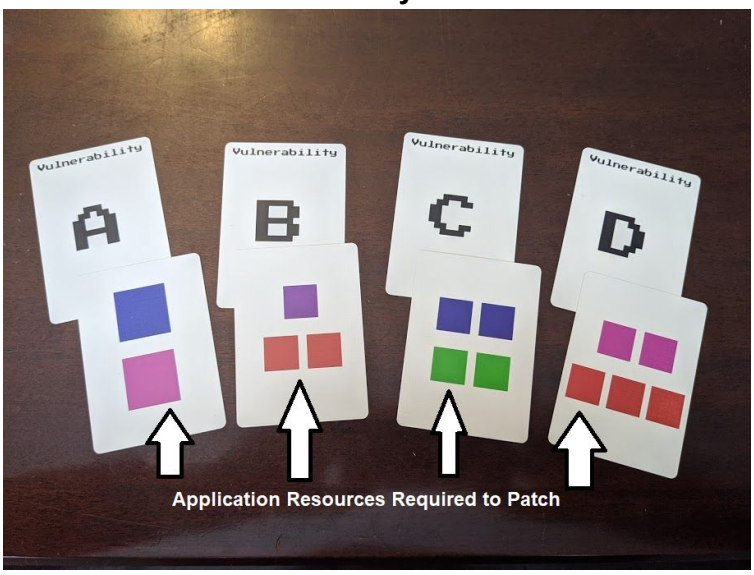
White Hat: Capture the Flag

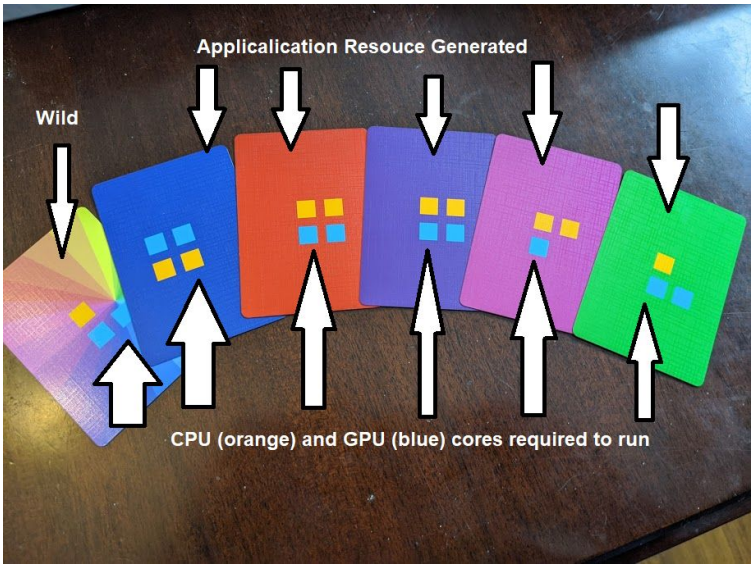
Components

- 6 player boards
- 1 vulnerability board
- 1 application offer board
- 6 player screens
- 6 cryptocurrency miner cards
- 24 CPU tokens
- 48 memory tokens
- 12 GPU tokens
- 12 end game scoring cards
- 60 application cards
- 100 vulnerability cards
- 1 score sheet
- 52 six-sided dice
  - 8 Red
  - 8 Blue
  - 8 Purple
  - 8 Green
  - 8 Yellow
  - 8 White
  - 4 Brown
- 60 meeples
  - 10 Red
  - 10 Blue
  - 10 Purple
  - 10 Green
  - 10 Yellow
  - 10 White
- 550 cubes
  - 48 orange cubes - CPU allocations
  - 24 light blue cubes - GPU allocations
  - 50 clear cubes - Written code
  - 50 large clear cubes - Written code x5
  - 50 yellow cubes - BTC
  - 50 large yellow cubes - BTC x5
  - 125 application cubes
    - 25 Red
    - 25 Blue
    - 25 Purple
    - 25 Green
    - 25 Pink

Vulnerability Cards



Application Cards



Theme

White Hat: Capture the Flag is an action selection style game where each player leads a team of hackers trying to patch vulnerabilities on their computer before their opponents do the same. Play happens simultaneously as all the players place their hackers and then complete actions at the same time.

White hats are hackers that aren't malicious in their hacking. If they find vulnerabilities in someone's code, they report it, often doing so for monetary rewards, rather than exploiting it to do damage. Black hats, by comparison, exploit vulnerabilities maliciously and use them for personal gain. The term comes from old wild west movies where the heroes wear white hats, and the villains wear black.

Capture the Flag is an event at many computer security and programming conferences and conventions. There are two types of Capture the Flag events. The most common is a Jeopardy style game where you get points for finding and patching vulnerabilities based on short clues. White Hat: Capture the Flag is based on a versus style event where each team of hackers is working against each other more directly.

White Hat: Capture the Flag also has a bit of cryptocurrency as a mechanic. Cryptocurrencies are generated by finding a hash that matches the next coin to be found. The game uses a simplistic version of this to simulate mining cryptocurrency and using that to purchase upgrades for a player's computer as well as additional hackers.

Goal

Have the most victory points after a player has patched all 4 vulnerabilities.

a) Each vulnerability is worth a number of points equal to the number of resources it takes to patch
   - 10, 15, 20, and 25 for a total of 70 possible points

b) Each player knows one unique end game scoring card
   - All players score each end game scoring card

Setup

1) Give each player a bag with the following contents:
   a) 16 dice
   b) 10 meeples
   c) A cryptocurrency miner application card
   d) 1 CPU Core
   e) 1 memory stick
   f) 2 orange cpu allocation cubes
2) Give each player a motherboard
   a) Place the cpu, cpu allocation cubes, and the memory on the designated spaces
3) Give each player a player screen
4) Shuffle the end game scoring cards and deal two to each player
   a) Each player picks one of the two and returns the other back to the box face down
   b) The other is kept face down by their motherboard
5) Each player starts with 4 hacker meeples. The rest are kept in their personal supply.
6) Place the vulnerability and application boards in the middle of the table
7) Shuffle each vulnerability deck and place them face down on the corresponding place on the vulnerability board

8) Shuffle the deck of application cards and place it face down on the blank spot on the application offer
9) Fill the application offer with the top 3 cards of the application deck

The Main Program Loop

1) Place hackers
    c) Each player starts with 4 hacker meeples
    d) Any number of hackers can be placed on any space
    e) In the case of write and sell code spaces, hackers start on the leftmost square
2) Activate hackers from left to right, top to bottom
    a) See Actions for full rules on each action
    b) Each action is taken simultaneously, with the exception of building applications (see Actions)
3) Retrieve or advance hackers
    a) Instead of retrieving hackers sent to write code, sell code, trade secrets, pen test, and qa test, you can advance them to the next square to increase the value of those hackers' actions on later turns
    b) If you ever retrieve a hacker on the write and sell code actions, you must restart that hacker at the leftmost square the next time they are placed
4) Check memory/CPU/GPU allocation
    a) Each unique running application uses a stick of memory. Each GPU counts as a stick of memory for the application running on it.
    b) Applications can not have more CPU/GPU allocation cubes than the player has available
        - The player purchased a CPU, for a total of 2 CPUs, and a GPU
        - They have a total of 4 orange CPU allocation cubes and 2 blue GPU allocation cubes to use
5) Applications generate resources
    a) For each application (not the cryptocurrency miner) running on the player's computer 1 resource cube is generated per set of allocation cubes on that application
        - Ex: The player has a pink resource generating application that takes 2 CPU allocations to run with 2 CPU allocation cubes on it. They get 1 pink resource cube.
        - Ex 2: The player has a purple resource generating application that takes 1 GPU allocation come to run with 3 allocation cubes on it. They get 3 purple resource cubes
6) Mine cryptocurrency
    a) Each player rolls 1 die per CPU allocation cube and 2 dice per GPU allocation cube on their cryptocurrency miner
    b) The four brown dice get rolled as well
    c) If a player rolled a die that matches one of the brown dice, they gain that amount of BTC. This can only happen for one die each turn per player.
        - Ex: The brown dice are rolled and show a 1, 4, 4, and 6
            (1) The player has 2 allocation cubes on their cryptocurrency miner and rolls 2 dice: a 2 and a 4. They receive 4 BTC
            (2) The player has 1 allocation cube on their cryptocurrency miner and rolls 1 die: a 3. They receive nothing.
7) Check for end of game
    a) If any player has released fixes for all 4 vulnerabilities, the game ends and score is calculated
    b) Winner is determined by calculating victory points
        - Each vulnerability is worth a number of points equal to the number of application resources it takes to patch. 10, 15, 20, and 25 for a total of 70 possible points
        - Each player reveals their secret end game scoring card and all players score each card
8) If the game hasn't ended, goto 1

Actions

Actions are taken in order from top left to bottom right of the player board. With the exception of Build Applications, the actions are independent of other players, meaning players may take these actions at their own pace.

- Write code
    a. Produces blocks of code (transparent cubes) for each square, square's value * number of hackers on that square
    b. Hackers are placed on the leftmost square and may be advanced during phase 3 to provide more code in future rounds
    c. Ex: In the first round the player places two hackers on the leftmost square of the write code action.
        - The player takes 4 blocks of code (transparent cubes), 2 for each hacker
    d. Ex 2: The player advanced their two hackers in the previous round and advanced them, and then placed two more on the leftmost square this round
        - The player takes 12 blocks of code (transparent cubes), 2 for each hacker on the leftmost square, and 4 for each hacker on the second square
- Sell code
    a. Produces cryptocurrency (yellow cubes) for each square, square's value * number of hackers on that square
    b. Hackers are placed on the leftmost square and may be advanced during phase 3 to provide more cryptocurrency in future rounds
    c. Ex: The player placed 2 hackers on sell code in round one, and advanced them in rounds 1 and 2.
        - In rounds one and two, the hackers can each sell 3 blocks of code for 1 BTC, for a total of 2 BTC
        - In round three, the hackers can each sell 3 blocks of code for 2 BTC, for a total of 4 BTC
- Trade Secret
    a. The player may sell applications, discovered vulnerabilities, and application resources a number of times equal to the value on that square
    b. Each hacker allows for one trade
        - Application resources -> 1 cryptocurrency
        - Unrevealed confirmed vulnerability -> 3 cryptocurrency
        - Application -> 5 cryptocurrency
            - If the application is running, any CPU or GPU allocation cubes on it go back to their CPU or GPU
            - Bitcoin miner can be sold, but not repurchased
        - Convert 2 different application resource to 1 of any resource
        - Convert 3 of the same application resource to 1 of any resource
- Build application
    a. If multiple players take this action, players secretly bid how many blocks of code they are spending
        - As players finish getting their resources from the previous four actions, they secretly choose how many blocks of code to bid and hold their hands out. When all players with hackers on this space have held their hands out, their bids are revealed and applications are purchased
        - The player spending the most gets first choice of which application they purchase
        - All players must return all the blocks of code they bid to the supply regardless of the cost on the offer board
        - The player must spend a minimum of the cost on the offer board, but there is no maximum that they can bid and spend
        - Players with multiple hackers building an application must allow all players with hackers on the action to build before getting their second application.
    b. Each hacker on this space allows the player to buy one application

    c. If multiple players have multiple hackers, each extra application is also bid for which applications are built.
    d. Cost in blocks of code is on the application offer boards
    e. Return to the supply a number of transparent cubes equal to the offer board
    f. As soon as an application is built, the offer refreshes.
        - Move each application to the next lowest cost until it fills the empty slot
        - Draw a new application from the deck to fill the highest cost slot in the offer
    g. A player may bid zero, and in doing so flush the offer once players who bid more had their opportunities to buy.
        - By placing two hackers it's possible to bid zero the first time and immediately buy from the new offer with the second action.
    h. In the case of a tie bid, the player with the most unspent code blocks breaks the tie and must spend one additional code block. If there is a continued tie all tied players roll a die, and the highest die role wins.
- Upgrade Machine
    a. Spend cryptocurrency to purchase a computer upgrade
        - 3 cryptocurrency -> CPU core or memory stick
            - Allows more applications to run on your computer
        - 8 cryptocurrency -> GPU
            - Counts as both a CPU core and a memory stick
            - GPU allocations are marked with light blue cubes
            - When the cryptocurrency miner is allocated to the GPU, each allocation cube provides 2 dice for cryptocurrency mining instead of just 1
            - Applications running on GPU use the memory of that GPU.
                - Ex: A player has 1 CPU, 1 Memory, and 1 GPU. That player may run one application on CPU, and one application on GPU.
        - 15 cryptocurrency -> A new hacker up to a maximum of 6 total hackers
- Disable/Enable Application
    a. The player may enable or disable a single instance of an application per hacker placed on this action
    b. An application that takes 2 or more cores to run only takes one action to enable or disable
    c. Players may not partially enable or disable an application
    d. Applications may be enabled more than once, but take the full core cost for each instance
    e. Applications can run on CPU, GPU, or both.
        - If running on both CPU and GPU, the player chooses whether to use a memory stick or the GPU for the application’s memory requirement
- Discover Vulnerability
    a. Code cost of making an attempt at discovering the vulnerability is on the offer board
    b. Return to the supply a number of transparent cubes equal to the value on the offer board
    c. Place a die of the players’ color on the vulnerability board with the die value equal to the vulnerability being discovered
        - A: 3
        - B: 4
        - C: 5
        - D: 6
- Pen/QA Test
    a. Pen Test
        - Removes a number of pips from vulnerability dice equal to the value on the square
        - Hackers are placed on the leftmost square and may be advanced during phase 3 to provide better testing efficiency
        - Ex: The player placed 1 hacker on the QA Test in round one, and advanced them in rounds 1, 2, and 3.

        - In round one, the hacker is simply performing tests looking for vulnerabilities
        - In rounds two and three, the hacker removes one pip from one of the 4 vulnerability dice
        - In round four, the hacker removes one pip from either two separate dice, or two pips from one die
    b. QA Test
        - Adds a number of pips to the vulnerability dice equal to the value on the square
        - Only after a vulnerability has been patched
        - Hackers are placed on the leftmost square and may be advanced during phase 3 to provide better testing efficiency
        - Ex: The player placed 1 hacker on the QA Test in round one, and advanced them in rounds 1, 2, and 3.
            - In round one, the hacker is simply performing tests looking for vulnerabilities
            - In rounds two and three, the hacker adds one pip to one of the 4 vulnerability dice
            - In round four, the hacker adds one pip to either two separate dice, or two pips to one die
- Confirm Vulnerability
    a. Roll a die. If the value rolled is equal to or greater than the vulnerability die, the vulnerability has been discovered.
        - Place a spare hacker standing up on that vulnerability space
        - Any player with a hacker on the vulnerability space may at any time peak at the vulnerability to remember what resources are required to patch it
- Patch Vulnerability
    a. Application resource cost of making an attempt at patching the vulnerability is on the vulnerability card.
        - Before attempts are made, flip over the cards that are attempting to be patched
    b. If multiple players take this action, multiple players may attempt to patch the same vulnerability on the same turn
    c. Each attempt consists of rolling two dice. Both dice must roll equal to or greater than the vulnerability die for the vulnerability being patched.
        - If a player has assigned multiple hackers to patching, they may retry a failed roll rather than attempt to patch a different vulnerability
    d. Once a vulnerability has been patched players return the designated application resource cubes to the supply and the top card of the patched vulnerability deck is discarded
        - This causes the same vulnerability to require new application resources to be patched again
        - Players who failed to patch the vulnerability after this round ends must use the new application resources to make another attempt on future turns
    e. Players who have already discovered the vulnerability may look at the new requirements for patching
- Release
    a. If a vulnerability has been discovered, patched, and the testing die has returned to 6, the patch can be released.
    b. Remove the laying down meeple and die, the vulnerability has been released.