
  - Metamask can be configured to connect to the Codebreaker RPC proxy and import the private key for a student account
  - Remix IDE runs locally and uses Metamask as an Injected Web3 Provider
  - This enables all blockchain-related work to be done within a web browser window

Ransomware Overview

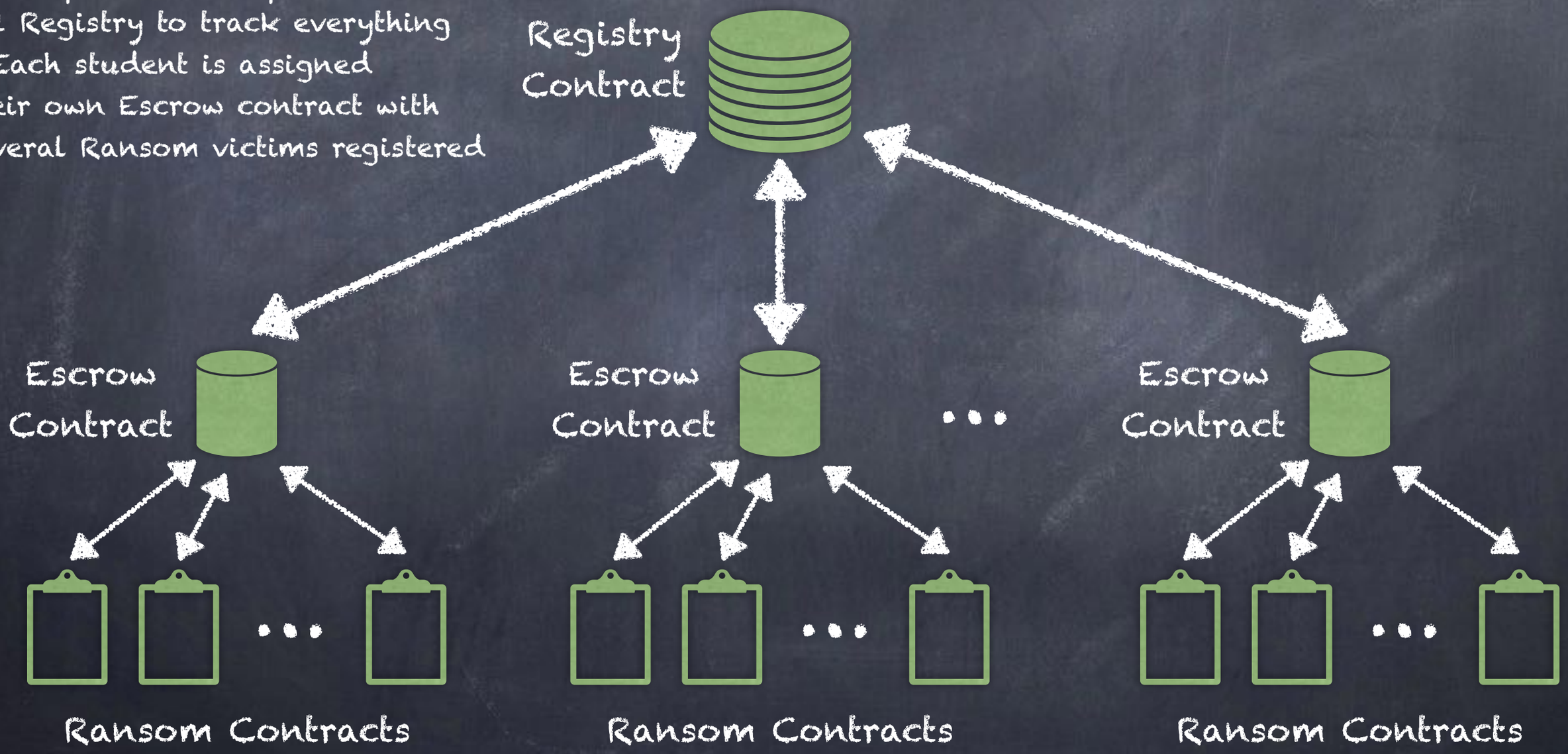
- Students will be provided with the following:
  - 2 shared libraries associated with the ransomware
  - Source code to the Ransom, Escrow, and Registry contracts
  - 1 encrypted (ransomed) file
- No infection logic is provided (for safety)
- The next slides provide a high-level overview of the infection process and smart contract relationships

Ransomware Infection Process



Ransomware Contract Relationships

- 1 Ransom Contract per Victim
- Multiple Ransoms per Escrow
- 1 Registry to track everything
- Each student is assigned their own Escrow contract with several Ransom victims registered



Authentication & Decryption Oracle

- There is an off-chain oracle that monitors the Registry and Escrow contracts for certain events:
  - Authentication Event: occurs when a new Ransom contract is deployed and tries to register with the Registry contract
  - Decryption Event: occurs when a victim has paid the ransom into Escrow and provided a test file to decrypt
- Purpose of oracle is to protect private keys so they aren't exposed on the blockchain
- The oracle responds to events by transacting directly with the Registry and Escrow contracts

Q & A

- If you encounter any problems, send an email to codebreaker@nsa.gov
- Good luck!

2017 Codebreaker Challenge Walkthrough

- Special thanks to Jonathan Armer for sharing his detailed writeup at:

https://armerj.github.io/CodeBreaker-Overview/