Back to Eric's FAQs Page Up to Site Map $Date: 1997/12/01 18:18:01 $

How To Become A Hacker

Why This Document?

As editor of the Jargon File, I often get email requests from enthusiastic network newbies asking (in effect) "how can I learn to be a wizard hacker?". Oddly enough there don't seem to be any FAQs or Web documents that address this vital question, so here's mine.

If you are reading a snapshot of this document offline, the current version lives at href="http://www.ccil.org/~esr/faqs/hacker-howto.html.

What Is A Hacker?

The Jargon File contains a bunch of definitions of the term `hacker', most having to do with technical adeptness and a delight in solving problems and overcoming limits. If you want to know how to *become* a hacker, though, only two are really relevant.

There is a community, a shared culture, of expert programmers and networking wizards that traces its history back through decades to the first time-sharing minicomputers and the earliest ARPAnet experiments. The members of this culture originated the term `hacker'. Hackers built the Internet. Hackers made the UNIX operating system what it is today. Hackers run Usenet. Hackers make the World Wide Web work. If you are part of this culture, if you have contributed to it and other people in it know who you are and call you a hacker, you're a hacker.

The hacker mind-set is not confined to this software-hacker culture. There are people who apply the hacker attitude to other things, like electronics or music -- actually, you can find it at the highest levels of any science or art. Software hackers recognize these kindred spirits elsewhere and may call them "hackers" too -- and some claim that the hacker nature is really independent of the particular medium the hacker works in. But in the rest of this document we will focus on the skills and attitudes of software hackers, and the traditions of the shared culture that originated the term `hacker'.

There is another group of people who loudly call themselves hackers, but aren't. These are people (mainly adolescent males) who get a kick out of breaking into computers and phreaking the phone system. Real hackers call these people `crackers' and want nothing to do with them. Real hackers mostly think crackers are lazy, irresponsible, and not very bright, and object that being able to break security doesn't make you a hacker any more than being able to hotwire cars makes you an automotive engineer. Unfortunately, many journalists and writers have been fooled into using the word `hacker' to describe crackers; this irritates real hackers no end.

The basic difference is this: hackers build things, crackers break them.

If you want to be a hacker, keep reading. If you want to be a cracker, go read the alt.2600 newsgroup and get ready to do five to ten in the slammer after finding out you aren't as smart as you think you are. And that's all I'm going to say about crackers.

The Hacker Attitude

Hackers solve problems and build things, and they believe in freedom and voluntary mutual help. To be accepted as a hacker, you have to behave as though you have this kind of attitude yourself. And to behave as though you have the attitude, you have to really believe the attitude.

But if you think of cultivating hacker attitudes as just a way to gain acceptance in the culture, you'll miss the point. Becoming the kind of person who believes these things is important for *you* -- for helping you learn and keeping you motivated. As with all creative arts, the most effective way to become a master is to imitate the mind-set of masters -- not just intellectually but emotionally as well.

So, if you want to be a hacker, repeat the following things until you believe them:

1. The world is full of fascinating problems waiting to be solved.

Being a hacker is lots of fun, but it's a kind of fun that takes lots of effort. The effort takes motivation. Successful athletes get their motivation from a kind of physical delight in making their bodies perform, in pushing themselves past their own physical limits. Similarly, to be a hacker you have to get a basic thrill from solving problems, sharpening your skills, and exercising your intelligence.

If you aren't the kind of person that feels this way naturally, you'll need to become one in order to make it as a hacker. Otherwise you'll find your hacking energy is sapped by distractions like sex, money, and social approval.

(You also have to develop a kind of faith in your own learning capacity -- a belief that even though you may not know all of what you need to solve a problem, if you tackle just a piece of it and learn from that, you'll learn enough to solve the next piece -- and so on, until you're done.)

2. Nobody should ever have to solve a problem twice.

Creative brains are a valuable, limited resource. They shouldn't be wasted on re-inventing the wheel when there are so many fascinating new problems waiting out there.

To behave like a hacker, you have to believe that the thinking time of other hackers is precious -- so much so that it's almost a moral duty for you to share information, solve problems and then give the solutions away just so other hackers can solve *new* problems instead of having to perpetually re-address old ones.

(You don't have to believe that you're obligated to give *all* your creative product away, though the hackers that do are the ones that get most respect from other hackers. It's consistent with hacker values to sell enough of it to keep you in food and rent and computers. It's consistent to use your hacking skills to support a family or even get rich, as long as you don't forget you're a hacker while you're doing it.)

3. Boredom and drudgery are evil.

Hackers (and creative people in general) should never be bored or have to drudge at stupid repetitive work, because when this happens it means they aren't doing what only they can do -- solve new problems. This wastefulness hurts everybody. Therefore boredom and drudgery are not just unpleasant but actually evil.

To behave like a hacker, you have to believe this enough to want to automate away the boring bits as much as possible, not just for yourself but for everybody else (especially other hackers).

(There is one apparent exception to this. Hackers will sometimes do things that may seem repetitive or boring to an observer as a mind-clearing excercise, or in order to acquire a skill or have some particular kind of experience you can't have otherwise. But this is by choice -- nobody who can think should ever be forced into boredom.)

4. Freedom is good.

Hackers are naturally anti-authoritarian. Anyone who can give you orders can stop you from solving whatever problem you're being fascinated by -- and, given the way authoritarian minds work, will generally find some appallingly stupid reason to do so. So the authoritarian attitude has to be fought wherever you find it, lest it smother you and other hackers.

(This isn't the same as fighting all authority. Children need to be guided and criminals restrained. A hacker may agree to accept some kinds of authority in order to get something he wants more than the time he spends following orders. But that's a limited, conscious bargain; the kind of personal surrender authoritarians want is not on offer.)

Authoritarians thrive on censorship and secrecy. And they distrust voluntary cooperation and information-sharing -- they only like `cooperation' that they control. So to behave like a hacker, you have to develop an instinctive hostility to censorship, secrecy, and the use of force or deception to compel responsible adults. And you have to be willing to act on that belief.

5. Attitude is no substitute for competence.

To be a hacker, you have to develop some of these attitudes. But copping an attitude alone won't make you a hacker, any more than it will make you a champion athlete or a rock star. Becoming a hacker will take intelligence, practice, dedication, and hard work.

Therefore, you have to learn to distrust attitude and respect competence of every kind. Hackers won't let posers waste their time, but they worship competence -- especially competence at hacking, but competence at anything is good. Competence at demanding skills that few can master is especially good, and competence at demanding skills that involve mental acuteness, craft, and concentration is best.

If you revere competence, you'll enjoy developing it in yourself -- the hard work and dedication will become a kind of intense play rather than drudgery. And that's vital to becoming a hacker.

Basic Hacking Skills

The hacker attitude is vital, but skills are even more vital. Attitude is no substitute for competence, and there's a certain basic toolkit of skills which you have to have before any hacker will dream of calling you one.

This tookit changes slowly over time as technology creates new skills and makes old ones obsolete. For example, it used to include programming in machine language, and didn't until recently involve HTML. But in late 1996 it pretty clearly includes the following:

1. Learn how to program.

This, of course, is the fundamental hacking skill. In 1997 the one language you absolutely must learn is C (though it's not the one to try learning first thing). But you aren't a hacker or even merely a programmer if you only know one language -- you need to learn how to think about programming problems in a general way, independent of any one language. To be a real hacker, you need to have gotten to the point where you can learn a new language in days by relating what's in the manual to what you already know. This means you should learn several very different languages.

Besides C, you should also learn at least LISP and Perl (and Java is pushing hard for a place on the list). Besides being the most important hacking languages, these each represent very different approaches to programming, and all will educate you in valuable ways.

I can't give complete instructions on how to learn to program here -- it's a complex skill. But I can tell you that books and courses won't do it (many, maybe *most* of the best hackers are self-taught). What will do it is (a) *reading code* and (b) *writing code*.

Learning to program is like learning to write good natural language. The best way to do it is to read some stuff written by masters of the form, write some things yourself, read a lot more, write a little more, read a lot more, write some more ... and repeat until your writing begins to develop the kind of strength and economy you see in your models.

Finding good code to read used to be hard, because there were few large programs available in source for fledgeling hackers to read and tinker with. This has changed dramatically; free software, free programming tools, and free operating systems (all available in source, and all built by hackers) are now widely available. Which brings me neatly to our next topic...

2. Get one of the free UNIXes and learn to use and run it.

I'm assuming you have a personal computer or can get access to one (these kids today have it so easy :-)). The single most important step any newbie can take towards acquiring hacker skills is to get a copy of Linux or one of the free BSD-Unixes, install it on a personal machine, and run it.

Yes, there are other operating systems in the world besides Unix. But they're distributed in binary -- you can't read the code, and you can't modify it. Trying to learn to hack on a DOS or Windows machine or under MacOS is like trying to learn to dance while wearing a body cast.

Besides, Unix is the operating system of the Internet. While you can learn to use the Internet without knowing Unix, you can't be an Internet hacker without understanding it. For this reason, the hacker culture today is pretty strongly Unix-centered. (This wasn't always true, and some old-time hackers aren't happy about it, but the symbiosis between Unix and the Internet has become strong enough that even Microsoft's muscle doesn't seem able to seriously dent it.)

So, bring up a Unix -- I like Linux myself but there are other ways. Learn it. Run it. Tinker with it. Talk to the Internet with it. Read the code. Modify the code. You'll get better programming tools (including C, Lisp, and Perl) than any Microsoft operating system can dream of, you'll have fun, and you'll soak up more knowledge than you realize you're learning until you look back on it as a master hacker.

For more about learning Unix, see The Loginataka.

To get your hands on a Linux, see the Where To Get Linux.

3. Learn how to use the World Wide Web and write HTML.

Most of the things the hacker culture has built do their work out of sight, helping run factories and offices and universities without any obvious impact on how non-hackers live. The Web is the one big exception, the huge shiny hacker toy that even *politicians* admit is changing the world. For this reason alone (and a lot of other good ones as well) you need to learn how to work the Web.

This doesn't just mean learning how to drive a browser (anyone can do that), but learning how to write HTML, the Web's markup language. If you don't know how to program, writing HTML will teach you some mental habits that will help you learn. So build a home page.

But just having a home page isn't anywhere near good enough to make you a hacker. The Web is full of home pages. Most of them are pointless, zero-content sludge -- very snazzy-looking sludge, mind you, but sludge all the same (for more on this see The HTML Hell Page).

To be worthwhile, your page must have *content* -- it must be interesting and/or useful to other hackers. And that brings us to the next topic...

Status in the Hacker Culture

Like most cultures without a money economy, hackerdom runs on reputation. You're trying to solve interesting problems, but how interesting they are, and whether your solutions are really good, is something that only your technical peers or superiors are normally equipped to judge.

Accordingly, when you play the hacker game, you learn to keep score primarily by what other hackers think of your skill (this is why you aren't really a hacker until other hackers consistently call you one). This fact is obscured by the image of hacking as solitary work; also by a hacker-cultural taboo (now gradually decaying but still potent) against admitting that ego or external validation are involved in one's motivation at all.

Specifically, hackerdom is what anthropologists call a *gift culture*. You gain status and reputation in it not by dominating other people, nor by being beautiful, nor by having things other people want, but rather by giving things away. Specifically, by giving away your time, your creativity, and the results of your skill.

There are basically five kinds of things you can do to be respected by hackers:

1. Write free software.

The first (the most central and most traditional) is to write programs that other hackers think are fun or useful, and give the program sources to the whole hacker culture to use.

Hackerdom's most revered demigods are people who have written large, capable programs that met a widespread need and given them away, so that now everyone uses them.

2. Help test and debug free software

They also serve who stand and debug free software. In this imperfect world, we will inevitably spend most of our software development time in the debugging phase. That's why any free-software author who's thinking will tell you that good beta-testers (who know how to describe symptoms clearly, localize problems well, can tolerate bugs in a quickie release, and are willing to apply a few simple diagnostic routines) are worth their weight in rubies. Even one of these can make the difference between a debugging phase that's a protracted, exhausting nightmare and one that's merely a salutory nuisance.

If you're a newbie, try to find a program under development that you're interested in and be a good beta-tester. There's a natural progression from helping test programs to helping debug them to helping modify them. You'll learn a lot this way, and generate good karma with people who will help you later on.

3. Publish useful information.

Another good thing is to collect and filter useful and interesting information into Web pages or documents like FAQs (Frequently Asked Questions lists), and make those generally available.

Maintainers of major technical FAQs get almost as much respect as free-software authors.

4. Help keep the infrastructure working.

The hacker culture (and the engineering development of the Internet, for that matter) is run by volunteers. There's a lot of necessary but unglamorous work that needs done to keep it going -- administering mailing lists, moderating newsgroups, maintaining large software archive sites, developing RFCs and other technical standards.

People who do this sort of thing well get a lot of respect, because everybody knows these jobs are huge time sinks and not much fun as playing with code. Doing them shows dedication.

5. Serve the hacker culture itself.

Finally, you can serve and propagate the culture itself (by, for example, writing an accurate primer on how to become a hacker :-)). This is not something you'll be positioned to do until you've been around for while and become well-known for one of the first four things.

The hacker culture doesn't have leaders, exactly, but it does have culture heroes and tribal historians and spokespeople. When you've been in the trenches long enough, you may grow into one of these. Beware: hackers distrust blatant ego in their tribal elders, so visibly reaching for this kind of fame is dangerous. Rather than striving for it, you have to sort of position yourself so it drops in your lap, and then be modest and gracious about your status.

The Hacker/Nerd Connection

Contrary to popular myth, you don't have to be a nerd to be a hacker. It does help, however, and many hackers are in fact nerds. Being a social outcast helps you stay concentrated on the really important things, like thinking and hacking.

For this reason, many hackers have adopted the label `nerd' and even use the harsher term `geek' as a badge of pride -- it's a way of declaring their independence from normal social expectations. See The Geek Page for extensive discussion.

If you can manage to concentrate enough on hacking to be good at it and still have a life, that's fine. This is a lot easier today than it was when I was a newbie; mainstream culture is much friendlier to techno-nerds now. There are even growing numbers of people who realize that hackers are often high-quality boyfriend/girlfriend/husband/wife material. For more on this, see Girl's Guide to Geek Guys.

If you're attracted to hacking because you don't have a life, that's OK too -- at least you won't have trouble concentrating. Maybe you'll get one later.

Points For Style

Again, to be a hacker, you have to enter the hacker mindset. There are some things you can do when you're not at a computer that seem to help. They're not substitutes for hacking (nothing is) but many hackers do them, and feel that they connect in some basic way with the essence of hacking.

- Read science fiction. Go to science fiction conventions (a good way to meet hackers and proto-hackers).
- Study Zen, and/or take up martial arts. (The mental discipline seems similar in important ways.)
- Develop an analytical ear for music. Learn to appreciate peculiar kinds of music. Learn to play some musical instrument well, or how to sing.
- Develop your appreciation of puns and wordplay.
- Learn to write your native language well. (A surprising number of hackers, including all the best ones I know of, are able writers.)

The more of these things you already do, the more likely it is that you are natural hacker material. Why these things in particular is not completely clear, but they're connected with a mix of left- and right-brain skills that seems to be important (hackers need to be able to both reason logically and step outside the apparent logic of a problem at a moment's notice).

Finally, a few things *not* to do.

- Don't use a silly, grandiose user ID or screen name.
- Don't get in flame wars on Usenet (or anywhere else).
- Don't call yourself a `cyberpunk', and don't waste your time on anybody who does.
- Don't post or email writing that's full of spelling errors and bad grammar.

The only reputation you'll make doing any of these things is as a twit. Hackers have long memories -- it could take you years to live it down enough to be accepted.

Other Resources

Translations of this document are available in French, Spanish and Japanese.

The Loginataka has some things to say about the proper training and attitude of a Unix hacker.

I have also written A Brief History Of Hackerdom.

Peter Seebach maintains an excellent Hacker FAQ for managers who don't understand how to deal with hackers.

I have written a paper, *The Cathedral and the Bazaar*, which explains a lot about how the Linux culture works. You can find it on my writings page.

Frequently Asked Questions

Q: Will you teach me how to hack?

Since first publishing this page, I've gotten several requests a week from people to "teach me all about hacking". Unfortunately, I don't have the time or energy to do this; my own hacking projects take up 110% of my time.

Even if I did, hacking is an attitude and skill you basically have to teach yourself. You'll find that while real hackers want to help you, they won't respect you if you beg to be spoon-fed everything they know.

Learn a few things first. Show that you're trying, that you're capable of learning on your own. Then go to the hackers you meet with questions.

Q: Where can I find some real hackers to talk with?

Well, not on IRC, that's for sure -- it's nothing but flamers and crackers there as far as the eye can see. The best way is to find a Unix or Linux user's group local to you and go to their meetings (there's a directory at the Linux Users' Group page on Sunsite).

Q: What language should I learn first?

HTML, if you don't already know it. There are a lot of glossy, hype-intensive *bad* HTML books out there, and distressingly few good ones. The one I like best is HTML: The Definitive Guide.

When you're ready to start programming, I would recommend starting with Perl or Python. C is really important, but it's also much harder.

Q: How can I get started? Where can I get a free Unix?

Elsewhere on this page I include pointers to where to get a Linux. To be a hacker you need motivation and initiative and the ability to educate yourself. Start now...

Back to Eric's FAQ Page Up to Site Map $Date: 1997/12/01 18:18:01 $

Eric S. Raymond <esr@snark.thyrsus.com>