Hacking

Hacking is the process of creating and modifying programs in the program memory of your flotation device and using those programs to accomplish various tasks.

Hacking Challenges

The basic unit of work in this mechanic is called a “Hacking Challenge.” Anytime you attempt to do anything non-trivial you will need to attempt a hacking challenge. To do this, you will need a deck of playing cards. If you have this green sheet, you should try to bring one to game. The GMs will have a small number to lend out, but you shouldn’t count on that.

All hacking challenges have a difficulty level which corresponds to a target poker hand. Your goal is to achieve that poker hand using the following method:

1. Shuffle your deck of cards 7 times.
2. Take your deck and place it in front of you.
3. Deal out 5 cards in front of you, face up. This is your working hand.
4. If your working hand meets the requirements of your target hand, you have successfully accomplished the challenge. If not, continue to the next step.
5. Discard a card face down from your working hand.
6. Deal 5 more cards from your deck. This is your dynamic library.
7. Take one card from your dynamic library and place it into your working hand.
8. Discard the other 4 cards of your dynamic library face down.
9. Repeat from step 4 above. If you run out of cards, decide to give up, or are attacked or otherwise interrupted, you fail the challenge. You may not attempt any other challenges again for a full minute and may suffer other consequences depending on what you were trying to do.

Here is the table of hand requirements for differing difficulty levels of challenges:

<table>
<thead>
<tr>
<th>Challenge Level</th>
<th>Hand Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 or less</td>
<td>Automatic (no hand required)</td>
</tr>
<tr>
<td>1</td>
<td>Pair</td>
</tr>
<tr>
<td>2</td>
<td>Three of a Kind</td>
</tr>
<tr>
<td>3</td>
<td>Four of a Kind</td>
</tr>
<tr>
<td>4</td>
<td>Straight Flush</td>
</tr>
<tr>
<td>5</td>
<td>Straight Flush + Pair</td>
</tr>
<tr>
<td>6</td>
<td>Straight Flush + Three of a Kind</td>
</tr>
<tr>
<td>7</td>
<td>Straight Flush + Four of a Kind</td>
</tr>
<tr>
<td>8 or more</td>
<td>Impossible</td>
</tr>
</tbody>
</table>

“Straight Flush + . . .” entries mean that you must first get a straight flush in your working hand and then continue swapping

---

1 If you have fewer than 5 cards left in your deck, just deal as many as you have left. However, you may not choose to deal less than 5 if you have at least 5 available.
in cards from your dynamic library until you have changed the straight flush into the second hand. You do not get to re-shuffle between the two hands. Yes, this means that a level seven challenge is very hard.

Programs and Memory

As we mentioned above, hacking is the art of creating programs and using them to do stuff. A hacking program consists of a sequence of up to seven capital letters with no letters repeated. The ordering of the letters is unimportant.

Your flotation device has enough memory to store up to three programs at any time. If you already have three programs and want to do something that would give you another program, you must delete one of your existing programs first. Deleting a program is instantaneous, always successful and may be done as often as you like. However, it is also irreversible (other than through going through the work to recreate the program from scratch), so think carefully before you do it.

Known Letters

An important concept to the rest of this mechanic is the idea of “known letters.” This is a list of the letters that you know how to use in creating and modifying programs. Your known letters will be listed on your hacking ability card. Whenever this sheet talks about “letters that you know how to use” or similar, it is referring to the letters listed on your ability card.

Ways to Get Programs

There are four basic ways to get a program:

- **Write a New Program:** You may create a new program and save it to a blank slot in your flotation device memory. The new program may be as long as you like (up to the seven letter maximum) but it may only contain letters that you know how to use and may not contain any duplicate letters. To create the program you must perform a hacking challenge with difficulty equal to the number of letters in the program you are trying to write. If you succeed at the hacking challenge, the new program is saved to memory. If you fail, you don’t get the program and cannot attempt to create that particular program again for at least 24 hours. You may attempt to create other programs so long as the differ by at least 1 letter.

- **Modifying an Existing Program:** You may modify a program you have saved in memory by replacing any letter in the program with any letter that you know, provided that the new letter does not already appear in the program. You do not need to know the letter that you are replacing or any other letters in the program. Note that you may not change the length of the program, only which letters it contains. To modify a program you must perform a hacking challenge with difficulty equal to the number of letters in the program minus 2. If you succeed at the hacking challenge, the program is modified in memory. If you fail, you are unable to modify that program and cannot attempt to modify that particular program for at least 24 hours. You may attempt to modify other programs so long as they differ by at least 1 letter.

- **Copying an Existing Program:** Sometimes you might want to make a copy of a program (if you wanted to modify it and still have the original, for example). Copying a program is instantaneous and always successful, but requires a free memory slot to hold the copy.

- **Be Sent a Program from Someone Else:** People may send programs to other people with flotation devices. If you want to send someone a program, simply tell them you are sending them a program and what the contents of the program are (i.e. what letters are in it). You must be within ZOC of them to do this. If someone sends you a program, you may save it to memory provided that you have a blank slot available. Sending or receiving a program is this way is always successful (assuming the recipient has a memory slot available). You may deny a program being sent to you.
Things to Do with Programs

There are four things that you can do with a program that you have saved in memory. They are:

- **Defeat Computer Security Protocols:** You may encounter mechanics in the game that assert that you can do something if you can “defeat computer security protocol <foobar>" (where <foobar> is some sequence of capital letters). You may do this with an appropriate program and hacking challenge. The program that you use must be in memory and must either include all the letters in the security protocol or all but one of the letters. If it has all the letters, the difficulty of the hacking challenge is the length of the security protocol minus two. If it has all but one letter, the difficulty is simply the length of the security protocol. If you succeed at the hacking challenge, you have defeated the security protocol (the mechanic that told you what the protocol was should also tell you what you get for defeating it). If you fail, the security protocol backlashes against your flotation device and deletes the program you tried to use from memory. You may try again to defeat the security protocol if you have or can create another suitable program.

- **Perform Simulations:** You may encounter mechanics in the game that assert that you can do something if you can "perform computer simulation <foobar>" (where <foobar> is some sequence of capital letters). Once again, you may do this with an appropriate program and hacking challenge. Unlike with security protocols, performing simulations requires a program that is an exact match (i.e. the program you are using must include all the letters in the simulation). The difficulty of the hacking challenge to run a simulation is equal to the number of letters in the simulation. If you succeed at the hacking challenge, you have successfully run the simulation (the mechanic that told you what the simulation was should also tell you what you get for defeating it). If you fail, you fail to run the simulation. You may not attempt to run that particular simulation again for at least 24 hours.

- **Implanting Worms in Networks:** You may encounter items described as “computer terminal that acts as a trusted node on the <network> computer network.” If you can physically access such a computer you may implant a worm on it that will search the network for data on a specific topic and then save it so that you can pick it up later. The computer will have a table with two columns on it, labeled S and D. When you implant a worm, you should write the subject of the worm in the S column and the defense program you are using in the D column. The defense program can be any program you have in memory. Implanting a worm is automatic; there is no hacking challenge. The challenge is in creating a defense program ahead of time that will be hard for others to duplicate; see “Destroying Network Worms” below. When you implant a worm, you should make a note of the time (you don’t need to write it on the computer, but you do need to keep track of it). At any time that is at least 24 hours after you initially implanted the worm, you may come back to pick up your data. To pick up the data, you must gain physical access to the computer again and your worm must not have been destroyed in the meanwhile (see below). Assuming that your worm is still there, you can remove it and collect your data by crossing out it’s line on the computer table. When you do this, see a GM and we’ll get you your data. Alternatively, you may wish to tell us about worms you’ve launched in advance so that we can preemptively give you your info in a sealed packet. Then you can open your packet once you retrieve your worm and get the data right away.

- **Destroying Network Worms:** You may also use your hacking skills to destroy worms written by other hackers from computers you encounter. To destroy a worm, you must defeat the security protocol of the worm (the entry in the D column) as described above under “defeat computer security protocols.” If you succeed at the hacking challenge to defeat the security protocol, you have removed the worm. Cross out its line in the table so its owner will know it has been destroyed. If you fail at the hacking challenge, your program is deleted (as normally happens when you fail to defeat a security protocol) and you can not attempt to destroy that particular worm again for 24 hours.

The following table summarizes the difficulties of various tasks:
<table>
<thead>
<tr>
<th>Task</th>
<th>Difficulty Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>create a new program</td>
<td>program length</td>
</tr>
<tr>
<td>modify a program</td>
<td>program length minus two</td>
</tr>
<tr>
<td>defeat security protocol with an exact match</td>
<td>program length minus two</td>
</tr>
<tr>
<td>defeat security protocol with a partial match</td>
<td>program length</td>
</tr>
<tr>
<td>run a simulation</td>
<td>simulation length</td>
</tr>
</tbody>
</table>

Some Examples

Example 1

Bob knows the letters ABCE and has no programs in memory. He wants to defeat security protocol ABEF. He decides to do this by writing program ABE. This program is 3 letters long so the hacking challenge difficulty is 3. Bob succeeds at this challenge and now has the program ABE in memory. This program contains all but one letter of the security protocol so Bob can perform a difficulty 4 hacking challenge to defeat the security protocol (because the security protocol is 4 letters long).

Example 2

Bob knows the letters ABCE and has the program BFGHI saved in memory. He wants to defeat security protocol ABEF. He cannot use his saved program because it is missing two letters. He could modify his saved program by replacing the one of the letters he doesn’t need (lets say the G) with one that he does need (lets say the E). He is modifying a length 5 program so he must complete a difficulty 3 hacking challenge. Lets assume he succeeds at this. His memory now contains the program BFEHI which is only one letter off from what he wants. He can now perform a difficulty 4 hacking challenge to defeat the security protocol (because the security protocol is 4 letters long). Alternatively, he could perform another difficulty 3 hacking challenge to modify the H (for example) into an A. If he succeeded at this he, would then have the program BFEAI in memory. Since this program contains all of the letters in the security protocol, Bob would only need to succeed at a level 2 challenge (4 minus 2) to defeat the protocol.

Searching for Information

If you are looking for general information, implant a worm into what you believe is the correct network with the subject of the information you want. The data you get back will likely be a file or document that is most relevant to your search. Different networks have different information (if any at all) on the same subject, so make sure you find the right network to search.

Looking for Scandals

Another common use for network worms is to try to find dirt or scandals on other people. To do this, implant a worm with the subject “scandal on <name>” (where <name> is the name of a character you want dirt on). The data you get back will then point you towards any scandalous information about that person that this network knows about. Once again, the data you get back is based on the network on which your ran the worm, you’ll want to do this on a network that has some connection to the person you want dirt on.

Deleting Scandals

After (and only after) you have successfully found dirt on a character, you may choose to delete all evidence of the scandal. Deleting a scandal is just like searching for one, except your worm’s subject should say “delete scandal on <name>.” Inform the
GMs 24 hours after the deletion, or tell us when 24 hours will pass.

Some information is protected by extra security. Although your worms are powerful enough to give you read access anywhere, they may not be able to achieve write access on all high-level information. A delete command may return failure if the security is too great.

**Learning New Letters**

You may add new letters to your “known letters” list through the following process. Decide which letter you want to learn and count how many letters come before it in alphabetical order that you don’t already know. Divide this number by three (round up) to get the number of lessons it will take to learn the new letter.

Now you must find someone who already knows the letter and get them to agree to teach you. This person can teach you a lesson in the use of that letter by sending you a program with that letter and by watching you use that program to do something useful (either defeating a security protocol, running a simulation or implanting a worm). The program that you receive for each lesson must be different from the program used on any previous lesson.

You may only have one lesson per day and may only be working on learning one letter at a time, but you may receive lessons from as many different teachers as you like. Once you have had enough total lessons (i.e. number of earlier letters you don’t know divided by three), you learn the letter and may add it to your known letters list permanently.

Note that you can only have one lesson per day total and not one per letter. This means that you can not learn more than one new letter per day, even if all the letters only take one lesson.

**Hacking and Netdiving**

You may use netdiving (see your Net Diving green sheet for details) to augment your hacking skills. You may generate the following effects with netdiving:

- **With a white dive,** you can load a “trivial program” into an empty memory slot in your flotation device. A trivial program is any program that starts with the letter A and contains only consecutive letters (e.g. ABC or ABCDEFG). This process is automatically successful and there is no hacking challenge.

- **With a green dive,** you may attempt to create a program that contains up to one letter that you don’t know. You must still perform the hacking challenge to create the program as normal. The difficulty of the hacking challenge is unchanged and, if you fail, the dive is wasted. If you could learn the letter within 2 days, this effect only requires a white dive to produce.

- **With a yellow dive,** you may attempt to modify a program by replacing any letter with another letter that you don’t know. You must still perform the hacking challenge to create the program as normal. The difficulty of the hacking challenge is unchanged and, if you fail, the dive is wasted. If you could learn the letter within 2 days, this effect only requires a green dive to produce.

- **With a red dive,** you may instantly and automatically create any program of up to seven letters with up to three letters that you don’t know in it. This process is automatically successful and there is no hacking challenge.