

# (Eric Raymond) Come diventare un Hacker - matrobriva

Short description of this text:

This document is released under the Specificated in description license

Upload date:2012-03-16 Original date:2002-10-03

---

## Come Diventare Un Hacker

(How To Become A Hacker, rev. 1.15, October 03, 2002)

**Eric Steven Raymond**

[Thyrsus Enterprises](#)

<[esr@thyrsus.com](mailto:esr@thyrsus.com)>

Copyright © 2001 Eric S. Raymond

### Indice dei Contenuti

[Perché Questo Documento? Cosa è Un Hacker? L'Atteggiamento dell'Hacker](#) 1. [Il mondo è pieno di affascinanti problemi che aspettano di essere risolti.](#) 2. [Nessun problema dovrebbe mai aver bisogno di essere risolto due volte.](#) 3. [Noia e ripetitività sono il male.](#) 4. [La libertà è un bene.](#) 5. [L'attitudine non è un sostituto della competenza.](#) [Capacità di Base dell'Hacking](#) 1. [Impara a programmare.](#) 2. [Procurati un sistema Unix open-source e impara ad utilizzarlo.](#) 3. [Impara come usare il World Wide Web e scrivere in HTML.](#) 4. [Se non comprendi l'Inglese, imparalo.](#) [La Posizione Sociale nella Cultura Hacker](#) 1. [Scrivi software open-source](#) 2. [Aiuta ad eseguire test e debug di software open-source](#) 3. [Pubblica informazioni utili](#) 4. [Aiuta a mantenere efficiente l'infrastruttura](#) 5. [Servi la cultura hacker](#) [Il Binomio Hacker/Nerd](#) [Appunti di Stile](#) [Altre Risorse](#) [Frequently Asked Questions](#)

## Perché Questo Documento?

Come curatore dello [Jargon File](#) e autore di pochi altri (ma ben conosciuti) documenti di questo genere, spesso ricevo delle richieste via email da parte di entusiasti principianti della rete che mi chiedono (praticamente)

"come posso diventare un mago dell'hacking?". Poiché, singolarmente, non sembra esserci alcun'altra FAQ o documento web che affronti questa vitale questione, ecco qua il mio.

Se stai leggendo questo documento offline, o una sua traduzione, la versione corrente si trova all'indirizzo: <http://www.tuxedo.org/~esr/faqs/hacker-howto.html>.

Nota: c'è una lista di [Frequently Asked Questions](#) alla fine di questo documento. Per favore leggila *due volte* prima di mandarmi qualsiasi richiesta relativa a questo documento.

Sono anche disponibili numerose traduzioni di questo documento: [Bulgarian](#), [Catalan](#), [Chinese \(Simplified\)](#), [Chinese \(Traditional\)](#), [Danish](#), [Dutch](#), [French](#), [German](#), [Hebrew](#), [Hungarian](#), [Indonesian](#), [Italian](#), [Japanese](#), [Korean](#), [Portuguese \(Brazilian\)](#), [Portuguese \(European\)](#), [Russian](#), e [Swedish](#). Considera che poiché questo documento viene talvolta modificato, esse potrebbero essere anche abbastanza superate.

## Cosa è Un Hacker?

Lo [Jargon File](#) contiene una quantità di definizioni del termine 'hacker', molte delle quali hanno a che fare con l'attitudine tecnica e il piacere di risolvere problemi e superare limiti. Se vuoi sapere come *diventare* un hacker, comunque, solo due definizioni sono realmente rilevanti.

Esiste una comunità, una cultura condivisa, di programmatori esperti e maghi della rete per rintracciare le origini della quale dobbiamo andare indietro di decenni, fino ai tempi dei primi minicomputer funzionanti in time-sharing o degli iniziali esperimenti con ARPAnet. I membri originari di questa comunità crearono il termine 'hacker'. Gli hacker hanno costruito Internet. Gli hacker hanno portato il sistema operativo Unix a quello che esso oggi è diventato. Gli hacker hanno dato l'avvio a Usenet e al lavoro per il World Wide Web. Se fai parte di questa cultura, se hai contribuito ad essa e altre persone che ne fanno parte ti conoscono e ti chiamano hacker, allora sei un hacker.

La predisposizione mentale di un hacker non è confinata a quest'unico ambito. Ci sono persone che applicano la propria attitudine hacker ad altre cose, come l'elettronica o la musica - e puoi trovarle ai più alti livelli di qualsiasi scienza o arte. Gli hacker del software riconoscono ovunque queste anime affini e le possono anche definire 'hacker' - e qualcuno sostiene che la natura hacker sia realmente indipendente dal particolare mezzo con il quale l'hacker si esprime. Tuttavia, nel resto di questo documento, noi focalizzeremo l'attenzione sulle capacità e sugli atteggiamenti degli hacker del software, e sulle tradizioni di quella cultura condivisa che ha originato il termine 'hacker'.

C'è un altro gruppo di persone che ad alta voce si proclamano hacker, ma non lo sono. Queste sono persone (principalmente adolescenti maschi) che provano piacere a penetrare nei computer e a compiere azioni di phreaking al sistema telefonico. I veri hacker li chiamano 'cracker' e non vogliono avere nulla a che fare con loro. I veri hacker, per lo più, pensano che i cracker siano pigri, irresponsabili e non troppo svegli, e obiettano che essere capaci di violare la sicurezza informatica non ti fa hacker più di quanto essere in grado di scassinare macchine ti faccia diventare ingegnere meccanico. Sfortunatamente, molti giornalisti e scrittori sono stati fuorviati nell'usare il termine 'hacker' per descrivere i cracker, e questo irrita infinitamente i veri hacker.

La differenza fondamentale è questa: gli hacker costruiscono le cose, i cracker le rompono.

Se vuoi diventare un hacker, continua a leggere. Se vuoi diventare un cracker, vai a leggere il newsgroup [alt.2600](#) e sii pronto a passare da cinque a dieci anni in prigione dopo aver scoperto che non sei così intelligente quanto pensi di essere. E questo è tutto quello che avevo intenzione di dire riguardo ai cracker.

## L'Atteggiamento dell'Hacker

Gli hackers risolvono problemi e costruiscono cose, e credono nella libertà e nell'aiuto reciproco. Per essere accettato come hacker, devi comportarti come se anche tu avessi questo tipo di atteggiamento. E per comportarti come se avessi questo atteggiamento, devi crederci con convinzione.

Ma se pensi che coltivare l'atteggiamento da hacker sia semplicemente un modo per ottenere l'accettazione nella comunità, non hai colto il punto fondamentale. Diventare il tipo di persona che crede in queste cose è importante soprattutto *per te stesso* - per aiutarti a imparare e per mantenerti motivato. Come per tutte le arti creative, la maniera più efficace per diventare un maestro è adottare l'atteggiamento mentale dei maestri - non solo dal punto di vista intellettuale ma anche da quello emozionale.

Come dice il seguente poema Zen moderno,

Per seguire il percorso: osserva il maestro, segui il maestro, cammina affianco al maestro, guarda attraverso il maestro, diventa il maestro.

Pertanto, se vuoi essere un hacker, ripeti le seguenti cose finché non credi in loro:

### **1. Il mondo è pieno di affascinanti problemi che aspettano di essere risolti.**

Essere un hacker è molto divertente, ma è un tipo di divertimento che necessita di un sacco di fatica. La fatica richiede motivazione. Gli atleti di successo traggono la loro motivazione dalla possibilità di ottenere delle prestazioni che li portano a superare i propri limiti fisici. In maniera simile, per essere un hacker devi provare una forte eccitazione nel risolvere problemi, affinando le tue abilità ed esercitando la tua intelligenza.

Se non sei il tipo di persona che sente queste cose naturalmente, hai bisogno di diventarlo prima ancora di poterlo fare come hacker. Altrimenti scoprirai che la tua energia hacker è fiaccata da distrazioni come il sesso, il denaro e il desiderio di approvazione sociale.

(Dovresti anche sviluppare una forma di fede nelle tue capacità di apprendimento - un credo secondo cui anche se non puoi conoscere tutto ciò di cui hai bisogno per risolvere un problema, se ne affronti solo una parte e trai insegnamento da quella, hai imparato abbastanza da passare al pezzo successivo - e così via finché non ce l'avrai fatta del tutto.)

### **2. Nessun problema dovrebbe mai aver bisogno di essere risolto due volte.**

Le menti creative sono una risorsa di valore, ma in numero limitato. Esse non dovrebbero sprecarsi a reinventare la ruota quando ci sono molti nuovi affascinanti problemi che aspettano là fuori.

Per comportarti da hacker, devi convincerti che il tempo speso dagli altri hacker a pensare è prezioso - al punto che è quasi un dovere morale per te condividere le informazioni, risolvere problemi e successivamente diffonderne le soluzioni, in modo che gli altri hacker possano risolvere *nuovi* problemi invece di dover perennemente ri-risolvere quelli vecchi.

(Non devi credere di essere obbligato a regalare *tutto* il tuo prodotto creativo, sebbene gli hacker che lo fanno siano quelli che ottengono il maggior rispetto dagli altri hacker. È coerente con l'etica dell'hacker venderne abbastanza per ottenere cibo, affitto e computer. È bello usare le proprie abilità di hacker per sostenere una famiglia o anche diventare ricco, ma se lo fai non dimenticare mai la lealtà verso la tua arte e i tuoi compagni hacker.)

### **3. Noia e ripetitività sono il male.**

Gli hacker (e le persone creative in generale) non dovrebbero mai essere annoiati o doversi affaticare in un lavoro stupido e ripetitivo, poiché quando questo accade significa che non stanno facendo l'unica cosa che dovrebbero fare: risolvere nuovi problemi. Questo processo di spreco è dannoso per tutti. Pertanto la noia e la ripetitività non sono solo spiacevoli, sono realmente il male.

Per comportarti come un hacker devi credere in questo, abbastanza da voler automaticamente allontanare quanto più possibile tutto ciò che è noioso, non solo per te stesso ma per tutti gli altri (specialmente gli altri hacker).

(Esiste un'unica apparente eccezione a questa regola. Gli hacker, a volte, fanno delle cose che possono sembrare ripetitive o noiose ad un osservatore, come esercizi per liberare la mente o allo scopo di acquisire capacità o provare qualche particolare tipo di esperienza che non potresti avere altrimenti. Ma questa è una scelta - nessun pensatore dovrebbe mai essere forzato a operare in una situazione che lo annoia.)

### **4. La libertà è un bene.**

Gli hacker sono anti-autoritari per natura. Chiunque possa darti degli ordini, può fermarti dal risolvere problemi dai quali sei affascinato - e, visto il modo con cui le menti autoritarie funzionano, tali ordini generalmente saranno motivati da ragioni orribilmente stupide. Così, l'atteggiamento autoritario deve essere combattuto ovunque si trovi, affinché non soffochi te e gli altri hacker.

(Ciò non è la stessa cosa che combattere tutte le forme di autorità. I bambini hanno bisogno di essere guidati e i criminali devono essere fermati. Un hacker potrebbe essere d'accordo nell'accettare qualche tipo di autorità allo scopo di ottenere qualcosa alla quale tiene di più che al tempo speso ad eseguire ordini. Ma questo è un caso limitato e consapevole; il tipo di resa personale che gli autoritari vorrebbero non è in discussione.)

Gli autoritari prosperano sulla censura e sulla segretezza. Essi distruggono la cooperazione volontaria e lo scambio di informazioni - l'unica 'cooperazione' che gli piace è quella di cui hanno il controllo. Così, per agire come un hacker, devi sviluppare un'istintiva avversione verso la censura, la segretezza e l'uso della forza o dell'inganno utilizzati nei confronti delle persone adulte. E devi essere disposto a mettere in atto questo credo.

## 5. L'attitudine non è un sostituto della competenza.

Per essere un hacker, devi sviluppare alcuni di questi atteggiamenti. Ma l'atteggiamento da solo non basta a fare di te un hacker, non più di quanto ti faccia diventare un atleta di successo o una rockstar. Diventare un hacker richiede intelligenza, pratica, dedizione e duro lavoro.

Innanzitutto, devi imparare a diffidare delle apparenze e a rispettare le competenze di ogni genere. Gli hacker non perdono tempo con chi si atteggia, ma adorano la competenza - specialmente quella nell'hacking, ma va bene anche la competenza in ogni altro campo. Particolarmente apprezzata è la competenza in alcune attività che solo pochi sanno gestire, e la competenza che richiede una mente brillante, astuzia e concentrazione è la migliore.

Se veneri la competenza, ti divertirai a svilupparla in te stesso - il duro lavoro e la dedizione diventeranno una sorta di intenso gioco invece che una fatica. E ciò è vitale per diventare un hacker.

## Capacità di Base dell'Hacking

L'attitudine hacker è vitale, ma le capacità lo sono ancora di più. L'attitudine non è un sostituto della competenza, e c'è un insieme di capacità di base che devi avere prima che qualsiasi hacker si sogni di chiamarti tale.

Questo bagaglio di conoscenze cambia lentamente nel tempo, man mano che la tecnologia crea nuove abilità e ne rende obsolete altre. Per esempio, esso in genere include la programmazione in linguaggio macchina e solo di recente richiede l'HTML. Al momento attuale il bagaglio di conoscenze di base prevede quanto segue:

### 1. Impara a programmare.

Questa, ovviamente, è la capacità fondamentale dell'hacker. Se non conosci alcun linguaggio di programmazione, ti suggerisco di iniziare con Python. È progettato in maniera chiara, ben documentato e abbastanza adatto ai principianti. Malgrado sia un buon linguaggio con il quale cominciare, non è proprio un giocattolo; è molto potente e flessibile e adatto anche a grandi progetti. Ho scritto una più dettagliata [recensione su Python](#). Dei buoni [manuali](#) sono disponibili sul [sito web di Python](#).

Anche Java è un buon linguaggio per imparare a programmare. È più difficile di Python, ma produce un codice più veloce rispetto a Python. Lo ritengo un eccellente secondo linguaggio.

Ma devi essere conscio che non raggiungerai i livelli di abilità di un hacker o più semplicemente di un programmatore se conosci solamente uno o due linguaggi - hai bisogno di imparare a pensare ai problemi legati alla programmazione in maniera più generale, indipendentemente dal linguaggio specifico. Per essere un vero hacker, hai bisogno di arrivare al punto di poter apprendere un nuovo linguaggio in pochi giorni semplicemente confrontando il manuale con quanto già sai. Questo significa che dovrai imparare parecchi linguaggi differenti tra loro.

Se vuoi dedicarti seriamente alla programmazione, dovrai imparare il C, il linguaggio cardine di Unix. Il C++ è

molto vicino al C; se già ne conosci uno, non ti sarà difficile imparare l'altro. Nessuno di questi due linguaggi, comunque, è adatto per iniziare ad imparare. E, al momento, quanto più riesci a evitare di programmare in C tanto più sarai produttivo.

Il C è veramente efficiente, e utilizza al meglio le risorse della macchina. Sfortunatamente, C fornisce questa efficienza in cambio di una notevole mole di lavoro manuale per la gestione delle risorse (quali la memoria) ad un basso livello. Tutta la programmazione di basso livello è complessa e soggetta a bug, e ti richiederà un'enorme quantità di tempo per il debugging. Con le odierne macchine, così potenti, questo in genere rappresenta un cattivo compromesso - è più conveniente utilizzare un linguaggio che impiega in maniera meno efficiente il tempo della macchina ma che impiega *più efficacemente* il tuo tempo. Ad esempio, Python.

Altri linguaggi di particolare importanza per gli hacker sono [Perl](#) e [LISP](#). Perl è degno di essere appreso per ragioni pratiche; è largamente utilizzato per costruire pagine web dinamiche e per l'amministrazione di sistema; così, anche se non dovrai mai scrivere in Perl, dovresti imparare almeno a leggerlo. Molte persone usano Perl nella maniera in cui io suggerisco di usare Python, per evitare di programmare in C quei lavori che non richiedono l'efficienza della macchina che C fornisce. Avrai bisogno di essere in grado di capire il loro codice.

LISP è da imparare per un altro motivo: la profonda e illuminante esperienza che proverai quando finalmente riuscirai a farlo tuo. Questa esperienza ti renderà un programmatore migliore per il resto dei tuoi giorni, anche se non lo utilizzerai poi molto. (Puoi procurarti qualche esperienza iniziale con LISP in maniera abbastanza semplice realizzando e modificando le modalità di editing per l'editor di testo Emacs.)

È meglio, addirittura, imparare tutti questi cinque linguaggi (Python, Java, C/C++, Perl e LISP). Oltre che essere i linguaggi più importanti per l'hacking, essi rappresentano approcci molto differenti alla programmazione, e ognuno ti educerà in maniera preziosa.

Non posso fornire qui delle istruzioni complete su come imparare a programmare - è un compito complesso. Ma posso dirti che libri e corsi vari non te lo insegnano (molti, forse *la maggior parte* dei migliori hacker è autodidatta). Puoi imparare le caratteristiche dei linguaggi - frammenti di conoscenza - dai libri, ma l'approccio mentale che trasforma quella conoscenza in reali capacità può essere appreso solo mediante la pratica e l'apprendistato. Ciò che devi fare è (a) *leggere codice* e (b) *scrivere codice*.

Imparare a programmare è come imparare a scrivere bene nel linguaggio nativo. Il miglior modo per farlo è leggere del materiale scritto dai maestri della forma, scrivere qualcosa per conto proprio, leggere molto di più, scrivere qualcosa in più, leggere molto di più, scrivere qualcosa in più... e ripetere il tutto finché la tua scrittura inizia a sviluppare la forza e l'economia che vedi nei tuoi modelli.

Una volta, trovare del buon codice da leggere era un'impresa ardua, poiché esistevano ben pochi programmi disponibili anche sotto forma di codice sorgente, a disposizione degli hacker per la lettura e la manipolazione. Tutto ciò è cambiato radicalmente; software open-source, strumenti per la programmazione e sistemi operativi (tutti costruiti da hacker) sono oggi largamente disponibili. Questo mi porta al prossimo argomento...

## **2. Procurati un sistema Unix open-source e impara ad utilizzarlo.**

Sto assumendo che tu abbia un personal computer o che possa averne accesso (questi giovani d'oggi lo ottengono facilmente :-)). L'unico passo importante che un principiante possa compiere per acquisire delle capacità da hacker è procurarsi una copia di Linux o di uno degli Unix BSD, installarlo sulla propria macchina ed utilizzarlo.

Certo, ci sono altri sistemi operativi al mondo oltre a Unix. Ma sono distribuiti solo in forma di file binari - non puoi leggerne il codice sorgente e non puoi neppure modificarlo. Imparare l'hacking su macchine DOS o Windows o MacOS è come cercare di imparare a ballare mentre si è ingessati.

In ambiente OS/X questo è possibile, ma solo una parte del sistema è open source: sei sempre sul punto di urtare qualche muro, e devi fare attenzione a non sviluppare la cattiva abitudine di dipendere dal codice proprietario di Apple. Se ti concentri su Unix senza distrarti puoi imparare alcune utili cose.

Unix è il sistema operativo di Internet. Sebbene tu possa imparare ad utilizzare Internet senza conoscere Unix, non puoi essere un hacker di Internet senza capire Unix. Per questa ragione, la cultura hacker oggi è fortemente centrata su Unix. (Questo non è sempre stato vero, e alcuni hacker dei tempi andati non ne sono molto contenti, ma la simbiosi tra Unix e Internet è diventata forte abbastanza che neppure i muscoli della Microsoft sembrano in grado di rappresentare una seria minaccia.)

Pertanto, procurati un sistema Unix - io amo Linux, ma ci sono altre scelte (è vero, *puoi* anche far girare DOS/Windows e Linux sulla stessa macchina). Impara ad usarlo. Fallo girare. Armeggiaci sopra. Usalo per connetterti a Internet. Leggine il codice sorgente e modificalo. Avrai a disposizione i migliori strumenti di programmazione (incluso C, LISP, Python e Perl) che qualsiasi sistema operativo della Microsoft possa mai sognarsi di ospitare, ti divertirai, acquisirai più conoscenza di quanto saprai rendertene conto, finché non lo guarderai con gli occhi di un maestro hacker.

Per altre informazioni sull'apprendimento di Unix, vedi [The Loginataka](#).

Per mettere le mani su Linux, vedi [Where can I get Linux](#).

Puoi trovare aiuto e risorse relative a Unix BSD presso [www.bsd.org](http://www.bsd.org).

Ho scritto anche un'introduzione alle [basi di Unix e Internet](#).

(Nota: se sei un principiante non ti consiglio di installare Linux o BSD da solo. Per Linux, trova un Linux User Group nella tua zona e chiedi aiuto a loro oppure contatta l'[Open Projects Network](#). LISC mantiene dei [canali IRC](#) dove puoi ottenere aiuto.)

### **3. Impara come usare il World Wide Web e scrivere in HTML.**

La maggior parte delle cose che la cultura hacker ha costruito operano in posti assolutamente non in vista, contribuendo a far funzionare aziende, uffici e università senza alcun evidente impatto sulla vita dei non-hacker. Il Web è l'unica, grande eccezione, il gigantesco e splendido giocattolo degli hacker che, come anche *ipolitici* ammettono, sta cambiando il mondo. Basterebbe questa ragione (e ce ne sarebbero molte altre, altrettanto buone), per capire che hai bisogno di imparare come funziona il Web.

Questo non significa solamente imparare ad utilizzare un browser (chiunque può farlo), ma imparare a scrivere in HTML, il linguaggio di marcatori del Web. Se ancora non sai programmare, scrivere in HTML ti fornirà delle abitudini mentali che ti aiuteranno ad imparare. Costruisci quindi una home page. Cerca di migrare ad XHTML,

essendo questo un linguaggio più pulito del classico HTML. (Esistono alcuni buoni tutorial per principianti sul Web; [eccone uno](#).)

Ma avere solamente una home page non è sufficiente a fare di te un hacker. Il Web è pieno di home page. Molte sono inutili, poltiglia senza contenuti, bella da vedere, certo, ma sempre poltiglia (per maggiori dettagli vedi [The HTML Hell Page](#)).

Per essere meritevole, la tua home page deve avere *contenuti*: dovrebbe essere interessante e/o utile agli altri hacker. E questo ci porta al prossimo argomento...

#### **4. Se non comprendi l'Inglese, imparalo.**

In quanto Americano, avendo io stesso come lingua madre l'Inglese, sono stato in passato riluttante nel suggerire questo, temendo che fosse considerato come una sorta di imperialismo culturale. Ma numerose persone di diversa lingua madre mi hanno sollecitato a segnalare che l'Inglese è la lingua di base per la cultura hacker e per Internet, e che avrai bisogno di conoscerlo per essere funzionale nella comunità hacker.

Questo è profondamente vero. All'incirca nel 1991 ho imparato che molti hacker che conoscevano l'Inglese come seconda lingua utilizzavano questo nelle discussioni tecniche, anche quando condividevano informazioni con un loro connazionale; all'epoca mi fu detto che l'Inglese disponeva di un vocabolario tecnico più ricco di qualsiasi altra lingua e comunque è uno strumento migliore per il lavoro. Per ragioni analoghe, le traduzioni di libri tecnici scritti in Inglese sono spesso insoddisfacenti (quando sono fatte).

Linus Torvalds, un Finlandese, commenta il proprio codice in Inglese (e apparentemente non ha mai avuto bisogno di fare altrimenti). La sua dimestichezza con l'Inglese è stata un importante fattore nella sua capacità di aggregare una comunità mondiale di sviluppatori attorno a Linux. Questo rappresenta un esempio da seguire.

## **La Posizione Sociale nella Cultura Hacker**

Come molte culture prive di un'economia di mercato, l'universo hacker orbita attorno alla reputazione. Tu stai tentando di risolvere interessanti problemi, ma per quanto interessanti siano, e benché le tue soluzioni siano veramente il massimo, è un dato di fatto che solo le persone con la tua stessa preparazione tecnica o superiore sono in grado di apprezzarlo.

Coerentemente con questo, quando giochi il ruolo dell'hacker, impari a fare attenzione innanzitutto a cosa gli altri hacker pensano delle tue capacità (e questo perché non sei veramente un hacker finché gli altri hacker non ti chiamano tale). Questo fatto è messo in ombra dall'immagine dell'hacking come un lavoro solitario; e anche da un tabù della cultura hacker (oggi gradualmente in declino ma ancora consistente) contro il voler ammettere che l'ego o l'essere ben considerati dagli altri contribuiscano ad alimentare le proprie motivazioni.

Nello specifico, l'universo hacker è ciò che gli antropologi chiamano *cultura del talento*. In essa guadagni posizione sociale e reputazione non dominando altre persone o grazie alla bellezza o possedendo beni che altri vorrebbero, ma piuttosto donando cose. In particolare, donando il tuo tempo, la tua creatività e i risultati della tua abilità.

Principalmente, ci sono cinque tipi di cose che puoi fare per essere rispettato dagli altri hacker:

## **1. Scrivi software open-source**

La prima (la cosa più importante e più radicata nella tradizione) è scrivere programmi che gli altri hacker possano considerare utili o divertenti, e rilasciarne il codice sorgente all'intera comunità hacker.

(In genere, si usava chiamare questi programmi col termine "free software", ma questo confondeva troppe persone, che non erano mai sicure di cosa esattamente si intendeva con il termine "free". La maggior parte di noi, in una proporzione di almeno 2:1 secondo le analisi effettuate sul web, attualmente preferisce usare la definizione di software "[open-source](#)").

I più riveriti semi-dei dell'universo hacker sono persone che hanno scritto utili e impegnativi programmi che rispondevano ad un bisogno molto diffuso, regalandoli poi alla comunità, di modo che oggi tutti li usano.

## **2. Aiuta ad eseguire test e debug di software open-source**

C'è bisogno anche di chi prova ed esegue il debug del software open-source. In questo mondo imperfetto, inevitabilmente spenderemo la maggior parte del nostro tempo di sviluppo del software nella fase di debugging. Questa è la ragione per cui ogni programmatore open-source ti dirà che un buon beta tester (una persona capace di descrivere chiaramente i sintomi, di individuare con precisione i problemi, disposto a tollerare i bug nelle versioni instabili e in grado di applicare semplici routine diagnostiche) vale tanto oro quanto pesa. Anche uno solo di questi può mutare la fase di debugging da lungo e stancante incubo a una semplice e salutare scoccatura.

Se sei un principiante, cerca un programma in fase di sviluppo al quale sei interessato e diventa un buon beta-tester. Si genera una naturale evoluzione dall'aiutare a testare i programmi, aiutare ad effettuarne il debug fino a contribuire a modificarli. Imparerai molto in questa maniera e creerai un buon karma con le persone che in futuro potranno aiutarti.

## **3. Pubblica informazioni utili**

Un'altra buona cosa è selezionare e raggruppare informazioni utili e interessanti in pagine web o documenti tipo Frequently Asked Questions (FAQ), e rendere il tutto disponibile a chiunque.

I curatori delle FAQ più tecniche godono di un ampio rispetto, al pari degli autori di software open-source.

## **4. Aiuta a mantenere efficiente l'infrastruttura**

La cultura hacker (e lo sviluppo tecnico di Internet, a dire il vero) è sostenuta da volontari. C'è un sacco di lavoro indispensabile ma poco affascinante che necessita di essere svolto: amministrare mailing list, moderare newsgroup, mantenere siti web con ampi archivi di software, sviluppare Request For Comment e nuovi standard tecnici.

Le persone che fanno questo genere di cose ottengono molto rispetto, poiché tutti sanno che queste attività sono avide di tempo e non altrettanto divertenti come giocare con il codice sorgente. Farle mostra dedizione.

## 5. Servi la cultura hacker

Infine, puoi servire e diffondere la cultura hacker in sé (scrivendo, ad esempio, un accurato manuale su come diventare un hacker :-)). Non sei tenuto a fare questo finché non arriva il momento e non diventi ben conosciuto per uno dei precedenti quattro punti.

La cultura hacker non deve avere leader in senso stretto, ma deve avere eroi e capi-tribù, storici e portavoci. Quando sarai stato abbastanza a lungo in trincea, potresti divenire una di queste figure. Ricorda: gli hacker non amano gli ego chiassosi tra i loro capi-tribù, per cui la visibilità raggiunta grazie a questo genere di fama è pericolosa. Piuttosto che batterti per essa, devi scegliere una posizione umile, ed essere modesto e felice del tuo stato.

## Il Binomio Hacker/Nerd

Contrariamente al luogo comune, non hai bisogno di essere un nerd per essere un hacker. Ti può aiutare, comunque, e molti hacker sono effettivamente dei nerd. Essere estraneo ai rapporti sociali ti aiuta a rimanere concentrato sulle cose veramente importanti, quali il pensiero e l'hacking.

Per questa ragione, molti hacker hanno adottato l'etichetta 'nerd' e utilizzano anche il termine 'geek' con orgoglio: è una maniera di dichiarare la propria indipendenza dalle normali aspettative sociali. Vedi [The Geek Page](#) per una discussione più ampia.

Se riesci ad organizzarti in modo da concentrarti abbastanza sull'hacking tanto da diventare bravo e avere ancora una vita normale, è perfetto. Questo è oggi molto più facile di quanto lo fosse negli anni '70, quando io ero un hacker; la cultura ufficiale è molto più amichevole con i techno-nerd al giorno d'oggi. C'è anche un numero crescente di persone che ritiene che gli hacker siano spesso degli ottimi amanti e buoni da sposare.

Se sei attratto dall'hacking perché non hai una vita sociale, va bene comunque - almeno non avrai problemi di concentrazione. Forse avrai una tua vita più avanti.

## Appunti di Stile

Ancora, per essere un hacker, devi avere la mentalità dell'hacker. Ci sono alcune cose che puoi fare quando non sei davanti al computer che sembrano essere utili. Non sono sostituti dell'hacking (niente lo è), ma molti hacker le fanno e sentono che in qualche modo esiste un legame tra queste e l'essenza dell'hacking.

-

Impara a scrivere bene nella tua lingua. Sebbene sia un luogo comune che i programmatori non sappiano scrivere, un sorprendente numero di hacker (inclusi i più completi che io conosca) sono abili scrittori.

- Leggi riviste scientifiche. Partecipa a convegni scientifici (un buon modo per incontrare hacker e proto-hacker).
- Studia lo Zen, e/o le arti marziali. (Le discipline mentali in molti casi presentano delle analogie.)
- Sviluppa un orecchio analitico per la musica. Impara ad apprezzare diversi tipi di musica. Impara a suonare bene qualche strumento, o a cantare.
- Sviluppa il tuo apprezzamento per i giochi di parole e gli anagrammi.

Quante più di queste cose già fai, maggiore è la tua predisposizione naturale per diventare un hacker. Il perché di queste cose in particolare non è del tutto chiaro, esse sono tuttavia connesse in un intreccio di capacità della parte destra - e sinistra - del cervello che sembra essere importante; gli hacker hanno bisogno di essere in grado sia di ragionare in maniera logica, sia, al momento opportuno, di mantenersi estranei alla logica apparente di un problema.

Lavora così intensamente come quando ti diverti, e divertiti così intensamente come quando lavori. Per i veri hacker, i confini tra "gioco", "lavoro", "scienza" e "arte" tendono a scomparire, o a confondersi in un gioioso momento creativo di alto livello. Inoltre, non ritenerti soddisfatto di un limitato campo di conoscenze. Sebbene la maggior parte degli hacker si definiscano programmatori, probabilmente sono più che competenti in numerose attività correlate: amministrazione di un sistema, web design e risoluzione dei problemi hardware sono le più comuni. Un hacker che sia anche un amministratore di sistema, d'altra parte, è indotto ad essere particolarmente competente nella programmazione di script e nel web design. Gli hacker non fanno le cose con superficialità; se si impegnano a fondo in un settore, tendono a diventare molto bravi.

Infine, alcune cose *danon* fare.

- Non usare un nickname stupido o grandioso.
- Non partecipare alle guerre di insulti su Usenet (o in qualunque altro posto).
- Non autodefinirti un 'cyberpunk', e non perdere il tuo tempo con chiunque lo faccia.
- Non inviare lettere o e-mail piene di errori di ortografia o di grammatica.

L'unica reputazione che otterrai facendo una qualsiasi di queste cose è quella del cretino. Gli hackers hanno una memoria di ferro, e potrebbero volerci degli anni per essere riaccettato.

La questione dei nickname o dei soprannomi richiede un approfondimento. Nascondere la propria identità dietro un soprannome è una stupida ed infantile caratteristica di crackers, warez d00dz ed altre forme inferiori di vita. Gli hacker non fanno questo. Loro sono orgogliosi di ciò che fanno e vogliono che venga associato al loro *vero* nome. Pertanto, se usi un soprannome eliminalo. Nella cultura hacker esso ti identificherebbe come un perdente.

## Altre Risorse

Peter Seebach mantiene un eccellente [Hacker FAQ](#) per manager che non capiscono come trattare con gli hacker. Se il sito di Peter non dovesse risultare attivo, la seguente [ricerca con Excite](#) dovrebbe trovarne una copia.

Io ho scritto anche [A Brief History Of Hackerdom](#).

Ho poi scritto un documento, [The Cathedral and the Bazaar](#), che spiega parecchie cose su come funzionano Linux e la cultura open-source. Ho affrontato questo argomento ancora più direttamente nel suo seguito [Homesteading the Noosphere](#).

Rick Moen ha scritto un eccellente documento su [come gestire un Linux user group](#).

Rick Moen ed io abbiamo lavorato assieme ad un altro documento su [Come Porre Domande Intelligenti](#). Questo ti aiuterà a cercare assistenza in un modo che la farà diventare più piacevole di quanto attualmente sia.

Se hai bisogno di un insegnamento di base su come funzionano i personal computer, Unix e Internet, consulta [The Unix and Internet Fundamentals HOWTO](#).

Quando realizzi software o scrivi correzioni di software, cerca di seguire le linee guida contenute in [Software Release Practice HOWTO](#).

## Frequently Asked Questions

Q: [Mi insegnerai a fare dell'hacking?](#) Q: [Come posso iniziare, allora?](#) Q: [Quando si dovrebbe iniziare? È troppo tardi per me?](#) Q: [Quanto tempo ci impiegherò ad imparare?](#) Q: [Visual Basic o C# sono buoni linguaggi per iniziare?](#) Q: [Potresti aiutarmi a violare un sistema o insegnarmi come fare?](#) Q: [Come posso ottenere la password per un account di qualcun altro?](#) Q: [Come posso introdurmi/leggere/sorvegliare la posta di qualcun altro?](#) Q: [Come posso rubare canali o privilegi su IRC?](#) Q: [Sono stato crackato. Mi aiuterai a difendermi dagli attacchi futuri?](#) Q: [Sto avendo problemi con il software di Windows. Mi aiuterai?](#) Q: [Dove posso trovare dei veri hacker con cui parlare?](#) Q: [Puoi consigliarmi dei validi libri su argomenti relativi all'hacking?](#) Q: [Ho bisogno di essere bravo in matematica per diventare un hacker?](#) Q: [Quale linguaggio dovrei imparare per primo?](#) Q: [Quale tipo di hardware mi occorre?](#) Q: [Voglio contribuire. Puoi aiutarmi ad ottenere un problema su cui lavorare?](#) Q: [Ho bisogno di odiare e combattere Microsoft?](#) Q: [Ma non è possibile che il software open-source renda i programmatori incapaci di crearsi una posizione?](#) Q: [Come posso iniziare? Dove posso ottenere uno Unix gratuito?](#) Q:

Mi insegnerai a fare dell'hacking?

**A:**

Fin dalla prima pubblicazione di questo documento, ho ricevuto numerose richieste ogni settimana (spesso molte al giorno) da persone che mi chiedevano "insegnami tutto sull'hacking". Sfortunatamente non ho il tempo né l'energia per fare questo; i miei progetti personali di hacker, e i viaggi in veste di sostenitore dell'open-source, occupano il 110% del mio tempo.

Ma anche se lo facessi, l'hacking è un'attitudine e una capacità che, fondamentalmente, devi imparare da solo. Ti accorgerai che nonostante i veri hacker vogliano aiutarti, non avranno alcun rispetto per te se chiedi loro di essere imboccato di tutto ciò che sanno.

Impara prima alcune cose. Mostra che ci stai provando e che sei in grado di imparare da solo. Infine rivolgiti agli hacker con domande specifiche.

Se invii una richiesta di aiuto ad un hacker, devi avere presente due cose. Primo, abbiamo notato che le persone pigre o negligenti nello scrivere di solito sono anche troppo pigre e negligenti nel pensare per poter diventare buoni hacker: per questo, abbi cura di usare un linguaggio appropriato, una grammatica corretta e una buona punteggiatura, in caso contrario sarai probabilmente ignorato. Secondo, non richiedere una risposta presso un account *diverso* da quello dal quale stai scrivendo; le persone che fanno questo sono di solito dei ladri che stanno usando account rubati, e non abbiamo alcuna intenzione di aiutarli.

**Q:**

Come posso iniziare, allora?

**A:**

La maniera per te migliore di iniziare probabilmente sarebbe quella di andare ad un raduno di un LUG (Linux user group). Puoi trovare molti gruppi nella [LDP General Linux Information Page](#); probabilmente ce n'è uno nella tua zona, se possibile, associato con un college o una università. I membri del LUG probabilmente ti daranno una copia di Linux se gliela chiedi e di sicuro ti aiuteranno ad installarlo e cominciare ad usarlo.

**Q:**

Quando si dovrebbe iniziare? È troppo tardi per me?

**A:**

Qualsiasi età alla quale ti senti motivato per partire è una buona età. La maggior parte delle persone sembrano divenire interessati tra i 15 e i 20 anni, ma conosco eccezioni in entrambe le direzioni.

**Q:**

Quanto tempo ci impiegherò ad imparare?

**A:**

Questo dipende da quanto talento hai e quanto duramente lavori. La maggior parte delle persone, se si impegna, è in grado di acquisire capacità del tutto rispettabili entro 18-24 mesi. Non pensare comunque che finisca lì; se sei un vero hacker, passerai il resto della tua vita imparando e perfezionando quanto sai.

**Q:**

Visual Basic o C# sono buoni linguaggi per iniziare?

**A:**

No, perché non sono portabili. Non ci sono versioni open-source di questi linguaggi, per cui saresti vincolato a quelle sole piattaforme che i produttori decidono di supportare. Accettare questo tipo di monopolio non è un modo di agire da hacker.

Visual Basic è, in particolar modo, terribile. Il fatto che sia un linguaggio proprietario della Microsoft è già abbastanza per declassarlo, e al pari degli altri Basic è un linguaggio progettato in maniera scadente che ti insegna cattive abitudini di programmazione.

Una di queste cattive abitudini è diventare dipendenti da un unico produttore di librerie, widgets e tool di sviluppo. In generale, qualsiasi linguaggio che non sia supportato almeno sotto Linux o qualche Unix BSD, e/o da almeno tre diversi fornitori di sistemi operativi, è una cosa scadente per imparare l'hacking.

**Q:**

Potresti aiutarmi a violare un sistema o insegnarmi come fare?

**A:**

No. Chiunque possa ancora porre una domanda del genere dopo aver letto questa FAQ è troppo stupido per essere educato anche se io avessi il tempo per l'insegnamento. Ogni richiesta di questo tenore che mi arriverà via e-mail, sarà ignorata o riceverà una replica di estrema durezza.

**Q:**

Come posso ottenere la password per un account di qualcun altro?

**A:**

Questo è cracking. Vai via, idiota.

**Q:**

Come posso introdurmi/leggere/sorvegliare la posta di qualcun altro?

**A:**

Questo è cracking. Sparisci, deficiente.

**Q:**

Come posso rubare canali o privilegi su IRC?

**A:**

Questo è cracking. Vai via, cretino.

**Q:**

Sono stato crackato. Mi aiuterai a difendermi dagli attacchi futuri?

**A:**

No. Tutte le volte che mi è stata posta, questa domanda proveniva da qualche povero sfigato che utilizzava Microsoft Windows. Non è possibile rendere realmente sicuri i sistemi Windows contro gli attacchi informatici; semplicemente perché il codice e l'architettura hanno troppe lacune, che rendono l'operazione di rendere sicuro Windows simile a cercare di svuotare una nave con un secchio. La sola prevenzione possibile consiste nel passare a Linux o a qualche altro sistema operativo progettato quanto meno per offrire sicurezza.

**Q:**

Sto avendo problemi con il software di Windows. Mi aiuterai?

**A:**

Si. Vai al prompt del DOS e digita "format c:". Qualunque problema ti sia capitato scomparirà in pochi minuti.

**Q:**

Dove posso trovare dei veri hacker con cui parlare?

**A:**

Il modo migliore è trovare un gruppo di utenti Unix o Linux nella tua zona e andare ai loro incontri (puoi trovare i link a numerosi elenchi di tali gruppi sul sito [LDP](#) presso il Metalab).

(A questo punto ero solito dire che non troverai alcun vero hacker su IRC, ma mi sembra di capire che qualcosa stia cambiando. Alcune comunità di hacker, legate a progetti come GIMP e Perl, ora hanno dei canali IRC.)

**Q:**

Puoi consigliarmi dei validi libri su argomenti relativi all'hacking?

**A:**

Io mantengo un [Linux Reading List HOWTO](#) che tu potresti trovare utile. Anche [The Loginataka](#) potrebbe essere interessante.

Per un'introduzione al linguaggio Python, consulta il [materiale introduttivo](#) sul sito di Python.

**Q:**

Ho bisogno di essere bravo in matematica per diventare un hacker?

**A:**

No. Sebbene tu debba essere capace di pensare in maniera logica e di seguire i passaggi di un ragionamento esatto, l'hacking usa molto poco la matematica formale o l'aritmetica.

In particolare, non avrai bisogno della trigonometria, del calcolo infinitesimale o dell'analisi matematica (lasciamo questa roba agli ingegneri elettronici :-)). Alcune basi di matematica finita (compresa l'algebra Booleana, la teoria degli insiemi finiti, il calcolo combinatorio e la teoria dei grafi) potrebbero essere di aiuto.

**Q:**

Quale linguaggio dovrei imparare per primo?

**A:**

XHTML (il più recente dialetto dell'HTML) se ancora non lo conosci. Ci sono un sacco di enormi, lucenti e *scadenti* libri sull'HTML in giro e, dolorosamente, pochi buoni libri. Quello che io preferisco è [HTML: The Definitive Guide](#).

Ma HTML non è un linguaggio di programmazione completo. Quando sarai pronto a programmare, ti raccomanderei di iniziare con [Python](#). Sentirai molte persone raccomandarti Perl, e inoltre Perl è più popolare di Python, ma è più difficile da imparare e (a mio parere) è progettato meno bene.

Il linguaggio C è davvero importante, ma è anche molto più difficile sia di Python che di Perl. Non cercare di impararlo per primo.

Utenti di Windows, *non* optate per Visual Basic. Vi insegnerà cattive abitudini e non è portabile all'infuori di Windows. Evitatelo.

**Q:**  
Quale tipo di hardware mi occorre?

**A:**  
Era convinzione comune che i computer fossero sotto-dimensionati e poveri di memoria, tanto da imporre dei limiti artificiali ai progressi degli hacker. Questo ha cessato di essere vero tempo fa; qualsiasi macchina dall'Intel 486DX50 in su è potente abbastanza per lo sviluppo di software, per utilizzare X e connettersi a Internet, e i dischi più piccoli che puoi comprare oggi hanno spazio in abbondanza.

La cosa importante nello scegliere una macchina sulla quale imparare è verificare che il suo hardware sia Linux-compatibile (o BSD-compatibile, se hai scelto di seguire questa strada). Attualmente, ciò dovrebbe essere vero per la maggior parte delle macchine moderne; il solo campo che richiede attenzione è rappresentato dai modem; alcune macchine sono dotate di hardware specifico per Windows che non funziona con Linux.

C'è una FAQ sulla compatibilità hardware; la versione più recente è [qui](#).

**Q:**  
Voglio contribuire. Puoi aiutarmi ad ottenere un problema su cui lavorare?

**A:**  
No, poiché non conosco le tue capacità e i tuoi interessi. Hai bisogno di essere auto-motivato altrimenti non ti applicherai, che è la ragione per la quale quando altre persone scelgono il tuo lavoro non funziona quasi mai.

Prova in questo modo. Controlla gli annunci dei progetti che si susseguono su [Freshmeat](#) per qualche giorno, quando ne vedi uno che ti fa pensare "Carino! Mi piacerebbe lavorare su questo!", allora scegliilo.

**Q:**  
Ho bisogno di odiare e combattere Microsoft?

**A:**  
No, non devi. Non che Microsoft non sia detestabile, ma esisteva una cultura hacker molto prima di Microsoft e ne esisterà ancora una molto dopo che Microsoft diventerà un ricordo. Ogni energia che tu spendi per odiare Microsoft potrebbe essere utilizzata in maniera migliore amando il tuo lavoro. Scrivi del buon codice - questo darà fastidio a sufficienza a Microsoft senza inquinare il tuo karma.

**Q:**  
Ma non è possibile che il software open-source renda i programmatori incapaci di crearsi una posizione?

**A:**  
Questo appare improbabile - finora l'industria del software open-source sembra creare posti di lavoro piuttosto che farne perdere. Se scrivere un programma produce una rendita economica, il programmatore sarà comunque pagato sia che questo diventi o meno open-source dopo essere stato realizzato. E non importa quanto software "free" venga scritto, sembra che ci sia sempre maggiore richiesta di applicazioni nuove e personalizzate. A tale proposito, ho scritto di più nelle pagine del sito [Open Source](#).

**Q:**

Come posso iniziare? Dove posso ottenere uno Unix gratuito?

**A:**

Da qualche parte in questa pagina ho incluso i link verso i siti dove puoi trovare le più utilizzate versioni free di Unix. Per diventare un hacker, hai bisogno di motivazione, iniziativa e capacità di auto-educazione. Inizia ora ...