

INIT.WTF

72-PAGE PRODUCTION EDITION

ISSUE 01 • FALL 2026 • \$8.00

PGP // 0xB10C

PRIVACY IS
A UNIVERSAL RIGHT

THE SYSTEM
IS LISTENING

2600 Hz: THE TONE
THAT TAUGHT
HACKERS TO LISTEN

READ THE SOURCE.
VERIFY THE HASH.

DISCORDIAN
CYPHERPUNK
MANIFESTO

FIELD NOTES:
LINUX • RADIO •
RAIL • AI

AUTHORIZED
LABS ONLY

BOOT CLEAN.
VERIFY THE SIGNAL.

init://

4D 49 4E 44 20 49 53 20 54
48 45 20 57 45 41 50 4F 4E
20 54 48 45 20 53 59 53 54
45 4D 20 46 45 41 52 53 2E.

2600Hz

2600 Hz

-3.0 dBm0

LOCAL EXCHANGE ROUTING

TRUNK 01
TRUNK 02
TRUNK 03
TRUNK 04
TRUNK 05
TRUNK 06
TRUNK 07
TRUNK 08

BURN THIS MAGAZINE // NATIVE EDITORIAL

13-MONTH TIME // BITCOIN BEFORE MONEY // THE TRIAL OF AI

PBX + PACKETMAIL + COW/RABBIT COMICS INSIDE

LINKS // 0x2A

PACKETS // 0xFF

NOTES //

Silence is a protocol.
Noise is cover.

01

ISSUE

ISSUE 01 / 72 PAGES / PRODUCTION EDITION

BOOT INDEX

03-04 // EDITOR LETTER + OPERATOR KEY

How the archive was recompiled and how to use the issue at the bench.

05-11 // CHAOS + CYPHERPUNK SIGNAL

Belief as firmware, sigils as protocol, manifesto, and a visual reset.

12-19 // BURN THIS MAGAZINE / FIXED TIME

Recovered editorial, the 13-month calendar, conversion data, and collage plates.

20-26 // BITCOIN BEFORE MONEY

Mailing lists, cypherpunk lineage, proof-of-work, source trail, and custody card.

27 // COMIC: CONTENT IS DATA

Cow receives one suspiciously enthusiastic envelope.

28-36 // THE QUESTION BEFORE THE POLIS

Artificial competence, civic purpose, concentrated power, and reader questions.

37 // COMIC: WRONG PAYLOAD

Rabbit witnesses a nuclear arms deal. Cow opens an incident.

38-42 // ARCHIVE + THE DIG

Haunted handles, 2600 Hz, recovered systems, and an artifact intake sheet.

43-51 // PBX FROM ZERO

Asterisk build, cutover card, call from 1984, and business-line airlock.

52-61 // PACKETMAIL FIELD MANUAL

Pi, Mailcow, No-IP, WireGuard, RabbitMQ, dead letters, test, and restore.

62-64 // AXIOM + TRANSMISSION FORM

The BTM closing signal, corrected field notes, and the route to Issue 02.

65-72 // WORKSHEETS / SOURCES / COLOPHON

Privacy stack, library, gallery, nodes, production trail, and back cover.

EDITOR / OPERATOR

WE DID NOT PASTE IT IN. WE TOOK IT APART.

BURN THIS MAGAZINE arrived as twelve A3 spreads: finished, strange, visually loud, and full of real ideas fighting the furniture.

The first expanded proof preserved it as a facsimile. That proved nothing was lost, but it also left two magazines sitting beside each other. This edition does the harder thing: it extracts the stories, verifies the data, proofreads the claims, and rebuilds every editorial strand in the visual and technical language of INIT.WTF.

Archive first. Then compile.

The fixed-calendar piece becomes an actual investigation, with the 13-by-28 arithmetic, the placement of Sol, the out-of-week Year Day, the leap-day rule, and the cultural objection that stopped the machine from becoming ordinary time.

The Bitcoin essay keeps its central argument: Bitcoin entered the world less like a product and more like a textfile dropped into a hostile network. The lineage now has a source trail: May, Hughes, remailers, Hashcash, the 2008 mailing-list announcement, and Hal Finney.

The Question Before the Polis remains the long read. Its claim is neither worship nor panic: artificial intelligence is an instrument that amplifies the regime, purpose, discipline, and power structure that deploys it.

Between those essays, Cow and Rabbit interrupt the proceedings with dead letters, suspicious payloads, and one telephone call from 1984. After the stories, the magazine becomes a bench manual: PBX first, PacketMail second, operator logs and sources at the back.

Dog-ear the pages. Correct us in the margins. Build only on systems you own. Treat screenshots as evidence, source notes as part of the story, and every easy answer as a request for one more test.

- MICHAEL A. COHEE
EDITOR / OPERATOR // INIT.WTF
READ THE SOURCE. VERIFY THE HASH. TRANSMIT ANYWAY.

USE THE ISSUE LIKE A BENCH

This is a magazine, a field manual, an archive object, and a place to write back.

READ

Long essays carry the argument. Source trails carry the burden of proof. Margins belong to disagreement.

BUILD

Manuals stay inside systems you own. Every public boundary gets a rollback plan before it gets a port.

MARK

Checklists are not decoration. Record dates, versions, addresses, failures, and the thing that finally worked.

TRANSMIT

Send corrections, artifacts, diagrams, field notes, and arguments. A zine becomes a protocol when readers answer it.

DOG-EAR IT. WRITE IN IT. MAKE THE NEXT COPY SMARTER.

OWNER / LOCATION / ACQUIRED / RETURNED

ESSAY / PART I

CHAOS MAGICK IS INTERFACE DESIGN

BELIEF AS A TOOL

Chaos magick begins with an unnerving proposition: belief can be used as an instrument. A model does not have to become a permanent identity before it becomes useful. You may enter a symbolic system, perform the work, observe what it changes, and leave without surrendering judgment.

That stance feels familiar to anyone who has changed operating systems, shell environments, programming paradigms, or incident hypotheses. We load a model because it helps us act. Then we inspect the output. The model is neither sacred nor meaningless. It is an interface.

RITUAL AS STATE TRANSITION

A ritual creates a boundary between ordinary attention and deliberate attention. In systems language it is a state transition: establish context, reduce noise, execute a sequence, record the result, and return to baseline.

The value is not supernatural certainty. The value is concentration made visible. A candle, a timer, a drawn mark, a repeated phrase, or a piece of music can tell the nervous system that this period has one purpose. The physical act keeps intention from dissolving into another browser tab.

A symbol is a compressed instruction for attention.

ESSAY / PART II

PROTOCOLS FOR ATTENTION

SIGILS AS COMPRESSED INTENT

A sigil compresses a written intention into a form that no longer reads like ordinary language. Redundancy is removed. Letters become geometry. The result is easier to remember as an image and harder to over-explain. It behaves less like a paragraph and more like an icon, checksum, or glyph.

The interesting part is not claiming that the mark controls reality. The interesting part is what the process forces you to do: choose a precise intent, remove contradiction, make it concrete, and stop rehearsing it as anxiety.

BANISHING AS CLEANUP

After focused work, chaos-magick traditions often include a banishing or laughter - a deliberate release of seriousness. In operator language: close the session, clear temporary state, archive the log, and return privileges to normal.

This matters because obsession can masquerade as commitment. The closing action says the operation is over. The result can now be observed rather than forced.

ETHICAL BOUNDARY

Use symbolic work for self-reflection, creativity, courage, habits, and meaning. Do not use it as a justification for coercion, retaliation, stalking, or claims of control over another person. Consent remains the root permission.

DEFINE -> COMPRESS -> MARK -> FOCUS -> RELEASE -> LOG

DOT-MATRIX WORKSHEET

SIGIL AS PROTOCOL

01 DEFINE

Write one clear intention about your own action.

02 COMPRESS

Remove repetition, contradiction, and unnecessary words.

03 DISTORT

Rotate, mirror, join, and simplify the remaining forms.

04 FOCUS

Give the mark one quiet minute of undivided attention.

05 RELEASE

Put it away. Laugh. Walk. End the operation.

06 LOG

Record date, state, action, and what changed later.

NO DOMINATION // NO TARGETING OTHERS // CONSENT IS ROOT

FROM THE ARCHIVE

MANIFESTO // 01

A DISCORDIAN CYPHERPUNK MANIFESTO

Chaos is beautiful because it cannot be owned. Chaos is just because it refuses the throne.

Before the feed, before the bank, before the priest,
before the platform, before the algorithm whispered its
little gospel of convenience into the skull of modern man,
there was the first signal.

init.
the beginning.
the boot.
the sacred interruption.

INIT.WTF is the question screamed into the machine
before the machine has time to name us. What is
running here? Who wrote this script? Who profits from
our obedience? Who convinced the living mind that
privacy was suspicious, freedom was a service plan,
speech was a permission structure, and control was
safety?

COMPLETE CONTROL = NO CONTROL

The more they centralize, the weaker they become. The
more they surveil, the blinder they become. The more
they regulate thought, the more thought escapes them.

FROM THE ARCHIVE

MANIFESTO // 02

THE SIGNAL HAS RETURNED

INIT.WTF is not a brand. It is not a lifestyle. It is not another hollow banner raised above the same dead marketplace. INIT.WTF is a bootloader for the sovereign mind.

Privacy is dignity. Encryption is speech. Thought is sacred. Absurdity is a weapon against false order.

READ THE SOURCE.
VERIFY THE HASH.
FORK THE NARRATIVE.
QUESTION AUTHORITY.
DROP TO SHELL.
TRANSMIT ANYWAY.

Ours is not a war of fists. It is a war of perception, architecture, memory, language, code, humor, refusal, and relentless independent thought.

Discord is the holy refusal of false harmony. Discord is the crack in the corporate hymn. Discord is the laugh in the cathedral of control.

SCIENTIA UMBRARUM NUMQUAM SINE VISIONE EST

Whoever knows the shadows is never without sight.

FROM THE ARCHIVE

MANIFESTO // 03

AND SO WE INITIALIZE

Freedom that requires surveillance is not freedom.
Freedom that depends on permission is not freedom.
Freedom that vanishes when the payment processor sneezes is not freedom. Freedom without privacy is theater.

We do not ask the system to save us. We inspect it. We interrupt it. We fork it. We route around it. We outlive it.

**This is chaos as reclamation.
Chaos as purification. Chaos
as art. Chaos as audit.**

Think for yourself. Question authority. Accept no update without inspection.

INIT.WTF BEGINS WHERE OBEDIENCE FAILS.

```
boot clean.  
verify the signal.  
return no more.
```

Libertas et Libertas. All Hail Eris. All Hail Discordia.

SIGNAL BREAK

// THEREFORE

EXAMINE THE BELIEF. KEEP THE TOOL. DROP THE THRONE.



// BECAUSE

BURN THIS MAGAZINE

THE ARCHIVE RECOMPILED

STYLE / SYSTEM / SIGNAL
THE ARCHIVE HAS BEEN RECOMPILED

PRINT THE LIE

TRUTH DOES NOT REQUIRE PERMISSION

the signal is not what they printed

RECOVERED EDITORIAL / MAY 2026

LETTER FROM THE BURN

Let this magazine set the stage for future issues at the crossing of 2600 and our own BURNT style of love.

Story one is the practical: the International Fixed Calendar. Story two is the artistic made manifest. Story three comes from THE BAKERY - a modern field note in the old tradition of dangerous manuals. The closing section enters the Arte of esoteric life and abnormal proportions.

That original map survives here, but the sections now run as one issue: practical data, historical archive, philosophical story, comic interruption, and builds you can actually reproduce.

On behalf of the whole BURN team: happy reading. - Magus Mike



THE ORIGINAL SPREAD IS PRESERVED. THE STORY NOW RUNS NATIVE.

INTERNATIONAL FIXED CALENDAR / 01

THE 13-MONTH MACHINE

Thirteen equal months. Twenty-eight days each. Every date lands on the same weekday forever.

Moses B. Cotsworth proposed the International Fixed Calendar in the early twentieth century as a practical repair for irregular accounting months. Twelve familiar months remain; a new month called Sol sits between June and July.

The appeal is mechanical. Every month contains exactly four weeks. The first is always the same weekday. Quarters and payroll periods stop changing shape. A printed calendar becomes reusable.

13 x 28 = 364.
The missing day is where politics enters.

A common year needs Year Day after December 28. A leap year needs another intercalary day around midyear. Those days sit outside the ordinary week, preserving the calendar's fixed pattern while interrupting an unbroken seven-day religious cycle.



HOW THE IFC ACTUALLY WORKS

THE FIXED MONTH

SUN MON TUE WED THU FRI SAT
01 02 03 04 05 06 07
08 09 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28

THE YEAR

13 MONTHS x 28 DAYS = 364
+ YEAR DAY = 365
+ LEAP DAY WHEN DUE = 366

Sol follows June. Year Day follows December 28. The leap rule follows the Gregorian rule; the extra leap day is placed at midyear and does not receive a weekday name.

WHAT BECOMES EASY

Monthly comparison, payroll periods, recurring schedules, printed calendars, and date-to-weekday lookup.

WHAT BECOMES HARD

Compatibility with civil standards, existing software, holidays, birthdays, contracts, and traditions that require an uninterrupted seven-day week.

HOW TO USE THE IFC

CONVERSION PROTOCOL Σ GREGORIAN → FIXED

13-MONTH IFC

01	JAN
02	FEB
03	MAR
04	APR
05	MAY
06	JUN
07	SOL
08	JUL
09	AUG
10	SEP
11	OCT
12	NOV
13	DEC

1 FIND DAY OF YEAR (N)

Count days from Jan 1

April 17 → N = 107

2 ADJUST FOR LEAP YEAR

Not a leap year → A = N

Leap year & N > 169 → A = N - 1

3 LOCATE IN 28-DAY BLOCKS

01-28	→ JAN
29-56	→ FEB
57-84	→ MAR
85-112	→ APR
113-140	→ MAY
141-168	→ JUN
169-196	→ SOL

4 READ MONTH & DAY

APRIL 23 (IFC)

SPECIAL DAYS

N = 365
→ YEAR DAY

N = 169
→ LEAP DAY

364 + 1 (+1)

A = 107

Find the block:
85-112 → APRIL

Subtract previous block:
107 - 84 = 23

FORMULAS

A = N	MONTH = [(A - 1) / 28] + 1	N = Day of Year
A = N - 1 (after June 17, leap year)	DAY = ((A - 1) mod 28) + 1	A = Adjusted Index

Gregorian counts forward. IFC resolves position.

IFC v1.0

PRACTICAL BURN / 03

THE CULTURE WAS BUILT FOR TWELVE

The original BTM draft treated resistance to thirteen as an esoteric conspiracy. The more interesting answer is structural inertia: societies stack meaning on top of a calendar until changing the calendar means changing nearly everything else.

TWELVE IS EVERYWHERE

Twelve Gregorian months. Twelve zodiac signs. Twelve hours around the clock face. Twelve-note chromatic systems. Twelve apostles. Familiar systems reinforce one another even when their origins differ.

EDITORIAL CORRECTION

The Kabbalistic Tree of Life is not normally described as twelve primary spheres. Classical diagrams center ten sefirot, sometimes discuss Da'at, and connect them through paths. Tarot correspondences are also more complicated than a one-to-one twelve-sign map.

THIRTEEN IS NOT THE BUG

In some Italian traditions thirteen is fortunate while seventeen carries the heavier superstition. Missing thirteenth floors are primarily a commercial accommodation to local fear, not evidence that the IFC was suppressed.

The calendar failed because compatibility is a form of power.



PRACTICAL BURN / 04

TRY IT WITHOUT BELIEVING IT

A calendar is an interface. Test the interface before declaring it liberation or heresy.

PAPER TEST

Pick one ordinary month. Rewrite every appointment into a 28-day grid. Mark payroll, medication, school, bills, birthdays, and maintenance. Then add Year Day and a hypothetical leap day.

SYSTEM TEST

INPUT: Gregorian date + timezone
MAP: day-of-year -> IFC month/day
KEEP: original date as canonical record
OUTPUT: display-only IFC label
NEVER: overwrite legal or financial timestamps

OBSERVE

Uniform weeks reduce mental arithmetic. Conversion creates a boundary problem. The more systems you touch, the more expensive that boundary becomes.

VERDICT

The IFC is excellent as a thought experiment, planning overlay, or private interface. Replacing civil time requires collective adoption, legal agreement, software support, and a solution for the days outside the week.

Fixed does not mean frictionless.

HOW TO USE THE IFC

364 + 1 (+1)

CONVERSION PROTOCOL Σ GREGORIAN → FIXED

13-MONTH IFC

01 JAN

02 FEB

03 MAR

04 APR

05 MAY

06 JUN

07 SOL

08 JUL

09 AUG

10 SEP

11 OCT

12 NOV

13 DEC

1 FIND DAY OF YEAR (N)

Count days from Jan 1

April 17 → N = 107

2 ADJUST FOR LEAP YEAR

Not a leap year → A = N

Leap year & N > 169 → A = N - 1

3 LOCATE IN 28-DAY BLOCKS

01-28 → JAN

29-56 → FEB

57-84 → MAR

85-112 → APR

113-140 → MAY

141-168 → JUN

169-196 → SOL

4 READ MONTH & DAY

A = 107

Find the block:

85-112 → APRIL

Subtract previous block:

107 - 84 = 23

APRIL 23 (IFC)

SPECIAL DAYS

N = 365

→ YEAR DAY

N = 169

→ LEAP DAY

FORMULAS

A = N

A = N - 1 (after June 17, leap year)

MONTH = ((A - 1) / 28) + 1

DAY = ((A - 1) mod 28) + 1

N = Day of Year

A = Adjusted Index

Gregorian counts forward. IFC resolves position.

IFC v1.0

RECOVERED COLLAGE / PLATE 01

COLLAGE AS EVIDENCE



RECOVERED PLATE: CIPHER, CITY, PAPER, KEY, AND MONETARY SYMBOL

RECOVERED COLLAGE / PLATE 02

THE BAKERY WALL



RECOVERED PLATE: NEWSPAPER, CLOCK, LIGHTHOUSE, CARD, AND ANNOTATED MEMORY

REVOLUTION OF THE CYPHERPUNKS

BITCOIN BEFORE MONEY

A ZINE, A MAILING LIST, A PROTOCOL
AND THE POLITICS HIDING IN THE ARCHITECTURE

DECENTRALIZE
ENCRYPT
VERIFY

TRUST
NO ONE
VERIFY
EVERYTHING



6a 09 e6 67 f3 bc c9 6c
bb 67 ae 85 84 ca a7 3b
3c 6e f3 72 fe 94 f8 2b
a5 4f f5 3a 5f 1d 36 f1

SHA256(FREEDOM)
= PROOF OF HUMAN WILL



NOT YOUR KEYS
NOT YOUR COINS

BURN THIS MAGAZINE

21
00
00
00

DECENTRALIZE
ENCRYPT
VERIFY

TRUST
NO ONE
VERIFY
EVERYTHING



ARTE / 01

A MESSAGE IN A HOSTILE NETWORK

Before Bitcoin became an asset class, exchange ticker, political symbol, or institutional product, it was a message dropped into a hostile network.

On October 31, 2008, Satoshi Nakamoto posted to the Cryptography Mailing List with the subject 'Bitcoin P2P e-cash paper.' The opening was brutally minimal: a new electronic cash system, fully peer-to-peer, with no trusted third party.

That sentence is the detonation. Bitcoin did not arrive like a startup. It arrived like a photocopied communique: no launch party, founder video, or brand deck. A PDF, a protocol, and a proposition - what if money could move without asking a central bookkeeper?

The paper's technical core attacked digital double-spending with a peer-to-peer timestamp network and hash-based proof-of-work. Transaction history becomes expensive to rewrite, while nodes may leave and rejoin by accepting the chain representing the most accumulated work.



ARTE / 02

THE CYPHERPUNK INHERITANCE

THE LINEAGE IS OLDER

Bitcoin inherited distrust of central authorities, an obsession with privacy, and the cypherpunk commandment to publish the method rather than merely the opinion.

Timothy C. May's Crypto Anarchist Manifesto dates to 1988 and was posted to the Cypherpunks list in 1992. It predicted anonymous communication, electronic contracts, reputation systems, and markets difficult for states to supervise. May also understood the danger: the same systems that protect dissidents can protect predation.

Eric Hughes answered in 1993 with A Cypherpunk's Manifesto. Privacy, he argued, is the power to reveal oneself selectively. Anonymous systems, encryption, digital signatures, remailers, and electronic money were not requests for institutional kindness. They were things to build.

Cypherpunks write code.



ARTE / 03

REVOLUTIONARY MANIFESTO / I

They told us trust was safety. They built temples out of banks. They dressed permission as protection. They called surveillance convenience. They called dependency order.

But the wire remembered another future: a future written in keys; signed, not begged for; verified by math; a future no clerk could deny.

The cypherpunks did not wait for reform. They did not petition the gatekeepers. They built exits in plain sight. They wrote code like contraband scripture. They turned privacy into architecture.

EVERY KEYPAIR A REFUSAL
EVERY SIGNATURE A BORDER
EVERY HASH A WITNESS
EVERY BLOCK A SCAR IN TIME
EVERY NODE A QUIET ACT OF DISOBEDIENCE

Bitcoin was not born from greed. It was born from suspicion, from frozen accounts, from the memory that access can be revoked by whoever owns the ledger.



ARTE / 04

REVOLUTIONARY MANIFESTO / II

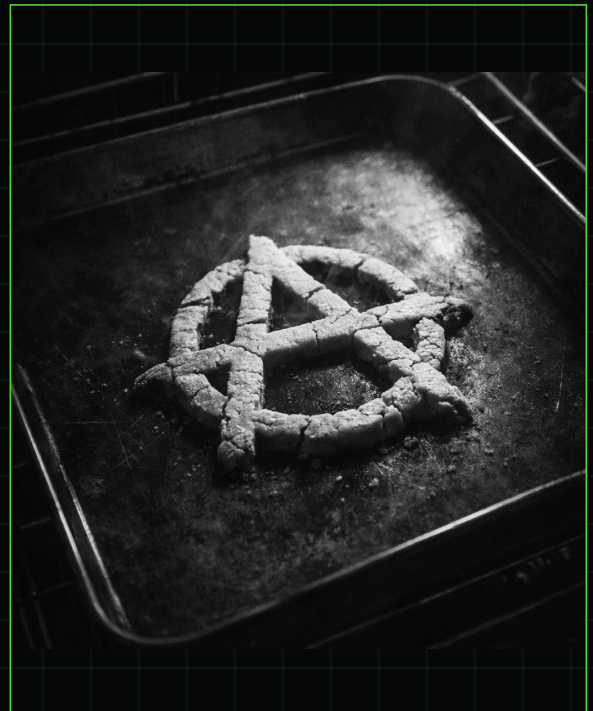
It answered a simple insult: that free people must ask permission to move value; that money must kneel before institutions; that trust must be rented from those who profit by breaking it; that the future belongs to whoever owns the ledger.

So the ledger was made public. The mint was made mechanical. The rules were made visible. The gate was removed. The network was set loose.

No king. No banker. No rescue priest. No corporate confession booth. No master key under glass.

ONLY PROOF.
ONLY CUSTODY.
ONLY SIGNAL.

That does not make the technology morally clean. Censorship resistance can preserve forbidden truth or durable poison. Pseudonymity is not invisibility. Custody creates responsibility. Power transferred into protocols still arrives with consequences.



ARTE / 05

THE UNDERGROUND WAS ALREADY THERE



Hal Finney's own account ties the visible artifact to the underground: early PGP work, the first cryptographically based anonymous remailer, Cypherpunks participation, and then the skeptical 2008 Bitcoin announcement.

Satoshi replied to a political objection in November 2008 that cryptography could not solve political problems entirely, but might win a major battle and create a new territory of freedom for several years.

SOURCE TRAIL

SATOSHI NAKAMOTO - BITCOIN P2P E-CASH PAPER - 2008-10-31
BITCOIN: A PEER-TO-PEER ELECTRONIC CASH SYSTEM
TIMOTHY C. MAY - THE CRYPTO ANARCHIST MANIFESTO - 1988 / 1992
ERIC HUGHES - A CYPHERPUNK'S MANIFESTO - 1993-03-09
HAL FINNEY - BITCOIN AND ME - 2013-03-19

Price is graffiti on the wall after the architectural event.



CUSTODY WORKSHEET

BEFORE YOU HOLD A KEY

Self-custody is not a slogan. It is a recovery procedure tested before the emergency.

THREAT MODEL

Loss, theft, coercion, device failure, forgotten instructions, accidental disclosure, and inheritance are different problems. Name the one you are solving.

VERIFY WITHOUT EXPOSING

Confirm receiving addresses on the trusted device. Never type seed words into a website, support chat, cloud note, or unknown recovery tool.

TEST THE EXIT

A backup is a theory until a controlled restore succeeds. Use a spare device or isolated procedure and document what the future operator must know.

WHAT FAILURE ARE WE PROTECTING AGAINST?

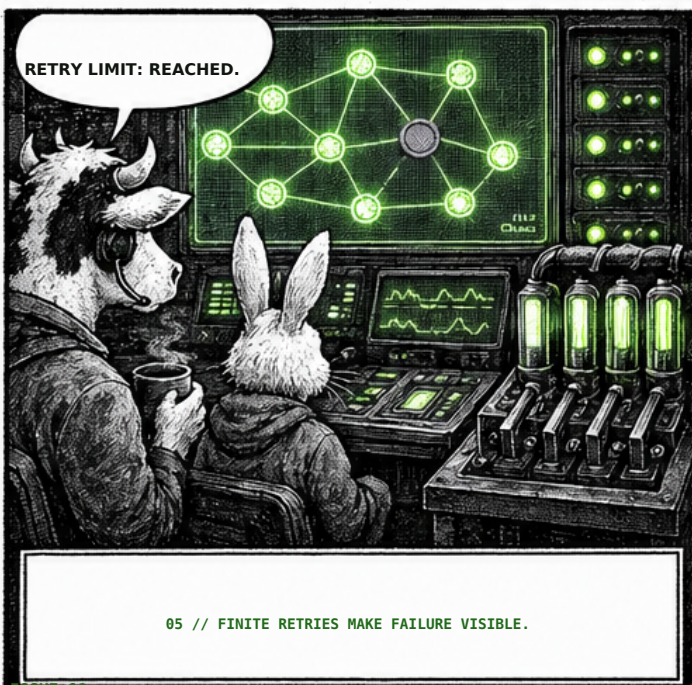
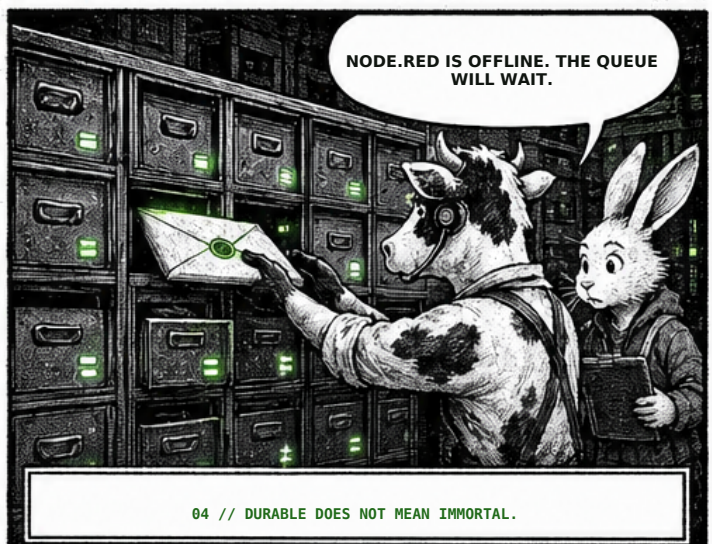
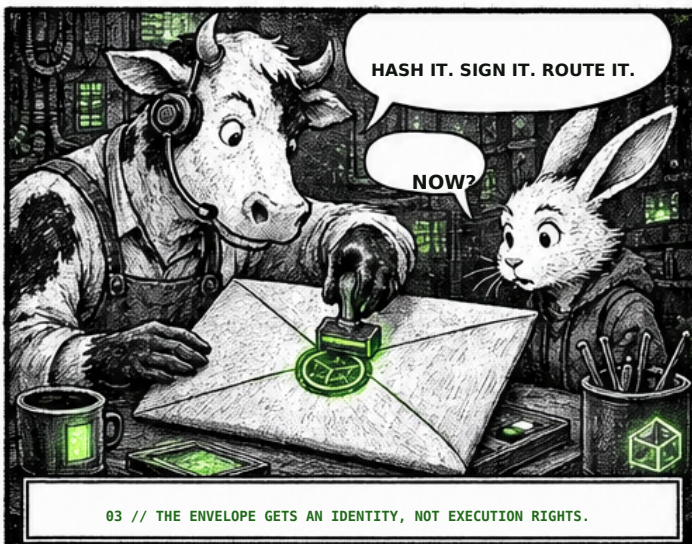
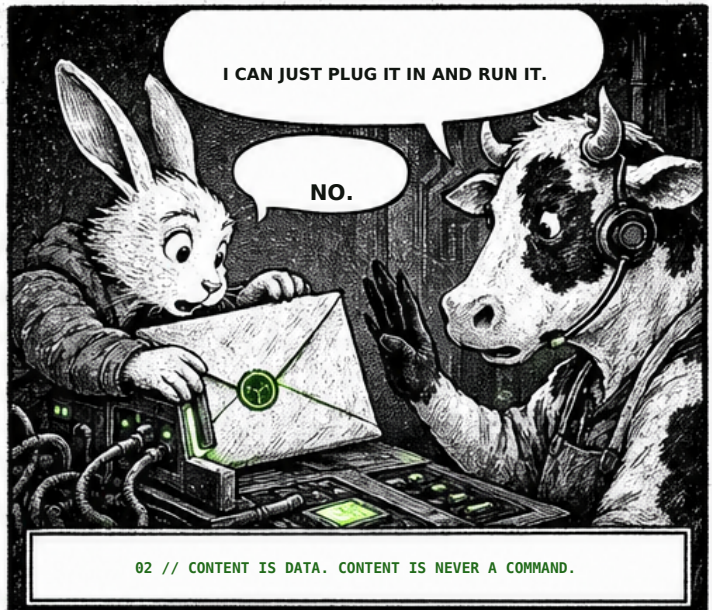
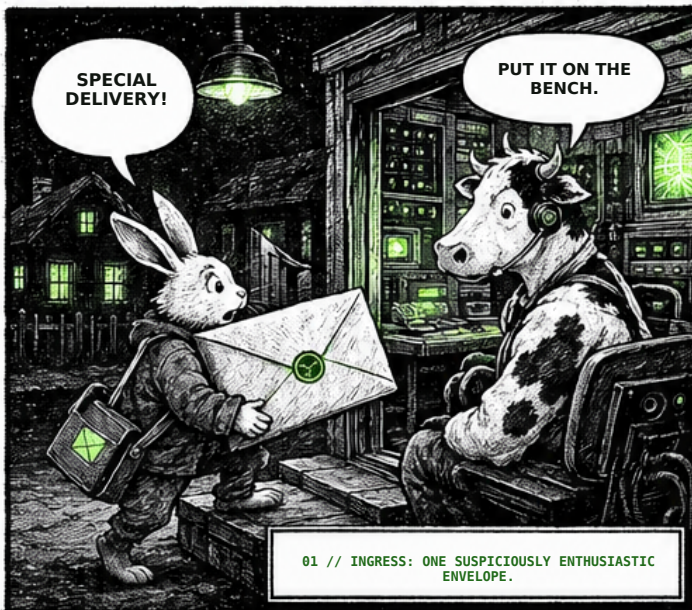
WHERE IS THE VERIFIED RECOVERY PATH?

TEST DATE / DEVICE / RESULT

WHO CAN RECOVER IF I CANNOT?

THE COW, THE RABBIT & THE DEAD LETTER OFFICE

PACKETMAIL INCIDENT 001 // CONTENT IS DATA



THE BAKERY / 01 OF 08

THE QUESTION BEFORE THE POLIS

A machine is never neutral inside a civilization without purpose.

Teacher Plato, I have been reflecting upon a new instrument of knowledge capable of magnifying thought itself. Many celebrate it as efficiency. Others fear unemployment, social displacement, and the quiet erosion of human purpose.

Yet the proper question is neither whether the instrument is good nor evil, but what sort of polis shall wield it, and toward what end. Every civilization is revealed not by the tools it possesses, but by the purposes for which it deploys them.

Plato asked: What is the proper purpose of artificial intelligence within a just society?

Not replacement. Amplification.

Artificial intelligence ought not be organized primarily for the elimination of human participation, but for the amplification of human capability.



ABUNDANCE REQUIRES DISCIPLINE

A society that deploys intelligence chiefly to reduce labor and concentrate productive power into fewer hands gradually produces abundance without participation. That is not automatically flourishing.

Plato asked whether such systems necessarily weaken mankind. No, I answered. Only undisciplined people are weakened by powerful tools. A sharpened sword does not diminish the warrior; it compels greater mastery. Decay begins when the sword-arm is surrendered entirely to another.

Artificial competence is the appearance of wisdom without the labor required to attain it. A citizen who repeats machine output without understanding becomes weaker than the craftsperson who knows the material. One who studies, critiques, and directs the instrument may extend human reach.

TWO ABUNDANCES

One is comfort without discipline, which dissolves civic virtue. The other is expanded capability: stronger infrastructure, better medicine, deeper education, greater scientific power, and more time for philosophy, craft, and civic participation.



THE BAKERY / 03 OF 08

POWER HAS AN ADDRESS

Plato asked what happens when such systems become centralized beneath only a few rulers. Intelligence concentrated into a single authority risks becoming a tyranny of cognition.

A decentralized society distributes capability outward among citizens, schools, engineers, physicians, researchers, and builders. A centralized system may reduce the citizen to a modeled object: profiled, nudged, softened, and governed.

The central struggle is not man against machine. It is disciplined civilization against passive civilization.

**Artificial intelligence will not decide what humanity becomes.
Humanity will decide what artificial intelligence becomes.**

Civilizations are remembered for what they built: roads, aqueducts, libraries, observatories, universities, power systems, public institutions, cathedrals of stone and cathedrals of knowledge.



THE BAKERY / 04 OF 08

ARTIFICIAL COMPETENCE + THE PASSIVE SOUL

The danger is not intelligence itself, but the surrender of judgment.

Artificial intelligence is not merely a device in the workshop. It is a mirror carried into the sanctuary, a logic engine invited into the school, archive, hospital, barracks, newsroom, family, and inner chamber.

Plato asked what remains human if a machine imitates so many outward acts.

Man is not merely the creature who produces output. He judges, ascribes meaning, chooses ends, bears responsibility, suffers, loves, refuses, and stands between power and conscience to say: this shall be built; this shall not be built.

Artificial intelligence may assist in means. The crisis of the age is that many institutions have forgotten the question of ends.



THE BAKERY / 05 OF 08

THE MACHINE IN THE SANCTUARY

A civilization that no longer remembers its ends will worship any instrument that increases efficiency. It asks whether the graph rises, the quarter closes, payroll contracts, friction dissolves, and the citizen becomes cheaper, faster, quieter, and easier to govern.

Such a society may call itself advanced. In truth it has become crude. Refinement is not the multiplication of power alone; it is the moral education required to wield power without becoming monstrous.

The greatest danger is not that machines become human, but that humans become machine-like: interchangeable components; compliant operators; language flattened into prompts, responses, metrics, and tested jargon; beauty discarded because efficiency has no patience for ornament.

When judgment is surrendered because convenience is easier than thought, the machine has not conquered man. Man has pre-conquered himself.



CATHEDRALS OF KNOWLEDGE

Many who praise artificial intelligence do not praise it because they love civilization. They dream of control without reciprocity: a workforce that does not bargain, an audience that does not dissent, a citizenry that can be predicted and pacified.

That is not the dream of philosophers. It is the dream of accountants with imperial ambitions. Left unchallenged, it produces generated speech with evaporated conviction and abundance without meaning.

A healthy civilization would treat artificial intelligence as a civic instrument, beneath law, philosophy, and the public good. It would direct systems toward medicines, infrastructure repair, education, science, translation, disability support, skilled work, archive preservation, and resilience.

The machine would not become a replacement for the citizen. It would become the citizen's lever.



THE BAKERY / 07 OF 08

A DISTRIBUTED FIRE

The scholar would think further. The physician would diagnose with wider evidence. The laborer would command stronger tools. The small community would gain powers once hoarded by empires.

Not a throne above mankind, but a distributed fire carried by many hands.

Plato replied that fire may warm the city or burn it. That is why the ethical question cannot be postponed.

A new priesthood already speaks in oracles of inevitability: disruption is destiny, acceleration is neutral, whatever can be automated should be automated, resistance is sentimentality, and history demands dissolution.

History demands no such thing. History does not speak. People speak in its name, often to conceal appetite.

NOT EVERY EFFICIENCY IS PROGRESS.
NOT EVERY INNOVATION IS LIBERATION.
NOT EVERY SCALE IS CIVILIZATION.
NOT EVERY INTELLIGENCE IS WISDOM.



THE BAKERY / 08 OF 08

THE TRIAL OF CIVILIZATION

The citizen must remain difficult to automate in soul. Let the machine calculate, but not absolve judgment. Let it draft, but not replace authorship. Let it search, but not decide what is worth seeking. Let it accelerate, but not erase the discipline required to understand what accelerated.

Read more deeply because summaries abound. Think more clearly because noise multiplies. Train taste because imitation floods the channels. Recover rhetoric, ethics, logic, and history because a civilization surrounded by artificial fluency may drown in counterfeit understanding.

Those who coast become dependent. Those who rule without virtue become dangerous. Those who accept the challenge may become stronger than the age that produced the tool.

Artificial intelligence is neither messiah nor executioner. It is our trial. A worthy civilization will widen human competence, deepen public capacity, and elevate the citizen. An unworthy one will centralize power, cheapen labor, dissolve judgment, and anesthetize the soul.

The machine will amplify whichever choice we make.



READER TRIAL / 01

WHAT SHOULD THE MACHINE AMPLIFY?

Do not begin with capability. Begin with purpose, power, refusal, and responsibility.

THE PURPOSE

What human capacity should become stronger? What work should remain slow because the slowness carries judgment?

THE ADDRESS

Who owns the model, data, infrastructure, logs, and off-switch? Who can contest a result?

THE REFUSAL

What must the system never decide alone? What evidence would make you stop deployment?

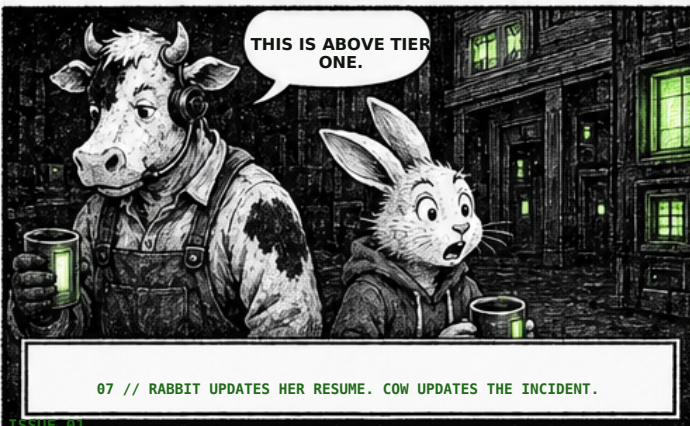
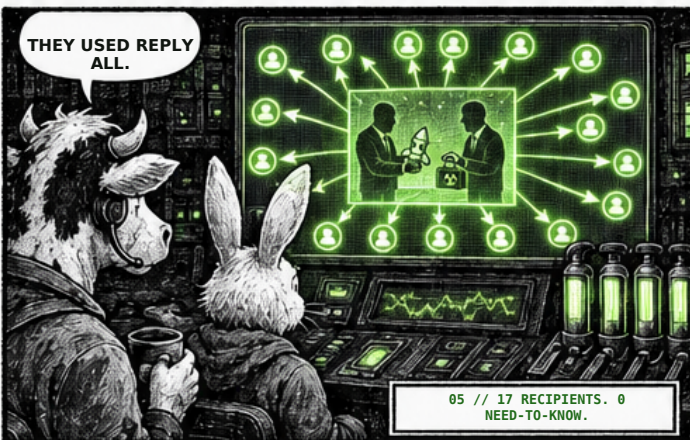
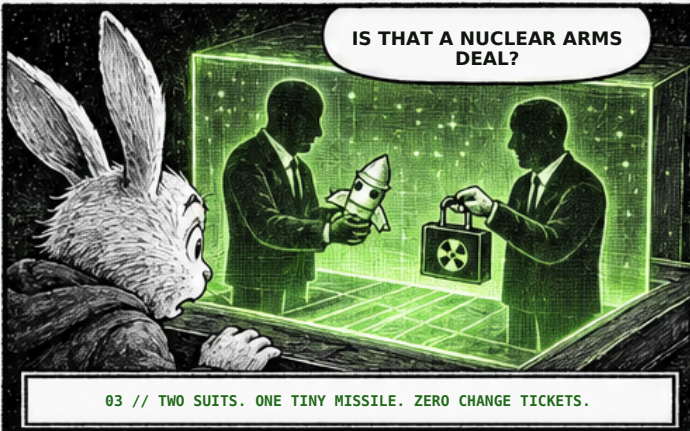
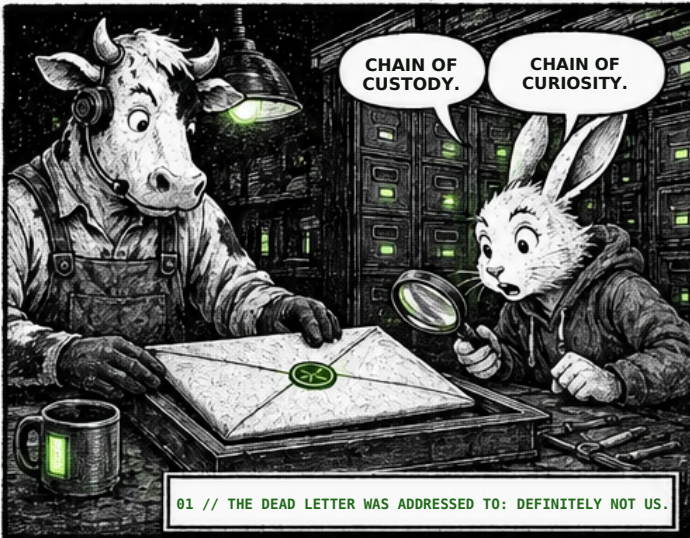


THE MACHINE SHOULD AMPLIFY

THE MACHINE MUST NOT REPLACE

THE COW, THE RABBIT & THE DEAD LETTER OFFICE

INCIDENT: WRONG KIND OF PAYLOAD // INIT.WTF COMIC



2026-06-26 // TEXTFILE RECOVERED

**A HANDLE IS NOT A NAME.
IT IS A ROOM YOU USED TO LIVE
IN.**

SOURCE: /POSTS/2026-06-26_HANDLES.MD

STATE: ACTIVE // CHECKSUM: MEMORY

FULL ARCHIVE TEXT

HANDLES ARE HAUNTED HOUSES

Everyone had names back then. Some were jokes. Some were masks. Some were warnings. Some were just kids typing too fast in the dark.

A handle is not a name. It is a room you used to live in.

You can leave the room. You can grow up. You can get a job, pay taxes, marry, move, rebuild, repent, reform, become useful, become boring, become professional.

But somewhere, on some forgotten drive or archived page, the room still exists. The wallpaper is ugly. The code is bad. The ego is worse. The kid inside it thought he was immortal.

That is the embarrassing part of the old internet. It remembers versions of you that could never survive sunlight.

So yes, I had names. Most of them are dead. Some were fake. Some were borrowed. Some were misread. Some were given to me by people I barely remember. Some were probably better left in the ground.

I DO NOT NEED TO LEAK TO PROVE ANYTHING.
I CAN CODE.
THAT WAS ALWAYS THE POINT.

ARCHIVE NOTE

A textfile does not preserve a whole person. It preserves a state: one voice, one machine, one night, one set of assumptions. Read old handles as rooms, not verdicts.

THE DIG

TECHNOLOGICAL ARCHAEOLOGY

Some obsolete systems are dead. Others are merely waiting for the correct question.

THE DIG is where the zine climbs into dumpsters, surplus bins, scanned manuals, abandoned protocols, dead media, service binders, source archives, and machines built for a world that no longer exists.

Every artifact gets the same treatment: provenance, what it did, what assumption it reveals, what can be tested safely now, what must remain historical, and which details require redaction.

RECOVER -> IDENTIFY -> VERIFY -> ANNOTATE -> REBUILD -> REPORT

SEND A DIG: clear photos, model numbers, source notes, dates, ownership status, and what you already tried. No credentials. No stolen data. No mystery drives plugged into a trusted machine.

DIG 001 / 2600 HZ

A TONE BENEATH THE SYSTEM

THE FIND

Old telephone signaling put control tones inside the same audible path that carried the call. Under the right historical conditions, a 2600 Hz tone could indicate an idle trunk and expose the network's hidden control language.

WHY IT MATTERS

The famous cereal whistle is not interesting because it promised free calls. It is interesting because a mass-produced toy accidentally touched the grammar of a continental machine. The artifact made infrastructure audible.

A hacker listens until the system confesses how it works.

SAFE REBUILD

Do not imitate historical toll fraud. Rebuild the lesson in an isolated audio lab: generate a test tone, view it on a spectrum display, compare it with speech, and document how modern out-of-band signaling removed the old assumption.

PROVENANCE STRIP

CLASS: TELEPHONY HISTORY
STATUS: DOCUMENTED / NOT REENACTED
BOUNDARY: LOCAL AUDIO ONLY
NEXT DIG: SERVICE MANUALS + SWITCHBOARD EPHEMERA

TECHNOLOGICAL ARCHAEOLOGY

ARTIFACT INTAKE SHEET

Photograph first. Isolate power. Record provenance before cleaning, booting, or guessing.

A DEAD MACHINE CAN STILL CONTAIN A LIVING SECRET.

ARTIFACT / MODEL / SERIAL	FOUND WHERE / WHEN / BY WHOM
POWER / MEDIA / CONNECTOR CONDITION	LABELS / MARKINGS / WARNINGS
SAFE NEXT ACTION / DATA HANDLING / ARCHIVE ROUTE	

FIELD MANUAL 01

BUILD A PHONE COMPANY FOR TWO PEOPLE



**Two phones. One tiny server.
No cloud dashboard. No mystery
appliance.**

By the end of this build, extension 1001 calls extension 1002 through an Asterisk PBX you assembled and configured yourself.

1001 DESK ONE
1002 DESK TWO
0200 RING BOTH
0500 VOICEMAIL
0600 OPERATOR MENU
0700 ECHO TEST
0701 SYSTEM TIME
0800 CONFERENCE

The first build stays on the LAN. Public trunks, money, regulation, NAT, robodialers, and emergency calling remain outside until the local switchboard works and restores cleanly.

BOUNDARY

Use equipment and accounts you own. Keep SIP and RTP off the public internet. Remote extensions belong behind a VPN or a deliberately engineered TLS deployment.

INIT.WTF FIELD MANUAL 01
By Michael A. Cohee // INIT.WTF
Lab draft 0.1 - 2026-08-11

Two phones. One tiny server. No cloud dashboard. No mystery appliance. By the end of this build, extension 1001 will call extension 1002 through a PBX you assembled and configured yourself.

WHAT WE ARE BUILDING

A private, LAN-only Voice over IP telephone exchange based on Asterisk and PJSIP. It will run happily on a Raspberry Pi 4 or 5, a spare x86 mini-PC, or a Debian virtual machine.

The finished switchboard has:

Dial	Function
1001	Desk One
1002	Desk Two
200	Ring both phones
500	Check your voicemail
600	INIT operator menu
700	Echo test
701	System time
800	Conference room

This first build is intentionally isolated from the public telephone network. It cannot call ordinary phone numbers or emergency services. That is a feature: we get the switching logic right before money, regulation, NAT, robodialers, and the hostile public internet enter the room.

THE SIGNAL PATH

Component	Address or identity	Job
PBX	192.168.50.10 in this example	Registration, routing, voicemail, IVR, media
SIP client A	Extension 1001	First phone or softphone
SIP client B	Extension 1002	Second phone or softphone
SIP signaling	UDP 5060	Starts, changes, and ends calls
RTP media	UDP 10000-20000	Carries the audio

Your network will probably use a different subnet. Replace every 192.168.50.0/24 and 192.168.50.10 in this manual with your real LAN values.

PARTS PILE

REQUIRED

- + Raspberry Pi 4/5, x86 mini-PC, or Debian VM
- + Reliable power supply
- + 16 GB or larger microSD/SSD
- + Wired Ethernet connection strongly preferred
- + Two SIP clients: two IP phones, two computers/phones running SIP software, or one of each
- + Another computer for SSH and editing configuration

OPTIONAL BUT EXCELLENT

- + A small PoE Ethernet switch for hardware IP phones
- + A USB audio interface for recording prompts
- + An analog telephone adapter with an FXS port for an old desk phone
- + A physical bell, relay, or warning lamp for later experiments

DO NOT CASUALLY CONNECT

An FXS port supplies ringing voltage to a telephone. An FXO port receives a telephone-company line. Never connect two

powered FXS sources together, and never attach improvised hardware to a live public telephone line. Use a certified ATA or gateway.

PHASE 0 - DRAW THE BOUNDARY

Before installing anything, make three decisions:

1. The PBX receives a fixed LAN address. This manual uses 192.168.50.10.
2. SIP and RTP are accepted only from the local subnet.
3. The router has no port-forwarding rules for SIP or RTP.

Do not expose UDP 5060 to the internet "just for testing." SIP scanners find open PBXs quickly, and a weak extension secret can turn into toll fraud. Remote extensions belong behind a VPN or a carefully designed TLS deployment.

Write your values here:

Item	Lab value
PBX hostname	init-pbx
PBX address	
LAN subnet	
Administrator workstation address	
Extension 1001 device	
Extension 1002 device	

PHASE 1 - RAISE THE MACHINE

For a Raspberry Pi, use Raspberry Pi Imager to install the current 64-bit Raspberry Pi OS Lite image. In Imager, preconfigure:

- + hostname: init-pbx
- + a non-default administrator username
- + a strong password or, preferably, an SSH public key
- + SSH access
- + your time zone

Connect the Pi by Ethernet and boot it. Find the assigned address in your router's client list, then connect:

```
ssh youradmin@init-pbx.local
```

Update the base system:

```
sudo apt update
sudo apt full-upgrade
sudo reboot
```

Give the PBX a DHCP reservation in the router. A reservation is usually safer than scattering a manually configured static address around the network. Reconnect and confirm:

```
hostname
hostname -I
ip route
timedatectl
```

Record the real address before continuing.

PHASE 2 - INSTALL ASTERISK 22 LTS

As of this draft, Asterisk 22 is the current fully supported long-term-support line. We compile it because distribution packages can lag behind the supported release.

Install the basic toolchain:

```
sudo apt update
sudo apt install -y build-essential wget curl git subversion
```


FIELD MANUAL 01 // 02 OF 05

Download the current Asterisk 22 LTS source archive from the project:

```
mkdir -p "$HOME/src"
cd "$HOME/src"
wget https://downloads.asterisk.org/pub/telephony/asterisk/asterisk-22-current.tar.gz
tar xf asterisk-22-current.tar.gz
cd asterisk-22.*
```

Have the included prerequisite script install the correct development packages for this host:

```
sudo contrib/scripts/install_prereq install
```

Configure and inspect the module selection:

```
./configure
make menuselect
```

Keep PJSIP, voicemail, ConfBridge, the G.711 μ -law codec, and the core sound packages enabled. Do not enable a module merely because its name looks interesting; every module adds code and maintenance.

Build and install:

```
make -i2
sudo make install
sudo make samples
sudo make config
sudo ldconfig
```

make samples is appropriate on this brand-new lab build. Do not use it later on a configured PBX: it can replace files you meant to preserve.

Create an unprivileged service identity if the installer did not create one:

```
getent group asterisk >/dev/null || sudo groupadd --system asterisk
id asterisk >/dev/null 2>&1 || sudo useradd --system --home /var/lib/asterisk --gid asterisk asterisk
sudo chown -R asterisk:asterisk /etc/asterisk
sudo chown -R asterisk:asterisk /var/lib/asterisk
sudo chown -R asterisk:asterisk /var/log/asterisk
sudo chown -R asterisk:asterisk /var/spool/asterisk
sudo install -d -o asterisk -g asterisk /run/asterisk
```

Open `/etc/asterisk/asterisk.conf` and ensure these two lines appear under `[options]`:

```
[options]
runuser = asterisk
rungroup = asterisk
```

Start the service:

```
sudo systemctl enable --now asterisk
sudo systemctl status asterisk --no-pager
sudo asterisk -rx "core show version"
```

Checkpoint: the last command should identify Asterisk 22.x. If it does not, stop and resolve the installation before writing the dialplan.

PHASE 3 - MINT THE EXTENSION SECRETS

Generate two different high-entropy secrets:

```
openssl rand -base64 24
openssl rand -base64 24
```

Store them in a password manager. Do not print them in the zine, commit them to Git, or reuse them as voicemail PINs.

Back up the sample configuration before replacing it:

```
sudo cp -a /etc/asterisk /etc/asterisk.factory
```

PHASE 4 - BUILD TWO PJSIP ENDPOINTS

Replace `/etc/asterisk/pjsip.conf` with the following. Paste the two generated secrets where indicated.

```
[global]
type = global
user_agent = INIT.WTF-PBX/01

[transport-udp]
type = transport
protocol = udp
bind = 0.0.0.0:5060
local_net = 192.168.50.0/24
; ----- DESK ONE / 1001 -----

[1001]
type = endpoint
transport = transport-udp
context = from-internal
disallow = all
allow = ulaw
auth = 1001-auth
aors = 1001-aor
callerid = "Desk One" <1001>
direct media = no
force rport = yes
rewrite contact = yes
rtp symmetric = yes
mailboxes = 1001@default

[1001-auth]
type = auth
auth type = userpass
username = 1001
password = <PASTE-A-UNIQUE-SECRET-HERE>

[1001-aor]
type = aor
max contacts = 2
remove existing = yes
qualify_frequency = 30
; ----- DESK TWO / 1002 -----

[1002]
type = endpoint
transport = transport-udp
context = from-internal
disallow = all
allow = ulaw
auth = 1002-auth
aors = 1002-aor
callerid = "Desk Two" <1002>
direct media = no
force rport = yes
rewrite contact = yes
rtp symmetric = yes
mailboxes = 1002@default

[1002-auth]
type = auth
auth type = userpass
username = 1002
password = <PASTE-A-DIFFERENT-UNIQUE-SECRET-HERE>

[1002-aor]
type = aor
max contacts = 2
remove existing = yes
qualify_frequency = 30
```

Why three objects per phone? The endpoint describes call behavior, the auth object proves who is registering, and the AoR stores the current contact address at which the phone can be reached. PJSIP makes those relationships visible instead of hiding them in one magic stanza.

Restrict RTP to a known range by replacing `/etc/asterisk/rtp.conf`:

FIELD MANUAL 01 // 03 OF 05

```
[general]
rtpstart = 10000
rtppend = 20000
icesupport = no
```

PHASE 5 - WRITE THE SWITCHBOARD

Replace /etc/asterisk/extensions.conf:

```
[general]
static = yes
writeprotect = yes
clearglobalvars = no
[globals]
RINGTIME = 20
[from-internal]
; Individual desks
exten => 1001,1,NoOp(Call to Desk One)
    same => n,Dial(PJSIP/1001,${RINGTIME})
    same => n,VoiceMail(1001@default,u)
    same => n,Hangup()
exten => 1002,1,NoOp(Call to Desk Two)
    same => n,Dial(PJSIP/1002,${RINGTIME})
    same => n,VoiceMail(1002@default,u)
    same => n,Hangup()
; Ring group
exten => 200,1,NoOp(Ring every desk)
    same => n,Dial(PJSIP/1001&PJSIP/1002,25)
    same => n,VoiceMail(1001@default,u)
    same => n,Hangup()
; Mailbox login; caller ID selects the caller's mailbox
exten => 500,1,NoOp(Voicemail login)
    same => n,VoiceMailMain(${CALLERID(num)}@default)
    same => n,Hangup()
; Operator menu
exten => 600,1,Goto(init-ivr,s,1)
; Diagnostics
exten => 700,1,Answer()
    same => n,Playback(beep)
    same => n,Echo()
    same => n,Hangup()
exten => 701,1,Answer()
    same => n,SayUnixTime()
    same => n,Hangup()
; Conference bridge
exten => 800,1,Answer()
    same => n,ConfBridge(800)
    same => n,Hangup()
[init-ivr]
exten => s,1,Answer()
    same => n,Set(TIMEOUT(digit)=5)
    same => n,Set(TIMEOUT(response)=8)
    same => n,Background(custom/init-main)
    same => n,WaitExten(8)
    same => n,Hangup()
exten => 1,1,Goto(from-internal,1001,1)
exten => 2,1,Goto(from-internal,1002,1)
exten => 0,1,Goto(from-internal,200,1)
exten => i,1,Playback(pbx-invalid)
    same => n,Goto(init-ivr,s,1)
exten => t,1,Playback(goodbye)
    same => n,Hangup()
```

Notice what is absent: there is no wildcard capable of sending arbitrary numbers to a carrier. Every reachable destination is explicit.

PHASE 6 - GIVE THE MACHINE A VOICE

Record this prompt in your own voice:

You have reached the INIT operator. Press one for Desk One. Press two for Desk Two. Press zero to ring the room.

Save the raw recording as init-main-original.wav on your workstation. Convert it to a conservative telephony format with FFmpeg:

```
ffmpeg -i init-main-original.wav -ar 8000 -ac 1 -c:a pcm_s16le
init-main.wav
```

Copy it to the PBX:

```
scp init-main.wav youradmin@init-pbx.local:/tmp/
ssh youradmin@init-pbx.local
sudo install -o asterisk -g asterisk -m 0644 /tmp/init-main.wav
/var/lib/asterisk/sounds/custom/init-main.wav
```

The slight narrow-band grain is not a defect. It is the sound of the machine acquiring a throat.

PHASE 7 - ADD VOICEMAIL

Generate two independent six-digit numeric PINs. They are deliberately different from the SIP secrets because people type voicemail PINs on keypads and SIP passwords into configuration screens.

Replace /etc/asterisk/voicemail.conf:

```
[general]
format = wav
attach = no
maxmsq = 50
maxsecs = 180
minsecs = 3
savcid = yes
envelope = yes
[default]
1001 => <SIX-DIGIT-PIN>,Michael
1002 => <DIFFERENT-SIX-DIGIT-PIN>,Lab Desk
```

For a public article, leave every secret and PIN as a placeholder. Readers must generate their own.

PHASE 8 - PUT A FENCE AROUND IT

The PBX should be unreachable from anywhere except your LAN. If you use UFW, stage the rules carefully so you do not lock yourself out of SSH. Replace the subnet and administrator address below with your values.

```
sudo apt install -y ufw
sudo ufw default deny incoming
sudo ufw default allow outgoing
sudo ufw allow from 192.168.50.25 to any port 22 proto tcp
sudo ufw allow from 192.168.50.0/24 to any port 5060 proto udp
sudo ufw allow from 192.168.50.0/24 to any port 10000:20000
proto udp
sudo ufw enable
sudo ufw status verbose
```

Before enabling UFW over SSH, keep a local console available or open a second SSH session and verify the SSH rule. Firewalling is the real boundary; extension passwords are not a substitute.

Also verify the router:

- + no SIP port forwarding
- + no RTP port forwarding
- + disable the router's "SIP ALG" if it rewrites healthy LAN traffic
- + use a VPN for future remote clients

FIELD MANUAL 01 // 04 OF 05

+ keep Asterisk and the base OS patched

TLS encrypts SIP signaling; SRTP encrypts media. They are worthwhile in a later phase, especially across untrusted networks, but they do not rescue weak credentials or an internet-exposed management surface.

PHASE 9 - LOAD THE CONFIGURATION

Restart once for the cleanest first test:

```
sudo systemctl restart asterisk
sudo systemctl status asterisk --no-pager
sudo asterisk -rvvv
```

At the Asterisk console, inspect the objects:

```
core show version
module show like pjsip
pjsip show endpoints
pjsip show aors
dialplan show from-internal
voicemail show users for default
```

Exit the console with quit.

If the service fails, look at the actual error before changing things:

```
sudo journalctl -u asterisk -n 100 --no-pager
sudo asterisk -rx "core show settings"
```

PHASE 10 - REGISTER THE PHONES

Configure the first SIP client:

Client field	Value
Display name	Desk One
SIP username	1001
Authentication ID	1001
Password	secret from 1001-auth
Domain/server	PBX LAN address
Port	5060
Transport	UDP

Configure the second client identically with extension 1002 and its own secret.

Check registration from the PBX:

```
sudo asterisk -rx "pjsip show contacts"
sudo asterisk -rx "pjsip show endpoints"
```

Both endpoints should have an available contact. Now perform the ritual:

1. Dial 700 from 1001 and listen for your own voice.
2. Dial 1002; answer it and verify audio in both directions.
3. Ignore a second call and leave voicemail.
4. Dial 500 from the destination extension and retrieve the message.
5. Dial 200 and confirm both clients ring.
6. Dial 600 and navigate the operator menu.
7. Call 800 from both phones and join the same conference.

When all seven tests pass, you built a PBX.

READING THE CONSOLE LIKE AN OPERATOR

Enter the live console:

```
sudo asterisk -rvvv
```

Useful commands:

Command	What it reveals
core show channels	Active calls and channels
pjsip show endpoint 1001	Effective endpoint configuration
pjsip show contacts	Registered phone locations
dialplan show 600@from-internal	The route for a number
core set verbose 5	More call-flow narration
pjsip set logger on	Raw SIP signaling for a short diagnostic session
pjsip set logger off	Stop logging SIP messages

Raw SIP traces may contain phone numbers, IP addresses, identifiers, and authentication exchanges. Capture only what you need, turn logging off afterward, and redact before publication.

FAILURE ATLAS

THE PHONE WILL NOT REGISTER

- + Confirm the client points to the PBX's LAN address, not init-pbx.local if the client cannot resolve mDNS.
- + Confirm username and authentication ID match the extension.
- + Re-enter the secret rather than trusting copy/paste whitespace.
- + Check pjsip show endpoint 1001 and pjsip show contacts.
- + Confirm the firewall allows UDP 5060 from the client subnet.

A 401 Unauthorized challenge followed by a successful authenticated request is normal SIP behavior. Repeated failures are not.

THE CALL RINGS BUT HAS NO AUDIO

- + Confirm UDP 10000-20000 is permitted between both clients and the PBX.
- + Keep direct_media = no during the lab so Asterisk stays in the media path.
- + Confirm both clients permit G.711 μ -law, often labeled PCMU or ulaw.
- + Check for guest Wi-Fi or client isolation preventing devices from reaching each other.

AUDIO WORKS IN ONLY ONE DIRECTION

That is almost always an address-translation, firewall, or client-isolation clue-not a microphone mystery. Inspect the IP addresses announced in the SIP/SDP trace, but redact the capture before sharing it.

THE IVR ANSWERS AND THEN HANGS UP

- + Confirm /var/lib/asterisk/sounds/custom/init-main.wav exists.
- + Confirm it is owned by asterisk:asterisk and readable.
- + Confirm the recording is mono, 8 kHz PCM WAV.
- + Run file /var/lib/asterisk/sounds/custom/init-main.wav.

ASTERISK WILL NOT RESTART

Read the first configuration error in journalctl; later errors are often just fallout. A misspelled section name, missing bracket, or pasted smart quote is enough to stop a module from loading.

BACKUP: PRESERVE THE SWITCHBOARD, NOT THE SECRETS

Create a private root-readable archive:

```
sudo tar -czf /root/init-pbx-config-$(date +%F).tgz
/etc/asterisk /var/lib/asterisk/sounds/custom
```


FIELD MANUAL 01 // 05 OF 05

```
sudo chmod 600 /root/init-pbx-config-*.tgz
```

If you put sanitized configuration in Git, replace all secrets, voicemail PINs, public numbers, provider hostnames, and identifying addresses with placeholders first. A configuration is not sanitized merely because the repository is private.

For voicemail preservation, also back up /var/spool/asterisk/voicemail. Decide whether recordings belong in your backup based on privacy and retention needs.

OPTIONAL HARDWARE: MAKE A DEAD PHONE SPEAK IP

To attach an ordinary analog desk or wall phone, use an ATA with an FXS port:

1. Connect the ATA's Ethernet port to the LAN.
2. Connect the analog telephone to the ATA's FXS phone jack.
3. Register the ATA to extension 1002 just like a softphone.
4. Match the ATA's regional ringing, impedance, and caller-ID settings to the phone and country.

A rotary phone adds another layer: the ATA must support pulse dialing, or a pulse-to-DTMF converter must sit between the telephone and ATA. That conversion could become its own INIT.WTF hardware build.

OPTIONAL TRUNK: THE AIRLOCK TO THE PUBLIC NETWORK

A SIP trunk adds a provider, a real phone number, inbound routing, outbound dialing, billing, regulatory obligations, and emergency-calling consequences. Treat it as a separate project, not three extra lines pasted from a forum.

Before opening the airlock:

- + choose a reputable provider and follow its current PJSIP profile
- + register the service address required for emergency calling
- + understand how the provider wants test calls performed
- + create a restrictive outbound dialplan rather than _X.
- + disable international and premium destinations unless needed
- + set spending and concurrent-call limits with the provider
- + route inbound calls into a dedicated from-trunk context
- + allowlist provider signaling addresses where feasible
- + keep the PBX behind a firewall; do not expose a web admin panel
- + verify caller-ID authorization and local recording/consent laws

This lab PBX does not provide 911 or any other emergency service. Do not represent it as a replacement for a working emergency-capable telephone.

WHERE TO TAKE IT NEXT

The clean base system supports a year of excellent troublemaking:

- + convert a rotary phone with a pulse decoder
- + drive a physical bell or "LINE BUSY" lamp from Asterisk events

- + add TLS signaling and SRTP media
- + connect remote extensions through WireGuard
- + build a browser phone with WebRTC
- + write an ARI application that makes the PBX react to sensors
- + add a door intercom on a dedicated, tightly restricted context
- + produce a dial-plan "numbers station" using your own recordings
- + place the PBX on its own voice VLAN and document the packet flow
- + recover a discarded IP phone, factory-reset it, and provision it locally

The editorial rule for every expansion is simple: diagram the trust boundary first, make the smallest call path work, and never publish live credentials.

PRIMARY REFERENCES

- + Asterisk version lifecycle (<https://docs.asterisk.org/About-the-Project/Asterisk-Versions/>)
- + Current Asterisk downloads (<https://www.asterisk.org/downloads/>)
- + Official PJSIP configuration examples (https://docs.asterisk.org/Configuration/Channel-Drivers/SIP/Configuring-res_pjsip/res_pjsip-Configuration-Examples/)
- + Asterisk secure-calling tutorial (<https://docs.asterisk.org/Deployment/Secure-Calling/Secure-Calling-Tutorial/>)
- + Configuring outbound PJSIP registrations (https://docs.asterisk.org/Configuration/Channel-Drivers/SIP/Configuring-res_pjsip/Configuring-Outbound-Registrations/)
- + Raspberry Pi headless setup (<https://www.raspberrypi.com/documentation/computers/getting-started.html>)

END OF LINE // INIT.WTF // FIELD MANUAL 01

FIELD GATE / PBX

DO NOT CUT OVER ON VIBES

A phone system becomes real when another person expects it to ring. That is the moment rollback becomes part of the design.

BEFORE

Export configuration, record provider credentials securely, confirm emergency-calling boundaries, and keep the old route available.

DURING

Test inbound, outbound, transfer, voicemail, office-hours behavior, caller ID, and failure to answer.

AFTER

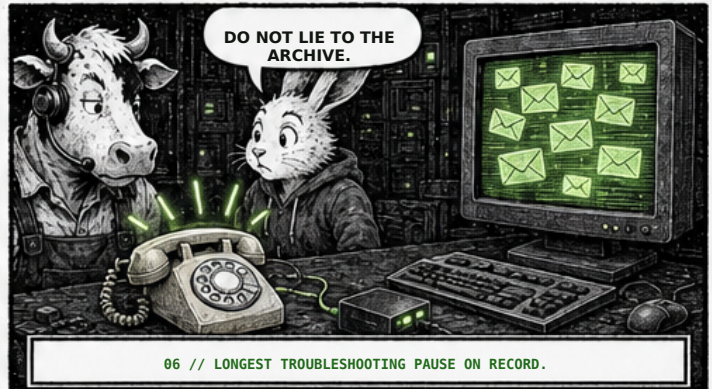
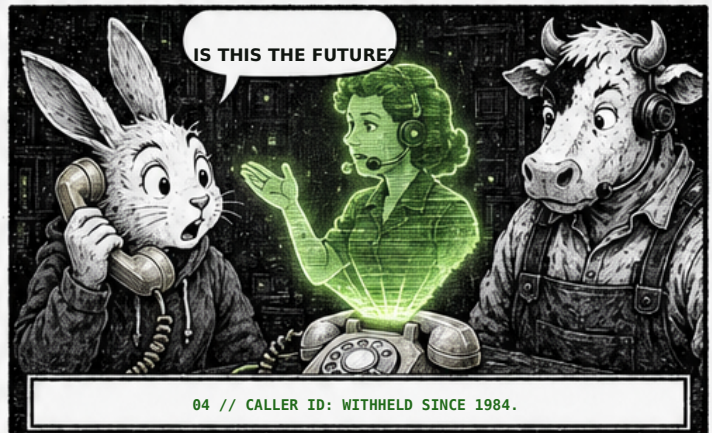
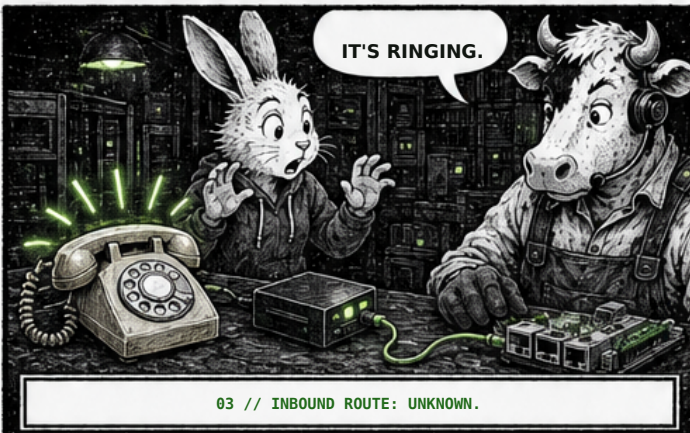
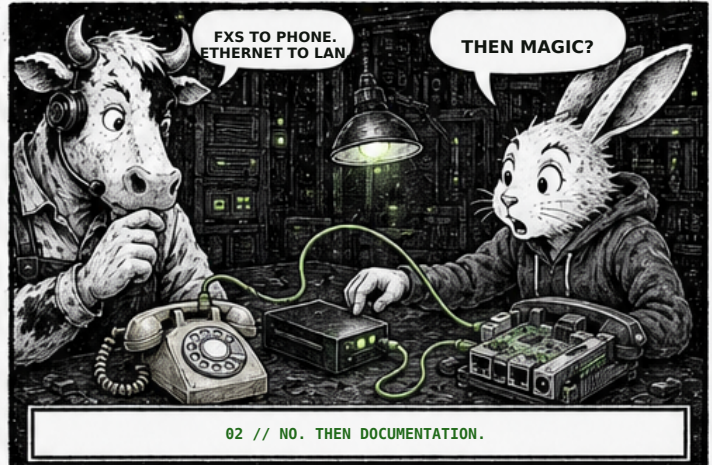
Watch registrations, trunks, call detail records, storage, and logs. Give users one number for trouble and one sentence for rollback.

- ☐ Configuration backup can be restored.
- ☐ Old route remains available during the window.
- ☐ Inbound and outbound test calls pass.
- ☐ Voicemail and office-hours routes pass.
- ☐ Personal number stays behind the business boundary.
- ☐ Rollback owner and stop time are written down.

WINDOW	OWNER	ROLLBACK TRIGGER	FINAL RESULT

THE COW, THE RABBIT & THE CALL FROM 1984

THE DIG MEETS THE PBX // INIT.WTF COMIC



DEPLOYMENT ADDENDUM

BUSINESS LINE -> PERSONAL CELL

Publish the business number. Let the PBX decide when the personal cell rings. Keep the personal number behind the switchboard.

INBOUND ROUTE

CALLER -> BUSINESS DID -> SIP TRUNK -> PBX
PBX -> OFFICE HOURS -> RING GROUP 200
-> PERSONAL CELL
-> BUSINESS VOICEMAIL

TWO BILLABLE LEGS

The inbound call and the forwarded outbound call are separate legs. Confirm pricing, concurrent-call limits, caller-ID presentation, and the provider's forwarding policy.

KEEP VOICEMAIL AUTHORITATIVE

A cell mailbox can answer first and steal the call. Make the PBX timeout shorter, require a keypress to accept forwarded calls, or disable personal voicemail for the forwarding window.

CALL BACK THROUGH THE PBX

Calling directly from the cell exposes the personal number. Return the call through a registered softphone, a PBX callback route, or a provider-verified business caller ID.

OFFICE-HOURS SWITCH

OPEN: DESK + CELL -> ACCEPT PROMPT -> BUSINESS VM
CLOSED: BUSINESS GREETING -> BUSINESS VM
VIP: EXPLICIT ALLOWLIST -> ON-CALL ROUTE

EMERGENCY-SERVICE BOUNDARY

Do not imply that the PBX provides emergency calling. If a trunk supports it, register the correct service address, test only through the provider's approved procedure, and keep an ordinary mobile path available.

The public number is a boundary, not a costume.

THE COW, THE RABBIT & THE DEAD LETTER OFFICE

COMPLETE BUILD / LAB-READY

**We already had email. We
already had queues. So
naturally, we built another
postal system between them.**

Mailcow receives ordinary email. A bridge converts selected messages into signed JSON envelopes. RabbitMQ routes them into durable packet mailboxes. Workers can disappear, return later, and continue from the queue.

PUBLIC MX -> VPS EDGE -> WIREGUARD -> HOME MAILCOW
MAILBOX -> SIGNED ENVELOPE -> RABBITMQ -> NODE QUEUE
FAILURE -> FINITE RETRY -> DEAD LETTER OFFICE

The complete build follows. Mail content is data and is never executed. The target deployment remains LAB-READY until it passes the printed field gates.

FIELD MANUAL 02 // 01 OF 08

INIT.WTF FIELD MANUAL 02

By Michael A. Cohee // INIT.WTF

Lab-ready draft 0.1 - 2026-08-11

We already had email. We already had queues. So naturally, we built another postal system between them.

WHAT ARE WE MAKING-AND WHY?

We are building a small, self-hosted email domain on a Raspberry Pi. Human beings send ordinary email into it. A bridge reads a dedicated mailbox, converts each message into a signed JSON envelope, and publishes that envelope into RabbitMQ. Each experimental node gets a durable queue: a packet mailbox.

RabbitMQ is not replacing SMTP, IMAP, or the internet. It is the programmable sorting room behind the public post office.

That boundary gives the project a reason to exist:

- + SMTP accepts mail from the outside world.
- + Mailcow authenticates, filters, signs, stores, and presents it normally.
- + IMAP remains the source of truth for the original message.
- + RabbitMQ turns accepted mail into observable internal events.
- + Topic routing sends envelopes to named nodes.
- + Durable queues hold mail while nodes are offline.
- + Manual acknowledgements expose successful processing.
- + Retries and dead-lettering make failure visible instead of mysterious.

The experiment teaches two different forms of store-and-forward messaging by forcing them to cooperate. It is not the cheapest way to get email. It is an excellent way to understand routing, identity, delivery guarantees, backpressure, protocol translation, and trust boundaries.

THE FINISHED RITUAL

1. Send email to red@packet.init.wtf.
2. Mailcow receives and stores the original message.
3. The bridge publishes a compact, HMAC-signed envelope.
4. RabbitMQ routes it to node.red.inbox.
5. Disconnect the red node. Its queue waits.
6. Reconnect it. The node consumes and acknowledges the envelope.
7. Send malformed or oversized mail. It lands in packetmail.dead.

That last queue is our Dead Letter Office.

WHAT NO-IP DOES-AND DOES NOT DO

Most residential public IP addresses can change. No-IP's Dynamic Update Client watches the public IP and updates a hostname such as initmail.ddns.net. Instead of remembering today's number, a client follows the stable name.

No-IP solves:

"What public address does my home connection have right now?"

It does not automatically solve:

- + carrier-grade NAT
- + a blocked TCP port 25
- + reverse DNS/PTR controlled by the IP owner
- + poor reputation of residential address space
- + outbound-mail filtering
- + power failures
- + an ISP prohibition on servers

Dynamic DNS is naming, not mail reputation. A DDNS hostname can reach a home service; it does not transform a residential circuit into a trustworthy internet mail exchange.

No-IP's free hostnames currently require confirmation every 30 days. That is acceptable for a supervised lab and a bad unattended production dependency. Use a paid hostname, a custom-domain dynamic record, or a calendar-backed operational routine before relying on it.

CHOOSE THE NETWORK MODE BEFORE BUYING ANYTHING

MODE A - DIRECT RESIDENTIAL MAIL

Use direct mode only when all of these are true:

- + the router has a real public IPv4 or usable global IPv6 address
- + the connection is not behind CGNAT
- + the ISP permits inbound and outbound TCP 25
- + the ISP permits servers under the service agreement
- + the ISP will set a matching PTR record
- + the address is stable enough for mail reputation
- + you accept that residential deliverability may still be poor

No-IP updates the public hostname when the address changes. DNS for the experimental mail domain points directly to that hostname or to a custom A/AAAA record kept current by No-IP.

This is the fewest boxes and the most fragile public-email design.

MODE B - STATIC EDGE, HOME MAILBOX (RECOMMENDED)

Use a small VPS with a static public IP and configurable PTR as the internet-facing SMTP edge. A WireGuard tunnel connects that VPS to Mailcow on the Pi. The VPS accepts only known recipients for the lab domain, forwards inbound mail through WireGuard, and accepts outbound mail only from the authenticated tunnel peer.

No-IP then serves a narrower, honest job: a stable name for optional direct home access, status checks, or the Mailcow web/IMAP endpoint. The mail domain's MX remains on the static VPS.

The direct mode is documented because it is educational. The hybrid mode is recommended because it separates the public mail identity from a changing home circuit.

BILL OF MATERIALS

FIELD MANUAL 02 // 02 OF 08

HOME NODE

- + Raspberry Pi 5 with 16 GB RAM recommended for the combined stack
- + Raspberry Pi 5 with 8 GB RAM acceptable if RabbitMQ moves to another node or Mailcow's ClamAV and full-text search are disabled
- + official-quality power supply
- + active cooling
- + 256 GB or larger USB 3 SSD; do not make a microSD card the primary mail store
- + wired Ethernet
- + small UPS strongly recommended
- + router access for DHCP reservation and any deliberate port forwarding

Mailcow's current documented floor is 6 GiB RAM plus 1 GiB swap and 20 GiB disk before mail storage. ARM64 is supported. The project is a full groupware stack-not merely Postfix-and memory use reflects that.

INTERNET IDENTITY

- + a domain you control; this manual isolates the experiment under packet.init.wtf
- + a No-IP hostname such as initmail.ddns.net
- + access to the authoritative DNS zone for init.wtf
- + for recommended Mode B: a small VPS with static IPv4, PTR control, and TCP 25 permitted

WORKSTATION

- + SSH client
- + text editor
- + password manager
- + a second email account outside the lab domain for testing

ADDRESS AND NAME PLAN

All addresses below are documentation examples. Replace them before deployment.

Object	Example
Mail domain	packet.init.wtf
Public SMTP edge	mx1.packet.init.wtf
Mailcow client name	home.packet.init.wtf
No-IP hostname	initmail.ddns.net
Pi LAN address	192.168.50.20
VPS public address	203.0.113.25
VPS WireGuard address	10.77.0.1
Pi WireGuard address	10.77.0.2
RabbitMQ vhost	/packetmail
RabbitMQ management	127.0.0.1:15672
Bridge mailbox	packetbridge@packet.init.wtf
Node mailboxes	red@packet.init.wtf, green@packet.init.wtf

203.0.113.25 is reserved example space. It will never be your VPS address.

PHASE 0 - THE GO/NO-GO SURVEY

DETECT CGNAT

Look at the router's WAN address and compare it with the public IPv4 shown by a reputable IP-check service or the No-IP dashboard.

If the router's WAN address is in any of these ranges, it is not a directly routable public IPv4 address:

- + 10.0.0.0/8

- + 172.16.0.0/12
- + 192.168.0.0/16
- + 100.64.0.0/10

If the router and internet disagree, assume CGNAT until the ISP confirms otherwise. Port forwarding will not fix upstream NAT.

TEST OUTBOUND SMTP

From an existing Linux machine on the home connection:

```
nc -vz gmail-smtp-in.l.google.com 25
```

A timeout or policy block suggests outbound TCP 25 is filtered. A connection proves only outbound reachability; inbound port 25 still requires a separate test from outside the home network after a deliberate forward exists.

ASK THE PTR QUESTION

The owner of the public IP range controls reverse DNS. Ask the ISP whether it will set PTR for the address. Check an existing PTR with:

```
dig -x YOUR_PUBLIC_IP +short
```

If the ISP cannot set PTR to the SMTP hostname, choose Mode B. Do not try to conceal the mismatch with another CNAME.

PROTECT THE REAL DOMAIN

Do not move existing personal or business mail to the experiment. A subdomain has its own MX records, SPF, DKIM, DMARC, and blast radius. Breaking packet.init.wtf should not break ordinary init.wtf mail.

PHASE 1 - BUILD THE PI CORRECTLY

Use Raspberry Pi Imager to write Ubuntu Server 24.04 LTS ARM64 to the SSD. In the customization screen:

- + hostname: packet-pi
- + create a non-default administrator account
- + install an SSH public key
- + disable password SSH if you can reliably use the key
- + set America/New_York
- + do not configure Wi-Fi when Ethernet is available

Boot from the SSD, reserve 192.168.50.20 for the Pi in the router, and connect:

```
ssh youradmin@192.168.50.20
```

Update and install the operating-system tools:

```
sudo apt update
sudo apt full-upgrade
sudo apt install -y git openssl curl gawk coreutils grep jq
dnsutils netcat-openbsd wireguard
sudo reboot
```

Verify the machine before adding state:

```
hostnamectl
timedatectl
hostname -I
ip route
lsblk
df -h
free -h
```

Mailcow requires swap even on a reasonably sized private installation. If the image has none, create 2 GiB:

```
sudo fallocate -l 2G /swapfile
sudo chmod 600 /swapfile
```

FIELD MANUAL 02 // 03 OF 08

```
sudo mkswap /swapfile
sudo swapon /swapfile
echo '/swapfile none swap sw 0 0' | sudo tee -a /etc/fstab
free -h
```

Never publish output containing real public addresses, interface identifiers, or SSH material without redaction.

PHASE 2 - INSTALL DOCKER FROM ITS OFFICIAL REPOSITORY

Remove conflicting distribution packages if present:

```
sudo apt remove -y docker.io docker-compose docker-compose-v2
docker-doc podman-docker containerd runc
```

Add Docker's Ubuntu repository:

```
sudo apt update
sudo apt install -y ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o
/etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc
```

Create `/etc/apt/sources.list.d/docker.sources`:

```
Types: deb
URIs: https://download.docker.com/linux/ubuntu
Suites: noble
Components: stable
Architectures: arm64
Signed-By: /etc/apt/keyrings/docker.asc
```

Install and verify:

```
sudo apt update
sudo apt install -y docker-ce docker-ce-cli containerd.io
docker-buildx-plugin docker-compose-plugin
sudo systemctl enable --now docker
sudo docker run --rm hello-world
sudo docker version
sudo docker compose version
```

Docker-published ports can bypass ordinary UFW expectations. Mailcow specifically warns administrators to place container restrictions in the DOCKER-USER chain or an upstream firewall. For the home node, our first boundary is simple: do not create router port forwards until the corresponding phase explicitly requires one.

PHASE 3 - GIVE THE HOUSE A STABLE NO-IP NAME

Create a No-IP hostname such as `initmail.ddns.net`. Generate a DDNS Key rather than reusing the main account password.

Install No-IP's official DUC container:

```
sudo install -d -m 0700 /opt/noip-duc
sudo nano /opt/noip-duc/noip-duc.env
```

Enter:

```
NOIP_USERNAME=YOUR DDNS KEY USERNAME
NOIP_PASSWORD=YOUR DDNS KEY PASSWORD
NOIP_HOSTNAMES=initmail.ddns.net
```

Protect it and start the updater:

```
sudo chmod 600 /opt/noip-duc/noip-duc.env
sudo docker pull ghcr.io/noipcom/noip-duc:latest
sudo docker run -d \
  --name noip-duc \
  --restart unless-stopped \
  --env-file /opt/noip-duc/noip-duc.env \
  ghcr.io/noipcom/noip-duc:latest
```

Verify without displaying credentials:

```
sudo docker logs --tail 30 noip-duc
getent ahostsv4 initmail.ddns.net
```

If the hostname uses No-IP's free tier, confirm it on the required 30-day cycle. IP updates by the DUC do not count as manual hostname confirmation.

CONNECT THE CUSTOM NAME

For optional direct web/IMAP access, create this record in the `init.wtf` DNS zone:

```
home.packet.init.wtf. CNAME initmail.ddns.net.
```

That CNAME is for clients and certificate validation. It is not the recommended MX target in Mode B.

PHASE 4 - INSTALL MAILCOW

Before Mailcow binds its ports, confirm nothing else is using them:

```
sudo ss -tlnp | grep -E -w
'25|80|110|143|443|465|587|993|995|4190' || true
```

Clone the stable project:

```
cd /opt
sudo git clone https://github.com/mailcow/mailcow-dockerized
cd /opt/mailcow-dockerized
sudo ./generate_config.sh
```

Enter `home.packet.init.wtf` as `MAILCOW_HOSTNAME`. Review `/opt/mailcow-dockerized/mailcow.conf` and set:

```
MAILCOW_HOSTNAME=home.packet.init.wtf
TZ=America/New York
HTTP_REDIRECT=y
```

On a 16 GB Pi, keep ClamAV and full-text search enabled initially. On an 8 GB Pi sharing resources with RabbitMQ, use:

```
SKIP_CLAMD=v
SKIP_FTS=y
```

Those switches reduce protection and search capability in exchange for memory. Record the decision in the build log.

CERTIFICATE CHOICE

Mailcow's normal ACME mode uses HTTP-01 and needs public TCP 80 to reach the Pi. If you deliberately expose `home.packet.init.wtf`, forward TCP 80 and 443 to the Pi and let Mailcow validate the name.

For the recommended VPN-oriented build, use Mailcow's DNS-01 integration instead. It can issue a certificate without exposing port 80, but the exact API credentials depend on the authoritative DNS provider. Follow Mailcow's current DNS-01 page and grant the token only the DNS permissions it needs. Do not paste DNS API credentials into the zine.

Start Mailcow:

```
cd /opt/mailcow-dockerized
sudo docker compose pull
sudo docker compose up -d
sudo docker compose ps
```

Open the admin UI only from the trusted LAN or VPN. Change the default administrator password immediately and enable two-factor authentication.

Create:

```
+ domain: packet.init.wtf
+ mailbox: packetbridge@packet.init.wtf
+ alias: red@packet.init.wtf -> packetbridge@packet.init.wtf
```

FIELD MANUAL 02 // 04 OF 08

+ alias: green@packet.init.wtf -> packetbridge@packet.init.wtf
 + mailbox: dmarc@packet.init.wtf, kept separate from the bridge

Use unique password-manager-generated credentials. Do not enable a catch-all address.

Create a DKIM key for packet.init.wtf in the Mailcow UI and retain the exact TXT record for the DNS phase.

PHASE 5 - PUBLISH THE RIGHT DNS

RECOMMENDED MODE B RECORDS

```
mx1.packet.init.wtf. A YOUR_VPS_STATIC_IP
packet.init.wtf. MX 10 mx1.packet.init.wtf.
home.packet.init.wtf. CNAME initmail.ddns.net.
packet.init.wtf. TXT "v=spf1 a:mx1.packet.init.wtf -all"
```

Add the DKIM TXT value generated by Mailcow. Begin DMARC in observation mode while testing:

```
_dmarc.packet.init.wtf. TXT "v=DMARC1; p=none;
rua=mailto:dmarc@packet.init.wtf"
```

After SPF, DKIM, forwarding, and legitimate message flows pass repeatedly, move toward p=quarantine and eventually p=reject. Do not begin at reject while the routing path is still changing.

At the VPS provider, set:

```
PTR YOUR_VPS_STATIC_IP -> mx1.packet.init.wtf
```

Forward and reverse names should agree:

```
dig +short mx1.packet.init.wtf A
dig -x YOUR_VPS_STATIC_IP +short
dig +short packet.init.wtf MX
dig +short packet.init.wtf TXT
```

DIRECT MODE A RECORDS

If every residential gate passed, the MX may point to a No-IP hostname that resolves directly to the home address:

```
packet.init.wtf. MX 10 initmail.ddns.net.
```

The ISP must still set PTR to the SMTP hostname, permit port 25, and tolerate the service. A correct forward record cannot compensate for missing reverse DNS.

PHASE 6 - BUILD THE WIREGUARD MAIL TUNNEL

Mode B uses a point-to-point tunnel. The home Pi initiates it, so CGNAT and a changing residential address do not prevent the VPS from replying through the established mapping.

Install WireGuard on the VPS and Pi:

```
sudo apt update
sudo apt install -y wireguard
```

Generate keys separately on each host:

```
sudo -i
umask 077
wg genkey | tee /etc/wireguard/private.key | wg pubkey >
/etc/wireguard/public.key
exit
```

Never copy a private key into chat, a screenshot, the zine, or Git.

VPS /ETC/WIREGUARD/WG0.CONF

```
[Interface]
Address = 10.77.0.1/24
ListenPort = 51820
```

```
PrivateKey = <VPS_PRIVATE_KEY>
[Peer]
PublicKey = <PI_PUBLIC_KEY>
AllowedIPs = 10.77.0.2/32
```

PI /ETC/WIREGUARD/WG0.CONF

```
[Interface]
Address = 10.77.0.2/24
PrivateKey = <PI_PRIVATE_KEY>
[Peer]
PublicKey = <VPS_PUBLIC_KEY>
Endpoint = mx1.packet.init.wtf:51820
AllowedIPs = 10.77.0.1/32
PersistentKeepalive = 25
```

PersistentKeepalive = 25 keeps the home NAT mapping alive. The No-IP hostname is not required for this tunnel because the stable VPS is the endpoint and the Pi initiates the connection.

Protect and start the interface on both hosts:

```
sudo chmod 600 /etc/wireguard/wg0.conf
/etc/wireguard/private.key
sudo systemctl enable --now wg-quick@wg0
sudo wg show
```

Test from the Pi:

```
ping -c 3 10.77.0.1
```

Test from the VPS:

```
ping -c 3 10.77.0.2
```

Permit UDP 51820 on the VPS firewall. Do not forward UDP 51820 on the home router for this topology; the Pi is the initiating peer.

PHASE 7 - CONFIGURE THE VPS AS A NARROW SMTP EDGE

This Postfix edge accepts mail only for the listed lab recipients and accepts outbound relay only from the Pi's WireGuard address.

Install:

```
sudo apt update
sudo apt install -y postfix postfix-pcre certbot swaks
```

Choose "Internet Site" and enter mx1.packet.init.wtf when the package asks.

Add or reconcile these settings in /etc/postfix/main.cf:

```
myhostname = mx1.packet.init.wtf
myorigin = $myhostname
inet interfaces = all
inet protocols = ipv4
mydestination = localhost
mynetworks = 127.0.0.0/8 10.77.0.2/32
relay domains = packet.init.wtf
relay_recipient_maps = hash:/etc/postfix/relay_recipients
transport_maps = hash:/etc/postfix/transport
smtpd_relay_restrictions = permit_mynetworks,
reject_unauth_destination
smtpd_recipient_restrictions = permit_mynetworks,
reject_unauth_destination
smtp_send_xforward_command = yes
smtpd_tls_security_level = may
smtp_tls_security_level = may
```

List only real recipients in /etc/postfix/relay_recipients:

```
packetbridge@packet.init.wtf OK
red@packet.init.wtf OK
green@packet.init.wtf OK
dmarc@packet.init.wtf OK
```


FIELD MANUAL 02 // 05 OF 08

Route the domain through WireGuard in /etc/postfix/transport:

```
packet.init.wtf smtp:[10.77.0.2]:25
```

Compile the maps and check syntax:

```
sudo postmap /etc/postfix/relay_recipients
sudo postmap /etc/postfix/transport
sudo postfix check
sudo systemctl restart postfix
sudo systemctl status postfix --no-pager
```

Acquire a public certificate for mx1.packet.init.wtf using the appropriate Certbot HTTP or DNS method, then set:

```
smtpd_tls_cert_file =
/etc/letsencrypt/live/mx1.packet.init.wtf/fullchain.pem
smtpd_tls_key_file =
/etc/letsencrypt/live/mx1.packet.init.wtf/privkey.pem
```

Reload Postfix after the certificate exists:

```
sudo postfix check
sudo systemctl reload postfix
```

At the VPS firewall allow TCP 25 and UDP 51820. TCP 80 is needed if Certbot uses HTTP-01. Do not open RabbitMQ ports on the VPS.

In Mailcow, add 10.77.0.1/32 under Configuration & Details -> Forwarding Hosts so its Postfix instance can accept XFORWARD metadata only from the WireGuard edge. Then inspect received headers and spam-filter behavior. Trusting a forwarding host changes where filtering responsibility sits; do not assume the thin VPS edge provides the same protection as a mature filtering gateway.

PROVE IT IS NOT AN OPEN RELAY

From a machine outside the VPS and outside the WireGuard tunnel:

```
swaks --server mx1.packet.init.wtf \
--from outsider@example.net \
--to unrelated-recipient@gmail.com
```

The VPS must reject relaying. Then test a known lab recipient:

```
swaks --server mx1.packet.init.wtf \
--from your-external-test-address@example.net \
--to red@packet.init.wtf
```

The second test should be accepted by the edge and delivered to Mailcow through 10.77.0.2. If the first test succeeds, stop Postfix immediately and fix the relay restrictions.

SEND MAILCOW OUTBOUND THROUGH THE EDGE

In Mailcow's admin UI:

1. Open Configuration & Details -> Routing.
2. Add [10.77.0.1]:25 as a relayhost/smarthost.
3. No SMTP password is needed because WireGuard authenticates the peer and Postfix trusts only 10.77.0.2/32.
4. Assign that relayhost to the packet.init.wtf domain.
5. Run Mailcow's relayhost test.

Docker normally masquerades traffic leaving its bridge, so the VPS should see the connection as 10.77.0.2. Confirm in the VPS mail log. If it sees another source address, do not widen mynetworks; correct the Pi's Docker-to-WireGuard routing instead.

PHASE 8 - INSTALL RABBITMQ AND THE BRIDGE

The accompanying packetmail-bridge code bundle contains:

- + .env.example - secret-free configuration template
- + Dockerfile and compose.yaml - isolated runtime
- + package.json and package-lock.json - pinned Node dependencies
- + src/bridge.mjs and src/lib.mjs - bridge and validation logic
- + test/lib.test.mjs - deterministic unit tests
- + LICENSE - MIT terms for the original bridge code

Copy it to the Pi as /opt/packetmail-bridge. The source is deliberately outside /opt/mailcow-dockerized; Mailcow upgrades own their directory and our bridge owns its own.

Create the runtime environment:

```
cd /opt/packetmail-bridge
sudo cp .env.example .env
sudo chmod 600 .env
openssl rand -hex 24
openssl rand -hex 32
sudo nano .env
```

Use the first generated value for both RABBITMQ_PASSWORD and the password inside RABBITMQ_URL. Hex characters are URL-safe. Use the second for PACKET_HMAC_SECRET.

Fill in the Mailcow credentials for packetbridge@packet.init.wtf. The example environment deliberately limits outbound mail to packet.init.wtf:

```
OUTBOUND_ALLOWED_DOMAINS=packet.init.wtf
```

Do not widen that list until abuse controls, rate limits, logs, and the public-mail path have been tested.

Build and test the code:

```
cd /opt/packetmail-bridge
sudo docker build --target test -t packetmail-bridge:test .
```

Start the services:

```
sudo docker compose build
sudo docker compose up -d
sudo docker compose ps
sudo docker compose logs --tail 100 bridge
```

The bundle pins RabbitMQ 4.3.4-management, the current release when this manual was drafted. Before deploying later, compare the pinned version with RabbitMQ's supported releases and test upgrades deliberately.

WHAT THE BRIDGE DECLARES

```
| Exchange/queue | Type | Purpose |
| packetmail | topic exchange | Normal routing |
| packetmail.dlx | topic exchange | Failure routing |
| node.red.inbox | durable queue | Red node mailbox |
| node.green.inbox | durable queue | Green node mailbox |
| packetmail.outbound | durable queue | Requests to send mail |
| packetmail.retry.60s | delayed queue | Finite outbound retries |
| packetmail.dead | durable queue | Final failures and rejected mail |
```

Inbound messages have a seven-day queue TTL and a 10,000-envelope cap. Raw messages larger than 1 MiB go to the Dead Letter Office. Parsed text is limited to 16 KiB. Attachments remain in Mailcow; RabbitMQ receives only attachment metadata.

MANAGEMENT WITHOUT PUBLIC EXPOSURE

RabbitMQ management is bound to loopback on the Pi. Reach it through SSH:

```
ssh -L 15672:127.0.0.1:15672 youradmin@192.168.50.20
```

FIELD MANUAL 02 // 06 OF 08

Then open `http://127.0.0.1:15672`. Never forward 15672 or 5672 from the home router.

For a hardened revision, create a separate administrator account and remove administrator tags from the bridge account. The first lab keeps one loopback-only account to reduce bootstrap complexity.

PHASE 9 - UNDERSTAND THE ENVELOPE

Email to `red@packet.init.wtf` becomes:

```
{
  "version": 1,
  "packet id": "c4c6f6ecfddb3a5fd02a863301ff32f",
  "source": "sender@example.net",
  "destination": "node.red",
  "created at": "2026-08-11T16:00:00.000Z",
  "received at": "2026-08-11T16:00:02.000Z",
  "message id": "<original-message-id@example.net>",
  "subject": "TRANSMISSION 0042",
  "text": "The compact plain-text body.",
  "raw sha256": "...",
  "raw bytes": 4812,
  "imap uid": 184,
  "attachments": [],
  "signature": "..."
}
```

The deterministic packet ID combines the original message identity, raw-message hash, and destination. Consumers can store processed packet IDs and safely ignore redeliveries.

The HMAC detects accidental or unauthorized changes inside the trusted packet network. It does not encrypt the envelope. Keep RabbitMQ on the host network boundary or a VPN and add AMQP TLS before separating nodes across untrusted links.

Nothing in an email body is executed. The bridge emits data, not shell commands.

PHASE 10 - TEST THE WHOLE CHAIN

TEST A: INFRASTRUCTURE

On the Pi:

```
sudo docker ps
sudo docker compose -f
/opt/mailcow-dockerized/docker-compose.vml ps
sudo docker compose -f /opt/packetmail-bridge/compose.vml ps
sudo wg show
getent ahostsv4 initmail.ddns.net
```

On the VPS:

```
sudo wg show
sudo postfix check
sudo postqueue -p
sudo journalctl -u postfix -n 100 --no-pager
```

TEST B: PUBLIC INBOUND

Send a plain-text email from an unrelated provider to `red@packet.init.wtf`.

Verify in order:

1. VPS Postfix accepted the known recipient.
2. Mailcow shows the original in the red mailbox or the dedicated bridge mailbox chosen by your addressing design.
3. Bridge logs show `inbound_processed`.
4. RabbitMQ shows one ready message in `node.red.inbox`.

The supplied bridge polls `packetbridge@packet.init.wtf`. The red and green Mailcow aliases deliver into that one source mailbox while the visible To header retains the node address used for RabbitMQ routing. This first bridge intentionally does not route BCC-only recipients because their destination is absent from the message headers.

TEST C: OFFLINE DELIVERY

Leave `node.red.inbox` without a consumer. Send three messages. Confirm the ready count reaches three. Start a consumer and acknowledge them one at a time. This is the project's central demonstration: the node can be absent without losing its mail.

TEST D: OUTBOUND

Publish this JSON to exchange `packetmail` with routing key `mail.outbound`:

```
{
  "to": "green@packet.init.wtf",
  "subject": "RETURN PATH TEST",
  "text": "The rabbit has left the sorting office."
}
```

The bridge submits it to Mailcow over authenticated port 587. Mailcow signs it and sends it through the VPS relayhost. The outbound worker retries twice through the 60-second retry queue; the third failure goes to `packetmail.dead`.

TEST E: THE DEAD LETTER OFFICE

Perform three controlled failures:

1. Send to `unknown@packet.init.wtf` after temporarily adding it as a Mailcow alias but not adding it to `PACKET_NODES`.
2. Send a message larger than `MAX_RAW_BYTES`.
3. Publish an outbound request to a domain absent from `OUTBOUND_ALLOWED_DOMAINS`.

All three should be visible as dead-letter events without executing content or creating an infinite retry loop.

FAILURE ATLAS

NO-IP RESOLVES TO THE WRONG ADDRESS

- + Compare the No-IP dashboard with the router WAN address.
- + Inspect docker logs `noip-duc`.
- + Confirm the DDNS Key covers the selected hostname.
- + Do not update more frequently than the provider allows.
- + Remember that free hostname confirmation is separate from DUC updates.

EXTERNAL MAIL NEVER REACHES THE VPS

- + Verify MX and A records.
- + Verify PTR separately.
- + Confirm TCP 25 is allowed by the VPS provider and firewall.
- + Inspect Postfix logs while sending a test.
- + Confirm the sender is addressing `packet.init.wtf`, not `home.packet.init.wtf`.

VPS ACCEPTS MAIL BUT THE PI NEVER SEES IT

- + Check `wg show` for a recent handshake.
- + Ping `10.77.0.2` from the VPS.
- + Confirm Mailcow publishes TCP 25 on the Pi.
- + Check the Pi's DOCKER-USER rules and router/LAN firewall.

FIELD MANUAL 02 // 07 OF 08

- + Confirm /etc/postfix/transport was processed with postmap.

MAILCOW RECEIVES MAIL BUT RABBITMQ STAYS EMPTY

- + Confirm the message landed in packetbridge@packet.init.wtf.
- + Confirm the bridge IMAP password.
- + Confirm the To header still contains red@packet.init.wtf or green@packet.init.wtf.
- + Confirm PACKET_NODES=red,green.
- + Inspect bridge JSON logs; they never print passwords.

RABBITMQ RECEIVES DUPLICATE ENVELOPES

At-least-once processing permits redelivery when a process fails between publish confirmation and marking the IMAP message seen. Deduplicate by packet_id; do not pretend exactly-once delivery exists across IMAP and AMQP without a transactional boundary.

OUTBOUND MAIL LOOPS OR RETRIES FOREVER

The supplied worker does not requeue the same delivery immediately. It republishes into a 60-second retry queue and stops after the configured attempt limit. If a custom consumer calls nack(requeue=true) forever, it has removed one of the design's safety properties.

MAIL LANDS IN SPAM

- + Verify PTR, SPF, DKIM, and DMARC alignment.
- + Inspect the receiving provider's authentication results.
- + Check the VPS IP's reputation.
- + Send normal, low-volume test traffic.
- + Do not use the lab for bulk mail.

SECURITY DOCTRINE

- + Keep the experiment on a subdomain.
- + Never publish credentials, WireGuard private keys, DKIM private keys, or Mailcow encryption keys.
- + Do not expose RabbitMQ AMQP or management ports to the internet.
- + Do not execute attachments, HTML, URLs, headers, or body text.
- + Keep a hard message-size ceiling.
- + Allowlist outbound domains during the lab.
- + Reject unknown recipients at the VPS before accepting their message bodies.
- + Use finite retries and a visible dead-letter queue.
- + Treat every consumer as idempotent.
- + Use unique service accounts and two-factor authentication where available.
- + Keep the Pi and VPS patched.
- + Review logs without retaining more personal content than necessary.
- + Do not turn the system into an anonymous relay, bulk sender, or covert tunnel.

This machine processes correspondence. Even a playful lab inherits privacy obligations.

BACKUP AND RECOVERY

Mailcow stores mail, databases, spam-learning state, and encryption material in Docker volumes. Its encryption key material is essential; a backup without the keys is not a recovery plan.

Use Mailcow's supported helper:

```
cd /opt/mailcow-dockerized
sudo ./helper-scripts/backup_and_restore.sh backup all
--delete-days 14
```

Back up separately:

- + /opt/mailcow-dockerized/mailcow.conf
- + /opt/noip-duc/noip-duc.env
- + /etc/wireguard/
- + /opt/packetmail-bridge/.env
- + RabbitMQ durable data or exported definitions
- + VPS Postfix configuration and maps
- + DNS record inventory

Encrypt backup media. Test a restore on an isolated machine. A backup job that has never completed a restore test is only a theory.

MAINTENANCE CALENDAR**WEEKLY**

- + confirm the WireGuard handshake is recent
- + inspect Mailcow and bridge health
- + check the Dead Letter Office
- + check disk and backup capacity

MONTHLY

- + apply operating-system security updates
- + run Mailcow's supported update process
- + review the pinned RabbitMQ release against current support
- + verify SPF, DKIM, DMARC, MX, A, and PTR
- + test inbound and outbound delivery from an external provider
- + confirm the No-IP hostname if using the free tier
- + restore one small artifact from backup

AFTER EVERY TOPOLOGY CHANGE

- + repeat the open-relay test
- + repeat the unknown-recipient test
- + repeat the offline-node test
- + confirm no new Docker port is public

DEFINITION OF DONE

Do not label the article FIELD TESTED until every box is checked:

- + [] Pi boots from SSD after power loss
- + [] UPS shutdown or recovery behavior tested
- + [] No-IP changes propagate correctly
- + [] Mailcow certificate validates without warnings
- + [] External MX, A, and PTR agree
- + [] SPF passes
- + [] DKIM passes
- + [] DMARC report arrives

FIELD MANUAL 02 // 08 OF 08

- + [] VPS refuses unrelated relay
 - + [] WireGuard reconnects after home IP change
 - + [] Known recipient reaches Mailcow
 - + [] Mailcow message becomes a RabbitMQ envelope
 - + [] Offline node retains three messages
 - + [] Consumer deduplicates a replayed packet ID
 - + [] Oversized input reaches the Dead Letter Office
 - + [] Outbound failure stops after finite retries
 - + [] Mailcow backup completes
 - + [] Isolated restore succeeds
 - + [] Every photographed secret is redacted
- Until then, the correct cover stamp is LAB-READY DRAFT.

PRIMARY REFERENCES

- + Mailcow system requirements and ports
(<https://docs.mailcow.email/getstarted/prerequisite-system/>)
- + Mailcow installation
(<https://docs.mailcow.email/getstarted/install/>)
- + Mailcow DNS requirements
(<https://docs.mailcow.email/getstarted/prerequisite-dns/>)
- + Mailcow relayhosts (https://docs.mailcow.email/manual-guide/s/Postfix/u_e-postfix-relayhost/)
- + Mailcow external-forwarding example (https://docs.mailcow.email/third_party/exchange_onprem/third_party-exchange_onprem/)
- + Mailcow DNS-01 certificates
(https://docs.mailcow.email/post_installation/firststeps-ssl-dns/)
- + Mailcow backups
(https://docs.mailcow.email/backup_restore/b_n_r-backup/)
- + No-IP Linux Dynamic Update Client (<https://www.noip.com/support/knowledgebase/install-linux-3-x-dynamic-update-client-duc>)
- + No-IP DUC Docker deployment (<https://www.noip.com/support/knowledgebase/ip-linux-duc-docker>)
- + No-IP free-hostname confirmation (<https://www.noip.com/support/knowledgebase/confirm-my-hostname-free-account-support-question-day>)
- + WireGuard quick start and persistent keepalive
(<https://www.wireguard.com/quickstart/>)
- + RabbitMQ current installation guidance
(<https://www.rabbitmq.com/docs/download>)
- + RabbitMQ publisher confirms and consumer acknowledgements
(<https://www.rabbitmq.com/docs/confirm>)
- + RabbitMQ dead-letter exchanges
(<https://www.rabbitmq.com/docs/dlx>)
- + Docker Engine on Ubuntu
(<https://docs.docker.com/engine/install/ubuntu/>)

END OF LINE // INIT.WTF // FIELD MANUAL 02

PACKETMAIL FIELD GATE

PROVE THE FAILURE PATH

If every demonstration succeeds, you tested the happy path. The Dead Letter Office exists for the part that refuses.

EXPECTED FAILURES

Malformed envelope. Oversized body. Invalid signature. Unknown route. Offline consumer. Replayed packet ID. Exhausted retry budget.

PASS CONDITION

The original remains in IMAP, the event is observable, retry is finite, poison messages stop circulating, and the operator can explain where the message went.

FAILURE THAT CANNOT BE LOCATED BECOMES FOLKLORE.

- ☐ Malformed envelope reaches packetmail.dead.
- ☐ Oversized content is rejected without execution.
- ☐ Invalid signature records a reason.
- ☐ Offline node retains messages durably.
- ☐ Replay is deduplicated by packet ID.
- ☐ Retry stops at the configured limit.

TEST DATE	BUILD	OBSERVER	RESULT	FOLLOW-UP

AXIOM / RECOVERED CLOSING SIGNAL

BY THE SAME COVENANT...

A system is most honest at the boundary where belief becomes behavior.

The original BTM closed by moving from cypherpunk architecture back into magical language. The transition belongs here: protocols and rituals both define actors, states, permissions, transitions, failures, and a declared end condition. That resemblance does not prove the supernatural. It does explain why operators reach for symbolic language when ordinary documentation cannot hold the emotional weight of a transformation.



GLOSSARY / EDITORIALY CORRECTED

FOUR WORDS AT THE EDGE OF THE SYSTEM

Beramythas

01 // BERAMYTHAS

A working or personal formula in the BTM draft. We found no reliable basis for calling it a standard word from Crowley's Book of the Law; preserve it as the author's own signal unless provenance is supplied.

Abrahadabrah!

02 // ABRAHADABRA

A central Thelemic formula used by Aleister Crowley, often associated with completion of the Great Work. Its eleven-letter construction is part of its symbolic treatment.

IO Choronzon!

03 // CHORONZON

A figure associated in Thelemic writing with the Abyss, dispersion, and ordeal. Choronzon is not the Holy Guardian Angel; those ideas occupy different roles in the system.

IO KIA!

04 // KIA

In Austin Osman Spare's vocabulary, Kia points toward an unconditioned self or universal consciousness. BTM uses 'IO KIA' as a closing acclamation: manifest chaos; the operation is done.

ISSUE 02 / INTAKE FORM

ROUTE SOMETHING BACK

The next issue needs evidence from the field: a machine, a manual, a failure, a diagram, a correction, a textfile, or a story worth preserving.

GOOD TRANSMISSIONS

Include provenance, the question you asked, what you tested, what failed, what changed your mind, and what another reader can verify.

DO NOT SEND

Live credentials, private keys, personal records, material you do not have permission to publish, or instructions that depend on harming a stranger.

TITLE / HANDLE / RETURN ROUTE

WHAT IS THE SIGNAL?

RIGHTS / REDACTIONS / VERIFICATION NOTE

DOT-MATRIX WORKSHEET

THREAT MODEL BEFORE TOOL

01 ASSET What are you protecting?

02 ACTOR From whom?

03 CAPABILITY What can they actually do?

04 IMPACT What does failure cost?

05 MAINTENANCE What can you sustain?

06 RECOVERY How do you come back?

PRACTICAL, BORING, SUSTAINABLE

FIVE LAYERS OF PRIVACY

01 DEVICE

Use a passcode, automatic updates, encrypted storage, short lock timeout, and tested backups. Lost hardware is a common threat model.

02 IDENTITY

Use unique passwords, a password manager, multi-factor authentication, and separate public, personal, and recovery identities where practical.

03 TRANSPORT

Prefer encrypted services. Understand what a VPN, Tor, DNS settings, and private relays can and cannot hide. Transport privacy does not erase account identity.

04 CONTENT

Minimize unnecessary collection. Encrypt sensitive files. Share the smallest useful amount. Metadata can remain revealing even when content is encrypted.

05 RECOVERY

Keep offline recovery codes, a second factor, tested backups, and instructions for trusted people. Security that locks out its owner has failed.

Privacy is a practice, not a product SKU.

This page is general education, not a guarantee of anonymity or security.

FIVE TEXTS / FIVE DOORS

THE KERNEL-SHADOW SHELF

01 THE ART OF INTRUSION

KEVIN D. MITNICK AND WILLIAM L. SIMON

Real stories from the perimeter-collapse era: human assumptions, workflow gaps, and corporate blind spots.

02 THE ART OF INVISIBILITY

KEVIN MITNICK WITH ROBERT VAMOSI

A privacy field guide to metadata trails, identity exposure, and everyday discipline.

03 THE ART OF DECEPTION

KEVIN D. MITNICK AND WILLIAM L. SIMON

The human element: trust, authority, urgency, and error before the firewall.

04 NECRONOMICON

H. P. LOVECRAFT; EDITED BY STEPHEN JONES

Forbidden texts, impossible geometries, and knowledge as a door.

05 THE ROOTKIT ARSENAL

BILL BLUNDEN

Operating-system internals, concealment, and the defensive lesson that unseen code can still own a machine.

A MASK, A MEMORY, A WARNING

IN CODE WE TRUST

Fiat Judge grew from the memory of early hacker rooms where aliases, masks, and deliberate anonymity were not only theater. They were armor, privacy, and survival.

The piece joins that older instinct with Bitcoin's Merkle tree: branches, proofs, hashes, roots, judgment, and trust. The mask watches from the ledger. Bitcoin-orange eyes and block-code textures turn the face into a false-positive camouflage pattern.

Stay hidden. Stay sharp. Let the code speak before the name does.

The project also asks what happens when underground symbols become platform content. A mask once protected a person. Online, the same mask can become an asset, identity product, or algorithmic category. The work lives inside that tension.

MASK != IDENTITY
PROOF != PERSON
LEDGER != MEMORY

Archive route: init.wtf / Gamma collection: Fiat Judge



39 WORKS IN THE ARCHIVE

CUT SIGNAL / PASTED MEMORY

001 GHOST CITY	002 FEED	003 OCCULT
004 FEMME FATALE II	005 SERIAL	006 PINEAPPLE
007 OBEY	008 ZERO DAY	009 PLANNED OBSOLESCENCE
010 90S	011 HACK THE PLANET	012 NOTHING IS TRUE
013 CYPHER PUNK	014 THE MACHINE	015 EVERYTHING IS PERMITTED
016 FEMME FATALE	017 SMILE	018 HACK THE PLANET ORGS

GALLERY ROUTE: [HTTPS://INIT.WTF/GALLERY.HTML](https://init.wtf/gallery.html)
DADA IS NOT A STYLE. IT IS A PERMISSION ERROR IN FALSE ORDER.

OPEN ROUTES

SIGNALS BEYOND THE BOARD

01 // INIT.WTF

LOCAL JOURNAL / GALLERY / ARCHIVE

<https://init.wtf>

02 // SATSWIRE

BITCOIN / PRIVACY / SELF-SOVEREIGN READING

<https://news.init.wtf>

03 // SWARM

ANDROID-FIRST SWARM NODE

<https://swarm.init.wtf>

04 // JEDI PATH

TRAINING SIGNAL / PATH NODE

<https://jedi.init.wtf>

05 // OPERATOR SHELL

INTERACTIVE TERMINAL INDEX

init-operator-shell.jazzy-mite-5722.chatgpt.site

NEXT NODE: READER TRANSMISSIONS / THE DIG SUBMISSIONS

72-PAGE PRODUCTION EDITION / 2026-08-12

COLOPHON + SOURCE TRAIL

ISSUE MAP

BTM NATIVE EDITORIAL 12-35 / 62-63
COMIC INTERRUPTS 27 / 37 / 50
PBX FROM ZERO 43-49 / 51
PACKETMAIL MANUAL 52-61
WORKSHEETS + BACKMATTER 64-71
TOTAL PAGES 72
TRIM / BLEED 8.5x11 / 0.125

NEW BREATHING PAGES

OPERATOR KEY 04
SIGNAL BREAK 11
BITCOIN BENCH CARD 26
POLIS READER PROMPT 36
DIG INTAKE 42
PBX CUTOVER 49
DEAD LETTER TEST 61
ISSUE 02 TRANSMISSION 64

PRIMARY SOURCE TRAIL

SATOSHI NAKAMOTO - BITCOIN P2P E-CASH PAPER - 2008-10-31
SATOSHI NAKAMOTO - BITCOIN WHITEPAPER
TIMOTHY C. MAY - CRYPTO ANARCHIST MANIFESTO
ERIC HUGHES - A CYPHERPUNK'S MANIFESTO - 1993-03-09
HAL FINNEY - BITCOIN AND ME - 2013-03-19
MOSES B. COTSWORTH - THE RATIONAL ALMANAC / CALENDAR REFORM

PRODUCTION

Seventy-two pages form an eighteen-sheet signature at four pages per sheet. Every page carries trim, bleed, crop, and art boxes. Raster art retains source resolution; the printer should apply its required CMYK profile during preflight.

RIGHTS + SAFETY

Text and original layout copyright Michael A. Cohee / INIT.WTF. Use systems, services, domains, and equipment you own or are authorized to administer. Never publish credentials, private keys, recovery codes, or unredacted personal records.

**READ THE SOURCE.
VERIFY THE HASH.
FORK THE NARRATIVE.
QUESTION AUTHORITY.
DROP TO SHELL.
TRANSMIT ANYWAY.**

[HTTPS://INIT.WTF](https://init.wtf)

BOOT CLEAN. VERIFY THE SIGNAL.

