White Hat

© Mattie Schraeder 2018-2021, @AFrozenPeach, mattie@mattie.lgbt





Theme

White Hat is an action selection, worker placement, and set collection style game where each player leads a team of hackers trying to find vulnerabilities in major corporations' networks. Players race against each other, as well as the nefarious Black Hat, to collect bug bounties hoping to earn the most BitCubes to win the game.

These are not your Hollywood hackers. Real hacking is done heads down at a computer, writing code, building applications to aid research, and maintaining computer hardware. White Hat does its best to simulate the real world of hacking and computer development in a way that anyone can understand and play, without having to know the intricacies of computer hacking.

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.

Components

- 1. 6 Motherboards
- 2. 1 BitHub Board
- 3. 1 Vulnerability Board
- 4. 24 CPU Tokens
- 5. 48 RAM Tokens
- 6. 12 GPU Tokens
- 7. 36 StarBits Coffee Tokens
- 8. 60 Application Cards
- 9. 15 Black Hat Cards
- 10. 100 Vulnerability Cards
- 11. 1 First Player Marker
- 12. 36 meeples
 - o 6 Red
 - o 6 Blue
 - 6 Purple
 - o 6 Green
 - 6 Yellow
 - o 6 White
- 13. 225 cubes

0

- 50 transparent clear cubes Written code
- 50 large transparent clear cubes 5x Written code
- 50 yellow cubes BitCube
- 50 large yellow cubes 5x BitCube
 - 125 application cubes
 - 25 Red
 - 25 Blue
 - 25 Purple
 - 25 Green
 - 25 Pink

Goal

Ultimately each player's goal is to have the most BitCubes at the end of the game. BitCubes are the primary currency in White Hat, and are primarily gained through completing Bug Bounties.

Each vulnerability has a set of exploits that are required to disclose it, and once disclosed are worth a Bug Bounty in BitCubes equal to five times the number of exploits required to disclose it.



Setup

- 1) Give each player a bag with the following contents:
 - a) 6 hacker meeples
 - b) 1 CPU
 - c) 1 memory
- 2) Give each player a motherboard
 - a) Place the CPU and the memory on their designated spaces
- 3) Each player starts with 3 hacker meeples. The rest are kept in their personal supply.
- 4) Place the vulnerability and application boards in the middle of the table
- 5) Shuffle each vulnerability deck and place them face up on the corresponding place on the vulnerability board
- 6) Shuffle the deck of application cards and place it face down on the draw deck spot on the application offer
- 7) Fill the application offer with the top 3 cards of the application deck
- 8) Shuffle the deck of Black Hat cards and place them face down on the Black Hat space of the vulnerability board.
- 9) Randomly select the first player and give them the first player marker.
- 10) The first player takes 2 StarBits Coffee tokens. Each player after the first gains 1 StarBits Coffee token more than the previous player, up to a maximum of 6.

The Main Program Loop

Each player, on their turn, executes the following:

 Advance hackers Place hackers Activate hackers 	Free action: Spend 1 StarBits Coffee token to advance a worker one spot along a task.
4. Run Applications	
 Claim Vulnerability Disclose Vulnerability 	

1. Advance Hackers

Any hacker already working on a task advances along the task track to the next space.

2. Place Hackers

All the hackers a player has available that are not currently working on a task are placed on action spaces, which represent various **tasks** a hacker can perform, on their individual motherboards. Hackers can team up and be working on the same tasks, or can be split up amongst several different tasks. Hackers are placed on the left most space of the chosen actions.



3. Activate Hackers

Players pick up a hacker meeple from a task to activate it. Tasks can be activated in any order. These tasks are explained later in the rulebook. Activating a hacker is optional, even if the hacker is at the last place on a track. If a player chooses to activate no hackers on their turn, they may instead take one StarBits Coffee token.

4. Run Applications

Applications running on that computer generate exploits that can be used for completing Bug Bounties. **Applications** are finished programs that perform tasks automatically as long as they are running on a computer. The color of the application card matches the color of **Exploit** that it generates.

Each application takes a certain number of **CPU cores** (the orange squares) or **GPU cores** (the blue squares) to run. If using **CPU cores**, one **memory stick** is also needed to run the application (Each



GPU counts as memory in addition to GPU cores). If using **GPU cores**, the application generates **two exploits** instead of one.

5. Claim Vulnerability

The player may claim one vulnerability from the four visible vulnerabilities for a number of code blocks based on the exploit they wish to claim. Claiming vulnerabilities means that only the player who claimed it can complete it. Claimed vulnerabilities are placed face up near their player board. Beware: the more vulnerabilities that players claim, the more targets a Black Hat has to exploit.

6. Disclose Vulnerability

If a player has exploits that match one of the four face up vulnerabilities on the vulnerability board, or any vulnerability that they have personally claimed, they can spend those exploits to disclose that vulnerability. The vulnerability is placed in a face down pile next to their player board and the player gains a number of BitCubes equal to 5x the number of exploits it took to disclose.

Tasks

Hackers get assigned to various tasks in the **Place Hackers** function. Each task represents a small bit of work that the hacker has completed, usually resulting in a resource that can be used in other tasks.

Write Code



The **Write Code** task produces **code blocks** represented by clear transparent cubes. A hacker working on the write code task produces a number of **code blocks** equal to the value of the space they are on.

In White Hat, **code blocks** represent nonfunctioning, but still useful, code that the hackers have written and can be used

for various nondescript purposes, as well as code that has been studied well enough that the hacker knows how it works without having to think about it. Individual **code blocks** aren't worth much, but when combined into useful applications, they can be used to automate work on behalf of the hacker.

Mine BitCubes



A hacker assigned to the **Mine BitCube** produces a number of BitCubes equal to the value of the space they are on. If a hacker wishes, they may discard a **GPU** installed on their computer to gain twice the listed value.

Trade



The **Trade** task allows a hacker to trade exploits, BitCubes, applications, and hardware on the BitHub. A number of trades can be taken depending on the value of the space the hacker is on. The ratio listed corresponds to the number of certain exploits or BitCubes that can be traded in order to gain the exploit in question.

Each of the following counts as one trade:

- Exploits -> Different exploit at rate listed on the action taken
- BitCubes -> Any exploit at the rate listed on the action taken
- Application -> 5 BitCube or 3 exploits
- Hardware -> Half the cost of the hardware, rounded down or 3 exploits

If an application is traded, it is placed on top of the Black Hat deck rather than in the discard pile.

Build Application



Hackers assigned to this task build a single application when activated. They may draw a number of cards from the Application Deck based on the value of the action taken. Applications each have a cost, in Code Blocks, in the top left corner of the card.

That player may build a face up application in the BitHub for

a discount, or one of the cards drawn. When an application is built from the BitHub, applications slide down filling the gap, and a new application card is revealed on the -1 space. Any drawn cards are then discarded to the discard pile.

Before drawing cards or choosing a BitHub application, the player may choose to spend 3 code blocks to remove all three cards in the BitHub and add them to the Black Hat deck. Alternatively, they can choose to shift the -3 space application into the Black Hat deck, and refill the BitHub.

Shopping



The **Shopping** task allows the hacker to either purchase a new component for their computer, or hire a new hacker to join their team. A number of individual purchases can be made depending on the value of the action taken.

- 3 BitCubes -> purchase 6 StarBits Coffee
 - A player can never have more than 6 **StarBits Coffee** tokens at any given time.
 - As a free action, a single **StarBits Coffee** token can be spent to move a hacker one additional space along a task track.
- 8 BitCubes -> purchase one CPU or Memory
 - CPU runs applications up to capacity
 - Memory required for each application running on CPU
- 15 BitCubes -> purchase one GPU
 - \circ $\,$ Counts as both memory and GPU for an application running on it
 - Generates twice as many exploits
- 25 BitCubes -> hire one additional hacker
 - A total of 3 additional hackers may be purchased throughout the game, bringing a player's team count to 6 hackers. The newly hired hacker can be placed in the next round's Place Hackers function.

Black Hat's Turn

After each player has taken their turn, the Black Hat takes a turn. The Black Hat is an AI that acts as the end game timer. When the black hat has completed a certain number of exploits, the game ends.

- 1. Shuffle the Black Hat deck and reveal the top card
- 2. If the revealed card is a colored application card, take one exploit matching the revealed card and add it to the Black Hat space on the vulnerability board. If a wild is revealed, the Black Hat takes the cube that will complete a vulnerability the fastest. If two vulnerabilities are tied, the higher vulnerability takes precedence.
- 3. If the black hat has enough exploits to exploit any face up vulnerability they claim that vulnerability and the game is one step closer to ending.

Scoring

The player with the most BitCubes at the end of the game wins.