# HACKERS' TOOLKIT: ESSENTIAL BOOKS, TOOLS & PROGRAMMING RESOURCES

Hacking Essentials: Books, Programming Guides, and Tools You Need is a curated PDF resource packed with top hacking books, programming tutorials, and essential tools for all skill levels.

By
**Abdul Wahab Junaid**

## For Contact Me

- Subscribe to Mailing List
- Instagram
- Youtube
- Github
- Facebook
- https://www.reddit.com/user/aw-junaid/
- http://twitter.com/awJunaid_
- https://vk.com/aw.junaid/
- https://buymeacoffee.com/awjunaid/
- https://www.instagram.com/awjunaid_
- https://www.twitch.tv/awjunaid

## Books Link:

- **Ethical Hacking**
- **Bug Bounty**
- **Malware Development**
- **Algorithms**
- **Android Security**
- **Arm - Assembly**
- **Bash Scripting**
- **Black Hat Hacking**
- **Buffer Over Flow**
- **Books by Dummies**
- **C & C++**
- **CCNA & Security**
- **Computer Circuits**
- **Computer Forensics**
- **Books for Computer Scientist**
- **Cryptography & Blockchain**
- **Dart & Flutter**
- **Database & Server**

- Docker
- Embedded System
- Extra Books
- Git
- Golang
- Hacking Articles 0
- Hacking Articles 1
- Hacking Articles 2
- Hacking Articles 3
- Hacking Extra
- Java
- Linux
- Machine Learning & AI
- Computer Networking
- Operating System
- Penetration Testing
- Phishing Attacks
- PHP
- Python
- Ransomware
- RedHat
- Ruby
- Rust
- Social Engineering
- SQL Injection
- Web Attacks
- Backend, Framework and Hosting
- Game Development
- Head First Books
- Html & CSS
- JavaScript
- Mobile App Development
- Programming Language Theory
- R Programming
- Random Programming Books
- Scala
- Server & Hosting
- TypeScript
- **Anime Girls Holding Books Pics**

# Ethical Hacking Tools List

## Information Gathering Tools

## DNS Analysis

- **dnsenum:** DNS enumeration tool.
- **dnsmap:** DNS mapping and analysis tool.
- **dnsrecon:** DNS reconnaissance tool.
- **fierce:** DNS reconnaissance and network misconfiguration discovery tool.

## IDS/IPS Identification

- **lbd:** Load balancer detector.
- **wafw00f:** Web application firewall detection tool.

## Live Host Identification

- **arping:** ARP-level ping utility.
- **fping:** Fast ICMP ping tool.
- **hping3:** Packet generator and network testing tool.
- **masscan:** Fast network scanner.
- **netcat:** Versatile networking tool for reading from and writing to network connections.
- **thcping6:** Ping for IPv6 networks.
- **unicornscan:** Comprehensive port scanning tool.

## Vulnerability Search

- **SPLOITUS:** Sploitus is a convenient central place for identifying the newest exploits and finding attacks that exploit known vulnerabilities
- **SearchSploit:** The official Exploit Database repository
- **Getsploit:** - Command line utility for searching and downloading exploits
- **Houndsploit:** - An advanced graphical search engine for Exploit-DB
- **OSV:** Open source vulnerability DB and triage service.

## Network & Port Scanners

- **nmap:** A popular network discovery and security auditing tool.

## OSINT (Open Source Intelligence) Analysis

- **maltego:** OSINT and link analysis tool.
- **spiderfoot:** OSINT automation tool.

## Route Analysis

- **netdiscover:** Active/passive network address discovery.
- **netmask:** Network subnet analysis tool.

## SMB Analysis

- **nbtscan:** NetBIOS name service scanner.
- **smbscan:** SMB share scanner.

### SMTP Analysis

- **smtp-user-enum:** SMTP user enumeration tool.
- **swaks:** SMTP test tool for debugging.

### SNMP Analysis

- **onesixtyone:** SNMP analysis tool.
- **snmp-check:** SNMP information extraction tool.

### SSL Analysis

- **ssldump:** SSL/TLS network traffic analyzer.
- **sslh:** Multi-protocol support over a single port.
- **sslscan:** SSL/TLS service scanner.
- **sslyze:** Fast SSL vulnerability scanner.
- **amass:** In-depth network mapping tool.
- **dmitry:** Deepmagic information gathering tool.
- **ike-scan:** IKE protocol scanner.
- **recon-ng:** Web reconnaissance framework.

## Vulnerability Analysis Tools

- **generic_chunked:** Tool for chunked transfer encoding vulnerabilities.
- **voiphopper:** VLAN hopping and VoIP exploitation tool.
- **nikto:** Web server vulnerability scanner.
- **nmap:** Vulnerability analysis through scripting.
- **unix-privesc-check:** Unix privilege escalation checker.

## Web Application Analysis Tools

- **cutycapt:** Web page screenshot capture utility.
- **dirb:** Web content scanner.
- **dirbuster:** Directory and file brute-forcer.
- **ffuf:** Fast web fuzzer.
- **cadaver:** WebDAV client.
- **davtest:** WebDAV vulnerability testing tool.
- **skipfish:** Web application security scanner.
- **wapiti:** Web vulnerability scanner.
- **whatweb:** Website fingerprinting tool.
- **wpscan:** WordPress vulnerability scanner.
- **burpsuite:** Web application security testing platform.
- **commix:** Command injection vulnerability exploitation tool.
- **sqlmap:** Automated SQL injection exploitation tool.
- **webshells:** Collection of backdoor web shells.

## Password Attack Tools

### Offline Attacks

- **chntpw:** NT password reset tool.
- **hash-identifier:** Hash recognition tool.
- **hashcat:** High-performance password cracking tool.
- **hashid:** Hash identification tool.
- **john:** John the Ripper password cracker.
- **ophcrack-cli:** Windows password cracker.
- **samdump2:** Dumps the contents of the Windows SAM file.
- **truecrack:** Brute-force tool for TrueCrypt volumes.

### Online Attacks

- **hydra:** Parallelized network login cracker.
- **medusa:** Modular network login brute-forcer.
- **ncrack:** Network authentication cracking tool.
- **thc-pptp-bruter:** PPTP VPN brute-force tool.

### Passing the Hash Tools

- **crackmapexec:** Post-exploitation tool for attacking Windows/Active Directory networks.
- **evil-winrm:** Exploitation tool for WinRM endpoints.
- **mimikatz:** Credential extraction tool.
- **smbmap:** SMB enumeration tool.
- **xfreerdp:** Free RDP client.

### Password Profiling & Wordlists

- **cewl:** Custom wordlist generator from websites.
- **crunch:** Wordlist generation tool.
- **rsmangler:** Wordlist mangling tool.
- **wordlists:** Predefined wordlists for password attacks.

## Wireless Attack Tools

- **bully:** WPS brute-force tool.
- **fern-wifi-cracker:** GUI tool for wireless network cracking.
- **wash:** WPS-enabled router scanner.
- **spooftooph:** Bluetooth spoofing tool.
- **aircrack-ng:** Wireless security audit tool.
- **kismet:** Wireless network detector and packet sniffer.
- **pixiwps:** WPS brute-force utility.
- **reaver:** WPS PIN recovery tool.

- **wifite:** Automated wireless attack tool.

## Reverse Engineering Tools

- **clang:** Compiler for C language.
- **clang++:** Compiler for C++.
- **msf-nasm_shell:** Shellcode assembler and disassembler.
- **radare2**: Open-source reverse engineering framework.
- **Ollydbg:** OllyDbg is a 32-bit assembler level analysing debugger for Microsoft Windows

## Exploitation Tools

- **crackmapexec:** Post-exploitation and network attack tool.
- **metasploit-framework**: Comprehensive exploitation framework.
- **msfpc:** Metasploit Payload Creator.
- **searchsploit:** Offline search tool for Exploit-DB.
- **setoolkit:** Social Engineering Toolkit.
- **sqlmap:** SQL injection exploitation tool.

### Cross-site Scripting（XSS）

- **BeeF:** The Browser Exploitation Framework Project
- **BlueLotus_XSSReceiver:** XSS Receiver platform without SQL
- **XSStrike:** Most advanced XSS scanner.
- **xssor2:** XSS'OR - Hack with JavaScript.
- **Xsser-Varbaek:** From XSS to RCE 2.75 - Black Hat Europe Arsenal 2017 + Extras
- **Xsser-Epsylon:** Cross Site "Scripter" (aka XSSer) is an automatic framework to detect, exploit and report XSS vulnerabilities in web-based applications.
- **Xenotix:** An advanced Cross Site Scripting (XSS) vulnerability detection and exploitation framework
- **PwnXSS:** PwVulnerability (XSS) scanner exploit
- **dalfox:** DalFox is an powerful open source XSS scanning tool and parameter analyzer, utility
- **ezXSS:** ezXSS is an easy way for penetration testers and bug bounty hunters to test (blind) Cross Site Scripting.

### Sql Injection

- **Sqlmap:** Automatic SQL injection and database takeover tool
- **SSQLInjection:** SSQLInjection is a SQL injection tool , support Access/MySQL/SQLServer/Oracle/PostgreSQL/DB2/SQLite/Informix Database.
- **Jsql-injection:** jSQL Injection is a Java application for automatic SQL database injection.
- **NoSQLMap:** Automated NoSQL database enumeration and web application exploitation tool.
- **Sqlmate:** A friend of SQLmap which will do what you always expected from SQLmap

- **SQLiScanner:** Automatic SQL injection with Charles and sqlmap api
- **sql-injection-payload-list:** SQL Injection Payload List
- **Advanced-SQL-Injection-Cheatsheet:** A cheat sheet that contains advanced queries for SQL Injection of all types.

### Command Injection

- **Commix:** Automated All-in-One OS command injection and exploitation tool

### File Include

- **LFIsuite:** Totally Automatic LFI Exploiter (+ Reverse Shell) and Scanner
- **Lfi-Space:** Lfi Scan Tool
- **Kadimus:** Kadimus is a tool to check sites to lfi vulnerability , and also exploit it
- **Shellfire:** Exploitation shell for exploiting LFI, RFI, and command injection vulnerabilities
- **LFIter2:** LFIter2 Local File Include (LFI) Tool - Auto File Extractor & Username Bruteforcer
- **FDsploit:** File Inclusion & Directory Traversal fuzzing, enumeration & exploitation tool.

### File Upload vulnerability

- **Fuxploider:** File upload vulnerability scanner and exploitation tool

### XML External Entity Attack （XXE）

- **XXEinjector:** Tool for automatic exploitation of XXE vulnerability using direct and different out of band methods
- **Oxml_xxe:** A tool for embedding XXE/XML exploits into different filetypes

### Cross-site request forgery （CSRF）

- **Deemon:** Deemon is a tool to detect CSRF in web application

### Deserialization exploit framework

- **Ysomap:** A helpful Java Deserialization exploit framework.

### Exploit Framework

- **POC-T:** Pentest Over Concurrent Toolkit
- **Pocsuite3:** pocsuite3 is an open-sourced remote vulnerability testing framework developed by the Knownsec 404 Team.
- **Metasploit:** The world's most used penetration testing framework
- **Venom:** Shellcode generator/compiler/handler (metasploit)
- **Empire:** Empire is a PowerShell and Python post-exploitation agent
- **Starkiller:** Starkiller is a Frontend for PowerShell Empire.
- **Koadic:** Koadic C3 COM Command & Control - JScript RAT
- **Viper:** metasploit-framework UI manager Tools
- **MSFvenom-gui:** gui tool to create normal payload by msfvenom
- **MYExploit:** A GUI Tools for Scanning OA vulnerabilities

- **ronin-exploits:** A Ruby micro-framework for writing and running exploits and payloads.

## Sniffing & Spoofing Tools

- **dnschef:** DNS proxy for pentesting.
- **dsniff:** Tools for network traffic sniffing.
- **netsniff-ng:** Linux network toolkit.
- **dns-rebind:** DNS rebinding attack tool.
- **sslsplit:** Man-in-the-middle for SSL/TLS connections.
- **tcpreplay:** Replays packet captures.
- **ettercap:** Comprehensive MITM attack tool.
- **macchanger:** MAC address spoofing tool.
- **minicom:** Serial communication program.
- **responder:** LLMNR, NBT-NS, and MDNS query poisoner.
- **scapy:** Network packet manipulation tool.
- **tcpdump:** Network packet analyzer.

## Post Exploitation

- **dbd:** Simple reverse shell connection tool.
- **powersploit:** PowerShell post-exploitation framework.
- **sbd:** Backdoor for reverse shells.
- **dns2tcpc:** DNS tunneling tool.
- **exe2hex:** Convert executables to hex.
- **iodine-client-start:** DNS tunneling client.
- **miredo:** Teredo tunneling client.
- **proxychains:** Tool to route connections through proxies.
- **proxytunnel:** Tunnels traffic over HTTP proxies.
- **ptunnel:** ICMP-based TCP tunneling tool.
- **pwnat:** NAT traversal tool.
- **sslh:** Multi-protocol service handling over a single port.
- **stunnel4:** SSL tunneling tool.
- **udptunnel:** UDP tunneling over TCP.
- **laudanum:** Web-based backdoors.
- **weevely:** PHP web shell.
- **evil-winrm:** Post-exploitation tool for WinRM.

## Forensics Tools

- **magicrescue:** Recover files by signature recognition.
- **scalpel:** File carving tool.
- **scrounge-ntfs:** NTFS file recovery tool.
- **guymager:** Disk imaging tool.
- **pdf-parser:** Tool for analyzing PDF files.

- **pdfid:** Tool for checking PDF security.
- **autopsy:** Digital forensics platform.
- **binwalk:** Firmware analysis tool.
- **bulk_extractor**: Extracts features from digital evidence.
- **hashdeep:** Recursive directory hashing tool.

## Reporting Tools

- **cherrytree:** Hierarchical note-taking application.
- **cutycapt:** Web page capture tool.
- **pipal:** Password analysis tool.

## Social Engineering Tools

- **msfpc:** Metasploit Payload Creator.
- **setoolkit:** Social Engineering Toolkit for phishing and other attacks.

### Code Audit

- **Cloc:** cloc counts blank lines, comment lines, and physical lines of source code in many programming languages
- **Cobra:** Source Code Security Audit
- **Cobra-W:** Cobra for white hat
- **Graudit:** Grep rough audit - source code auditing tool
- **Rips:** A static source code analyser for vulnerabilities in PHP scripts
- **Kunlun-M:** KunLun-M is a static code analysis system that automates the detecting vulnerabilities and security issue.
- **Semgrep:** Semgrep is a fast, open-source, static analysis engine for finding bugs, detecting vulnerabilities in third-party dependencies, and enforcing code standards.

## Intranet penetration

### Service Detection

- **Netspy:** A tool to quickly detect the reachable network segments of the intranet.
- **Cube:** Intranet penetration testing tools, weak password blasting, information collection and vulnerability scanning.

### Port Forwarding & Proxies

- **EarthWorm:** Tool for tunnel
- **Termite:** Tool for tunnel (Version 2)
- **Frp:** A fast reverse proxy to help you expose a local server behind a NAT or firewall to the internet
- **Nps:** A lightweight, high-performance, powerful intranet penetration proxy server, with a powerful web management terminal.

- **Goproxy:** A high-performance, full-featured, cross platform proxy server
- **ReGeorg:** The successor to reDuh, pwn a bastion webserver and create SOCKS proxies through the DMZ. Pivot and pwn
- **Neo-reGeorg:** Neo-reGeorg is a project that seeks to aggressively refactor reGeorg
- **Venom:** A Multi-hop Proxy for Penetration Testers
- **Stowaway:** Stowaway -- Multi-hop Proxy Tool for pentesters
- **rport:** Manage remote systems with ease.
- **PortForward:** The port forwarding tool developed by Golang solves the problem that the internal and external networks cannot communicate in certain scenarios.
- **Suo5:** A high-performance http proxy tunneling tool

## RootKit

- **Beurk:** BEURK Experimental Unix RootKit
- **Bedevil:** LD_PRELOAD Linux rootkit (x86 & ARM)

## Audit Tools

- **DevAudit:** Open-source, cross-platform, multi-purpose security auditing tool

## Penetration Testing Distribution

- **Backbox Linux:** penetration testing and security assessment oriented Linux distribution
- **Kali Linux:** Debian-based pentesting distribution
- **BlackArch Linux**: Arch Linux-based penetration testing distribution
- **Parrot Security:** The ultimate framework for your Cyber Security operations
- **ArchStrike:** Arch Linux respository for security professionals

## Cyber Range

### Vulnerability application

- **DVWA:** Damn Vulnerable Web Application (DVWA)
- **WebGoat:** WebGoat is a deliberately insecure web application maintained by OWASP designed to teach web application security lessons
- **DSVW:** DSVW is a deliberately vulnerable web application written in under 100 lines of code, created for educational purposes
- **DVWS:** Damn Vulnerable Web Services is an insecure web application with multiple vulnerable web service components that can be used to learn real world web service vulnerabilities
- **XVWA:** XVWA is a badly coded web application written in PHP/MySQL that helps security enthusiasts to learn application security
- **BWAPP:** A buggy web application whit more than 100 vulnerabilities
- **Sqli-lab:** SQLI labs to test error based, Blind boolean based, Time based

- **HackMe-SQL-Injection-Challenges:** Hack your friend's online MMORPG game - specific focus, sql injection opportunities
- **XSS-labs:** Small set of scripts to practice exploit XSS and CSRF vulnerabilities
- **SSRF-lab:** Lab for exploring SSRF vulnerabilities
- **SSRF_Vulnerable_Lab:** This Lab contain the sample codes which are vulnerable to Server-Side Request Forgery attack
- **LFI-labs:** Small set of PHP scripts to practice exploiting LFI, RFI and CMD injection vulns
- **Commix-testbed:** A collection of web pages, vulnerable to command injection flaws
- **File-Upload-Lab:** Damn Vulnerable File Upload V 1.1
- **Upload-labs:** A summary of all types of uploading vulnerabilities for you
- **XXE-Lab:** A XXE vulnerability Demo containing language versions such as PHP, Java, python, C#, etc
- **Vulnerable-Flask-App:** Erlik2 Vulnerable-Flask-App provided by anil-yelken.

### Simulation Range

- **Fopnp:** A Network Playground for 《Foundations of Python Network Programming》
- **CyberRange:** The Open-Source AWS Cyber Range
- **Pentest-Ground:** Pentest-Ground is a free playground with deliberately vulnerable web applications and network services.

### Honeyhots

- **DecoyMini:** A highly scalable, safe, free enterprise honeypots

## CTF challenges

- **Vulnhub:** VulnHub provides materials allowing anyone to gain practical hands-on experience with digital security, computer applications and network administration
- **TryHackMe:** TryHackMe is a free online platform for learning cyber security, using hands-on exercises and labs, all through your browser!
- **Hackthebox:** Hack The Box is a massive, online cybersecurity training platform, allowing individuals, companies, universities and all kinds of organizations around the world to level up their hacking skills.
- **Root Me:** Root Me allows everyone to test and improve their knowledge in computer security and hacking.
- **Pentestit:** Penetration testing laboratories "Test lab" emulate an IT infrastructure of real companies and are created for a legal pen testing and improving penetration testing skills
- **Pentesterlab:** Learn Web Penetration Testing: The Right Way
- **Cyberseclabs:** At CyberSecLabs, we aim to provide secure, high-quality training services that allow information security students the opportunity to safely learn and practice penetration testing skills.
- **Web Security Academy:** Free, online web security training from the creators of Burp Suite
- **Vulnmachines:** A place to learn and improve penetration testing/ethical hacking skills for FREE

**Excellent project**

- **Rawsec's CyberSecurity Inventory:** An inventory of tools and resources about CyberSecurity.
- **All-Defense-Tool:** A List for Defense Tools.
- **Awesome-POC:** A POC knowledge base of various vulnerabilities.

# From GitHub: Arctic Code Vault: Tech Tree

## Introduction: A Guide To The Tech Tree

What follows, which we call the Tech Tree, is a selection of works intended to describe how the world makes and uses software today, as well as an overview of how computers work and the foundational technologies required to make and use computers. The purpose of the GitHub Archive Program is to preserve open source software for future generations. This implies also preserving the knowledge of other technologies on which open-source software runs, along with a depiction of the open-source movement which brought this software into being.

This initial version of the Tech Tree will consist almost entirely of copies of pre-existing works, none of which were written for an unknown audience a long way into the future. As such it is not so much a guide as a collection of resources that we hope will be historically interesting and/or useful. We have tried to strike a balance between abstract/theoretical and concrete/practical work, and to provide at least an overview of the entire technical stack on which modern software engineering rests.

In addition to this technical documentation, we have also included a selection of artistic, cultural, and historical works, to help describe the overall cultural context in which this archive was created. A data dump of an entire 2020 snapshot of Wikipedia in the archive's five primary languages, along with a snapshot of Stack Overflow and sundry other smaller data collections, is also stored alongside the works itemized below.

**The current iteration of the Tech Tree is loosely divided into the following thirteen sections:**

1. ***Fundamentals of computing and the Internet:*** the essentials of how computers work, and, at least as important to today's world, how they are connected together into a single planetary network which includes most of the computers on Earth.
2. ***Algorithms and data structures:*** processes, sets of rules, and methods of arranging data to solve common categories of problems in efficient ways. Metaphorically,

algorithms are the intelligence in a software program, and data structures are its storage.

3. ***Compilers, assembler, and operating systems:*** how written source code becomes the machine code which causes the electrical signals inside a computer to change in a controlled manner, and the theory of operating systems, the software which supports a computer's basic functions and provides the fundamental, low-level functionality that all other software ultimately calls upon.
4. ***Programming languages:*** some of the world's most popular and widely used programming languages described in detail. While, fundamentally, any program can be written in any language, certain languages are better or worse at particular tasks.
5. ***Networking and connectivity:*** how computers connect to one another, via physical wires and radio signals, both one-on-one and in larger networks. Includes descriptions of the structure of the global "network of networks" known as the Internet, which connects most of the computers on Earth.
6. ***Modern software development:*** the processes and procedures of dealing with software projects, tools, and services at scale, with constant monitoring and communication, at assured levels of quality.
7. ***Modern software applications:*** in-depth description of applications such as Web development (the Web is, essentially, that part of the Internet used to display output and receive input from human beings); scientific research and analysis; image processing; pattern recognition and generation via neural networks; software distributed across many different computers; cryptocurrencies, which can be used as a platform for trustless decentralized software; and the new field of quantum computing.
8. ***Hardware architectures:*** the concepts, structures, and layout of computer hardware. Hardware refers to physical electronic components; hardware architecture refers to how those components are structured and connected in order to run software; and software ultimately becomes ephemeral patterns of electricity within those physical components.
9. ***Hardware development:*** how to build simple computers from collections of electronic components.
10. ***Electronic components, transistors, semiconductor manufacturing:*** those electronic components which predated computers, along with individual transistors, the component from which computers are made, and an overview of the technologies and processes of fabricating interconnected transistors at scale.
11. ***Pre-industrial technologies:*** technologies of eras which predated electricity.
12. ***Fiction, culture, and history:*** human histories and changing human cultures, mostly through the lens of celebrated fictional narratives written over the last 150 years.
13. ***Cultural context:*** information about humanity at the time the Tech Tree was created; in particular, a snapshot of Wikipedia, a collectively generated repository of all sorts of information about our world. Due to Wikipedia's enormous size, this section is provided as encoded data, like the rest of the archive, rather than as visual/readable pages.

The first seven sections are devoted to software, the purpose and content of the GitHub Arctic Code Vault, and its uses and applications. The next four sections describe the technologies required to construct computers on which software might run. The remaining two are intended to illustrate the human context in which these technologies have been developed, the stories the cultures of our era told, the languages in which we told them, and the factual background and descriptions of the world in which we lived.

The Tech Tree is part of the much larger GitHub Arctic Code Vault. As such, it also includes, as an appendix, visual copies of the Guide to the GitHub Code Vault, along with an index of the archive's fifteen thousand most significant code repositories, including brief descriptions and locations within the archive.

It is perhaps worth noting that our advisory board stressed that ours is likely to be the best-documented era in human history by far, so bundling the Tech Tree with the archive is likely to be more convenient than essential for its inheritors. As such, it is entirely possible -- indeed quite likely -- that its value will consist largely of providing context regarding the era and culture in which the archive was created, rather than as a source of new and unavailable knowledge, though of course there are imaginable futures in which it plays the latter role.

What follows is a brief summary of each section, describing both the general topics it covers, and the works the Tech Tree includes to document our current understanding of those topics.

## Fundamentals of computing and the Internet

These books describe what computers are, from the silicon up -- electricity, transistors, binary logic, digital gates, bits, bytes, chips, ALUs, microprocessors, software -- as well as introducing what they can do. It also includes books which describe, at a high level, how computers can be connected together, and what that means. The works in question are:

- *The Pattern On The Stone* by W. Daniel Hills (Basic Books)
- *But How Do It Know?* by J. Clark Scott (John C Scott)
- *Code: The Hidden Language of Computer Hardware and Software* by Charles Petzold (Pearson Education)
- *Computer Fundamentals* by Anita Goel (Pearson Education)
- *How The Internet Really Works* by Ulrike Uhlig, Mallory Knodel, Niels ten Oever, Corinne Cath, and Catnip (No Starch)

## Algorithms and data structures

These are the fundamentals of computer science, and hence the foundation of software engineering; describing how data is structured and stored, and the most effective and efficient ways in which it can be processed.

- *The Art of Computer Programming* by Donald Knuth (Pearson)
- *Algorithmic Thinking* by Daniel Zingaro (No Starch)

- *Sequential and Parallel Algorithms and Data Structures* by Peter Sanders, Kurt Mehlhorn, Martin Dietzfelbinger, Roman Dementiev (Springer)
- *Cryptography* by Simon Rubinstein-Salzedo (Springer)
- *Mastering SciPy* by Francisco J. Blanco-Silva (Packt)
- *Everyday Data Structures* by William Smith (Packt)
- *Database Internals* by Alex Petrov (O'Reilly)
- *Introduction to the Theory of Computation* by Michael Sipser (Cengage)
- *Think Like A Programmer* by V. Anton Spraul (No Starch)
- *Write Great Code* by Randall Hyde (No Starch)

## Compilers, assembler, and operating systems

The purpose of the Archive Program is to preserve software, and these are the fundamental building blocks of software. These books help to explain how high-level written software becomes low-level electrical impulses:

- *A Practical Approach To Compiler Construction* by Des Watson (Springer)
- *The Secret Life Of Programs* by Jonathan Steinhart (No Starch)
- *The Art of Assembly Language* by Randall Hyde (No Starch)
- *Understanding the Linux Kernel* by Daniel P. Bovet and Marco Cesati (O'Reilly)
- *Mastering Linux Kernel Development* by Raghu Bharadwaj (Packt)
- *The Linux Programming Interface* by Michael Kerrick (No Starch)

## Programming languages

There are hundreds of programming languages; the enormous chart visualizing their evolution at the Computer History Museum is worth visiting if you're a developer, and we don't intend to document them all. Still, accessible book-length descriptions of a selection of the world's major languages seems desirable.

- *Introducing Python* by Bill Lubanovic (O'Reilly)
- *Comprehensive Ruby Programming* by Jordan Hudgens (Packt)
- *LISP, Lore, and Logic* by W. Richard Stark (Springer)
- *The C Programming Language* by Kernighan and Ritchie (Pearson)
- *Learn C The Hard Way* by Zed Shaw (Pearson)
- *Head First C* by David Griffiths, Dawn Griffiths (O'Reilly)
- *Effective C* by Robert Seacord (No Starch)
- *The C++ Primer* by Stanley B. Lippman, Josée Lajoie, and Barbara E. Moo (Pearson)
- *Programming Rust* by Jim Blandy and Jason Orendorff (O'Reilly)
- *The Rust Programming Language* by Steve Klabnik and Carol Nichols (No Starch Press)
- *The Go Programming Language* by Alan A. A. Donovan and Brian W. Kernighan (Pearson)
- *Head First Go* by Jay McGavren (O'Reilly)

- *Learning Java* by Patrick Niemeyer and Daniel Leuck (O'Reilly)
- *The Java Virtual Machine Specification* by Tim Lindholm, Frank Yellin, Gilad Bracha, and Alex Buckley (Pearson)
- *JavaScript: The Definitive Guide* by David Flanagan (O'Reilly)
- *JavaScript: The Good Parts* by Douglas Crockford (O'Reilly)
- *Automate the Boring Stuff with Python* by Al Sweigart (No Starch)
- *Learning Swift* by Jonathon Manning, Paris Buttfield-Addison, and Tim Nugent (O'Reilly)
- *Introducing Erlang* by Simon St. Laurent (O'Reilly)
- *Clojure Programming* by Chas Emerick, Brian Carper, and Christophe Grand (O'Reilly)
- *Clojure for the Brave and True* by Daniel Higginbotham (No Starch)
- *The Art of R Programming* by Norman Matloff (No Starch)
- *Mastering Scientific Computing with R* by Paul Gerrard and Radia M. Johnson (Packt)
- *Learn You A Haskell For Great Good* by Miran Lipovaca (No Starch)

## Networking and connectivity

Computers are great, but in a way, so 20th century; it's *networked* computers which are, at least arguably, the real technical revolution of the 21st. As such our networking protocols and technologies deserve considerable attention. We might hope our inheritors will either have long surpassed our networking, or will have the freedom to design anew rather than be shackled by all the compromises we've needed to make for the sake of backwards compatibility but either way, hopefully they can learn something from what we've done. Which is described by:

- *Cabling: The Complete Guide To Copper and Fiber-Optic Networking* by Andrew Oliviero and Bill Woodward (Wiley)
- *Ethernet: The Definitive Guide* by Charles E. Spurgeon and Joann Zimmerman (O'Reilly)
- *Understanding TCP/IP* by Alena Kabelová and Libor Dostálek (Packt)
- *TCP/IP Essentials* by Shivendra S. Panwar, Shiwen Mao, Jeong-dong Ryoo, and Yihan Li (Cambridge)
- *DNS and BIND* by Cricket Liu and Paul Albitz (O'Reilly)
- *BGP* by Iljitsch van Beijnum (O'Reilly)
- *HTTP: The Definitive Guide* by David Gourley, Brian Totty, Marjorie Sayer, Anshu Aggarwal, and Sailu Reddy (O'Reilly)
- *Implementing SSL / TLS Using Cryptography and PKI* by Joshua Davies (Wiley)
- *Nginx HTTP Server* by Martin Fjordvald and Clement Nedelcu (Packt)
- *sendmail* by Bryan Costales, Claus Assmann, George Jansen, and Gregory Neil Shapiro (O'Reilly)
- *Programming Internet Email* by David Wood (O'Reilly)
- *Data Communications and Networking* by Behrouz A. Forouzan (McGraw-Hill)
- *Computer Networking: Principles, Protocols, and Practice* by Olivier Bonaventure (Pearson)
- *Computer Networking: A Top-down Approach* by Jim Kurose (Pearson)

## Modern software development

The line-by-line act of writing software is quite different from the team-by-team process of developing, testing, integrating, and deploying it. A few key approaches, tools, and roles are described here, including, for obvious reasons, unpacking Git itself.

- *Working in Public: The Making and Maintenance of Open Source Software* by Nadia Eghbal (Stripe Press) /
- *The Manager's Path* by Camille Fournier (O'Reilly)
- *The Missing README* by Chris Riccomini and Dmitriy Ryaboy (No Starch)
- *Learning Agile* by Andrew Stellman and Jennifer Greene (O'Reilly)
- *Professional Git* by Brent Laster (Wiley)
- *Tangled Web: A Guide to Securing Modern Web Applications* by Michal Zalewski (No Starch)
- *Metasploit* by David Kennedy, Jim O'Gorman, Devon Kearns, and Mati Aharoni (No Starch)
- *Effective DevOps* by Jennifer Davis and Ryn Daniels (O'Reilly)
- *Site Reliability Engineering* edited by Betsy Beyer, Chris Jone, Jennifer Petoff & Niall Richard Murphy (O'Reilly)
- *Designing Distributed Systems* by Brendan Burns (O'Reilly)
- *Designing Data-Intensive Applications* by Martin Kleppmann (O'Reilly)
- *Exercises in Programming Style* by Cristina Videira Lopes (CRC Press)

## Modern software applications

It would take a tech forest, not a tree, to even try to describe all of the uses to which software is put. However, some depictions of how individual projects and libraries are knit together into powerful networked applications seem valuable, as do overviews of e.g. virtualization, "big data" software, and especially machine learning.

### Web development

- *Web Development with Node and Express* by Ethan Brown (O'Reilly)
- *Flask Web Development* by Miguel Grinberg (O'Reilly)
- *RESTful Web APIs* by Leonard Richardson, Mike Amundsen, Sam Ruby (O'Reilly)
- *Ruby on Rails Tutorial* by Michael Hartl (Pearson)
- *Django for Professionals: Production Websites with Python & Django* by William S. Vincent (Still River)

### Machine learning

- *Deep Learning from Scratch* by Seth Weidman (O'Reilly)
- *Deep Learning: A Visual Approach* by Andrew Glassner
- *Fundamentals of Deep Learning* by Nikhil Buduma and Nicholas Locascio (O'Reilly)
- *Practical Convolutional Neural Networks* by Mohit Sewak, Md. Rezaul Karim, and Pradeep Pujari (Packt)

- *Pattern Recognition and Machine Learning* by Christopher Bishop (Springer)
- *Generative Deep Learning* by David Foster (O'Reilly)
- *Strengthening Deep Neural Networks* by Katy Warr (O'Reilly)

### Virtualization and containers

- *Mastering Docker* by Scott Gallagher (Packt)
- *Kubernetes: Up and Running* by Brendan Burns, Joe Beda, and Kelsey Hightower (O'Reilly)
- *Spark: The Definitive Guide* by Bill Chambers, Matei Zaharia (O'Reilly)

### Reliability and scaling

- *Database Reliability Engineering* by Laine Campbell and Charity Majors (O'Reilly)
- *The Art of Capacity Planning* by Arun Kejariwal and John Allspaw (O'Reilly)

### Economics and sociotechnical systems

- *The Economics of Information Technology* by Hal Varian, Joseph Farrell and Carl Shapiro (Cambridge University Press)
- *Mastering Bitcoin* by Andreas Antonopoulos

## Hardware architectures

The spectrum of complexity from a single analog transistor to a modern multicore processor is, needless to say, difficult to summarize. This section tries to describe the basics of digital circuits and microprocessors, along with a few key references, before going on to hardware architectures and hardware design languages.

### Modern hardware architecture

- *Microprocessor Design* by Grant McFarland (McGraw-Hill)
- *Microprocessor Architecture* by Jean-Loup Baer (Cambridge)
- *Inside the Machine* by Jon Stokes
- *Introduction to Parallel Processing* by Behrooz Parhami (Springer)

### HDLs

- *IEEE Standard VHDL Language Reference Manual* (IEEE)
- *IEEE Standard for SystemVerilog* (IEEE)

### Example architecture details

- *Arduino: A Technical Reference* by J. M. Hughes (O'Reilly)
- *RISC-V Specifications* by the RISC-V International Technical Committee
- *Learning FPGAs* by Justin Rajewski (O'Reilly)

### Hardware development

- *Digital Computer Electronics* by Albert P. Malvino and Jerald A Brown (Career Education)
- *Computer Time Travel* by JS Walker (Oldfangled)
- *Theory, Design, and Applications of Unmanned Aerial Vehicles* by A.R. Jha (CRC Press)
- *Modern Robotics* by Kevin Lynch and Frank Park (Cambridge University Press)
- *Mastering ROS for Robotics Programming* by Lentin Joseph (Packt)

## Electronic components, transistors, semiconductor manufacturing

A more low-level analysis of fundamental electronic components and transistor-based circuitry, along with textbooks describing lithography and chip manufacturing. Obviously such manufacturing is essentially impossible to recreate from scratch (Moore's lesser-known second law described how fabricator costs increase just as chip density decreases) but these works could conceivably be of historical or even practical significance.

- *Fundamentals of Semiconductor Manufacturing* by Gary S. May and Simon M. Sze (Wiley)
- *Semiconductor Manufacturing Handbook* (both editions) by Hwaiyu Geng (McGraw-Hill)

## Pre-industrial technologies

These are the works which address the "romantic catastrophe" image of the archive's inheritors, who seek to reboot all of modern technological civilization from pre-industrial scratch. Such possible futures do exist, although they seem unlikely; furthermore, it seems possible that these works might help fill in gaps which arise in historical knowledge.

- *The Knowledge* by Lewis Dartnell (Penguin)
- *Caveman Chemistry* by Kevin Dunn (Universal)
- *Practical Blacksmithing* by M.T. Richardson (Weathervane)
- *Materials Handbook* by George S. Brady Henry R. Clauser, and John A. Vaccari (McGraw-Hill)
- *Practical Self-Sufficiency* by Dick and James Strawbridge (DK)
- *Oxford Handbook of Infectious Diseases and Microbiology* by Estée Török, Ed Moran, and Fiona Cooke (OUP)

## Fiction, culture, and history

It is our belief that culture is often best expressed through great works of fiction. As such, we sought to assemble a list of notable literary works (including / beginning with a few books of nonfiction) to convey, on a human level, the history and culture of our time. These are:

- *Chapman's Homer*
- *The Complete Works of William Shakespeare*
- *The Tale of Genji* by Murasaki Shikibu
- *Crime and Punishment* by Fyodor Dostoevsky
- *Extraordinary Popular Delusions and the Madness Of Crowds* by Charles Mackay
- *I, Claudius* by Robert Graves
- *Brave New World* by Aldous Huxley (Harper Perennial)
- *1984* by George Orwell (Signet Classics)
- *Cyberiad* by Stanislaw Lem (Mariner)
- *One Hundred Years Of Solitude* by Gabriel García Márquez (Harper Perennial)
- *Foucault's Pendulum* by Umberto Eco (Mariner)
- *Anathem* by Neal Stephenson (William Morrow)
- *Magic for Beginners* by Kelly Link

## Cultural context

This section of the Tech Tree is intended to convey both useful practical information from our culture, and a depiction of what it was like at the time the archive was written. It will consist of encoded data, rather than imaged pages, largely because its centerpiece, a snapshot of Wikipedia, is far too large for the latter format.

Wikipedia, while not without its flaws and omissions, is the most readily available proxy for "a written summary of our world." Note that this section is by no means intended as a complete depiction of humanity today: as our advisors stressed, this era is likely to be the best documented in all of human history, and such information is very unlikely to be difficult to find. Rather, it is intended as a convenience to indicate to the archive's inheritors the specific, particular context of the era in which the archive was written.