

# SELF-HOSTING — ON THE — DARK WEB

Building Private, Anonymous, and  
Resilient Services on the Tor Network



**David König**

# Self-Hosting on the Dark Web

Building Private, Anonymous, and Resilient Services on the Tor Network

Steve Publications

This book is available at <https://leanpub.com/self-hostingonthedarkweb>

This version was published on 2026-09-28



This is a [Leanpub](#) book. Leanpub empowers authors and publishers with the Lean Publishing process. [Lean Publishing](#) is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

© 2026 Steve Publications

# Contents

- Building Private, Anonymous, and Resilient Services on the Tor Network** 1
- Introduction** . . . . . 2
- Chapter 1: Landscapes of the Invisible Web** . . . . . 5
  - The Internet’s Layers: Surface Web . . . . . 5
  - The Deep Web: Everything Behind Login Forms . . . . . 6
  - The Dark Web: Purpose-Built Anonymity Networks . . . . . 7
  - Anonymity vs Pseudonymity vs Privacy . . . . . 8
  - Why Self-Host in the First Place . . . . . 9
  - Legitimate Use Cases and Real-World Motivations . . . . . 10
- Chapter 2: Origins and Design Philosophy** . . . . . 13
  - DARPA’s Purple Gang and Early Onion Routing . . . . . 13
  - Pathway, Mixmaster, and Cryptographic Precursors . . . . . 14
  - The Tor Project is Born: 2002 to Open Source . . . . . 15
  - Funding, Governance, and the Modern Tor Project . . . . . 16
  - Design Goals: Anonymity, Resistance, Usability . . . . . 17
  - The Political and Social Context . . . . . 18
- Chapter 3: The Tor Network Architecture** . . . . . 21
  - What is a Tor Node: Client, Relay, Bridge, Authority . . . . . 21
  - Directory Authorities: The Seven Guardians . . . . . 21
  - Directory Caches and Consensus Distribution . . . . . 21
  - Guard Relays and Stable Path Building . . . . . 21
  - Middle Relays: The Workhorse Tier . . . . . 21
  - Exit Relays: Touchpoints with the Clearnet . . . . . 21
  - Onion Services: Servers on the Tor Network . . . . . 22
- Chapter 4: The Tor Protocol Stack** . . . . . 23
  - The Transport Layer: TCP and TLS Handshakes . . . . . 23

## CONTENTS

|                                                                      |           |
|----------------------------------------------------------------------|-----------|
| Onion Skin Encryption: Layered Keys . . . . .                        | 23        |
| Link Keys and Circuit Keys . . . . .                                 | 23        |
| Key Exchange Protocols: NTor and Older Schemes . . . . .             | 23        |
| Cell Types and Protocol Messages . . . . .                           | 23        |
| Version Negotiation and Authentication . . . . .                     | 23        |
| <b>Chapter 5: Circuit Construction and Routing . . . . .</b>         | <b>25</b> |
| Guard Node Selection and Persistence . . . . .                       | 25        |
| Relay Weighting: Bandwidth, Flags, and History . . . . .             | 25        |
| Building a Three-Hop Circuit Step by Step . . . . .                  | 25        |
| Exit Node Selection Policies . . . . .                               | 25        |
| Circuit Lifetimes, Expiry, and Refresh . . . . .                     | 25        |
| Path Isolation and Connection Mapping . . . . .                      | 25        |
| <b>Chapter 6: Onion Services: Concept and Evolution . . . . .</b>    | <b>27</b> |
| From Hidden Services to Onion Services: Naming Changes . . . . .     | 27        |
| v2 vs v3: Architecture and Security Improvements . . . . .           | 27        |
| The Rendezvous Protocol: Meeting in the Middle . . . . .             | 27        |
| Introduction Points and Rendezvous Points . . . . .                  | 27        |
| Service Identifiers and Public Keys . . . . .                        | 27        |
| Onion Addresses: Encoding, Structure, and Validation . . . . .       | 27        |
| <b>Chapter 7: The v3 Onion Service Protocol . . . . .</b>            | <b>29</b> |
| Private Key Generation and Descriptor Signing . . . . .              | 29        |
| Publishing Onion Service Descriptors . . . . .                       | 29        |
| Introducing Points: Registration and Maintenance . . . . .           | 29        |
| Client Discovery: Hash-Based Directory Lookup . . . . .              | 29        |
| Rendezvous Point Selection and Negotiation . . . . .                 | 29        |
| End-to-End Circuit Establishment . . . . .                           | 29        |
| Descriptor Refresh and Fault Tolerance . . . . .                     | 30        |
| <b>Chapter 8: Threat Models and Attack Surfaces . . . . .</b>        | <b>31</b> |
| Tor's Formal Threat Model: What It Promises . . . . .                | 31        |
| Passive Network Observers and Local ISPs . . . . .                   | 31        |
| Global Adversaries and Correlation Attacks . . . . .                 | 31        |
| Malicious Relays: Observation and Manipulation . . . . .             | 31        |
| Endpoint Compromise and Application Leaks . . . . .                  | 31        |
| The Limits of Network-Layer Anonymity . . . . .                      | 31        |
| <b>Chapter 9: Traffic Analysis and Correlation Attacks . . . . .</b> | <b>33</b> |

CONTENTS

|                                                                      |           |
|----------------------------------------------------------------------|-----------|
| Passive Correlation: Timing and Volume . . . . .                     | 33        |
| Active Attacks: Padding and Probe Techniques . . . . .               | 33        |
| Statistical Fingerprinting and Machine Learning . . . . .            | 33        |
| Network-Wide Correlation at Scale . . . . .                          | 33        |
| Defenses: Padding, Guard Diversity, Path Length . . . . .            | 33        |
| Theoretical Guarantees vs Practical Risk . . . . .                   | 33        |
| <b>Chapter 10: Fingerprinting and Side Channels . . . . .</b>        | <b>35</b> |
| Browser Fingerprinting and Tor Browser Countermeasures . . . . .     | 35        |
| TLS Fingerprinting and JA3 Identifiers . . . . .                     | 35        |
| JavaScript Tracking and DOM Fingerprinting . . . . .                 | 35        |
| Application-Level Metadata Leaks . . . . .                           | 35        |
| Timing Side Channels in Web Applications . . . . .                   | 35        |
| Weeping Cryptography and Hidden Data . . . . .                       | 35        |
| <b>Chapter 11: Sybil Attacks, Censorship, and Bridges . . . . .</b>  | <b>37</b> |
| Sybil Relays and Identity Fabrication . . . . .                      | 37        |
| Directory Authority Consensus and Weighting . . . . .                | 37        |
| ISP-Level Blocking and Keyword Filtering . . . . .                   | 37        |
| Bridges: Unlisted Entry Points . . . . .                             | 37        |
| Pluggable Transports: obfs4 and WebTunnel . . . . .                  | 37        |
| <b>Chapter 12: Infrastructure Planning and Design . . . . .</b>      | <b>38</b> |
| Jurisdiction and Legal Environment . . . . .                         | 38        |
| Hosting Options: Colocation, VPS, Home Hosting . . . . .             | 38        |
| Hardware Requirements and Sizing . . . . .                           | 38        |
| Network Topology: Isolation and Segmentation . . . . .               | 38        |
| Power, Connectivity, and Redundancy . . . . .                        | 38        |
| <b>Chapter 13: Operating System Hardening and Security . . . . .</b> | <b>39</b> |
| OS Selection: Debian, Alpine, or Specialized Distributions . . . . . | 39        |
| Minimal Installation and Surface Reduction . . . . .                 | 39        |
| Kernel Parameters and Sysctl Hardening . . . . .                     | 39        |
| User, Group, and Permission Design . . . . .                         | 39        |
| Full Disk Encryption and Key Management . . . . .                    | 39        |
| Intrusion Detection and Log Integrity . . . . .                      | 39        |
| <b>Chapter 14: Network Configuration and Firewalls . . . . .</b>     | <b>41</b> |
| Firewall Fundamentals: iptables and nftables . . . . .               | 41        |
| Default-Deny Policies and Minimal Open Ports . . . . .               | 41        |

|                                                                                   |           |
|-----------------------------------------------------------------------------------|-----------|
| Blocking All Exit Relay Traffic . . . . .                                         | 41        |
| DNS Resolution Without Information Leakage . . . . .                              | 41        |
| Network Interface Isolation . . . . .                                             | 41        |
| Detecting and Blocking Direct Access Attempts . . . . .                           | 41        |
| <b>Chapter 15: Tor Installation and Core Configuration . . . . .</b>              | <b>43</b> |
| Installing Tor: Package Managers and Compiling . . . . .                          | 43        |
| The torrc File: Structure and Directives . . . . .                                | 43        |
| Configuring HiddenServiceDir and HiddenServicePort . . . . .                      | 43        |
| Running Tor as a Systemd Service . . . . .                                        | 43        |
| Validating Circuit and Service Status . . . . .                                   | 43        |
| <b>Chapter 16: Deploying Your First Onion Service . . . . .</b>                   | <b>44</b> |
| Minimal Web Server: Nginx Bound to Loopback . . . . .                             | 44        |
| First Hidden Service Configuration . . . . .                                      | 44        |
| Testing with Tor Browser from a Remote Machine . . . . .                          | 44        |
| Verifying No Direct IP Accessibility . . . . .                                    | 44        |
| Checking Descriptor Publication . . . . .                                         | 44        |
| Troubleshooting First-Time Deployments . . . . .                                  | 44        |
| <b>Chapter 17: Architectural Pattern: Single-Host Minimal Service . . . . .</b>   | <b>46</b> |
| Single Machine Topology Diagram . . . . .                                         | 46        |
| Nginx Configuration for Loopback Binding . . . . .                                | 46        |
| Tor Hidden Service Configuration . . . . .                                        | 46        |
| Systemd Service Units . . . . .                                                   | 46        |
| SSL/TLS Considerations for Onion Services . . . . .                               | 46        |
| Backup and Recovery Procedure . . . . .                                           | 46        |
| <b>Chapter 18: Architectural Pattern: Hardened Network Segmentation . . . . .</b> | <b>48</b> |
| Three-Tier Network Topology . . . . .                                             | 48        |
| DMZ Host: Tor and Reverse Proxy . . . . .                                         | 48        |
| Application Tier Isolation . . . . .                                              | 48        |
| Database Tier and Access Controls . . . . .                                       | 48        |
| Internal Firewall Rules Between Zones . . . . .                                   | 48        |
| End-to-End Request Flow Diagram . . . . .                                         | 48        |
| <b>Chapter 19: Architectural Pattern: Containerized Multi-Service . . . . .</b>   | <b>50</b> |
| Docker Networking and Host vs Container Mode . . . . .                            | 50        |
| Tor in a Container vs on the Host . . . . .                                       | 50        |
| Docker Compose for Multi-Service Stacks . . . . .                                 | 50        |

CONTENTS

|                                                                              |           |
|------------------------------------------------------------------------------|-----------|
| Reverse Proxy Routing to Multiple Onion Services . . . . .                   | 50        |
| Persistent Volumes and Secrets Management . . . . .                          | 50        |
| Build, Deploy, and Update Workflow . . . . .                                 | 50        |
| <b>Chapter 20: Private and Authorized Onion Services . . . . .</b>           | <b>52</b> |
| Private Onion Services Concept . . . . .                                     | 52        |
| Client Authorization and ClientAuthFile . . . . .                            | 52        |
| Generating and Distributing Authorization Tokens . . . . .                   | 52        |
| Combining Tor with Web-Based Authentication . . . . .                        | 52        |
| Multi-Factor Authentication over Tor . . . . .                               | 52        |
| Managing Revocation and Key Rotation . . . . .                               | 52        |
| <b>Chapter 21: Application-Specific Deployments . . . . .</b>                | <b>54</b> |
| File Hosting: Nextcloud and FileBrowser . . . . .                            | 54        |
| Messaging: Matrix, XMPP, and IRC over Tor . . . . .                          | 54        |
| REST and GraphQL APIs behind Onion Services . . . . .                        | 54        |
| Remote Administration: SSH and Web Consoles . . . . .                        | 54        |
| Personal Cloud and Sync Services . . . . .                                   | 54        |
| Monitoring and Observability Stacks . . . . .                                | 54        |
| <b>Chapter 22: Performance, Scaling, and Optimization . . . . .</b>          | <b>56</b> |
| Tor Latency Characteristics and Benchmarks . . . . .                         | 56        |
| Bandwidth Limits and Congestion Behavior . . . . .                           | 56        |
| Caching and CDN Considerations . . . . .                                     | 56        |
| Load Balancing Multiple Onion Services . . . . .                             | 56        |
| Database Performance Behind Tor . . . . .                                    | 56        |
| When to Leave Tor Behind for Internal Traffic . . . . .                      | 56        |
| <b>Chapter 23: Operations, Monitoring, and Maintenance . . . . .</b>         | <b>58</b> |
| Update Cadence and Security Patches . . . . .                                | 58        |
| Onion Service Key Rotation . . . . .                                         | 58        |
| Log Aggregation and Alerting . . . . .                                       | 58        |
| Backup Strategies: Encrypted and Offsite . . . . .                           | 58        |
| Incident Response and Breach Containment . . . . .                           | 58        |
| Operational Security Procedures . . . . .                                    | 58        |
| <b>Chapter 24: Troubleshooting, Diagnostics, and Failure Modes . . . . .</b> | <b>60</b> |
| Verifying Onion Service Reachability . . . . .                               | 60        |
| Descriptor Publication Failures . . . . .                                    | 60        |
| Circuit Construction Problems . . . . .                                      | 60        |

|                                                                          |           |
|--------------------------------------------------------------------------|-----------|
| Backend Connectivity Issues . . . . .                                    | 60        |
| HiddenServiceDir Permission Errors . . . . .                             | 60        |
| Network-Level Issues . . . . .                                           | 60        |
| Application-Level Issues . . . . .                                       | 61        |
| Verifying No IP Address Leakage . . . . .                                | 61        |
| Debugging Client-Side Access Issues . . . . .                            | 61        |
| <b>Chapter 25: Future Directions and Emerging Technologies . . . . .</b> | <b>62</b> |
| Post-Quantum Cryptography and Proposal 353 . . . . .                     | 62        |
| Quantum-Safe Onion Service Addresses . . . . .                           | 62        |
| Network Growth and Resilience . . . . .                                  | 62        |
| Decentralized Alternatives and Complementary Technologies . . . . .      | 62        |
| Browser Fingerprinting Evolutions . . . . .                              | 62        |
| Machine Learning and Traffic Analysis . . . . .                          | 62        |
| Regulatory and Legal Landscape . . . . .                                 | 63        |
| Practical Skills for the Next Decade . . . . .                           | 63        |
| <b>Conclusion . . . . .</b>                                              | <b>64</b> |
| <b>References . . . . .</b>                                              | <b>65</b> |

# Building Private, Anonymous, and Resilient Services on the Tor Network

This book takes you from the foundational concepts of the Tor network through deep technical understanding of onion routing protocols to hands-on deployment of production-grade self-hosted services accessible only through onion addresses. You will learn how Tor's layered encryption and multi-hop routing actually work at the protocol level, how v3 onion services enable servers to communicate without exposing their physical location, what real threats and attack vectors exist, and how to architect, configure, and maintain resilient onion service infrastructures. Every chapter is grounded in Tor specifications, research literature, and working configurations. Whether you are hosting a personal website, a private messaging system, a file server, or any other privacy-critical service, this book provides the architectural knowledge and practical skills to build it correctly.

# Introduction

In 2013, Edward Snowden's disclosures revealed that the National Security Agency operated programs capable of collecting bulk internet traffic across the globe. Among the details was that NSA analysts used a tool called XKeyscore to target one of The Tor Project's own onion services and intercept user communications, not by breaking Tor's cryptography, but by compromising the endpoint server itself. The episode was instructive in two ways. First, it showed that network-layer anonymity alone is never sufficient if the edges of communication are weak. Second, it showed that even the most capable intelligence agencies found it worth monitoring Tor, which meant that for people facing state surveillance, corporate data collection, or internet censorship, Tor remained one of the most powerful tools available to protect privacy and resist control.

The world has changed since those revelations, but the motivations for using Tor have only grown more urgent. Governments deploy deep packet inspection and IP-level blocking. Corporations harvest browsing histories and sell behavioral profiles. Internet service providers log connections and, in some jurisdictions, are compelled to hand data over to authorities. Meanwhile, activists, journalists, whistleblowers, researchers, ordinary citizens, and system administrators increasingly need ways to communicate and host services without revealing their location, identity, or relationship to others.

Self-hosting on the Tor network addresses these needs directly. By deploying a Tor onion service, you gain a server that users can reach without your physical IP address ever being exposed, a service that resists ISP-level blocking because the traffic is encrypted end-to-end within the Tor network, and a presence on the internet that does not depend on conventional domain name systems, certificate authorities, or public cloud providers. You own the infrastructure, you control the keys, and you can design the architecture to match your specific threat model.

This is not a book about illegal activity, criminal anonymity, or evading law enforcement. Onion services are legitimate infrastructure tools. The Tor Project explicitly supports their use for journalistic secure drop systems like SecureDrop, for anti-censorship in countries where independent media

are blocked, for private communications between healthcare providers and patients, for academic research into censorship and surveillance, and for any number of privacy-preserving services that simply should not be accessible from the public internet or discoverable by casual network reconnaissance. This book treats onion services as what they are: serious, technically sophisticated networking infrastructure that happens to provide strong anonymity guarantees.

What follows is organized as a journey from understanding to implementation. The first part lays the foundations. We will explore where onion routing came from, how the Tor Project grew from military research into a global nonprofit, and what the Tor network actually looks like in practice. We will define the vocabulary: surface web, deep web, dark web, relays, bridges, directory authorities, circuits, guard nodes, and onion services. Without a clear mental model of how these pieces fit together, configuring and securing an onion service becomes a matter of copying someone else's configuration without understanding what it does or why it matters.

The second part goes deep into Tor's technical internals. We will walk through the protocol stack from the TCP and TLS connections between relays to the layered encryption of onion skin circuits, the public-key operations of the ntor key exchange, the relay selection algorithms that build three-hop paths, and the complete lifecycle of establishing and accessing a v3 onion service. You will learn exactly how introduction points, rendezvous points, and onion service descriptors work together to allow a client to reach a server without either party learning the other's public IP address. Understanding these mechanisms is essential for troubleshooting, for making informed security decisions, and for recognizing the real boundaries of Tor's guarantees.

The third part examines threats. We will define Tor's formal threat model, which is important because what Tor protects against is not the same as what Tor promises, and there is a gap between them that matters. We will cover traffic analysis and correlation attacks, browser and TLS fingerprinting, Sybil attacks, censorship techniques, and the practical deanonymization vectors that have succeeded in real-world incidents. This chapter is deliberately unsentimental. The goal is to help you form an accurate threat model for your own situation, which is the only way to design a genuinely secure deployment.

The fourth and largest part is the practical guide to deploying onion services. We will start with infrastructure planning and operating system hardening, move through network configuration and firewall rules, and then

configure Tor itself as an onion service. You will see complete, working configurations for a minimal single-host service, a hardened architecture with network segmentation, and a containerized multi-service deployment using Docker Compose. Each architecture comes with explanations of every significant technical decision, including why you bind to loopback instead of a public IP, how to use Unix sockets to avoid local network leaks, how to configure client authorization to restrict access to known users, and how to verify that your server's real IP address is not accidentally exposed.

The final sections cover long-term operation: monitoring, logging practices that do not undermine anonymity, backup strategies for onion service keys, disaster recovery procedures, troubleshooting workflows for common failure modes, and performance optimization under Tor's inherent latency constraints. We will close by looking at the future of Tor's anonymity infrastructure, including post-quantum cryptography migration plans and emerging protocol improvements that will affect how we deploy onion services in the years ahead.

You should approach this book with a clear understanding of your own prerequisites and assumptions. I assume you are comfortable with Linux command-line administration, understand basic networking concepts like IP addresses, TCP ports, and DNS, have worked with configuration files in formats like Nginx or systemd, and know how to edit files in a terminal editor. I do not assume you know anything about Tor beyond the basic idea that it is an anonymity network. I do assume you have access to a server or a machine you can use for testing, though many concepts can be explored on a local system before deploying to production.

Let us begin by mapping the terrain. Before we configure a single onion service, we need to understand the landscapes where onion services live: the surface web, the deep web, and the dark web, and why the distinctions matter.

# Chapter 1: Landscapes of the Invisible Web

## The Internet's Layers: Surface Web

The internet is not a single thing. It is a layered ecosystem of networks, protocols, and services, each with its own visibility properties and access controls. When people talk about the “surface web” they mean the portion of the internet that is indexed by conventional search engines and accessible through standard browsers without special software or credentials. This includes everything from Wikipedia and news sites to corporate webpages and government portals.

The surface web has three defining characteristics. First, it is discoverable. Search engine crawlers follow hyperlinks and populate searchable databases. If you want to find a website on the surface web, you can type keywords into Google and expect relevant results. Second, it is identified by the Domain Name System, which maps human-readable names like `example.com` to numeric IP addresses like `93.184.216.34`. Third, it is publicly routable. Any internet-connected device can send a packet to a surface web server's IP address and expect to reach it, assuming firewalls and network policies do not block the traffic.

These properties make the surface web convenient, fast, and widely interoperable. They also make it inherently visible. When you host a service on the surface web, you are making a public commitment to a specific IP address in a specific jurisdiction. That address can be scanned by automated tools looking for open ports, vulnerable software, or misconfigurations. It can be blocked by ISPs at the request of governments or copyright holders. It can be correlated with your identity through WHOIS records, hosting provider databases, or traffic analysis. For many services this is perfectly acceptable. For privacy-critical services it is often unacceptable.

Consider a journalist setting up a secure drop system for sources in an authoritarian regime. If the drop system is hosted on a conventional server

with a public IP address, an adversary who can observe network traffic can see which devices connect to which server, potentially linking sources to their activity. An adversary who can pressure the hosting provider can obtain server logs, DNS records, and billing information. An adversary who controls national internet infrastructure can block access to the server's IP address entirely. None of these attacks require breaking any cryptography. They exploit the inherent visibility of surface web infrastructure.

This is not to say that all surface web services should be paranoid. Most web applications have threat models that comfortably include public IPs and centralized hosting. An e-commerce site, for example, benefits from being findable and reachable. The point is that the surface web's design optimizes for accessibility and discoverability, not for anonymity or censorship resistance. When your requirements include hiding your location, protecting your users' identities, or resisting network-level blocking, the surface web becomes the wrong tool for the job.

## The Deep Web: Everything Behind Login Forms

The term “deep web” is frequently misunderstood. In popular discourse it is sometimes used interchangeably with “dark web” to evoke mysterious and illicit online spaces. Technically, however, the deep web refers to a much simpler and more mundane concept: any part of the internet that is not indexed by search engines.

Your email inbox is on the deep web. Your bank's online portal is on the deep web. Your company's internal wiki is on the deep web. Medical records, academic databases, government intranets, and password-protected forums are all on the deep web. Search engines cannot crawl content behind authentication walls, so they never appear in search results. This is not because the content is hidden for nefarious reasons. It is because it is protected for practical reasons: privacy, security, access control, or licensing restrictions.

The deep web has no single defining technology. It is simply the complement of the surface web: everything on the internet that is not publicly indexed. Most estimates place the deep web as orders of magnitude larger than the surface web in terms of total data, simply because the volume of private, authenticated content dwarfs the volume of public webpages.

The distinction between surface web and deep web is primarily about discoverability. A deep web service may still be hosted on a conventional

server with a public IP address. It may still use standard domain names and be accessible over the conventional internet. The fact that it is not indexed does not make it anonymous. An adversary who knows the URL can still find it. An adversary who can monitor network traffic can still observe connections to it. The protection provided by non-indexing is minimal compared to the protections provided by anonymity networks like Tor.

Confusing the deep web with the dark web leads to inaccurate threat models. Hosting a password-protected website behind a login form does not make it anonymous, even though it is technically on the deep web. Anonymity requires architectural features like multi-hop routing, layered encryption, and hidden server locations, which the deep web as a concept does not provide.

## The Dark Web: Purpose-Built Anonymity Networks

The dark web is a subset of the deep web with a specific and deliberate design: it consists of services that are only accessible through purpose-built anonymity networks. These networks are engineered to provide strong guarantees about the privacy of both clients and servers, often at the cost of convenience and performance.

The most widely known anonymity network is Tor. Others include I2P, Freenet, Fingerprint, and Zephyr. Each of these networks uses different architectural approaches, but they all share a common goal: making it difficult or impossible for observers to determine who is communicating with whom.

Dark web services are typically accessed through specialized clients. Tor Browser is the most famous example, but there are also mobile clients, command-line tools, and application libraries that route traffic through anonymity networks. Dark web services themselves have special addresses that cannot be resolved by the public DNS system. On Tor these addresses end in `.onion`. On I2P they end in `.b32.i2p`. They are not subdomains in the traditional sense. They are identifiers that encode cryptographic keys and are resolved internally by the anonymity network's own discovery mechanisms.

When you connect to a dark web service, your traffic never leaves the anonymity network. It is routed through multiple volunteer-operated nodes that each know only the immediately previous and next hop in the chain. The final destination's physical IP address is never revealed to the client. The client's IP address is never revealed to the destination. This is fundamentally

different from a deep web service behind a login form, where both parties' addresses are visible to network infrastructure.

The dark web's reputation has been shaped by its use in illicit activities. Criminal marketplaces, darknet forums, and services facilitating illegal transactions have historically operated on dark web networks. This has attracted law enforcement attention, led to high-profile takedowns, and contributed to public associations between dark web and criminality. However, the same technologies that protect criminal actors also protect legitimate users: whistleblowers submitting evidence to journalists, political dissidents communicating under surveillance, researchers conducting sensitive studies, and ordinary users resisting pervasive data collection.

The fact that a technology has dual-use properties does not make it inherently good or bad. Encryption protects banking transactions and hacker ransomware. Virtual private networks protect corporate remote workers and malware command-and-control channels. Anonymity networks protect journalists in repressive regimes and marketplace operators evading law enforcement. The architecture is neutral. The outcomes depend on who uses it and for what purpose.

This book takes anonymity infrastructure seriously as a legitimate tool for legitimate purposes. The technical mechanisms are the same regardless of use case, and understanding them thoroughly is valuable whether you are deploying a personal blog on Tor, a secure whistleblower drop system, or an internal service you do not want exposed to the public internet.

## **Anonymity vs Pseudonymity vs Privacy**

These three terms are frequently conflated, but they describe distinct properties. Clarity about the differences is essential for understanding what Tor can and cannot do.

Privacy is the broadest concept. It refers to the ability to control what information about you is revealed to others and under what conditions. Privacy can mean not disclosing your browsing history, not revealing your location, not having your identity linked to your financial transactions, or any number of other protections depending on context. Privacy is about boundaries: keeping certain information within a circle of your choosing.

Pseudonymity is a form of privacy in which you operate under an identity that is not your real-world legal identity. This pseudonym can be persistent, allowing you to build reputation and relationships over time while still separating your online actions from your legal name. Cryptocurrency addresses, forum usernames, and GitHub accounts are examples of pseudonymous identifiers. Pseudonymity protects against direct identification but does not necessarily protect against linking your activities together under the same pseudonym. If an adversary can determine that pseudonym X belongs to real person Y, then the protection is broken.

Anonymity is the strongest condition. It means that your identity cannot be determined from your actions, even in principle given the information available. A truly anonymous system does not leak information that allows an observer to distinguish between two users or to trace actions back to a specific individual. Anonymity is harder to achieve than pseudonymity because it requires defending against linkage attacks as well as direct identification.

Tor's goal is anonymity, but it achieves this through a combination of mechanisms, each addressing a different threat vector. At the network layer, Tor's multi-hop routing and layered encryption protect against network observers who would try to link your IP address to your destination. At the application layer, Tor Browser's fingerprinting protections prevent websites from identifying you based on your browser configuration. These layers work together, and weakness in any one layer can undermine the whole system.

It is also important to understand what Tor does not provide. Tor does not provide cryptographic authentication of the other party unless you are connecting to a Tor onion service whose identity key you have independently verified. Tor does not protect against malware on your device that exfiltrates your browsing history. Tor does not protect against websites serving you identifying content based on your account login. Anonymity on Tor requires discipline from both the technology and the user.

## Why Self-Host in the First Place

Self-hosting means running your own infrastructure instead of relying on third-party providers. In the context of the dark web and Tor, this means deploying and maintaining your own onion services rather than using managed hosting services or cloud providers that offer Tor integration.

There are several reasons why self-hosting is attractive for onion services, even for people who might otherwise use managed hosting for regular services.

First, control. When you self-host, you control the entire stack: the operating system, the applications, the configuration, the keys, and the data. There is no intermediary that could be compelled by legal process to hand over your data, no terms of service that could be changed retroactively to forbid your content, and no corporate entity that could decide your service violates its acceptable use policy. This matters significantly if you are hosting services that might be controversial, politically sensitive, or legally precarious in certain jurisdictions.

Second, security. When you host your own infrastructure, you are responsible for its security, but you are also the only party that can understand and optimize for your specific threat model. Managed providers must design for the lowest common denominator and cannot account for edge cases that might be critical to you. A self-hosted onion service can be hardened, audited, and configured to minimize attack surface in ways that a shared hosting environment cannot match.

Third, cost. For many use cases, self-hosting is significantly cheaper than commercial alternatives. Running a modest onion service on a single server or even a home-hosted machine costs only the expense of electricity and internet bandwidth. There are no monthly fees, no premium tiers, and no per-user charges. For organizations with limited budgets but high privacy requirements, self-hosting can be the only economically viable option.

Fourth, principle. For some people and organizations, self-hosting is a statement of values. It is a commitment to digital autonomy, to resisting the centralization of the internet into a handful of corporate platforms, and to building infrastructure that aligns with the ethos of the tools they use. If you care about privacy and censorship resistance enough to use Tor, you may also care about whether the services you access are themselves operated on a decentralized, independently controlled infrastructure.

Self-hosting is not without drawbacks. It requires technical skill. It demands ongoing maintenance. If you misconfigure something, the consequences fall on you. If your server goes down, there is no support team to call. The bar is higher than it is for managed services, and that is a real cost. But for privacy-critical services, the trade-off is often worth it.

## Legitimate Use Cases and Real-World Motivations

It is worth stating explicitly that onion services are used for many legitimate and valuable purposes. Listing real-world examples helps establish that this technology is not inherently associated with illicit activity.

SecureDrop is perhaps the most famous legitimate onion service. Used by major news organizations including The Guardian, The New York Times, and The Washington Post, SecureDrop provides a way for whistleblowers to submit confidential information without revealing their identity. The SecureDrop onion service is hosted on an air-gapped server, receives submissions through Tor, and stores encrypted messages for journalists to review. Without Tor and onion services, SecureDrop would be far less secure and far less usable for sources in high-risk situations.

Journalists in authoritarian regimes routinely use onion services to communicate with sources, publish uncensored reporting, and receive information from within their countries. When conventional websites are blocked by national firewalls, onion services can remain accessible through bridges and pluggable transports that evade censorship.

Academic researchers use onion services to study censorship patterns, to deploy experiments in countries where research participation might carry risks, and to communicate with colleagues across politically sensitive borders. The anonymity guarantees of onion services enable research that would otherwise be impossible or unethical to conduct.

Healthcare providers have deployed onion services for patients in repressive environments who need access to medical information or telehealth services without revealing their conditions to surveillance systems. Mental health providers, HIV testing services, and reproductive health organizations have all used Tor to reach vulnerable populations.

System administrators and security researchers use onion services to expose internal monitoring systems, vulnerability disclosure portals, and collaboration tools that should not be accessible from the public internet. Some security operations use onion services for incident response coordination across organizations that would prefer not to establish direct network connections.

Privacy-conscious individuals use onion services to host personal websites, blogs, and file sharing systems without exposing their location or having their browsing activity tracked by advertisers and data brokers.

These examples are not exhaustive, and they demonstrate a fundamental point: onion services are infrastructure technology. They solve real problems for real people facing real constraints. Understanding how to self-host an onion service well is a valuable skill for anyone whose work involves privacy, security, journalism, activism, research, or digital autonomy.

# Chapter 2: Origins and Design Philosophy

## DARPA's Purple Gang and Early Onion Routing

The story of Tor begins not with activists or privacy advocates, but with military researchers at the United States Naval Research Laboratory in the mid-1990s. At the time, the internet was growing rapidly, but its lack of security was glaring. Traffic moved largely unencrypted, routing information was trivially observable, and anyone with access to network infrastructure could see who was communicating with whom. For the US Navy, which routinely transmitted sensitive communications over public networks, this was a serious problem.

The answer came from three researchers: Paul Syverson, a mathematician; Michael G. Reed, a computer scientist; and David Goldschlag, another computer scientist at the Naval Research Laboratory. In 1995 they began working on a scheme to protect the routing information of network communications. Their insight was elegant: if traffic could be routed through a sequence of intermediate servers, each of which removed a layer of encryption before forwarding to the next, then no single server could see both the origin and the destination of the communication. Each hop would know only its immediate predecessor and successor, like a game of telephone where each participant repeats a message but does not know who originally sent it or who will ultimately receive it.

The analogy of peeling layers off an onion led to the name “onion routing.” Each intermediate server in the chain strips away one layer of encryption, revealing only the address of the next hop. By the time the message reaches its final destination, all the intermediate layers have been removed, but no single node along the path knows both where the message came from and where it was going.

The team published their work in 1996 at the First International Workshop on Information Hiding in Cambridge, United Kingdom. The paper, titled “Hiding Routing Information,” presented the first formal description of onion

routing as a mechanism for protecting communications privacy. The work was funded by the Office of Naval Research and later supplemented by DARPA's High Confidence Networks Program. The US Navy obtained a patent on the technique in 1998.[1]

The original onion routing design was research software, not a production system. It required participants to have relatively technical expertise and was intended as a proof of concept rather than something ordinary people could use. But the fundamental idea was solid, and it demonstrated that practical, low-latency anonymity on the internet was achievable through clever protocol design rather than hardware modifications or kernel-level changes.

## Pathway, Mixmaster, and Cryptographic Precursors

Onion routing did not emerge in isolation. It was part of a broader research tradition in anonymous communications that dated back decades. Understanding these precursors helps clarify what onion routing contributed and what trade-offs it made compared to alternative designs.

The concept of mix networks originated with David Chaum in 1981. A mix network routes messages through a sequence of “mix” servers that collect messages in batches, shuffle their order, strip off layers of encryption, and forward them to the next mix. Chaum's design provided strong anonymity guarantees because the batching and reordering made it computationally infeasible to correlate incoming and outgoing messages. The problem was that this approach introduced significant latency, often measured in minutes or even hours, because messages had to wait for a batch to fill before being processed. Mix networks were well-suited for applications like anonymous email but entirely unsuitable for interactive tasks like web browsing.

Mixmaster, developed in the 1990s, was a widely used implementation of Chaum's ideas for anonymous email. It became a cornerstone of early cypherpunk privacy tools. The same latency trade-off applied: Mixmaster was excellent for sending anonymous messages but useless for real-time communication.

Tarzan was another early anonymous communication project from MIT's Free Haven Project in the late 1990s. It used a layered encryption scheme similar to onion routing but with different protocol design choices. Tarzan

demonstrated many of the architectural patterns that would later appear in Tor, including circuit-based routing and directory servers for relay discovery.

What onion routing brought to the table was low latency. Unlike mix networks, onion routing did not rely on batching and reordering. Each circuit was built on demand and could be used for interactive communication with response times similar to conventional internet connections. This made onion routing usable for web browsing, instant messaging, and other real-time applications, which was a significant advantage over earlier designs. The trade-off was that low latency made onion routing more vulnerable to timing-based traffic analysis than high-latency mix networks, a tension that continues to shape Tor's design to this day.

## The Tor Project is Born: 2002 to Open Source

By the early 2000s, the concept of onion routing had proven itself in research but still lacked a practical, deployable implementation. That changed when Roger Dingledine, then a graduate student at MIT, began working on a second-generation onion routing system. Dingledine had been involved in the Free Haven Project and was interested in building privacy tools that ordinary people could actually use.

In 2001 and 2002, Dingledine, working with his classmate Nick Mathewson and in collaboration with Paul Syverson from the Naval Research Laboratory, began developing what would become Tor. They received funding from the Naval Research Laboratory to build a production-quality implementation of onion routing. Dingledine committed the first code in 2002, and the network was deployed in October of that year under a free software license.[1]

The name “Tor” is an acronym for “The Onion Router,” though Dingledine has also noted that he liked that it is a standalone word rather than just an acronym. The initial network was tiny: a few dozen nodes, most of them in the United States, running on machines operated by the researchers themselves. But it was functional, and it demonstrated that a volunteer-operated onion routing network could work at scale.

A pivotal moment came in 2004 when the Electronic Frontier Foundation began funding Tor's development. The EFF's involvement brought Tor to the attention of the privacy advocacy community and provided the financial stability needed for continued development. That same year, Dingledine, Mathewson,

and Syverson published the paper “Tor: The Second-Generation Onion Router” at the USENIX Security Symposium, which became the canonical description of Tor’s design and architecture.[1] The paper presented Tor not just as an implementation but as a research contribution: a circuit-based, low-latency anonymous communication system with perfect forward secrecy, configurable exit policies, integrity checking, and directory servers for decentralized relay discovery.

## **Funding, Governance, and the Modern Tor Project**

As Tor’s user base grew, the need for organizational structure became apparent. Individual researchers could not sustainably manage the development, operations, and legal affairs of an increasingly important piece of global privacy infrastructure. The solution was to create a dedicated nonprofit organization.

The Tor Project, Inc. was founded on December 22, 2006, as a 501(c)(3) nonprofit corporation. It was co-founded by Roger Dingledine, Nick Mathewson, and five others. The Electronic Frontier Foundation acted as the project’s fiscal sponsor in its early years, and early financial supporters included the US Navy, the US State Department, the MacArthur Foundation, and various corporate donors interested in internet privacy and security.[1]

Tor Project’s funding has always been a mix of government and non-government sources, and this fact has occasionally been the subject of scrutiny and debate. Government funding from US agencies has been particularly controversial, with some critics questioning whether an anonymity tool funded by intelligence agencies could be trusted. Tor Project’s response has been consistent: the software is open source, and anyone can audit the code for backdoors or surveillance mechanisms. The source code is publicly available, and independent researchers have reviewed it extensively. The argument is that transparency and verifiability provide stronger guarantees than speculation about funding sources.

The Tor Project has grown into an organization with paid staff and volunteers working on a wide range of activities: core protocol development, Tor Browser engineering, bridge operations and pluggable transports, network health monitoring, user education, legal advocacy, and grant management. It operates directory authorities, runs bridge servers to help users in censored countries, and supports the broader ecosystem of tools that depend on Tor, including SecureDrop, PGP, and various anonymity-preserving applications.

Today, the Tor Project is one of the most important privacy infrastructure organizations in the world. Its software is used by millions of people, its protocols are studied by researchers, and its infrastructure is critical for journalists, activists, and dissidents in countries where the internet is monitored or controlled. Understanding the organization's history and motivations provides context for understanding the design choices that shaped the network we use today.

## Design Goals: Anonymity, Resistance, Usability

Tor's architecture is the product of deliberate design decisions, each motivated by specific goals and constrained by specific requirements. Understanding these goals is essential for understanding why Tor works the way it does and what trade-offs it has made.

The primary design goal is anonymity: ensuring that network observers cannot determine who is communicating with whom. This is achieved through multi-hop routing, layered encryption, and careful protocol design that prevents information from leaking at any single point in the circuit. The three-hop circuit architecture is a core mechanism: no single relay can see both the client and the destination, and the layered encryption ensures that each relay can only decrypt the layer intended for it.

The second goal is censorship resistance: ensuring that users in countries where Tor is blocked can still access the network. This goal has driven the development of bridges, which are unlisted relays that are not included in the public directory and therefore harder for censors to discover and block. It has also driven pluggable transports like obfs4, which obfuscate Tor traffic to make it look like something else and evade deep packet inspection.

The third goal is usability: ensuring that Tor is accessible to non-technical users. This goal has driven the development of Tor Browser, which bundles Tor with a configured Firefox browser and provides a simple interface for anonymous web browsing. It has also influenced the design of onion services, which can be accessed with the same browser without requiring special configuration on the client side.

The fourth goal is decentralization: ensuring that Tor does not depend on any single organization or infrastructure provider. This goal has led to a

volunteer-operated relay network, a distributed directory system, and open-source software that anyone can audit and contribute to. No single entity controls Tor.

These goals sometimes conflict. Usability can conflict with anonymity, because making something easier to use often means simplifying security controls that non-technical users might otherwise disable. Censorship resistance can conflict with performance, because pluggable transports add overhead and bridges may be slower than regular relays. Decentralization can conflict with coordination, because making decisions across a distributed community is slower and messier than centralized development.

Tor's designers have had to make trade-offs between these goals, and some of those trade-offs are explicit and stated while others are implicit consequences of practical constraints. The result is a system that is not perfect in any single dimension but is well-balanced across multiple objectives. Understanding these trade-offs is important for anyone who wants to deploy onion services effectively, because the security and behavior of your service depends on the underlying network's design choices.

## The Political and Social Context

Technology does not develop in a vacuum. Tor was created and evolved in response to a specific political and social context: the growing surveillance capabilities of governments and corporations, the increasing centralization of the internet, and the escalating use of internet censorship to suppress dissent.

In the late 1990s and early 2000s, the internet was often characterized as a frontier, a space where traditional power structures had not yet fully extended their reach. That perception was partly accurate and partly illusory. Governments had always been able to monitor communications at the infrastructure level, but the encryption tools available to ordinary citizens were limited. The situation changed in the 2000s with the widespread adoption of HTTPS, the availability of strong encryption libraries, and the development of tools like Tor that made anonymity accessible to non-technical users.

The rise of internet censorship has been particularly significant. Countries like China, Iran, Russia, and Turkey have developed increasingly sophisticated tools for monitoring, blocking, and filtering internet traffic. National firewalls, deep packet inspection systems, and legal frameworks requiring ISPs to log

and provide user data have made it difficult for citizens in these countries to access uncensored information or communicate freely. Tor has become one of the primary tools for circumventing these controls, and the development of bridges and pluggable transports has been driven largely by the need to help users in censored environments.

The Snowden revelations of 2013 dramatically increased public awareness of government surveillance programs and the capabilities of intelligence agencies to monitor internet traffic at scale. The disclosure that the NSA targeted Tor users and operated malware against Tor Browser specifically led to a surge in interest and downloads. It also prompted the Tor Project and its contributors to invest heavily in improving Tor's security and closing vulnerabilities that had been exploited by adversaries.

The Arab Spring of 2010-2012 demonstrated the role of communication tools in political organizing and resistance. Activists used the internet to coordinate protests, document government actions, and communicate with international observers. In some cases, censorship and surveillance were used to suppress dissent, highlighting the importance of tools that could resist control.

Today, the political landscape surrounding internet privacy is more complex than ever. Governments are developing new legal frameworks for data retention and surveillance. Corporations are building ever more sophisticated tracking and profiling systems. At the same time, privacy tools are becoming more accessible and more powerful. Tor sits at the intersection of all of these trends, and its continued development is shaped by the ongoing tension between surveillance and privacy, control and freedom, visibility and anonymity.

Understanding this context is not about taking a political position. It is about understanding the motivations and requirements that shaped the technology you are using. The people who designed Tor did so because they believed that internet users should have private access to an uncensored web. They built a system that trades some convenience for privacy, some performance for anonymity, and some simplicity for security. Whether you agree with their politics or not, their design choices have produced a tool that is now indispensable for many people around the world.

Now that we have traced the history of onion routing from its military origins to the modern Tor ecosystem, we can turn to the technical architecture of the network itself. Before we can deploy an onion service, we need to

understand the pieces that make up the Tor network: the different types of nodes, how they interact, and what responsibilities each one has. This is the foundation for everything that follows.

# Chapter 3: The Tor Network

## Architecture

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

### What is a Tor Node: Client, Relay, Bridge, Authority

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

### Directory Authorities: The Seven Guardians

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

### Directory Caches and Consensus Distribution

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

### Guard Relays and Stable Path Building

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

### Middle Relays: The Workhorse Tier

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Exit Relays: Touchpoints with the Clearnet

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Onion Services: Servers on the Tor Network

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

# Chapter 4: The Tor Protocol Stack

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## The Transport Layer: TCP and TLS Handshakes

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Onion Skin Encryption: Layered Keys

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Link Keys and Circuit Keys

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Key Exchange Protocols: NTor and Older Schemes

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Cell Types and Protocol Messages

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Version Negotiation and Authentication

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

# Chapter 5: Circuit Construction and Routing

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Guard Node Selection and Persistence

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Relay Weighting: Bandwidth, Flags, and History

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Building a Three-Hop Circuit Step by Step

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Exit Node Selection Policies

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Circuit Lifetimes, Expiry, and Refresh

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Path Isolation and Connection Mapping

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

# Chapter 6: Onion Services: Concept and Evolution

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## From Hidden Services to Onion Services: Naming Changes

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## v2 vs v3: Architecture and Security Improvements

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## The Rendezvous Protocol: Meeting in the Middle

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Introduction Points and Rendezvous Points

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Service Identifiers and Public Keys

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Onion Addresses: Encoding, Structure, and Validation

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

# Chapter 7: The v3 Onion Service Protocol

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Private Key Generation and Descriptor Signing

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Publishing Onion Service Descriptors

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Introducing Points: Registration and Maintenance

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Client Discovery: Hash-Based Directory Lookup

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Rendezvous Point Selection and Negotiation

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## End-to-End Circuit Establishment

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Descriptor Refresh and Fault Tolerance

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

# Chapter 8: Threat Models and Attack Surfaces

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Tor's Formal Threat Model: What It Promises

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Passive Network Observers and Local ISPs

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Global Adversaries and Correlation Attacks

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Malicious Relays: Observation and Manipulation

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Endpoint Compromise and Application Leaks

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## The Limits of Network-Layer Anonymity

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

# Chapter 9: Traffic Analysis and Correlation Attacks

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Passive Correlation: Timing and Volume

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Active Attacks: Padding and Probe Techniques

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Statistical Fingerprinting and Machine Learning

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Network-Wide Correlation at Scale

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Defenses: Padding, Guard Diversity, Path Length

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Theoretical Guarantees vs Practical Risk

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

# Chapter 10: Fingerprinting and Side Channels

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Browser Fingerprinting and Tor Browser Countermeasures

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## TLS Fingerprinting and JA3 Identifiers

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## JavaScript Tracking and DOM Fingerprinting

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Application-Level Metadata Leaks

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Timing Side Channels in Web Applications

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Weeping Cryptography and Hidden Data

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

# Chapter 11: Sybil Attacks, Censorship, and Bridges

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Sybil Relays and Identity Fabrication

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Directory Authority Consensus and Weighting

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## ISP-Level Blocking and Keyword Filtering

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Bridges: Unlisted Entry Points

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Pluggable Transports: obfs4 and WebTunnel

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

# Chapter 12: Infrastructure Planning and Design

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Jurisdiction and Legal Environment

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Hosting Options: Colocation, VPS, Home Hosting

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Hardware Requirements and Sizing

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Network Topology: Isolation and Segmentation

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Power, Connectivity, and Redundancy

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

# Chapter 13: Operating System Hardening and Security

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## OS Selection: Debian, Alpine, or Specialized Distributions

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Minimal Installation and Surface Reduction

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Kernel Parameters and Sysctl Hardening

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## User, Group, and Permission Design

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Full Disk Encryption and Key Management

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## **Intrusion Detection and Log Integrity**

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

# Chapter 14: Network Configuration and Firewalls

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Firewall Fundamentals: iptables and nftables

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Default-Deny Policies and Minimal Open Ports

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Blocking All Exit Relay Traffic

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## DNS Resolution Without Information Leakage

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Network Interface Isolation

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Detecting and Blocking Direct Access Attempts

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

# Chapter 15: Tor Installation and Core Configuration

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Installing Tor: Package Managers and Compiling

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## The torrc File: Structure and Directives

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Configuring HiddenServiceDir and HiddenServicePort

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Running Tor as a Systemd Service

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Validating Circuit and Service Status

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

# Chapter 16: Deploying Your First Onion Service

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Minimal Web Server: Nginx Bound to Loopback

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## First Hidden Service Configuration

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Testing with Tor Browser from a Remote Machine

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Verifying No Direct IP Accessibility

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Checking Descriptor Publication

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Troubleshooting First-Time Deployments

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

# Chapter 17: Architectural Pattern: Single-Host Minimal Service

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Single Machine Topology Diagram

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Nginx Configuration for Loopback Binding

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Tor Hidden Service Configuration

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Systemd Service Units

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## SSL/TLS Considerations for Onion Services

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Backup and Recovery Procedure

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

# Chapter 18: Architectural Pattern: Hardened Network Segmentation

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Three-Tier Network Topology

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## DMZ Host: Tor and Reverse Proxy

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Application Tier Isolation

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Database Tier and Access Controls

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Internal Firewall Rules Between Zones

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## End-to-End Request Flow Diagram

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

# Chapter 19: Architectural Pattern: Containerized Multi-Service

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Docker Networking and Host vs Container Mode

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Tor in a Container vs on the Host

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Docker Compose for Multi-Service Stacks

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Reverse Proxy Routing to Multiple Onion Services

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Persistent Volumes and Secrets Management

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Build, Deploy, and Update Workflow

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

# Chapter 20: Private and Authorized Onion Services

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Private Onion Services Concept

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Client Authorization and ClientAuthFile

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Generating and Distributing Authorization Tokens

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Combining Tor with Web-Based Authentication

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Multi-Factor Authentication over Tor

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Managing Revocation and Key Rotation

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

# Chapter 21: Application-Specific Deployments

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## File Hosting: Nextcloud and FileBrowser

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Messaging: Matrix, XMPP, and IRC over Tor

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## REST and GraphQL APIs behind Onion Services

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Remote Administration: SSH and Web Consoles

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Personal Cloud and Sync Services

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Monitoring and Observability Stacks

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

# Chapter 22: Performance, Scaling, and Optimization

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Tor Latency Characteristics and Benchmarks

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Bandwidth Limits and Congestion Behavior

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Caching and CDN Considerations

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Load Balancing Multiple Onion Services

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Database Performance Behind Tor

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## When to Leave Tor Behind for Internal Traffic

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

# Chapter 23: Operations, Monitoring, and Maintenance

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Update Cadence and Security Patches

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Onion Service Key Rotation

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Log Aggregation and Alerting

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Backup Strategies: Encrypted and Offsite

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Incident Response and Breach Containment

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Operational Security Procedures

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

# Chapter 24: Troubleshooting, Diagnostics, and Failure Modes

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Verifying Onion Service Reachability

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Descriptor Publication Failures

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Circuit Construction Problems

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Backend Connectivity Issues

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## HiddenServiceDir Permission Errors

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Network-Level Issues

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Application-Level Issues

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Verifying No IP Address Leakage

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Debugging Client-Side Access Issues

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

# Chapter 25: Future Directions and Emerging Technologies

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Post-Quantum Cryptography and Proposal 353

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Quantum-Safe Onion Service Addresses

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Network Growth and Resilience

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Decentralized Alternatives and Complementary Technologies

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Browser Fingerprinting Evolutions

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Machine Learning and Traffic Analysis

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Regulatory and Legal Landscape

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

## Practical Skills for the Next Decade

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

# Conclusion

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.

# References

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/self-hostingonthedarkweb>.