**Hacker-TCG Rules version 0.0.2a**

**Overview**: you play as a white hat, gray hat or black hat and you are trying to reach your hacker's objective. Objectives range from crippling your opponents system, locking down your system, or compromising sensitive information. The objective it indicated on the hacker's card.

**Card Types**

* + **Hacker**: the hacker card represents the player, his abilities, and his motivations.
    - Card Layout
  + **Name**
  + **Hat**: white, gray, or black.
  + **Proficiencies**: all the OS's the user is proficient with. If a user isn't proficient with an OS, they can't use it effectively.
  + **Objective**: the condition for victory.
  + **CPU Goddess**: this card represents the current machine being used by the hacker. This card is placed in the CPU field.
    - Card Layout
  + **Name**
  + **Grade**: this symbol represents the grade of the CPU Goddess. Higher grades have better stats.
  + **Compatibility**: this symbol indicates what upgrades this system can use.
  + **Architecture**: this represents what software and operating systems a the CPU Goddess can run.
  + **CPU**: this is the number of cycles pass on this players turn as well as the number of cores.
  + **RAM**: this is the number of RAM slots the user has available to play tools into.
  + **HDD**: this is the maximum size of the player’s hand.
  + **SLT**: number of upgrades this CPU Goddess can accomodate.
  + **Additional Rules**: any special abilities the CPU has.
  + **Upgrade**: this card represents and upgrade to the CPU Goddess. These cards are placed under the CPU Goddess card.
    - Card Layout
  + **Name**
  + **Compatibility**: this represents the CPU type/types that can use this upgrade.
  + **Upgrade**: on the left side of the card, it lists the upgrades this confers to the CPU Goddess.
  + **OS**: this card represents the current OS being used by the CPU Goddess. This card is placed in the OS field.
    - Card Layout
  + **Name**
  + **REPO**: the number of cards in your deck.
  + **Architecture**: this represents the CPU architecture this OS is compatible with.
  + **Additional Rules**: any special abilities the OS has.
  + **Package**: this card represents libraries software requires as dependencies. These cards are places under the OS card.
    - Card Layout
  + **Name**
  + **Dependency**
  + **Software**: this card represents a tool used by the hacker to complete his objectives.
    - Card Layout
  + **Name**
  + **Color**
    - **Black**
      * Represents a malicious piece of software that is injected by a black hat’s software.
    - **White**
      * Represents any non-malicious piece of software run by a player.
  + **Hat**: the color hat can use this software guiltlessly.
  + **OS**: the OS that can run this software.
  + **Dependencies**: what libraries you must have installed on your OS to use this software.
  + **Cycle**: how many cycles must pass for this software to EXE.
  + **Vulnerability/Exploit**: this represents the weaknesses of the software.
  + **Protocol**: this represents the protocol the software card uses.
  + **Port**: this represents the port the software runs on.
  + **Class**: the type of software.
  + **Effect**: any effects or rules the card adds when in play.
  + **Type**
    - **Daemon (D)**: runs until process is killed.
      * Consumes cycles every turn.
      * Takes precedence over Execute software.
    - **Execute (E)**: runs once.
      * Consumes cycles the first time it is run.
  + **RAM Blank**: this card is used to cover any used RAM slots.
  + **RAM None**: this card is used to cover any unavailable RAM slots.
  + **Clock Counters**: these tokens are placed on running software to indicate how many cycles have passed.
    - Once a card has a number of clock counters equal to its Cycle number, it's EXE effect will activate.
  + **Password Token**: this allows you to play one card directly to your opponents RAM field.

**Card Effects**

* **NIT**: this indicates how the card must be played. (example: card must be played under a running process).
* **EXE**: this is the cards effect every time the number of clock counters on the card equals its cycle number.
  + Note: this effect doesn’t apply if the card is flipped or halted.
* **CONT**: this is a continuous effect.
  + Note: this effect doesn’t apply if the card is flipped or halted.
* **COND**: this is an effect that fires off when specified conditions are met.
* **FLIP:** some cards will flip from the face down position when a condition is met. This effect activates at that time.

**Terms**

* **Process**: a running program in the RAM Field
* **Halted**: a process that has been turned upside down (not flipped). This process doesn’t consume CPU cycles.

**Building a Deck**

When building a deck, you must take three things into account. Your chosen Hacker will determine what OS's you can use. The OS will determine what architecture your CPU Goddess must have and what Libraries and Software you can use in your deck. You also want to make sure the cards you use work towards your Hacker's objective.

**Deck Building Rules**:

1. Only 4 of each card.
2. Deck must be the size indicated on the CPU Goddess (SAN stat).

**White Hat Decks**: white hat decks tend to revolve around gathering evidence to indict the attacking hacker (Forensics win). White hat decks cannot play against another White Hat deck. These decks can use White Hat or Gray Hat cards.

**Gray Hat Decks**: gray hat decks tend to revolve around defending yourself (Forensics win), and fighting back to try to shut down the attacker (Infection win). These decks can use weaker White Hat and Black Hat Software cards, but not the stronger Software exclusive to each.

**Black Hat Decks**: black hat decks focus completely on attacking. These decks are usually centered around infecting a computer (Infection win) or compromising sensitive data (System Compromise win). These decks can use Gray Hat cards and Black Hat cards.

**Play Stages**

1. **Draw Phase**:
   * Draw a card from the SAN.
2. **Upgrade Phase**:
   * Play any number of upgrade cards under CPU Goddess.
   * Play any number of library cards under OS.
   * Play a CPU Goddess, one grade or higher than the current one and reshuffle all programs, upgrades, and libraries.
3. **Main Phase**:
   * Play any number of cards to the RAM field.
   * You may move the running programs around in RAM. Programs are given Clock Tokens in order and execute in order.
4. **Clock Tick Phase**:
   * Place one Clock token on each running program in order for each Cycle your CPU Goddess has.
   * If you have any left over after placing one on each running program, start over from the beginning and repeat.
5. **Resolution Phase**:
   * Each running program that has a number of Clock tokens on it equal to its Cycle number, that program starts running.
   * If a running program is an Execute program and it activates this turn, you place it into the trash can.
   * If a running program is a Daemon program, it remains in play and you remove the Clock Tokens from the card.
   * If a black card is removed from one of your RAM slots, take out of play and keep it until end of game.
   * If a card is taken by the effect of a card, remove from play and place to the side of the player who played the black card.
6. **Discard Phase**:
   * Discard down to HDD size.

**Starting Setup**

* Keep initial Grade 0 CPU Goddess card, Hacker card, and OS card separate from deck.
* Shuffle all other cards (excluding tokens and blanks) into the SAN (deck).
* Place CPU Goddess card onto CPU Field.
* Place Hacker card onto Hacker Field.
* Place OS card onto Hacker Field.

**Conditions for Losing**

* This losing condition is a given for any Hacker, regardless of their objective.
* Any player who has all RAM slots consumed by black cards loses via "Infection".

**Conditions for Winning**

* First and foremost, the Hacker's objective is the one that must be completed to win.
  + The following are some examples:
    - Any player who has captured x cards from the other player's SAN wins via "System Compromise".
    - Any player who has captured x black cards from their opponent wins via "Forensics".
    - Any player who has caused a player to lose via "Infection" wins via "Infection".
  + These examples are not all inclusive.

**Strategies for Winning:**

* Be Charlie Shean
* You don’t have to only be on the defensive or the offensive as long as you complete your objective.
* Capture opponent’s cards through the use of tools, or scanners to gain forensic evidence.
* Use captured cards to get password tokens and deliver cards into your opponents RAM Field.
* Be careful what software you install. Vulnerabilities will give your opponent the ability to deliver cards into your RAM Field via exploits!
* Patch vulnerabilities as quickly as possible!

**Additional Rules**

* Physical Memory: you can gain additional RAM slots by reducing your HDD size by 2 for each slot.
* What the card says goes: all rules can be broken by the effect of a card.

**Delivering Cards to Opponents RAM Field**

All non-trivial Daemon programs have vulnerabilities than a hacker can exploit. As long as a user is running this software, an opponent can exploit this vulnerability and deliver a card from their hand into the players RAM Field.

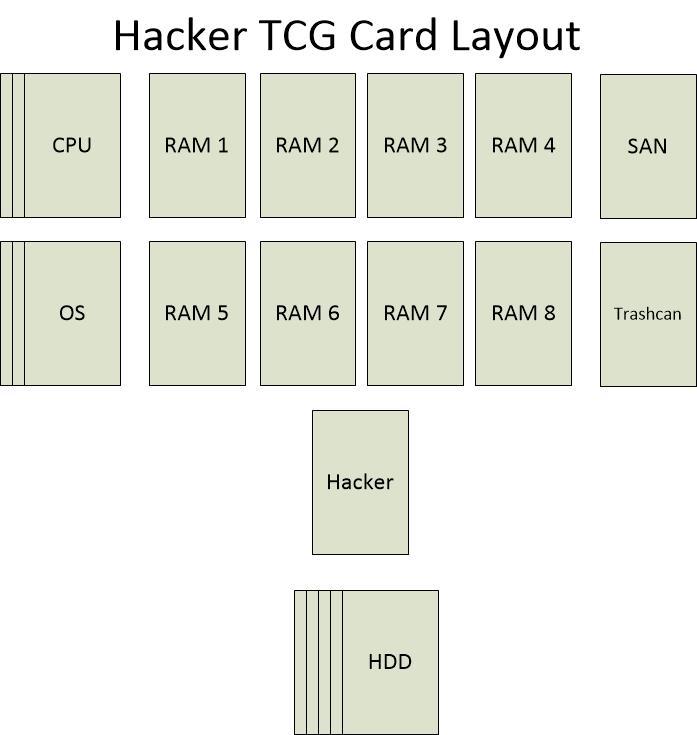
If an exploit has a protocol and/or port requirement, then the target of the exploit must have a process running on that port and/or protocol.

If a rootkit is installed and the opponent has network access to the port/protocol of the rootkit, they can deliver cards into the affected players RAM Field. The rootkit in question will show how many cards they can play into the RAM field each time it executes.

A captured card can be analyzed by programs like John the Ripper and converted into a password token. A password token can be used to delivery one card into the opponents RAM Field. Alternatively it can be used to pay the cost of some cards like ssh.

**Types of Software**

* **White**:
  + Scanner: these cards are generally played face down. And will have an effect when flipped.
  + Firewall: this limits access to the players CPU and disables cards that attack a specific protocol.
* **Black**:
  + Rootkit: allows continued access into opponents RAM Field until rootkit is removed.
  + Malware: software that runs in opponents RAM Field.
  + Delivery: delivers of another black card into the opponents RAM field if certain conditions are met.
* **Gray**:
  + Tool: a piece of software that can either be used to do a variety of actions.
  + Technique: a skill that is not necessarily related to computes (i.e. social engineering)
  + Server: a program that confers benefits to the player.



**Play Field Areas**

**Hacker Field**: the deck's chosen hacker.

**CPU Field**: the deck's chosen CPU Goddess and all upgrades are played here. Upgrade cards are placed under the CPU leaving the left side of the cards visible. This side of the card contains a condensed form of what the upgrade confers. It also has a symbol to match up with the CPU type of the CPU Goddess it is running under.

**OS Field**: the deck's chosen OS. Library cards are played under this card. The side of the card contains the symbol of the library (which matches up against the dependencies of the right side of each software card). Also included is the OS symbol which matches up with the OS card's symbol.

**RAM Fields**: A player can play as much software as can fit into the available RAM slots (size is determined by CPU Goddess's RAM stat).

**SAN Field**: A player's deck (size is determined by CPU Goddess's SAN stat).

**HDD Field**: A player's hand (size is determined by CPU Goddess's HDD stat).

**Trash Can**: A player's discard pile.

**Examples**

If I have a CPU with an 8GHz CPU and I have 4 processes that require 3, 3, 2, and 1 to execute...

Before Main Phase:

0/3, 0/3, 0/2, 0/1

After Main Phase:

3/3, 2/3, 2/2, 1/1

As you can see when a processes cycle count is met, the cycle is then moved forward to the next process. So on the resolution step process 1, 2, and 4 will execute. If you rearrange their order, you can cause many different effects!