PRODUCT BRIEF · 2026

Nexilis *Sentinel*

Mobile security for regulated institutions — Zero Trust application protection, hardware-rooted attestation, and continuous threat defense, architected for the way attacks actually happen.

PT Easysoft Indonesia

NEXILIS.IO · JAKARTA

PLATFORM	PROTECTION	CLASSIFICATION	VERSION
Android · iOS	Server-Delivered Keys	Confidential	1.0

01 INTRODUCTION

A serious answer to a serious *problem*.

For every regulated institution running a customer-facing mobile application, the mobile channel is now the single largest source of security incidents. Nexilis Sentinel is built to change that equation at the architectural level.

This document is organized so each reader can find what they need without reading the whole.

Who this document is for

READER	PRIMARY INTEREST	READ IN ORDER
CISO, CIO, Head of Digital Banking	Value, risk posture, regulatory fit	Sections 01 — 03 — 08
Security Architect, Head of IT Security	Architecture, threat model, differentiation	Sections 04 — 05 — 06
Procurement, Commercial Lead	Deployment, licensing, operational model	Sections 07 — 09 — 10

What Sentinel is, in one paragraph

Nexilis Sentinel is an integrated mobile security product that combines Zero Trust application protection with continuous mobile threat defense, delivered as a native SDK for Android and iOS with a supporting server-side authorization and intelligence platform. It is built for a specific problem that existing commercial products do not solve: the decryption key for a protected mobile application should never exist on the user's device without a live, hardware-attested server authorization. Every leading commercial mobile app protection SDK in this category — DexGuard and iXGuard from Guardsquare, SHIELD from Promon, Arxan from Digital.ai, Zimperium's zShield and zKeyBox, Appdome, AppSealing, Verimatrix, Talsec, and the comparable products across the category — derives or stores protection keys on the device. Sentinel does not. That single architectural property changes the attacker's economic model from *slow and expensive* to *requires compromising a live attested device in real time.*

THE ARCHITECTURAL PRINCIPLE

The decryption key for the protected application payload never exists on a user's device without a live server authorization that has verified hardware-rooted attestation of that *specific* device.

02 THE PROBLEM

Why existing protection *is not enough*.

Commercial mobile protection SDKs have not materially advanced their architectural model in five years. Attackers have.

Three shifts driving mobile as the dominant attack surface

Most

Urban bankable adults in Indonesia use mobile as their primary banking channel

Leading

Mobile fraud is the dominant source of financial fraud losses in Indonesian BFSI

0

Commercial SDKs that remove the protection key from the device

12 *layers*

Sentinel defensive layers on Android; 8 on iOS

For every regulated institution running a customer-facing mobile application, the mobile channel is now the single largest source of security incidents. Three structural shifts are driving this. Mobile banking in Indonesia has become the primary banking interaction for the majority of urban bankable adults, concentrating transaction volume on a channel most institutions do not fully control. Android rooting tools — Magisk, Zygisk, LSPosed — have reached a level of maturity where a moderately skilled attacker can bypass conventional root detection in hours. And commercial mobile protection SDKs, which most institutions rely on, have not materially advanced their architectural model in five years.

What commercial SDKs do not protect against

The leading commercial mobile app protection products — DexGuard and iXGuard from Guardsquare, SHIELD from Promon, Arxan from Digital.ai, Zimperium's zShield and zKeyBox, Appdome, AppSealing, Verimatrix, Talsec, and the comparable products across the category — share a common architectural limitation. All of them derive or store the protection key on the client device. The key may be obfuscated, split, white-box-encrypted, or tied to device characteristics; it is nonetheless present. An attacker with sufficient time, a rooted device, and standard reverse-engineering tools can — and regularly does — extract it. Once extracted, the protection scheme is defeated for every installation of that version of the application.

This is how most published mobile banking attacks unfold: not through novel vulnerabilities, but through patient extraction of protection keys from commercially protected applications. The commercial SDK raises the cost of attack; it does not eliminate it.

REGULATORY CONTEXT

What Indonesian regulators now expect

OJK POJK 11/2022 on digital banking resilience, POJK 29/2024 on consumer protection, SEOJK provisions on mobile risk management, and Bank Indonesia SNAP all require financial institutions to demonstrate device integrity verification, protection against instrumentation, and real-time detection of compromised environments. Conventional SDKs can claim partial compliance; they cannot demonstrate the cryptographic properties regulators increasingly expect.

03 VALUE PROPOSITION

Four outcomes that change *the equation*.

These are the outcomes Sentinel is designed against and the outcomes it is measured against. Each is architecturally specific, not aspirational.

01

No offline attack surface

The protection key does not exist on the device without a live, server-authorized delivery conditional on hardware attestation. Static binary analysis yields only ciphertext. There is no key to extract offline. This is the property that most directly addresses the extraction-based attack pattern responsible for the majority of real-world mobile banking breaches.

02

Cryptographic proof of integrity

Conventional root and jailbreak detection is a game the defender eventually loses. Sentinel supplements runtime detection with hardware-rooted attestation: Android Keystore attestation chained to Google's Hardware Attestation Root CA, and on iOS, Apple's DeviceCheck App Attest chained to Apple's certificate authority. Signed by hardware the attacker cannot forge.

03

Per-installation blast radius

Each installation receives a uniquely encrypted application payload bound to that device's attestation keypair. A compromise of one installation yields no information useful against any other installation. Targeted attacks remain possible; scalable attacks against the entire user base are not. An asymmetric defensive property no commercial SDK provides.

04

Real-time incident response

Every key issuance is logged with device fingerprint, attestation verdict, geography, and timestamp. Anomalous patterns trigger automatic alerts and can auto-revoke specific installations, users, or devices. When a compromise is detected, the institution can respond in minutes rather than waiting seven to fourteen days for an application store update cycle.

THE ONE SENTENCE WORTH REMEMBERING

Nexilis Sentinel is the only productized mobile security SDK where a stolen device, with stolen credentials, still cannot decrypt the protected application.

04 ARCHITECTURE

Twelve layers on Android, eight on *iOS*.

Each layer is independent and additive. The chain between Layers 1–6 delivers the architectural advantage: hardware attestation gates Zero Trust authentication, which gates key delivery.

Android protection layers

LAYER 0	**Supply chain integrity**	Verified build pipeline with signed manifest and air-gapped signing. The protector binary is reproducibly built from signed source; the per-install diverse payloads it generates (Layer 4.5) are derived deterministically from the institution's master build plus device-specific attestation parameters
LAYER 1	**Device integrity**	Hardware attestation via Keystore; requires the strongest Play Integrity verdict tier; emulator detection
LAYER 2	**OS integrity**	Verified boot, security patch level, developer mode and ADB state checks
LAYER 3	**Environment integrity**	Native detection of the major Android instrumentation frameworks — rooting tools, code-hooking frameworks, debugger attach, and GOT manipulation — combined with the hardware attestation verdict for authoritative determination
LAYER 3.5	**Watchman guard**	Secondary native library verifies the protector binary's own hash at runtime
LAYER 4	**Application integrity**	Play Integrity APP verdict, APK signature, protected payload hash vs manifest
LAYER 4.5	**Per-install diversity**	Each installation receives a uniquely encrypted payload bound to device attestation
LAYER 5	**Zero Trust authentication**	Challenge-response nonce, attestation package, user credentials with MFA
LAYER 6	**Server-side key delivery**	Fresh AES-256 session key wrapped with device hardware public key; only TEE can unwrap
LAYER 6.5	**Server-side logic execution**	Most security-critical operations execute server-side; client receives results, not code
LAYER 7	**Secure runtime & kill switch**	Protected payload loaded via secure in-memory class loading; decryption key wiped from memory immediately after load; per-device, per-user, and per-install revocation
LAYER 8	**Audit & anomaly detection**	Every key issuance logged; anomalous patterns auto-alert and auto-revoke

iOS protection layers

LAYER 0	**LLVM obfuscation**	Control-flow flattening, bogus control flow, instruction substitution, indirect calls
LAYER 1	**String encryption**	Every hardcoded string XOR-encrypted at build; decrypted only at call site
LAYER 2	**Linker hardening**	Dead-strip, symbol strip, PIE/ASLR, stack protector, Bitcode disabled
LAYER 3	**Runtime self-protection**	Jailbreak detection, debugger denial, Frida detection, code signature integrity
LAYER 4	**Certificate pinning**	SPKI SHA-256 pinning with backup pin; connection rejected on mismatch
LAYER 5	**Apple App Attest**	Hardware-rooted attestation via DeviceCheck; counter monotonicity prevents replay
LAYER 6	**Secure Enclave binding**	Cryptographic keys generated inside SEP; biometric-gated; never leave hardware
LAYER 7	**Server-side key delivery**	ZTA server delivers payload key only after attestation; key held in memory only

04.5 ZERO TRUST PROTOCOL

App launch to key delivery, in *six steps*.

This flow executes on every application launch. There is no cached or offline fallback. The nonce makes replay cryptographically impossible; the key wrap makes interception pointless.

I Client initializes and requests nonce

RASP preflight checks execute. Device requests a single-use nonce from the Sentinel server over a pinned TLS connection.

2 Hardware attestation generated

TEE (Android) or Secure Enclave (iOS) generates an attestation object signed by the device, chained to Google or Apple root certificate authorities.

3 Attestation package transmitted

Device sends attestation, Play Integrity or App Attest verdict, integrity results, and user credentials to the server.

4 Server verifies every verdict

Certificate chain, nonce freshness, device posture, and user credentials are independently verified. Any single failure rejects the request and logs the event.

5 Session key wrapped to device hardware

Server generates a fresh AES-256 session key and wraps it with the device's attested public key. Only the specific device's TEE or Secure Enclave can unwrap it.

6 Payload decrypted in memory, key wiped

The protected application payload is loaded into memory via secure in-memory class loading. The decryption key is wiped from memory immediately after load. Neither the key nor the plaintext payload is written to persistent storage by the application.

WHY THIS FLOW IS SECURE

The cryptographic chain an attacker would need to break

To obtain a decryption key, an attacker would need to simultaneously: forge an attestation certificate chain signed by Google or Apple root CAs; replay a nonce that the server tracks as single-use; compromise the device's hardware-rooted keypair without detection; and do all of this within a TLS session pinned to the Sentinel server's certificate. Each of these is individually hard; the conjunction is, in practice, the reason offline attacks against Sentinel do not succeed.

05 SHIELD

Protecting the user, not only the *application*.

The Shield module extends Sentinel into active threat defense for the end user. It catches the attacks that reach customers through channels outside the application itself.

PILLAR 01

Phishing and malware detection

URLs intercepted from four input channels — SMS, clipboard, browser intents, accessibility service on Android, and Message Filter Extension, clipboard, Share Extension, Safari Content Blocker on iOS. Each URL passes through a three-stage pipeline: local Bloom-filter blocklist, Google Safe Browsing hash-prefix matching, and an on-device ML classifier evaluating URL entropy, homoglyph patterns, and credential-harvesting indicators.

PILLAR 02

Network posture monitoring

Continuous evaluation of the device's network environment. WiFi security assessment, ARP spoofing detection on Android, DNS hijacking probe, captive portal detection, VPN status tracking. Detection is entirely local; network posture data never leaves the device. The institution receives summary telemetry only, never the specific networks a user connects to.

PILLAR 03

Identity theft monitoring

User PII — email, phone, card numbers, passport, NIK — monitored against a continuously updated dark-web breach database. The critical property: raw PII never leaves the device. The client transmits only the first five hex characters of a SHA-256 hash of the identifier, following the k-anonymity pattern (a stronger variant of the SHA-1 prefix model used by Have I Been Pwned). The server returns matching suffixes; the client performs the full comparison locally.

PILLAR 04

Server-side breach intelligence

A dedicated pipeline continuously crawls Tor hidden services, paste sites, and known breach dumps, normalizing and indexing discovered credentials against institutional PII hash registrations. When new data matches a registered prefix, an encrypted push notification reaches the device; the client performs suffix matching locally and raises an alert only if the user is actually affected.

PRIVACY ARCHITECTURE

Why k-anonymity matters to regulators and customers

The Shield identity monitoring architecture is designed so that even Nexilis cannot know which specific identifiers a particular user has registered. The server receives a five-character hash prefix that maps to hundreds of possible underlying identifiers; the full match happens on-device. This satisfies the data minimization principles embedded in POJK 29/2024 and PP 71/2019, and makes the product defensible under the increasingly strict privacy obligations banking supervisors are imposing on third-party processors.

06 COMPETITIVE POSITION

The two rows that *define the gap*.

The comparison below draws on each vendor's published architecture documentation. Two rows — offline attack surface and key extractability — determine real-world resistance to the attacks responsible for most published mobile banking compromises.

CAPABILITY	GUARDSQUARE (DEXGUARD / IXGUARD)	PROMON SHIELD	ARXAN (DIGITAL.AI)	ZIMPERIUM (ZSHIELD / ZKEYBOX)	SENTINEL
Application binary encryption	Client-derived	Client-derived	Client-derived	Client-derived	Server-delivered
Instrumentation detection (Frida, Magisk, Zygisk, debugger)	Partial	Yes	Partial	Yes	Yes
Hardware attestation (TEE / SEP)	No	No	No	No	Yes
Play Integrity STRONG verdict	No	Partial	No	No	Yes
Apple App Attest integration	No	No	No	No	Yes
Server-side key delivery	No	No	No	No	Yes
Per-install binary diversity	No	No	Yes	No	Yes
Offline attack surface	HIGH	MEDIUM	MEDIUM	MEDIUM	MINIMAL
Key extractable offline	Yes	Yes	Yes	Yes	No
Kill switch & revocation	No	No	No	No	Yes
Server-side logic execution	No	No	No	No	Yes
Integrated MTD	No	No	No	Partial	Yes

CAPABILITY	GUARDSQUARE (DEXGUARD / IXGUARD)	PROMON SHIELD	ARXAN (DIGITAL.AI)	ZIMPERIUM (ZSHIELD / ZKEYBOX)	SENTINEL
Audit trail & anomaly detection	No	No	No	Partial	Yes

The capabilities that warrant specific attention are server-side key delivery (Layer 6), per-install binary diversity (Layer 4.5), and server-side logic execution (Layer 6.5). These are the properties that change the attacker's economic model from *slow and expensive but eventually successful* to *requires compromising a specific, live, attested device with valid user credentials — and even that compromise yields nothing useful against any other installation.*

The four named competitors are representative of the category and shown for comparison detail; the architectural observations apply equally to Appdome, AppSealing, Verimatrix, Talsec, and other commercial mobile app protection SDKs in the same category, each of which stores or derives the protection key on the client device.

ADJACENT CATEGORIES

What Sentinel does not directly compete against

Two product categories sit adjacent to in-app protection without being direct competitors. Mobile Threat Defense products — Lookout, Pradeo, Zimperium's zDefend — operate at the device level rather than the application level and address a different attack surface; Sentinel integrates this capability through the Shield module rather than competing against it. API request attestation products — Approov (CriticalBlue) — implement server-side cryptographic proof of app genuineness for API authentication; Sentinel extends the same architectural principle from API requests to the protected application payload itself. Institutions currently using either category can complement rather than replace their existing investment.

06.5 THREAT MODEL

Honest about what it *does* and *does not* protect.

This is the threat model Sentinel is designed and audited against. The residual risks are named, not hidden. No architecture is unbreakable, and any supplier suggesting otherwise is overstating what is achievable.

THREAT	ATTACK METHOD	SENTINEL MITIGATIONS	RESIDUAL
Static binary analysis	Unzip, decompile, extract keys	AES-256-GCM ciphertext only; no plaintext payload at rest; no key on device	MINIMAL
Offline key extraction	Patient reverse engineering	Key is never present on device without live server authorization	NONE
Attestation token replay	Capture valid token, reuse	Single-use nonce bound to session and timestamp; replay cryptographically impossible	NONE
Emulator / virtual device	Run in controlled environment	Hardware attestation impossible; Play Integrity / App Attest fails	MINIMAL
Stub patching	Repack, patch integrity checks	Watchman guard verifies protector binary; app integrity invalidates repack	LOW
Compromised rooted device	Magisk with hiding to pass checks	MEETS_STRONG_INTEGRITY + hardware attestation + native Magisk detection	LOW-MED
Runtime memory dump	Dump decrypted payload from RAM	Layer 6.5 moves sensitive logic server-side; continuous RASP monitoring	MEDIUM
Supply chain compromise	Inject backdoor into build pipeline	Reproducible builds, signed manifest, air-gapped signing, hash verification	LOW
Credential theft via phishing	Social engineering for password	Credentials alone insufficient; hardware attestation independently required	LOW
Platform vulnerability (TEE/SEP)	Nation-state exploit of hardware	No software mitigation; outside realistic BFSI threat model	*ACCEPTED*

Sentinel is not claimed to be unbreakable. What the architecture materially achieves is the elimination of the offline attack surface — the largest and most common attack class — and the reduction of online attacks to an operational problem: one that can be detected, logged, revoked, and responded to in real time rather than through an application-store update cycle.

07 DEPLOYMENT

Eleven to nineteen weeks, signed engagement to *general availability*.

The eleven-to-nineteen-week range covers commercial Trusted Channel deployments from contract signature to production; sovereign TrustLink deployments for government and defense typically run 18–28 weeks due to additional infrastructure coordination. Pre-contract activities (executive briefing, technical deep dive, scoped proof-of-concept) typically add 8–16 weeks before signed engagement.

TOPOLOGY 01

Hosted

Nexilis-managed infrastructure in Indonesia. The institution consumes the service through signed API contracts. Fastest to deploy; suitable for commercial BFSI and fintech with standard regulatory exposure.

TOPOLOGY 02

Hybrid

Attestation and key delivery services deploy in the institution's own environment; threat intelligence and breach data supplied from the Nexilis-managed platform. Preferred for large banks with data residency policies.

TOPOLOGY 03

Sovereign

The entire platform deploys on infrastructure controlled by the institution. No external dependencies beyond the device-level Play Integrity and App Attest services. Designed for government and defense.

TOPOLOGY 04

Operational support

Standard operational support during Jakarta business hours with 24×7 incident escalation for production-severity issues. Platform updates for emerging threats within one platform release cycle. Dedicated technical account manager available at TrustLink and Enterprise tiers.

Implementation timeline

PHASE	ACTIVITY	DURATION	OUTCOME
01	Discovery	2 weeks	Architecture workshop, threat model review, integration scope
02	Integration	3–5 weeks	SDK integration into host application; server platform deployment
03	UAT & audit	2–4 weeks	UAT, adversarial security review, independent penetration test
04	Pilot	4–8 weeks	Limited production rollout to selected user segment. The pilot duration reflects the adversarial evaluation window: the institution's security team is explicitly invited to attempt to defeat the protection architecture during this phase, and findings are addressed before general availability.
05	General availability	Ongoing	Full production deployment under SLA

Operational considerations

The architectural properties that make Sentinel distinctive also introduce operational implications worth naming explicitly. The current generation of the product handles each of the scenarios below through documented mechanisms; detailed behaviour is covered in the technical evaluation package.

OPERATIONAL 01

Application updates

New application versions are registered with the Sentinel server before release; the server delivers the correct payload key for each known version. Users on older versions continue working through a grace window configurable per institution (typically 30–60 days), after which they are required to update.

OPERATIONAL 02

Launch latency

Server round-trip for attestation and key delivery typically adds 200–500 ms to cold start on a stable mobile network. Sentinel's caching of per-session material within the authorized session window minimizes impact on subsequent foreground resumptions. Measured against conventional-SDK baselines in real Indonesian network conditions.

OPERATIONAL 03

Offline operation

Sentinel requires live server authorization at application launch. Limited offline behaviour (viewing last-fetched data, queued actions pending reconnection) remains available within the authorized session; a fresh session cannot be established without network connectivity. For mobile banking specifically, this aligns with institutional preference — authenticated transactions already require connectivity.

OPERATIONAL 04

Device change & re-enrollment

New device, factory reset, or OS-level attestation reset invalidates the prior device binding. A standard re-enrollment flow (institution-defined: typically credentials plus one additional authentication factor) rebuilds the binding on the new device. The re-enrollment flow is integrated with the institution's existing customer onboarding infrastructure.

DEVICE COVERAGE

Play Integrity tier and hardware attestation coverage

Sentinel's strongest configuration requires the highest Play Integrity verdict tier on Android (which requires hardware-backed attestation, supported on GMS-certified devices from recent manufacturers) and Apple DeviceCheck App Attest on iOS (supported on devices running iOS 14 and later with Secure Enclave). On Android devices that cannot reach the highest tier — older devices, some regional-vendor devices without full GMS certification — Sentinel applies a configurable degraded trust path: the device is granted application access at a lower trust tier with correspondingly reduced transaction limits, rather than being excluded outright. The exact tier logic is defined per institution during Discovery.

08 REGULATORY ALIGNMENT

Built for Indonesian regulation, *first*.

Sentinel is designed to satisfy Indonesian financial services regulation and Indonesian government data protection regulation directly — not adapted from foreign frameworks.

REGULATION	RELEVANT REQUIREMENT	SENTINEL CAPABILITY
OJK POJK 11/2022 Digital Banking Resilience	Mobile channel integrity; tampering protection; compromised environment detection	Hardware attestation, Play Integrity, Layers 1–4 integrity checks, full audit trail
OJK POJK 29/2024 Consumer Protection	Protection of customer data and transactions on mobile channels	Shield phishing defense, network posture, breach monitoring, PII isolation
SEOJK Digital Banking Risk	Real-time mobile threat detection and response	Continuous RASP, server-side anomaly detection, kill switch, Shield MTD
Bank Indonesia SNAP SDK	Secure mobile SDK for open banking API consumption	Sentinel SDK with ZTA authentication, certificate pinning, attestation-bound sessions
Kominfo PP 71/2019 PSTE Data Sovereignty	Data sovereignty and secure handling in Indonesia	On-device PII isolation, k-anonymity, Indonesia-hosted platform option
NIST SP 800-163	Mobile application vetting framework	Architecture satisfies all Level 3 vetting criteria
PCI MPoC & PCI SSF	Secure payment applications on mobile	AES-256 at rest and in transit, TLS 1.3, certificate pinning, no plaintext on persistent storage
OWASP MASVS v2.0	Mobile Application Security Verification Standard (current category-based structure)	Controls mapped against MASVS-STORAGE, CRYPTO, AUTH, NETWORK, PLATFORM, CODE, RESILIENCE, and PRIVACY categories; detailed mapping provided under NDA

AVAILABLE UNDER NDA

Detailed control mapping

A detailed control-mapping document, showing each regulatory provision and the specific Sentinel architectural layer or capability that satisfies it, is available under mutual non-disclosure as part of the technical evaluation package. Institutions with active procurement cycles can receive this package within forty-eight hours of signed NDA.

09 COMMERCIAL STRUCTURE

Three tiers, *one platform*.

Sentinel is the platform; Trusted Channel and TrustLink are the commercial tiers under which institutions acquire it. Per-active-device annual subscription includes the full SDK, server platform access, threat intelligence, breach data, platform updates, and operational support.

TIER	INTENDED FOR	INCLUDED CAPABILITIES
Trusted Channel	Commercial banks, digital banks, payment processors, fintech	Full ZTA protection core, Shield MTD, hosted or hybrid deployment, standard support
TrustLink	Government agencies, intelligence, defense organizations	All Trusted Channel capabilities plus sovereign deployment, enhanced attestation thresholds, Layer 6.5 enabled by default, classified-operation support
Trusted Channel Enterprise	Tier-1 institutions with extended scale requirements	All Trusted Channel capabilities with dedicated infrastructure, custom SLA, dedicated security liaison

Included with all tiers

- Platform updates addressing new threats, device models, and Android / iOS releases
- Threat intelligence feeds, phishing signature updates, and dark-web breach data
- Standard operational support during Jakarta business hours; 24×7 for production-severity incidents
- Quarterly security posture review and anomaly reporting
- Annual architecture review with the institution's security team

COMMERCIAL ENVELOPE

Specific pricing discussed per engagement

Per-device pricing, volume tiers, multi-year discounts, and professional services rates are established per engagement following the initial discovery phase. Indicative commercial envelopes are available on request under mutual non-disclosure. Sentinel is competitive with commercial SDK alternatives on total cost of ownership, with materially stronger architectural protection and regional specialization.

The conversation starts *here.*

Institutions evaluating Sentinel typically follow a three-stage process from initial conversation to signed engagement.

STEP 01

Executive briefing

A 60–90 minute session with the institution's CISO, CIO, or Head of Digital Banking. Architectural value, regulatory fit, typical deployment. Includes a live demonstration of the protection architecture operating against instrumented attack scenarios.

STEP 02

Technical deep dive

A half-day workshop with the institution's security architecture, mobile engineering, and audit teams. Full layer-by-layer architecture, threat model, integration model, and operational tooling. Technical evaluation package provided under NDA.

STEP 03

Proof of concept

Scoped integration into the institution's mobile application in a controlled environment for six to eight weeks. The institution's security team is invited to attempt to defeat the protection architecture during the PoC. Findings are addressed prior to production engagement.

PT Easysoft Indonesia

Jakarta, Indonesia · nexilis.io

sentinel@nexilis.io