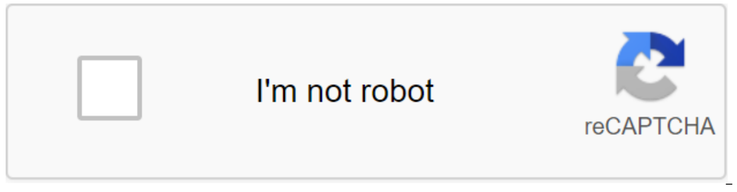

Continue

Jargon file español pdf

Thyrsus Enterprises Copyright © 2001 By Eric S. Raymond Original: How to Become a Hacker Translation: Mikel miquel@sindominio.net'gt;Vidal, from a previous version made by Cesar Ballardini. As an editor of Jargon File (Archive Hacker Jargon) and author of some famous documents of the same type, I often get emails from aspiring network enthusiasts asking: How can I learn to be an experienced hacker? It's sad to say that there doesn't seem to be any frequently asked questions or web documents that deal with this vital topic, so here comes mine. If you're reading this document offline, the original online version can be found esr/faqs/hacker-howto.html. Note: At the end of this document there is a list of frequently asked questions. Please read it - 2 times - before emailing me with questions about this document. Numerous translations of this document are available: Bulgarian, Catalan, Chinese (simplified), Chinese (traditional), Danish, Dutch, French, German, Hebrew, Hungarian, Indonesian, Italian Japanese, Korean, Portuguese (Brazilian), Portuguese (European), Russian and Swedish. Please note that as the content of this document changes from time to time, such translations may be more outdated. The Jargon file contains many definitions of the term hacker, most of which are based on the reputation of technical and pleasure in solving problems outside. If you want to know how to become a hacker, well, only 2 points are really relevant. There is a community, a common culture, experienced programmers and network masters whose history dates back decades to the days of the first timeshare of minicomputers and early experiments with ARPAnet. Members of this culture have created the term hacker. Hackers created the Internet. Hackers have made Unix the operating system as it is today. Hackers make Usenet walk. Hackers run WWW. If you are part of this culture, if you have contributed to it and other people know who you are and call you a hacker, then you are a hacker. The hacker mentality is not limited to this software culture. There are people who apply the attitude of hackers to other things, such as electronics or music, in fact, you can find it at the highest level of any science or art. Software hackers recognize these kindred spirits elsewhere and may call them hackers, and some claim that the hacker's nature is not really dependent on the specific environment in which the hacker operates. However, in the rest of this paper we will focus on the skills and attitudes of software hackers, as well as the tradition of a common culture that originated the term zlt;/miquel@sindominio.net'gt;There is another group of people who call themselves hackers but who are not. These are people (usually male teens) who have fun illegally hacking into computers and doing phreaking in the phone system. Real hackers have a name for these people: hackers and they don't want to hear from them. Real hackers believe that most hackers are lazy, irresponsible and not very bright, and base their criticism that the ability to hack security does not make a person a hacker, just as the ability to start a car with a bridge in the key does not make you a car engineer. Unfortunately, many journalists and writers mistakenly use the word hacker to describe hackers; this is very annoying for real hackers. The main difference is that hackers build things; Crackers destroy them. If you want to be a hacker, read on. If you want to be a cracker, go straight to read alt.2600 and be prepared to tolerate the harsh reality when you learn that you are not as smart as you think. And that's all I'm going to say about crackers. Hackers solve problems and build things, believe in freedom and mutual voluntary help. To be accepted as a hacker, you must behave as if you have such an attitude within you. And to behave as if you have such an attitude, you have to sincerely believe that attitude. But if you think of cultivating hacker relationships just as a way to gain acceptance in this culture, you're wrong. Become the kind of person who believes that these things are important to you to help you learn and stay motivated. As with all creative arts, the most effective way to become a teacher is to emulate the thinking of teachers, not only intellectually, but also emotionally. Or as the following modern poem zen says: So if you want to be a hacker, repeat what's next until you believe what you're saying: It's a lot of fun being a hacker, but it's the kind of fun that takes a lot of effort. Efforts require motivation. Successful athletes get their motivation from a kind of physical pleasure that arises from the work of their body, forcing themselves beyond their own physical limits. Similarly, to be a hacker you have to feel a primitive shudder when you solve problems, hone your skills and exercise your intelligence. If you are not the right person, you feel inclined to these things naturally, you should be able to test them to become a hacker. If not, you will find that your hacking energy will be exhausted by other distractions such as sex, money or social approval. (In addition to develop some faith in your own learning ability – the belief that even if you don't know everything you need to solve a problem, if you take part of it and learn the uuda, you'll learn enough to solve the next part, and so on, until you fully resolve.) Hackers are anti-authoritarian by nature. Anyone who can give you orders can make you stop solving this problem, which fascinates you, and given the way authoritarian minds work, they will find frighteningly stupid reasons to do so. So you have to fight authoritarian attitudes wherever it is, because if you leave it, it will stifle you and other hackers. (It's not the same as fighting all the authorities. Authoritarians thrive on censorship and secrecy. And they are suspicious of voluntary cooperation and information exchange - they only like cooperation under their control. So to behave like a hacker, you need to develop an instinctive hostility to censorship, secrecy, and the use of force or fraud to subdue responsible adults. And you have to be prepared to act confidentially. To be a hacker, you need to develop some of these relationships. But having just an attitude doesn't turn you into a hacker and it can't turn you into a champion athlete or rock star. To become a hacker, you will need intelligence, practice, dedication and hard work. Therefore, we must learn not to trust and respect competition in all its forms. No hacker likes to waste time on those who take a hacker pose, but they revere competition, especially competition when hacking, but competition in any field is good. The competition is especially good in demanding skills that few people master, and the best competition is in demanding skills that require mental acuity, agility and concentration. If you respect competition, you will like to develop it in yourself - hard work and dedication will become a kind of intense game, not a routine. This attitude is vital to becoming a hacker. The attitude of hackers is vital, but even more so skills. Attitude is no substitute for competition, and there is a certain set of basic tools that you have to master before any hacker dreams of calling it. This set slowly changing over time as technology creates new skills and discards others as outdated. For example, machine language programming was enabled, and until recently HTML was not mentioned. But by the end of 1996, it is clear that the following should be included: This is, of course, a fundamental skill of a hacker. If you don't know programming languages, I recommend starting with Python. It's clean, well documented, and relatively easy for beginners. Although it is a good native language, it is not a toy; it is very powerful, flexible and adapts well to large projects. I wrote a detailed analysis of Python. Good tutorials are available on the Python website. Java is also a good language to start programming. It's harder than Python, but it produces code faster than Python. I think it's a good second language to learn. But don't think you can be a hacker, even a programmer, if you only know one language, you need to learn to think about programming problems in general, regardless of any language. To be a real hacker, you need to get to the point where you learn a new language in the days regarding what's in the manual that you already know before. This means that you have to learn several languages that are very different from each other. If you want to be serious about programming, you'll have to learn C, the backbone of Unix. The NHS is very similar to C; If you know one, teaching the other won't be difficult for you. However, none of them are good at learning the program. And today, the more you can avoid programming in C, the more productive you will be. C is very efficient and works well with machine resources. Unfortunately, C is effective based on a large level of low-level resource management (such as memory) manually. All this low-level programming is very complex and error-prone, and you'll have to spend a lot of time debugging programs. With today's powerful machines, it's no longer so necessary - it's smarter to use a language that uses machine time less efficiently, but instead use your time more efficiently. I mean, Python. Other languages that are particularly important to hackers are Perl and LISP. Perl is worth learning for practical reasons: it is widely used by dynamic websites and management systems, so even if you've never written in Perl, you should learn to read. Many people use Perl the way I suggest using Python to avoid C programming in works that don't require C effectiveness. Worth studying LISP for another reason - the insightful, illuminating experience you'll get when you finally this will make you the best programmer for the rest of your days, even if you don't use LISP much. (You can easily get your first LISP experience by typing and changing Emacs editor editing modes.) The best thing today is to learn these five (Python, Java, C/C, Perl and LISP). In addition to being the most important in hacking, these languages are very different ways of approaching programming, and everyone will teach you differently. I can't give the full instructions in this document on how you can learn to program - it's a complex skill. But I can tell you that books and courses won't work (many, maybe most of the best hackers, self-taught). You can learn the characteristics of languages - part knowledge - books, but true knowledge is acquired in real life by applying what you already know. What will work is a) read the code and b) write the code. Learning programming how to learn to write well in a natural language. The best way to learn is to read some things written by style masters and then write some things yourself, read a lot more, write a little more... and repeat this until what you write starts to show the kind of strength and economics you value in your models. It used to be difficult to find good quality code to read because there were several programs of a certain magnitude available in the source code, so for the first time hackers could get their hands on them. This situation has changed dramatically: open source software, free programming tools and free operating systems (all made by hackers) are widely available today. Which brings me elegantly to our next topic... I suppose you have a personal computer or you can access one (these guys today find it so easy to get it... :-)). The most important step any beginner can take in the way of acquiring hacker skills is to get a copy of Linux or one of the other free BSD type Unices, install it on a personal computer and make it work. Yes, it is true that there are other operating systems in the world besides Unix. But they can only be found in binary format - you can't read the code or change it. If you try to learn how to hack a DOS, Windows or MacOS machine you will feel as if you are trying to learn to dance with a plastered body. Under OS/X it's possible, but only part of the open source system - it's like you're hitting a multi-layered concrete wall, and other than you have to be careful not to develop a bad habit of relying on Apple's own code. If you're concentrating on Unix under the hood, you can find out more Unix is an Internet operating system. While you can learn how to use the Internet without knowing Unix, you can never be a hacker on the internet without knowing it. For this reason, today's hacker culture is very unix-oriented. (This was not always the case, and even the situation is disliked by some of the oldest hackers, but the symbiosis between Unix and the Internet is so strong that even Microsoft's energy seems incapable of making a dent.) So get Unix -I personally like Linux, but there are others (and yes, you can run Linux and DOS/Windows on the same machine). Get to know him. Make it work. Put your hand in it, cheer. Communicate online through it. Read the code. Change it. This system includes the best programming tools (including C, Lisp and Perl) that any Microsoft operating system couldn't even dream of, plus you'll have fun and immerse yourself in a wealth of knowledge of such magnitude that you never thought until at some point, looking back, you'll realize that you're already an experienced hacker master. If you want to learn more about Learning Unix, go to The Loginataka.Si want to get your hands on Linux, check out where I can get Linux. You can find help and resources from BSD Unix at www.bsd.org.I wrote about the basics of Unix and the Internet. (Note: I don't recommend installing Linux or BSD alone if you're a beginner. for Linux it finds a local group of Linux users and asks for help; or contact it through a network of open projects. LISC supports IRC channels where you can get help.) Most of the things that hacker culture has built work out of the public's mind are, helping to operate factories, offices and universities, and the lack of obvious impact on the lives of non-hackers. The Internet is the only big exception, and this hacker toy is so huge and brilliant that even politicians recognize that it changes the world. It is for this reason (and there are many other equally good ones) that you have to learn to work online. I don't mean learning how to handle a browser (anyone can do it), but you have to learn to write HTML, web language markup. If you still don't know how to program, a training that involves writing HTML will teach you some of the mental habits that will help you with programming. So make yourself a personal page. Try XHTML, which is clearer than HTML classics. (There are good tutorials for beginners on the internet, here's one.) But don't believe that by having a personal page you are closer to being a hacker. The Internet is full of personal pages. Most are banal, rubbish-free - very ostentatious rubbish, but rubbish is finally (if you want more on this topic, see HTML Hell Page). In order to be useful, your Page needs to have content - it needs to be interesting and/or useful to other hackers. And that brings us to the next topic... As an American and native English speaker, he did not want to offer this if it was interpreted as a kind of cultural imperialism. But native speakers from other languages have encouraged me to point out that English is a language that hacking culture and the Internet work, and you need to know what to work in a hacking community. That's very true. Back in the day, around 1991, I learned that many hackers who had English as a second language used it in technical discussions, even when they shared their native language; I was told that English has a richer technical vocabulary than any other language, and just because of that it was the best tool to work with. For similar reasons, translations of technical books originally written in English are often unsatisfactory (if any). Linus Torvalds, a Finn, comments on his code in English (apparently he never thought of doing it any other way). His fluency in English was an important factor in his ability to recruit a global Linux development community. And that brings us to the next topic... As in many other cultures without the monetary economy, hacking is based on reputation. You are trying to solve interesting problems, but how interesting and good the solutions you find are what only your technically speaking peers or

superiors will be able to judge. In line with this, when you play a hacker game, you will learn to evaluate yourself primarily based on what other hackers think of your skills (which is why you can't be a real hacker until other hackers constantly call you that). This fact is tarnished by the depiction of hacking as a single job; also hacker cultural taboo (which is currently reduced but still strong) that prevents ego or external verification from being admitted as elements involved in the very motivation. In particular, hacking is what anthropologists call the culture of gifts. You gain status and reputation not through the domination of others without being beautiful, or having what other people desire, but by sacrificing things. In particular, sacrificing your time, your creativity, and the result of your skill. There are basically five kinds of things you can do to gain the respect of hackers: The first (most central and most traditional) is to write programs that other hackers find fun or useful, and donate the sources of hacker culture programs to be (We used to call it free software, but it confused too many people who weren't sure what the term free should have meant and could mean free or free. However, free software is still widely used in all of its documents, although in this version we respected naturally the change in Eric's terminology, which replaced free open source software in all of his documents, and we translated it into open source code - T.N.). The most revered demigods of hacking are the people who have written great programs, with great features that meet and sacrifice long-term needs, so that everyone can use them. People who debug open source software bugs are also recognized. In this imperfect world, we will inevitably spend most of our time developing at the debugging stage. This is why open source software developers believe that a good beta tester (beta tester, someone who knows how to clearly describe symptoms, who can correctly find problems, makes mistakes in quick delivery, and who is willing to apply a few simple diagnostic procedures) is worth his weight in gold. Even with one tester of them, the debugging process can be made of a long nightmare that leaves one exhausted only by healthy troubles. If you're new, try finding an evolving program that interests you and become a good beta tester. There is a natural development from assisting in testing programs and then helping to debug them and then helping to change them. You will learn a lot, and people will help you in the future. Contrary to popular myth, you shouldn't be a nerd literally nerd, but in hacker jargon it was ironically assumed, having lost the originally pejorative nuance, and ended up being used as a synonym for those who care about important things and don't entertain themselves in minutiae. - T.N. to be a hacker. Help, however, and many hackers are nerds. Being a social outsy, a nerd can focus on really important things like thinking and hacking. For this reason, many hackers have adopted nerd tags and even used the nasty term geek as a sign of pride- this is their way of declaring their own normal social expectations. If you can focus enough on hacking to be good at it, and on top of that enjoy your personal life, that's fine. It's much easier now than when I was a freshman in the 1970s; The dominant culture sees techno-nerds with the best eyes today. There are also a growing number of people who realize that hackers often have high quality material for a friend/girlfriend/husband/wife. If you are attracted to hacking because you don't have a life of your own, that's good too, at least you won't have any problems concentrating. You may later be able to get your own life like the rest of the people. Again, to be a hacker, you have to develop a hacker's mindset. There are some things you can do when you are without a computer that can help. These things are not a substitute for hacking the activity itself (nothing), but many hackers do them, and believe that they are somehow primitively connected to the essence of hacking. The more of these things you have done, the more likely you are to have natural hacker material. Why these things in particular and not others is something that is not entirely clear, but they are all related to the mixture of your left and right side abilities of your brain, which seems to be an important thing; hackers will be able to both logically reason and take steps beyond the obvious logic of the problem at any given time. Work so hard as you play and play as intensely as you work. For true hackers, the respite between games, work, science and art tends to disappear, or blend into a high level of creativity. They are also not satisfied with having a narrow range of skills. While most hackers self-describe as programmers, they are often more than competent in various activities-system management, web design and hardware solutions are common. The hacker, who on the one hand is a system administrator on the other hand, is also skilled in script programming and web design. Hackers don't do things halfway; if they get into the theme, they tend to be very good. Finally, a few things you shouldn't do: The only reputation you'll earn from these customs is that fool. Hackers have a great memory- it can take years for them to take you back after these things. The problem with virtual nicknames deserves some explanation. Hiding your identity behind the nickname is childish and stupid behavior that is typical of crackers, warez d00dz, and other lower forms of life. Hackers don't; they're proud of what they're doing and they want it with their real names. So if you have a name for them, give it up. In hacker culture they will only serve to mark you as a loser. Peter Seebach supports excellent Hacker frequently asked questions for managers who do not understand dealing with hackers. If Peter's page doesn't respond, you'll find a copy in the next search for Excite. I also wrote A Brief History of Hacking (there is a Castile translation: A Brief History of Hacker Culture - T.N.). I wrote an article by Cathedral and Bazaar that explains a lot about how Linux culture and open source software work. I studied this topic more directly in my second part, Manor Noosphere. Rick Moen wrote an excellent document on how to run a group of Linux users. Rick Moen and I collaborated on another paper on how to ask smart questions. This will help you find help along the way. If you need basics instructions like personal computers, Unix and the Internet, go to Unix and Internet Basics HOWTO. When you publish software or write patches, try following guidelines in how to release HOWTO software. The question is: Will you teach me how to hack? How can I start? When should I start? Is it too late to learn? The question is: How long will it take me to learn how to hack? In: Visual basic or C-good languages to get you started? The question is: Would you help me hack the system, or would you teach me how to do it? How do I get someone else's account password? How to access/read/control someone else's mail? In: How to steal the privileges of a channel operator in the IRC? B: I was hacked. Will you help me protect myself from future attacks? B: I have a problem with Windows, will you help me? Where can I find real hackers to communicate with them? The question is: Can you recommend useful books on hacking and related topics? A: Do I need to be good at math to become a hacker? What language should I learn first? What equipment do I need? Q:. I want to do my part. Can you help me choose a problem for the job? In: Do I need to hate and beat Microsoft? But won't open source software make it impossible for programmers to make a living? In: How to start? Where can I get a free Unix? The question is: Will you teach me how to hack? A: From the first post on this page, I get a few requests a week (often somewhat a day) from people who want me to teach you all about hacking. Unfortunately, I have neither the time nor the energy Do it. my own hacking projects consume 110% of my time. Even if you could do it, hacking attitude and skill you have to learn for yourself. You will realize that hackers really want to help you, but they will not respect you if you start asking them to feed you in the mouth of everything they know. Learn a few things first. Show them what you are trying, that you can learn on your own. Then go to the hackers you know, but already with specific questions. If you write a hacker's letter asking a hacker for advice, there are 2 things you should know first. First, we found that people who are lazy or careless in their writings are often too lazy or careless to think about being good hackers, so be careful to spell correctly, and use good grammar and punctuation. Otherwise, you will probably be ignored. Second, don't ask for an answer with another ISP account than the account you're sending out; We find people who do this, usually scammers using stolen accounts, and we are not interested in assisting or supporting theft. How can I start? A: The best way to get started is probably to go through LUG (Linux User Group, a local group of Linux users). Groups can be found, for example, on the common information page LDP Linux; There is probably one near your area related to college or university. LUG members will probably give you Linux if you order it, and they will definitely help you install and start it. When should I start? Is it too late to learn? A: Any age at which you are motivated to start is a good age. Most people seem to be interested in the ages of 15 to 20, but I know the exceptions in both directions. How long will it take me to learn how to hack? A: It depends on your talent and how hard you work. Most people can acquire a respectable skill level in eighteen months or two years if focused. However, don't think you ended up there; If you are a real hacker, you will spend the rest of your life learning and ruining your material. In: Visual basic or C-good languages to get you started? A:No, because they're not portable. There is no free implementation of these languages, so you will only be closed to platforms that the manufacturer decides to support. Accepting this monopoly situation is not the hacker's way. Visual Base is particularly huge. The fact that this Microsoft-owned language is enough to disqualify it, and like other versions of Basic it has a bad design and will teach you bad programming habits. One of these bad habits is to become addicted to libraries, modules and tools from one manufacturer. In general, any language that is not supported under Linux or one of the BSDs, and/or when it may be least run on 3 different operating systems, is a bad language to learn how to hack. The question is: Would you help me hack the system, or would you teach me how to do it? A:No. Anyone who asks about it after reading this document is too stupid to be instructed, even if he had time to do so. Any letter with such questions would ignore it or answer it with extreme rudeness. How do I get someone else's account password? A: It's a crack. Disappear, idiot. How can I access/read/control someone else's mail? A: It's a crack. Disappear. In: How to steal the privileges of a channel operator in the IRC? A: It's a crack. Get out of here. B: I was hacked. Will you help me protect myself from future attacks? A:No. All the time I wondered what this was because of the poor programming of Microsoft Windows. Windows cannot be effectively protected from hacker attacks; simply, its code and architecture have too many flaws. The only achievable prevention is the transition to Linux or some other operating system that is at least designed to protect. B: I have a problem with Windows, will you help me? Answer: Yes. Go to the DOS command line and say 'format c:'. Any problems you experience will go away in a few minutes. Where can I find real hackers to communicate with them? Answer: The best way is to find a localized group of Unix or Linux users and go to their meetings (you can find links to various user group lists on metalab's LDP site). (I used to say that you won't find real hackers in IRC, but I understand that is changing. apparently some real hacking communities associated with things like GIMP and Perl, have IRC channels now.) The question is: Can you recommend useful books on hacking and related topics? A: I support Linux Reading List HOWTO that you may find useful. Loginova can also be interesting. For the introduction to Python, check the introductory materials on Python.P: Should I be good at math to become a hacker? A:No. While you should be able to think logically and follow precise chains of reasoning, hacking uses very little formal or arithmetic mathematics. In particular, you don't need trigonometry, calculation or analysis (we leave this to electrical engineers :-)). Some final mathematicians (including bulia algebra, ultimate set theory, combinatorial and graph theory) can be useful. What language should I learn (the latest HTML dialect) if you don't already know. There are a lot of very bad HTML BOOKS outside, and unfortunately only a few good ones. The one that I think is the best is HTML: The Ultimate Guide.But HTML is not a complete programming language. When you're ready to start programming, I recommend starting with Python. You'll hear a lot of people recommend Perl for you, and Perl is even more popular than Python, but it's harder to learn and (in my opinion) worse designed. C is very important, but it's also much more complicated than Python and Perl. Don't try to recognize it in the first place. Windows users: Don't settle for Visual Basic. It will teach you bad habits and it is not portable outside of Windows. Avoid it. P: What equipment do I need? A: Previously cases that personal computers had little function and little memory that imposed artificial restrictions on the process of learning a hacker. This ceased to be true some time ago; Any Intel 486DX50 machine or higher is powerful enough to develop, X and Internet connections, and the smaller floppy disks you can buy today are enough. It is important when choosing a machine to find out whether your hardware supports Linux (or BSD compatible). This applies to the most modern machines. The only problem is modems: some machines have windows-specific hardware that doesn't work with Linux. There is a website called linmodems.org where you can determine if, despite being winmodem, you can get the device to work according to GNU/Linux - T.N.) There are frequently asked questions about hardware compatibility; The latest version is here. Q:. I want to do my part. Can you help me choose a theme for work? R:. No, because I don't know what your talents or interests are. You have to be single-minded or you won't be serious about it, so it almost never works for other people to choose for you. Try this. Watch Freshmeat ads for a few days. When you see one that makes you think: Wow, I'd like to work on this! Join us. In: Do I need to hate and beat Microsoft? A: No, you don't have to. And not because Microsoft is not disgusting, but because the hacking culture existed long before Microsoft and will exist for a long time after Microsoft is history. Any energy you spend hating Microsoft would be better spent getting excited about your skills. Write good code - it will be a tougher hit for Microsoft and doesn't require you to get karma dirty. But won't open source software make it impossible for programmers to make a living? A: It seems unlikely-so-so the software industry is open code seems to create jobs without removing them. If you have a program program In the network economy benefits from not having it written, the programmer will be rewarded regardless of whether the program will be open source after it is done. And no matter how many free software you've written, there seems to be always more demand for new and specialized applications. I wrote more about it on Open Source Pages. P:P: How can I get started? Where can I get a free Unix? A: Elsewhere on this page I include pointers to places where you can get the most used free Unix. To be a hacker you need motivation and initiative and the ability to educate yourself. Start now... Nwo... jargon file español pdf. the jargon file español