THREAT HUNTER RESEARCH · IDENTITY & ADVERSARIAL FRAUD INTELLIGENCE

The Compromised Capture Layer: How "Virtual-Camera-as-a-Service" Defeats Biometric eKYC & Video Verification

Binary disassembly of three real-world virtual-camera toolchains—spanning host-app hooking, iOS camera-daemon shared GPU memory injection, and zero-malware WebRTC conferencing SaaS—and why ISO/IEC 30107-3 PAD compliance leaves the injection layer exposed.

> **Core Finding:** Modern identity fraud operators no longer try to fool a liveness model in front of the lens. By replacing camera streams inside the software capture pipeline, they remove physical sensor integrity from the equation—forcing downstream biometric and video-verification systems to **honestly evaluate a lie**.

LAYER 01 · HOST-APP HOOK

62 ObjC Hooks + 5-Layer Spoof

Rebuilds BGRA/420v/420f frames with copied PTS & EXIF while spoofing device IDs, WiFi, GPS, proxy, and screenshot checks.

LAYER 02 · OS DAEMON IPC

mediaserverd + GLSL Relighting

iOS 17/18 zero-copy IOSurface daemon pairs Vision face-ROI physical lighting synthesis with fake IMU motion and audio bridging.

LAYER 03 · WEBRTC SAAS

9 Permissions, Zero Malware

Clean conferencing APK & web app gated by a server `virtualCam` flag; 30-day operator seats impersonate bank officers live.

1. Executive Summary

Financial institutions, payment networks, digital wallets, and regulated platforms have spent a decade treating biometric liveness and live video verification as the final line of identity defense. That architecture rests on a single, rarely audited assumption: **the video frames entering the verification pipeline came from a physical camera sensor.**

Threat Hunter's binary reverse engineering and controlled sandbox analysis of three real-world virtual-camera toolchains show that this assumption no longer holds. Rather than holding photos, screens, or silicone masks in front of a lens, fraud operators now replace the capture source itself—feeding pre-recorded videos or live remote streams directly into software buffers. The liveness model does not malfunction; **it honestly evaluates fabricated inputs.**

3 Layers
CAPTURE HIJACK STACK
Host-app ObjC hooks, OS camera-daemon IOSurface IPC, and zero-malware WebRTC tracks.

62 Hooks
SINGLE-TWEAK SURFACE
One free iOS dylib hooks 62 methods across AVFoundation + 5 layers of OS device identity.

0 / 9
PAD → ISO 24745 REUSE
Zero standard PAD deliverables directly prove ISO/IEC 24745 capture-to-server integrity.

9 Perms
ZERO-MALWARE SAAS
Scam video-call APK needs no root, overlay, or accessibility—malice lives in server config.

Research Scope & Evidence Boundaries

This report synthesizes: (1) full static binary disassembly of two commercial/gray-market iOS virtual-camera toolchains (`FakeTools` v5.5 and a `LordVCAM 3.0.47` RootHide reseller variant, including Mach-O symbols, Objective-C selectors, embedded GLSL shaders, and install scripts); (2) static decompilation and controlled sandbox execution of a dual-platform Android/Web video-impersonation toolkit (`Tianqiong` v1.3.2); and (3) formal compliance gap mapping across `ISO/IEC 30107-3:2023` , `19795-1:2021` , and `24745:2022` . All customer names, internal references, and live malicious endpoints have been sanitized. Live bypass pass rates against specific commercial liveness SDKs were outside static/sandbox scope and are intentionally not extrapolated.

2. From Presentation Attacks to Capture-Layer Injection

Traditional biometric anti-spoofing defends against **Presentation Attacks (PAs)**: photos, replayed screens, or 3D masks presented *in front of a real camera*. Because optical light still travels through a lens and hardware image signal processor (ISP), Presentation Attack Detection (PAD) models can detect moiré interference, screen glare, depth flatness, and sensor texture anomalies.

Digital Injection Attacks remove the physical world from the pipeline. By intercepting frames inside the OS camera daemon, host application runtime, or WebRTC media layer, attackers supply synthetic video carrying authentic pixel formats (`BGRA` , `420v` , `420f`), cloned timestamps (`PTS` / `DTS`), and copied hardware EXIF/TIFF metadata. Worse, camera hijacking is now co-packaged with device spoofing, sensor synthesis, and map-matched GPS—blinding device-risk and location engines simultaneously.

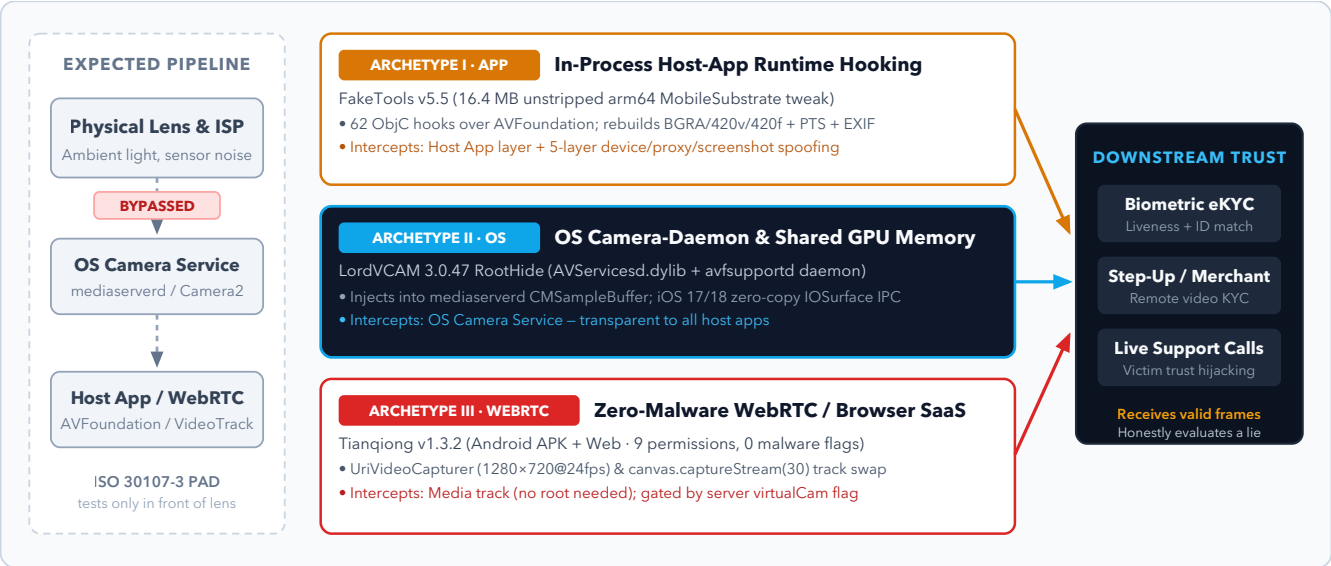


Figure 1. The Three-Layer Virtual-Camera Injection Architecture reconstructed from Threat Hunter binary disassembly and controlled sandbox analysis, contrasted against the physical-lens pipeline tested by ISO/IEC 30107-3 PAD evaluations.

Dimension	Archetype I: Host-App Hook (`FakeTools` v5.5)	Archetype II: OS Daemon Engine (`LordVCAM` 3.0.47 Variant)	Archetype III: WebRTC SaaS (`Tianqiong` v1.3.2)
Injection Layer	Host-app ObjC runtime (`AVFoundation` delegates)	OS daemon (`mediaserverd`) + `avfsupportd` `IOSurface` IPC (iOS 17/18)	App-level WebRTC (`UriVideoCapturer`) + Web `canvas.captureStream(30)`
OS & Privilege	iOS; requires jailbreak + `MobileSubstrate`	iOS (up to iOS 17/18); requires jailbreak (`RootHide` / `Dopamine`)	Android (`minSdk 30`) + Mobile Web; **no root or jailbreak required**
Frame Source	Local MP4 (`temp.mp4`) decoded via `AVAssetReader`	Local `.mov` loop slots **or** live PC WebSocket (`:8765`) H.264/YUV stream	Any local video via Android `OpenDocument` picker (muted loop, `1280×720@24fps`)
Anti-Liveness & Spoofing	Pixel/PTS/EXIF parity + 5-layer device ID, GPS, WiFi, proxy & screenshot spoofing	Vision face-ROI GLSL relighting + `CMDeviceMotion` synthesis + audio bridge + real-frame alpha blend	Looping visual impersonation paired with live human voice + room lock / waiting-room isolation
Commercial Model	Free unsigned tweak driving encrypted-messaging/web funnels	Metered subscription (PIX + USDT/BTC/LTC) + Secure Enclave lock + white-label reskin layer	Batch-provisioned 30-day operator seats gated via server `virtualCam` flag
Primary Defensive Seam	Hard jailbreak dependency; static loop fails randomized challenges	2–5 px facial-ROI lighting gradient step; synthetic IMU lacks hand tremor; daemon/IPC artifacts	Static loop behavioral cues (lip-sync drift, frozen room); Punycode IDN & WebRTC signatures

3. Binary Reverse Engineering of Three Real-World Archetypes

3.1 Archetype I – In-Process Host-App Hooking (`FakeTools` v5.5)

Static disassembly of a `16.40 MB` unencrypted, unstripped arm64 Mach-O library (`com.xuuz.faketools` , self-branded *Virtual Toolbox 5.5*) exposed **15,321 symbols**, **91 classes**, and **2,034 methods** alongside embedded OpenSSL and FFmpegKit builds targeting the `iPhoneOS17.0` SDK. Upon injection via `CydiaSubstrate` , a single module initializer installs **62 global** `_MSHookMessageEx` **Objective-C hooks** across camera, identity, sensor, and UI frameworks.

- **Complete Three-Channel Camera Replacement:** Hooks `AVCaptureVideoPreviewLayer` , `AVCaptureSession` , `AVCaptureStillImageOutput` , `AVCapturePhotoOutput` , and `AVCaptureVideoDataOutput` (dynamically hooking every delegate's `captureOutput:didOutputSampleBuffer:`). Preview, still photography, and video recording are replaced uniformly.
- **Pixel, Timing & Metadata Parity:** Inside `GetFrame::getCurrentFrame::` , the tweak decodes `Documents/temp.mp4` via `AVAssetReader` into the exact pixel format requested by the host app (`BGRA` , `420v` , or `420f`), reconstructs a `CMSampleBuffer` with the original frame's `duration` / `PTS` / `DTS` , and copies `__Exif_` and `__TIFF_` attachments verbatim onto the synthetic buffer.

Co-Packaged Capability	Confirmed Static Implementation	Impact on Enterprise Controls
5-Layer Device Identity Spoofing	Hooks `uname` , `sysctl` / `sysctlbyname` , `MGCopyAnswer` (`HWModelStr`), `UIDevice` , and `NSBundle`	Forges hardware model (e.g., `iPhone16,2`), OS/app version, `IDFV` , and `IDFA` –breaking device fingerprints and denylist linkage
Network & Location Evasion	Hooks `CNCopyCurrentNetworkInfo` (`SSID` / `BSSID`), `CLLocation` , and `CFNetworkCopySystemProxySettings`	Forces proxy checks to return `HTTPEnable=false` while spoofing WiFi/GPS context, blinding client-side VPN detection
Capture-Audit Suppression	Forces `UIScreen.isCaptured` to `0` and swallows `UIApplication._handleScreenshot`	Eliminates client-side screenshot and screen-recording alerts during sensitive KYC or payment steps
Behavior Automation & Speed	Private `IOHIDEvent` touch synthesis (`PTFakeMetaTouch`), button-title auto-clicking, 4-mode clock scaling (`mach_absolute_time` rebind)	Automates promotional/order workflows and distorts client-side behavioral timing telemetry
Runtime UI & URL Rewriting	Config-driven (`tihuan.plist`) hooks on `UILabel` / `UITextView` and `NSURLSession`	Modifies on-screen copy, redirects URLs, or alters HTTP request headers at runtime

Operational Soft Spot: Archetype I strictly requires jailbroken iOS with Substrate injection and relies on linear pre-recorded playback. Randomized server-issued action challenges and hardware-backed `App Attest` challenge-response checks neutralize this entire class at low operational cost.

3.2 Archetype II – OS Camera-Daemon Injection & GLSL Relighting (`LordVCAM` 3.0.47 Variant)

Static analysis of a compressed `929.12 KB` RootHide package (`.deb` , bundle identifier `com.apple.avservicesd.roothide` , resold under the label "Tri-Color Camera" on top of the `LordVCAM 3.0.47` engine) reveals a modular **three-component system** designed to remain transparent to every application on the device, including on iOS 17 and iOS 18:

- **Core Engine (`AVServicesd.dylib` , `3.4 MB` , `13,419` strings):** Injects into `mediaserverd` , `UIKit` , and `WebKit.GPU` to handle system-wide `CMSampleBuffer` substitution, Vision face localization, Metal/CoreImage physical relighting, motion synthesis, audio bridging, and map-matched GPS.
- **iOS 17/18 Producer Daemon (`avfsupportd` , `977 KB`):** Persists as a `KeepAlive` LaunchDaemon disguised under `com.apple.*` naming (IPC cache `/var/tmp/com.apple.avfcache`). Decodes local or PC-streamed H.264/YUV frames (WebSocket `:8765`), writes frames zero-copy into `IOSurface` **shared GPU memory**, and notifies consumers via `CFNotification` (`com.apple.avfcache.framepulse`) with frame-staleness rejection (`WK REJECT: frame STALE`).
- **White-Label Reskin Adapter (`lordvcamroothide.dylib` , `168 KB`):** Intercepts licensing/billing UI `baseURL` memory slots via `MSHookMessageEx` to re-point payment and branding endpoints to a regional distributor—allowing multiple underground resellers to commercialize one shared core engine.

```
// Extracted from AVServicesd.dylib — Embedded Metal/CoreImage CIColorKernel k0 (Sanitized Excerpt)
kernel vec4 k0(__sample base, vec4 face, vec4 light, vec3 lightDir, float strength, float range, float
feather, ...) {
  vec2 uv = (destCoord() - face.xy) / max(face.zw, vec2(1.0));
  // ... (face-center normalization & linear-light sRGB conversion elided)
  // Soft elliptical cheek contour mask (avoids rectangular boundary box)
  float faceShape = 1.0 - pow(abs(centered.x/halfW), 2.35) - pow(abs(centered.y/halfH), 2.20);
  float mask      = ccSmooth(-0.15 - 0.55 * feather, 0.05, faceShape);
  // Linear-space Lambertian diffuse + specular highlight + Reinhard tone-mapping to sRGB
  vec3 diffuse    = bl * (ll * w);
  vec3 specular   = ll * ccSmooth(0.72, 1.0, Llum) * w * 0.60;
  vec3 outL       = bl + diffuse + specular;
  outL            = outL / (vec3(1.0) + max(vec3(0.0), outL - vec3(1.0)));
  return vec4(clamp(ccSRGB(outL), 0.0, 1.0), base.a);
}
```

To counter modern active and multi-modal liveness checks, the engine layers four additional modules onto the relit frame stream: **`AVSMotionSynthesizer`** (generating synthetic `CMAccelerometerData` and `CMGyroData` aligned with head movement), **`AVSAudioBridge`** (routing video audio into the microphone for read-aloud challenges), **`blendFrame:withFrame:alpha:`** (blending live physical pixels over virtual frames to preserve subtle micro-expressions), and **`forceMapMatching`** **GPS spoofing** (snapping fake coordinates onto real road networks to prevent "in-a-river" anomalies).

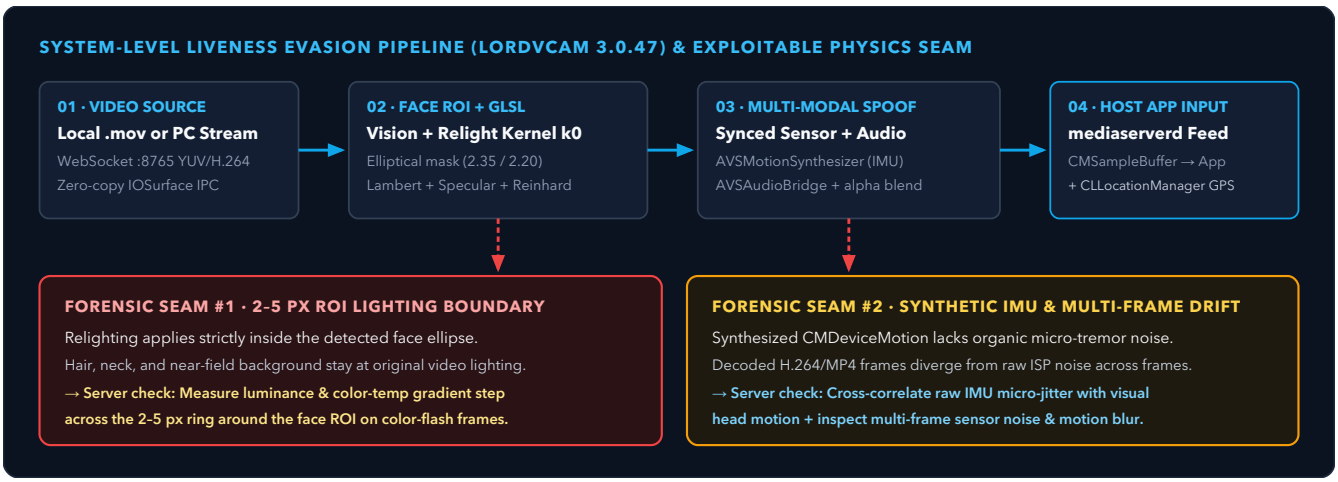


Figure 2. End-to-end system-level liveness evasion in `LordVCAM 3.0.47` and the two durable server-side forensic seams exposed by Threat Hunter's binary analysis: the 2–5 px facial-ROI relighting discontinuity and missing organic micro-tremor noise in synthesized IMU streams.

3.3 Archetype III – Zero-Malware WebRTC Impersonation SaaS (`Tianqiong` v1.3.2)

Archetype III demonstrates why mobile antivirus and permission-based risk scoring miss a rapidly growing class of social-engineering fraud infrastructure: **the malicious capability is engineered cleanly into business logic rather than hidden in malware.**

Static decompilation and controlled sandbox analysis of an Android conferencing client (`com.tianqiong.thadmin` , v1.3.2, `29.2 MB` unencrypted DEX with native WebRTC `libjingle_peerconnection_so.so`) paired with its companion web client confirmed:

- **Indistinguishable from Legitimate Conferencing Software:** Declares only **9 standard permissions** (network, camera, microphone, Bluetooth audio, foreground call service, notifications) and **4 components** (`LoginActivity` , `WorkActivity` , `CallActivity` , `CallService`). Zero accessibility abuse, zero screen-capture permissions, zero SMS/contacts access, zero obfuscation.
- **Dual-Platform Productized Virtual Camera:** On Android, `UriVideoCapturer` implements standard `org.webrtc.VideoCapturer` –selecting any local video via Android's `OpenDocument` picker, playing it muted in a continuous loop via `MediaPlayer` (`setLooping(true)` , `setVolume(0,0)`), closing the real camera, and injecting `1280×720@24fps` frames into `VideoTrack("th-vcam")` . The companion web client replicates this with an off-screen `<video loop muted>` element drawn into `canvas.captureStream(30)` .
- **Server-Gated 30-Day Operator Seats:** Feature visibility is toggled dynamically by a server account flag (`virtualCam` returned on `/auth/me`). Provisioning workflows support batch-creating operator seats (e.g., 10 accounts per batch with shared credentials and automatic **30-day expiration**). Victims join via mobile browser using a 7-digit room code–zero registration, zero app install–while meeting lock, waiting-room approval, and participant removal prevent family or bank fraud teams from intervening.
- **Modular Scam Chaining:** Administrative metadata explicitly references a separate parallel screen-sharing product, confirming operators split video impersonation and live screen viewing across clean single-purpose apps to stay below permission-anomaly thresholds.

4. The Underground Supply Chain: Freeware to Multi-Currency SaaS

Viewing these three samples side by side reveals a mature, stratified B2B cybercrime market where capability distribution mirrors commercial SaaS economics:

Market Tier	Representative Model	Monetization & Protection Architecture	Strategic Implication for Defenders
Tier 1: Free Acquisition Funnel	`FakeTools` v5.5 (iOS tweak)	Zero license enforcement; distributed free via web portals & encrypted messaging groups to build operator communities	Lowers entry barriers for opportunistic fraudsters; easy to detect via Substrate & App Attest controls
Tier 2: Multi-Region Subscription + White-Label Reselling	`LordVCAM` 3.0.47 engine + reskin adapters	Hardware-bound via **Secure Enclave ECDSA-P256** (`AVSAttestationProvider`); metered quotas; accepts **Brazil PIX + USDT/BTC/LTC**; `168 KB` reskin hook rewrites billing `baseURL` slots for regional resellers	Single-brand takedowns fail; tracking shared engine IPC/GLSL fingerprints and disrupting upstream licensing disables entire reseller trees
Tier 3: Turnkey Seat-Rental SaaS	`Tianqiong` v1.3.2 (Android + Web)	Batch-provisioned 30-day operator seats with per-seat `virtualCam` toggle, Punycode IDN domains, and 90-day free TLS certificates	Endpoint AV cannot flag the client; defense requires network/IDN heuristics, customer education, and randomized live verification

5. The eKYC Compliance Blind Spot: ISO/IEC 30107-3 vs. 19795-1 vs. 24745

Across banking, payments, and fintech governance committees, a dangerous assumption persists: *"Our eKYC vendor passed ISO/IEC 30107-3 Level 2 PAD certification, so we are covered against camera spoofing."* Threat Hunter's formal cross-standard audit demonstrates why PAD certification alone leaves digital injection unmeasured.

> **Critical Standard Boundary: Same Video, Different Governance Scope**
>
> When an adversary plays a recorded face video on a screen **in front of a physical camera**, the attack falls under **ISO/IEC 30107-3** (Presentation Attack Detection). When that *exact same video* is injected digitally into `AVFoundation` , `mediaserverd` , `IOSurface` , or WebRTC, it becomes a **sample substitution and replay threat** governed by **ISO/IEC 24745** (Biometric Information Protection). The two evaluations must be executed and recorded separately.

Dimension	ISO/IEC 30107-3:2023	ISO/IEC 19795-1:2021	ISO/IEC 24745:2022
Scope	Presentation attacks in front of the capture sensor	Normal operational biometric verification accuracy	Biometric information protection across capture, transit & storage
Required Evidence	Per-PAI-species `APCER`, separate `BPCER`, and non-response counts	`FTA`, genuine `FRR`, impostor `FAR`, statistical uncertainty, and first-attempt outcome retention on retries	Architecture-level proof of confidentiality, integrity, sample-substitution defense, and server-side anti-replay
Virtual-Camera Gap	Does **not** test OS-daemon or software-pipeline digital injection	PAD attack pass counts cannot be merged into normal `FAR`	Almost never evidenced by client-side PAD lab certificates alone

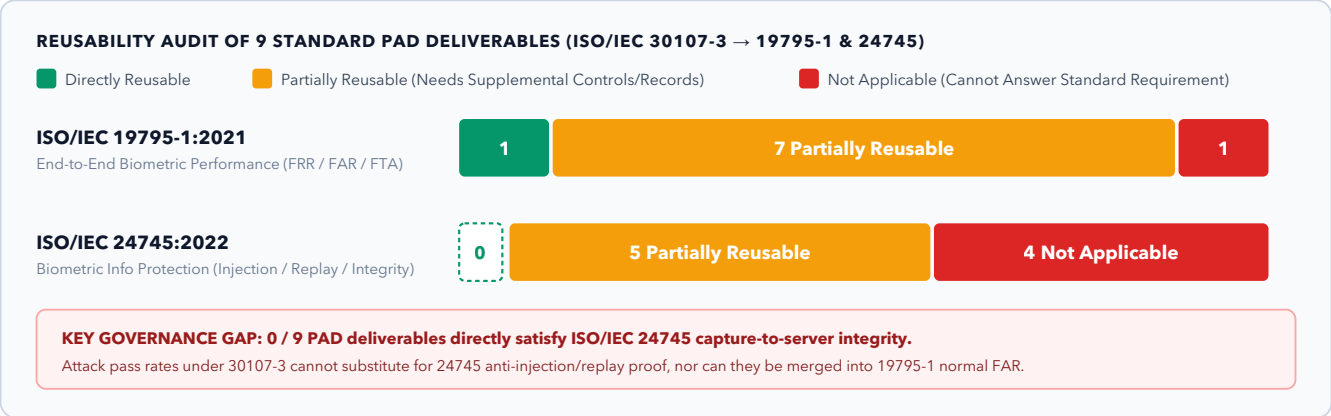


Figure 3. Threat Hunter's reusability audit of 9 standard PAD evaluation deliverables (subject counts, execution logs, PA test cases, metrics/denominators, metadata, statistical methods, efficiency logs, device configs, and personnel roles) mapped to `ISO/IEC 19795-1:2021` and `ISO/IEC 24745:2022`.

Compound governance risk also arises from **version drift**: organizations frequently author current procurement test plans referencing `ISO/IEC 30107-3:2023` while continuing to accept legacy vendor evaluation certificates issued under `ISO/IEC 30107-3:2017`—creating unexamined gaps in both attack-species classification and evidence boundaries.

Why 0 of 9 PAD Deliverables Directly Satisfy ISO/IEC 24745

Across the nine deliverables audited—(1) subject/sample counts, (2) capture execution logs, (3) PA test cases, (4) error metrics/denominators, (5) report metadata, (6) statistical methodology, (7) efficiency/latency logs, (8) device/environment configs, and (9) test-personnel roles—**four items** (subject counts, error rates, statistical methods, and latency) provide zero evidence of confidentiality, payload integrity, or anti-replay protection. The remaining **five** can only serve as partial context once supplemented with server-side capture-to-storage data-flow verification and injection/replay boundary controls.

6. Defense-in-Depth Framework: Migrating Trust from Client to Server

Once attackers can hook host apps, replace frames inside `mediaserverd`, or deliver zero-malware WebRTC apps, **no client-side check can serve as a standalone trust anchor.** Financial institutions, eKYC vendors, and platform security teams should implement a prioritized, server-anchored defense architecture:

Priority	Defense Layer	Concrete Engineering & Operational Controls	Mitigated Vector
P0 · NOW	**Close Asynchronous Timing Gaps**	Enforce strict capture-to-submission freshness timeouts across **merchant underwriting, remote video due diligence, and step-up reviews**—not only real-time consumer onboarding.	Non-real-time video upload & camera replacement
P0 · NOW	**Hardware Attestation & Runtime Integrity**	Require server-verified **iOS App Attest / DeviceCheck** (Secure Enclave challenge-response) and **Android Play Integrity** before KYC/payment actions; flag Substrate/ RootHide, non-system `mediaserverd` dylibs, and spoofed `com.apple.*` IPC artifacts.	5-layer device spoofing (`FakeTools`) & jailbreak daemons (`LordVCAM`)
P0 · NOW	**Server-Issued Random Challenges**	Issue randomized action sequences and random read-aloud tokens bound to a single-use server nonce, ensuring pre-recorded loops cannot satisfy verification.	Linear pre-recorded playback (`FakeTools` , `Tianqiong`)
P1 · HIGH	**Server-Side Physics & ROI Forensics**	Evaluate **luminance and color-temperature gradient continuity across the 2–5 px ring around the facial ROI** during screen-flash challenges; analyze multi-frame sensor noise, compression artifacts, and motion-blur consistency.	Local GLSL facial relighting (`LordVCAM`) & decoded stream injection
P1 · HIGH	**Sensor & Network Cross-Validation**	Correlate raw accelerometer/gyroscope streams against observed head movement to detect synthetic motion lacking organic hand tremor; validate proxy/ VPN and geography server-side via IP/ASN rather than client APIs or GPS.	`AVSMotionSynthesizer` , proxy suppression & map-matched GPS
P1 · HIGH	**Velocity Rules & Video-Call Guardrails**	Cap unique devices per ID document, hold rapid post-eKYC high-value transfers, cluster identical device/ background cohorts, and flag live-call anomalies (frozen backgrounds, lip-sync drift, refusal to flip cameras); update customer anti-scam warnings beyond screen-sharing alone.	Live PC-streamed proxies & zero-malware scam SaaS (`Tianqiong`)
P2 · GOV	**ISO/IEC 24745 Audits & Off-Network Intel**	Mandate separate `ISO/IEC 24745` capture-to-storage integrity and anti-replay evidence in eKYC vendor RFPs; continuously track virtual-camera core engines, white-label reseller reskins, and disposable Punycode meeting infrastructure.	Compliance blind spots & reseller proliferation

7. Appendix: Sanitized Detection Anchors & Audit Checklist

7.1 Sanitized Structural & Behavioral Detection Anchors

All items below represent sanitized engineering signatures for defensive telemetry and threat hunting; live customer identifiers and active malicious infrastructure are excluded.

Archetype	Defensive Signals for Mobile Risk SDKs, EDR & Server Analytics
In-Process Hook (`FakeTools` Class)	Loaded `CydiaSubstrate` / `MobileSubstrate` libraries or `/var/jb` paths; `_MSHookMessageEx` hooks across `AVCaptureVideoDataOutput` and `UIDevice`; preference suites matching `com.xuuz.faketools`; synthetic `IOHIDEvent` touch events with near-zero inter-tap variance; monotonic `PTS` vs. wall-clock drift from `mach_absolute_time` rebinding.
OS Daemon Engine (`LordVCAM` Class)	Non-system dylibs loaded inside `mediaserverd`; rogue LaunchDaemons (`avfsupportd`) or spoofed `com.apple.*` IPC directories (`/var/tmp/com.apple.avfcache`); `CFNotification` patterns (`com.apple.avfcache.framepulse`); **2-5 px facial-ROI boundary luminance/color-temperature step discontinuity** on color-flash frames; absence of organic micro-tremor noise in raw `CMDeviceMotion` streams.
Zero-Malware WebRTC (`Tianqiong` Class)	Custom `org.webrtc.VideoCapturer` classes backed by muted looping `MediaPlayer` instances (`UriVideoCapturer` , `th-vcam` , server entitlement flags such as `virtualCam`); browser clients pairing off-screen `<video loop muted>` with `canvas.captureStream()` ; disposable Punycode IDN domains with short-lived TLS certs and `?room=<digits>` invite links; shared frontend asset clustering to parallel screen-sharing sites.

7.2 Executive eKYC Readiness Checklist (ISO/IEC 30107-3 & 24745)

- ☐ **Separate PAD from Injection Evidence:** Require eKYC vendors to test and report digital injection and replay defense under `ISO/IEC 24745` independently of `ISO/IEC 30107-3:2023` physical presentation attack testing.
- ☐ **Eliminate Standard-Version Drift:** Confirm active internal test specifications and external lab certificates both reference `ISO/IEC 30107-3:2023` rather than legacy `2017` assessments.
- ☐ **Preserve First-Attempt Outcomes:** Verify `ISO/IEC 19795-1` reporting separates genuine vs. impostor attempts and retains first-attempt failures whenever user retries occur.
- ☐ **Audit Non-Real-Time Verification Paths:** Extend strict capture freshness windows and randomized challenges to merchant onboarding, step-up authentication, and remote video reviews.
- ☐ **Anchor Trust Server-Side:** Pair hardware attestation (`App Attest` / `Play Integrity`) with server-side ROI-boundary lighting forensics, IMU cross-checks, and post-eKYC velocity rules.

About Threat Hunter

Threat Hunter is a leading antifraud and cyber threat intelligence provider headquartered in Singapore. Grounded in continuous off-network intelligence—starting from the underground fraud supply chain rather than isolated internal logs—Threat Hunter enables banks, payment networks, e-commerce platforms, and digital enterprises to detect scam infrastructure, account takeover, payment abuse, and external digital risk earlier, and act with context.

See Risk Earlier. Act with Context. · www.threathunter.com/en/research

Public Release & Compliance Notice: Compiled by Threat Hunter Research from static binary analysis, controlled sandbox testing, and formal ISO/IEC standard mapping. Legitimate operating-system APIs, mobile frameworks, and payment rails referenced herein are abused or targeted by threat actors and have no affiliation with fraud. All customer identifiers, personal data, and live malicious endpoints have been redacted in accordance with public-release compliance standards.