



fragcap

BRAND & DESIGN SYSTEM

Passive process-attributed network capture for games.

Captured signal. Preserved evidence.

See what your game is actually saying

Your machine is already exchanging thousands of packets a second while you play. fragcap makes them readable: which process, which endpoint, how many bytes, right now. It reads traffic and never writes it — no injection, no modification, no automation, nothing hidden.

That is the whole pitch. Some people open fragcap once out of curiosity and close it again; some find the protocol quirk that has been bugging them for months. Both are fine outcomes.

CATEGORY	Game network capture and analysis tooling
ROLE	Passive capture and decode. fragcap reads; it does not write
AUDIENCE	Curious players first, then modders, server operators, developers, researchers and tool authors. The technical floor is real; being paid to do this is not
DESCRIPTOR	Passive process-attributed network capture for games
PARENT	ShruggieTech — shared typography and dark-first discipline, independent identity

Captured signal. Preserved evidence.

The brand treats packets as observable material. Every row fragcap shows you actually crossed the wire, timestamped and attributed to a process, and it stays readable long after the session ends.

TRAIT	EXPRESSION	AVOID
Precise	Exact nouns, units, formats, commands	Vague claims
Observational	Observation, capture, decode, traces	Combat metaphors
Dry	Calm statements and sparse wit	Hype, performed excitement
Competent	Assumes technical ability	Condescending tutorials
Transparent	Names limitations, prerequisites and sharp edges	Implied magic

- Capture is passive and inspectable.
- Every row is a packet that actually happened.
- Unknowns are labelled as unknowns.

- Technical boundaries are stated before convenience claims.
- Terminology stays consistent across CLI, docs, schemas and interfaces.

Marks and lockups

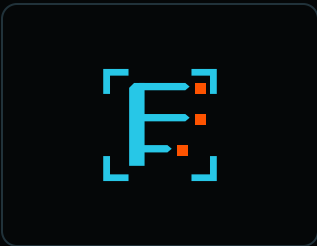
The mark carries three coordinated ideas: a triple-bladed F for parallel packet lanes; four separated corner segments that frame a capture without resorting to a weapon-like crosshair; and orange terminals for the bytes isolated by capture. The lower lane is deliberately shorter so the silhouette reads as F rather than E.



HORIZONTAL LOCKUP – HEADERS, REPOSITORY ARTWORK, WIDE SURFACES



STACKED · 96 PX



MARK · 24 PX



LIGHT SURFACE



REDUCED · ≤ 32 PX

CLEAR SPACE

One orange terminal, all the way round

The dashed boundary below is the minimum. No text, border, icon or crop may enter it.



MINIMUM SIZE

ASSET	MIN WIDTH
Mark	24 px
Horizontal lockup	160 px
Stacked lockup	96 px
Wordmark	120 px

At and below **32 px** the four reticle corners and three terminals collapse into noise. Small icons use the reduced mark — the F and its terminals — from the favicon exports. Never rasterise a large lockup down to browser-icon size at runtime.

The wordmark is custom vector lettering shipped as **filled outlines**. It does not depend on stroke rendering, it fills with `currentColor`, and it survives expand-to-outlines workflows. Never set it in live text or a substitute typeface.

Backgrounds and prohibited treatments

The primary presentation is Signal Cyan and Capture Orange on Void. The light variant uses the deeper accessible values on Light Surface. Single-ink white and black variants exist for reproduction that supports one ink only.



VOID · PRIMARY



LIGHT SURFACE



WHITE · SINGLE INK



BLACK · SINGLE INK

DO

- Use SVG wherever the surface supports it; PNG for social platforms and raster-only systems.
- Keep the Void tile in small icon exports — it protects the reticle corners and terminals against light and dark browser chrome alike.
- Treat logos/svg/ as the source of truth.
- Give every lockup its full clear space before anything else is placed.
- Pair every colour-coded state with text or an icon.

DO NOT

- Rotate, skew, stretch, outline, bevel or add glow.
- Recolour individual packet lanes.
- Move the orange terminals or make them appear explosive.
- Close the four reticle corners into a box or circle.
- Set the wordmark in live text or a substitute typeface.
- Place the logo over busy imagery.
- Combine the fragcap and ShruggieTech marks into one lockup.
- Add skulls, weapons, controllers, shields or exploit imagery.

WHY THE LAST RULE MATTERS

Reading game traffic is where protocol reverse-engineering starts, and protocol knowledge is how cheats get written. That road exists, and fragcap stands at one end of it. Anyone technical enough to run this works that out in about a minute, so the brand does not pretend otherwise. fragcap answers with scope: it captures and decodes, and it will not inject, modify, automate or hide. The project draws that line; the technology does not enforce it.

The visual rules follow. Moderators, storefronts and anti-cheat vendors triage on appearance long before they read a manifest, and a skull or a crosshair answers the question for them.

COLOUR

Palette

Dark-first and close to monochrome. Cyan is the observed signal. Orange is the captured terminal or a state requiring attention. Aim for roughly **80%** neutral surfaces, **15%** cyan and under **5%** orange in a typical view. Orange is not a general call-to-action colour.

ACCENTS — DARK SURFACES

Signal Cyan #27C7E7 Identity, links, focus, active data 10.01:1 on Void	Capture Orange #FF5300 Captured terminal, warning (scarce) 6.24:1 on Void	Fault #E9505F Failed capture, hard error 5.55:1 on Void
---	---	---

NEUTRALS — DARK SURFACES

Void #050708 Primary background	Surface #0B1115 Panels, code	Surface Raised #101A20 Selected rows	Line #21323A Borders, rails	Text #F2F7F8 18.68:1
---	--	--	---	--

LIGHT READING SURFACE

Text Muted #94A8B0 8.16:1 on Void	Light Surface #F5F8F9 Reading background	Light Text #102027 15.64:1	Light Cyan #006F82 5.47:1	Light Orange #C24100 4.86:1
---	--	--	---	---

THE ONE HARD RULE

Signal Cyan measures **1.89:1** on Light Surface. Never set it as text on a pale background — use Light Cyan. Capture Orange needs Light Orange, and Fault needs Fault Deep #C0293A (5.44:1).

SEMANTIC USE

COLOUR	MEANS
Signal Cyan	Selection, capture readiness, active filters, links, focus.
Capture Orange	Captured boundaries, warnings, dropped data, a state needing inspection.
Fault	A capture that failed. Always with explicit text and an icon.

FAULT IS NEW IN 1.1.0

Version 1.0.0 had no failure colour at all: success was cyan, orange covered warnings, and "this failed" had nothing to use.

Capture Orange and Fault sit only about 25 degrees apart in hue, so they are not reliably separable under colour-vision deficiency. That is acceptable only because status never rests on colour alone here — every state ships a label or a shape too.

● CAPTURED

■ INSPECT

× FAILED

Display, reading, payload

Typography is shared selectively with ShruggieTech, which creates lineage through craft rather than through copied layouts or parent-brand colour.

FRAGCAP TYPOGRAPHY

Observe the traffic.

Passive process-attributed network capture for games.

HEX READABILITY TEST

00000000 45 00 00 3C 1C 46 40 00 40 06 B1 E6
00000010 AC 10 0A 63 AC 10 0A 0C C2 03 00 50

GLYPH DISAMBIGUATION

0 0 1 1 I 8 B 5 S 2 Z { } [] ()

Space Grotesk Medium 500 carries subheads

Geist Regular 400 carries reading and interface text at a line height near 1.65.

Geist Mono 400 carries packet payloads, code, labels and metadata.

FUNCTION	TYPEFACE	WEIGHTS	BEHAVIOUR	VALUE
Display, headings	Space Grotesk	500, 700	Display tracking	-0.025em
Body, interface	Geist	400, 500	Label tracking (mono)	0.08em
Packet data, code	Geist Mono	400	Body line height	1.65
Wordmark	Custom vector	Filled outlines	Code line height	1.5

The scale, and the rules that do not bend

Sizes are carried as tokens in `tokens/typography.css`, so a heading level is chosen by name rather than by typing a number into a stylesheet.

TOKEN	SIZE	SPECIMEN
<code>--fc-text-display-lg</code>	52	Title pages
<code>--fc-text-display-md</code>	40	Page heads
<code>--fc-text-display-sm</code>	28	Section heads
<code>--fc-text-heading</code>	20	Subheads
<code>--fc-text-body-md</code>	16	Default reading size
<code>--fc-text-body-sm</code>	14	Dense tables and captions
<code>--fc-text-body-xs</code>	12	EYEBROWS AND METADATA

THE MONOSPACE DECISION

Geist Mono keeps 0, 1, I, 8, B, 5, S and 2 Z distinguishable at the actual interface size, which is what matters in a payload. Its family relationship to Geist also stops technical specimens from feeling detached from the documentation around them.

Evaluate any future replacement against the real-format specimen in `specimens/`, never a decorative alphabet.

NON-NEGOTIABLE

- Hex dumps never use ligatures.
`tokens/base.css` disables them on `code`, `pre`, `kbd` and `samp`.
- Never apply smart quotes or automatic character substitution to payloads, commands or file paths.
- Sentence case is the default. Uppercase is reserved for compact labels, capture states and table metadata.
- Geist has no 700 and Geist Mono has no bold. Asking for a weight that is not shipped makes the renderer synthesise a faux bold, which prints badly and forces outlined glyphs into exported PDFs.

Instrumentation

Visual references come from oscilloscopes, protocol analysers, packet lanes, capture gates, timing marks and network topology. Use exact alignment and quiet negative space. Every graphic should do a job: explain a state, or explain a relationship.

SIGNATURE COMPONENT – CAPTUREROW

One observed packet: direction, the endpoint pair, the attributed process, byte count, and a capture state. Endpoints are mono so columns align down a long trace. One endpoint may be tinted cyan to mark the active observation path. State is always a labelled badge, never a bare colour.

OUT	10.0.0.99:51820 → 172.16.10.12:27015	game.exe	1 314 B	CAPTURED
IN	172.16.10.12:27015 → 10.0.0.99:51820	game.exe	1 460 B	INSPECT
OUT	10.0.0.99:51821 → unknown	unattributed	0 B	FAILED

ICONOGRAPHY

Simple line icons, 1.5–2 px strokes, square or lightly chamfered terminals, minimal rounding. Prefer direct technical symbols: filter, file, clock, interface, endpoint, search. Avoid mascots, weapons, controllers, shields, hooded figures and generic circuit-board decoration.

SURFACES AND GEOMETRY

Panels use 1 px Line borders and radii between 4 px and 8 px. Avoid glass effects, large shadows, chrome and decorative gradients. Selected states may use a subtle cyan border or a low-opacity cyan fill. Orange backgrounds should be rare and small.

DATA VISUALISATION

Prefer lanes, traces, timelines, byte grids and topology diagrams. Cyan marks the active observation path. Orange marks the selected terminal, a discontinuity or a warning. Grey remains the default for context.

MOTION

Motion communicates capture state, filtering or continuity. Keep interface transitions between 120 ms and 240 ms on cubic-bezier(.2, .6, .2, 1). Avoid pulsing glow, glitch effects, screen shake and decorative scan lines. Always respect prefers-reduced-motion.

ICON SPECIMEN – NEW IN 1.1.0

Six starter icons ship in icons/: the exact symbols named above, on a 24 grid with a 1.5 stroke, butt caps and mitre joins. They stroke with `currentColor`, so an icon inherits whatever semantic token its container sets. Extend the set with direct technical symbols only.

					
FILTER	FILE	CLOCK	INTERFACE	ENDPOINT	SEARCH

Precise, dry, technically literate

Assume technical competence. Use active voice, concrete nouns and observable outcomes. State prerequisites before instructions. Distinguish supported behaviour, inferred behaviour and unknown behaviour.

Address the reader as someone playing a game on their own machine. "Your traffic" and "your game" beat "the operator" and "the end user" every time, and they cost nothing in precision. Where a capability has an obvious misuse, name it and say where the project's line falls — this audience will see what fragcap enables whether or not the copy admits it.

PREFERRED REGISTER

PREFER

your game, your machine, your traffic

capture, observe, inspect, decode

packet, frame, trace, session

supported, experimental, unknown

requires, emits, records, filters

AVOID

the operator, the end user

intercept, attack, exploit

traffic magic, secret data

flawless, revolutionary, effortless

unlocks, dominates, supercharges

WORKED EXAMPLES

PRODUCT STATEMENT

Passive process-attributed network capture for games.

PREREQUISITE

Packet capture requires an Npcap-compatible capture driver.

EMPTY STATE / ERROR

No packets matched the current filter. Capture stopped. Interface 'Ethernet 2' is gone.

REGISTER ON THE LANDING PAGE

fragcap does not use conventional marketing structure. State what the tool is, show one worked invocation with real output, name the prerequisite, link to documentation. No testimonials, no feature grids, no urgency. A curious player should get through it without looking anything up.

LIMITATION

Encrypted payloads remain opaque unless a supported decoder can derive the required session context.

CASING AND TERMINOLOGY

The product name is always lowercase: **fragcap**. Begin a sentence with fragcap rather than capitalising it. File formats, protocol names and command flags keep their canonical casing.

NEVER WRITE "X, NOT Y"

Evidence, not theatre. An instrument, not a weapon. Borrowed confidence, and the clearest tell of machine-written copy. Same for X over Y and rather than merely Z. State the thing you mean. **build/verify.py** fails the build when this pattern reaches brand copy.

A ShruggieTech project

fragcap is an independent ShruggieTech product identity. It shares Space Grotesk, Geist, Geist Mono, dark-first discipline and the parent's exact #FF5300 orange. It does not inherit ShruggieTech green, the shruggie mark, marketing layouts or verbal flourish.

The approved endorsement is A ShruggieTech project. Set it in Geist Mono, uppercase, with positive tracking. Keep it visually subordinate and outside the fragcap logo's clear space. It may appear in footers, About pages, repository metadata, title-page colophons and social previews. **Never combine both marks into one logo.**

A SHRUGGIETECH PROJECT

DIGITAL IMPLEMENTATION

Load the system

```
<link rel="stylesheet" href="styles.css">
<link rel="stylesheet" href="components/components.css">
```

`styles.css` is the single entry point — it pulls in the fonts, all four token files and the base element styles. The default surface is dark; add `class="fc-light"` to any container to switch that subtree to the light reading surface and the deeper accent values. No second stylesheet, no component changes.

FILE	PROVIDES
<code>tokens/colors.css</code>	Brand colours, transparent fills, semantic tokens, <code>.fc-light</code>
<code>tokens/typography.css</code>	Families, weights, display and body scales, tracking, leading
<code>tokens/spacing.css</code>	Spacing scale, radii, stroke widths, shadows, motion, layout
<code>tokens/base.css</code>	Element resets, focus ring, ligature suppression, reduced motion

ACCESSIBILITY

- Never use bright Signal Cyan as body text on a light surface — it measures 1.89:1. Use Light Cyan.
- Never communicate capture status through colour alone. Pair it with text or a glyph.
- Every interactive element needs a visible 2 px focus ring and a non-colour state change.
- Interface text should meet WCAG AA contrast at its rendered size.
- Respect `prefers-reduced-motion` on every transition and animation.
- Favicons ship the reduced mark at and below 32 px, so the identity is still readable at browser-tab size.

What ships in the kit

Treat the files in `logos/svg/` as the source of truth. Use SVG wherever the surface supports it; use PNG for social platforms, raster-only systems and external listings.

PATH	CONTENTS
<code>logos/svg/</code>	Vector masters: every lockup, mark and wordmark variant, plus the social preview. All filled paths — no live text, no stroked lettering
<code>logos/png/</code>	High-resolution raster exports and the 1280×640 social preview
<code>favicons/</code>	SVG, multi-resolution ICO, browser PNGs, Apple and Android icons, manifest
<code>fonts/</code>	WOFF2, static TTF, CSS declarations and OFL licences
<code>tokens/</code>	CSS and JSON design tokens, with measured contrast ratios
<code>icons/</code>	Six starter line icons on a 24 grid, stroked with <code>currentColor</code>
<code>components/</code>	<code>components.css</code> plus React core and form components
<code>guidelines/index.html</code>	Visual system reference, rendered live from the system itself
<code>ui_kits/fragcap-web/</code>	Demo product page built entirely from the system
<code>specimens/</code>	Typography and hex-readability specimen
<code>styles.css</code>	Single CSS entry point
<code>README.md</code>	The complete written manual
<code>SKILL.md</code>	Makes the kit invocable as a skill
<code>VERIFY.md</code>	Generated dimensions, ICO entries and SHA-256 checksums



A SHRUGGIETECH PROJECT · BRAND SYSTEM 1.1.0 · FRAGCAP.COM