Kryfo Dreadful: fork hardenizzato, anti-forensics dell'archivio isolato e deployment con Obtainium

Kryfo Dreadful: fork hardenizzato, anti-forensics dell'archivio isolato e deployment con Obtainium

Nel nostro precedente approfondimento abbiamo analizzato l'architettura innovativa di **Kryfo** (il software di messaggistica cypherpunk sviluppato da *GnosiWorks*), esaminandone il motore Tor onion v3 in-process, il protocollo di scambio chiavi post-quantum **PQXDH** e le difese a livello di kernel Linux tramite `PR_SET_DUMPABLE`.

Tuttavia, quando una tecnologia viene esaminata nell'ottica di scenari avversari estremi — come perquisizioni, sequestri fisici dell'hardware e perizie forensi condotte con suite avanzate (Cellebrite UFED, Magnet AXIOM, GrayKey) — emergono dettagli implementativi capaci di fare la differenza tra la libertà personale e la compromissione totale della riservatezza.

Da questa esigenza nasce **Kryfo Dreadful**: una variante rigorosamente hardenizzata, progettata per sanare le asimmetrie forensi dell'upstream, garantire la **plausibile negabilità** reale dell'archivio riservato, ottimizzare il trasporto delle notifiche push cieche e consentire una distribuzione continua e decentralizzata tramite **Obtainium**.

Il paradosso forense dell'upstream: perché il vault opzionale tradisce l'utente

L'implementazione ufficiale di Kryfo include una funzione potente: la creazione di un'area protetta (*vault*) per isolare contatti e conversazioni delicate sotto un secondo PIN indipendente. Nella logica standard, l'utente poteva decidere se attivare il vault durante l'onboarding oppure rimandarne o saltarne la configurazione.

Sebbene dal punto di vista dell'usabilità questa scelta sembri naturale, dal punto di vista dell'**analisi forense post-sequestro** introduce una vulnerabilità critica:

```
flowchart TD
    subgraph Upstream ["Kryfo Upstream (Comportamento Asimmetrico)"]
        U1["Configurazione iniziale"] --> U2{"L'utente vuole il vault?"}
        U2 -- No / Salta --> U3["Database halo.db standard\nNessun k3.db\nvault_meta.pub ASSENTE\nhidden_chats VUOTO"]
        U2 -- Sì --> U4["Database halo.db + k3.db\nvault_meta.pub PRESENTE\nhidden_chats POPOLATO"]
        U3 -.-> UF1["Evidenza peritale:\nIl perito SA che l'utente\nNON ha chat nascoste"]
        U4 -.-> UF2["Evidenza peritale:\nIl perito VEDE la chiave pub:\nSA che l'utente mente se nega l'esistenza"]
    end
```

L'asimmetria degli artefatti nel database

Nel momento in cui un dispositivo viene clonato a livello fisico (bit-stream image):

1. **se l'utente non aveva configurato il vault**: la tabella `vault_meta` non conteneva alcuna chiave pubblica (`pub`), la tabella `hidden_chats` era inerte e non esisteva alcun file `k3.db`;
2. **se l'utente aveva configurato il vault**: all'interno del database principale `halo.db` comparivano immediatamente la chiave pubblica del vault (`vault_meta.pub`) e i riferimenti crittografici dello slot secondario, oltre all'esistenza fisica del database secondario `k3.db`.

Davanti a un perito informatico forense, questa asimmetria distrugge la **plausibile negabilità**:

- se l'indagato fornisce soltanto il PIN principale (mostrando l'ambiente ordinario privo di conversazioni compromettenti) e dichiara di non possedere altri dati, il perito rileva immediatamente la presenza del record `pub` in `vault_meta`;
- la presenza della chiave dimostra inconfutabilmente che un'area protetta esiste ed è stata configurata. L'utente non può più sostenere che l'applicazione contenga soltanto le conversazioni visibili.

Le contromisure di Kryfo Dreadful: uniformità forense di stato

Per neutralizzare questo vettore di attacco, Kryfo Dreadful adotta il principio della **uniformità forense di stato** (*forensic state uniformity*): eliminare qualsiasi differenza rilevabile tra un'installazione che fa uso dell'archivio riservato e una che non lo utilizza affatto.

```
flowchart TD
    subgraph Dreadful ["Kryfo Dreadful (Uniformità Forense al 100%)"]
        D1["Onboarding"] --> D2{"Scelta dell'utente"}
        D2 -- "Ignora" (Default) --> D3["Generazione PIN casuale a 16 cifre\nInizializzazione e sigillo immediato\n(k3.db + vault_meta.pub creati)"]
        D2 -- "Configura archivio isolato" --> D4["Impostazione secondo PIN utente\n(k3.db + vault_meta.pub creati)"]
        D3 --> DU["Stato forense identico sul 100% dei dispositivi:\n• k3.db sempre presente\n• vault_meta.pub sempre presente\n• Strutture SQLCipher indistinguibili"]
        D4 --> DU
    end
```

1. Auto-sealing con PIN pseudocasuale sicuro

Quando l'utente completa la schermata di protezione premendo **"Ignora"**, Kryfo Dreadful non tralascia la creazione del vault:

- invoca un generatore crittografico ad alta entropia (`Random.secure()`) e produce un codice a 16 cifre casuale;
- esegue la normale procedura di provisioning: crea il database `k3.db`, configura le credenziali nello slot 3 di SQLCipher e inserisce il record crittografico `pub` in `vault_meta`;
- dichiara completato il setup e distrugge immediatamente il PIN casuale dalla memoria volatile.

Il risultato peritale è radicale: **ogni singola copia installata di Kryfo Dreadful presenta l'identico profilo strutturale su disco**. Di fronte a un dump forense, il perito trova sempre e comunque il file `k3.db` e il record `pub`. L'affermazione dell'indagato: "*Non ho mai configurato un archivio isolato: l'applicazione al primo avvio genera automaticamente le tabelle cifrate interne per impostazione predefinita*" diventa **tecnicamente e peritalmente inattaccabile**.

2. Ridenominazione in "Archivio isolato"

Ogni riferimento a "chat nascoste" o "cassaforte" (*vault*) è stato eliminato dall'interfaccia utente e localizzato in 17 lingue come **"Archivio isolato"** (*isolated archive*). Questo lessico neutro si allinea ai normali strumenti aziendali di separazione dei contesti (come archivi offline o cartelle di lavoro isolate), riducendo l'attrito e la rilevanza psicologica durante un controllo visivo superficiale.

3. Inversione del flusso: scelta sicura per impostazione predefinita

Nell'upstream, il pulsante in evidenza induceva l'utente ad andare avanti creando il vault con il rischio di configurazioni incomplete o leak operativi. In Kryfo Dreadful la gerarchia della schermata è invertita:

- il pulsante principale è un chiaro e inequivocabile **"Ignora"**;
- chi desidera attivare e memorizzare il secondo PIN deve compiere un'azione deliberata, cliccando sul testo secondario "*Configura archivio isolato*";
- l'archivio può essere configurato **soltanto in questa fase iniziale** o durante il ripristino di un profilo: una volta superato l'onboarding, non esiste alcun menu, sottomenu o impostazione nell'applicazione che permetta di riconfigurarlo, azzerando qualsiasi indizio nell'interfaccia grafica.

Ottimizzazioni architetturali: push relay gateway e disaccoppiamento

Oltre alla componente forense locale, il fork Dreadful risolve due criticità architetturali del client mobile:

1. Push relay gateway dedicato (r.dreadful.work)

Per recapitare notifiche push preservando l'anonimato su rete Tor, Kryfo supporta il disaccoppiamento tramite **UnifiedPush** (ntfy). Tuttavia, i gateway pubblici upstream o provider terzi generano spesso endpoint URL molto lunghi che superano il limite rigido di 512 byte

previsto dalla specifica dei messaggi di sincronizzazione o soffrono di rate-limiting aggressivo.

Kryfo Dreadful integra direttamente nelle patch il puntamento all'infrastruttura di relay dedicata:

```
https://r.dreadful.work
```

Questo garantisce un indirizzo compatto, latenze minime e la certezza che nessun log di connessione o metadato di transito venga persistito su disco.

2. Disaccoppiamento dai canali upstream

Nel fork sono stati rimossi i contatti di sviluppo hardcoded (*devchat*) e i link promozionali o di canale broadcast diretti verso l'autore originale, mantenendo pulito l'ambiente per contesti ad alta sicurezza. Al contempo, il file `README.md` preserva intatta e visibile la sezione di attribuzione e supporto verso il progetto madre **GnosiWorks**, incoraggiando le donazioni per sostenere la ricerca crittografica indipendente.

3. Identificatore di applicazione isolato

Il pacchetto Android è distribuito con il namespace univoco:

```
work.dreadful.kryfo
```

Ciò consente di installare Kryfo Dreadful in parallelo con la versione standard di Kryfo o all'interno di profili di lavoro isolati (come Shelter o Insular) senza alcun conflitto di firme o collisione di preferenze condivise.

Guida al deployment e aggiornamento con Obtainium

Per installare e mantenere aggiornato Kryfo Dreadful senza passare da store commerciali o attendere le tempistiche dei repository centralizzati, il metodo d'elezione è **Obtainium**.

Obtainium (https://github.com/ImranR98/Obtainium) è un gestore di pacchetti open source per Android che interroga direttamente le pagine delle release di GitHub, GitLab, Forgejo o sorgenti HTML, estrae gli APK firmati, ne verifica l'integrità e segnala (o installa automaticamente) i nuovi aggiornamenti.

```
flowchart LR
    Repo["GitHub / CI Workflow\nCyberPingU/Kryfo"] -->|Compilazione e Release\nTag: v*-dreadful| Releases["GitHub Releases\n(app-release.apk)"]
    Obtainium["Obtainium Client\n(su smartphone Android)"] -->|Polling periodico API| Releases
    Obtainium -->|Download diretto & Verifica SHA-256| Install["Installazione / Update\nwork.dreadful.kryfo"]
```

Procedura di installazione passo dopo passo

1. **installazione di Obtainium**: scaricare l'applicazione Obtainium dal suo repository ufficiale (https://github.com/ImranR98/Obtainium/releases) o tramite F-Droid;
2. **aggiunta della sorgente di Kryfo Dreadful**: aprire Obtainium, premere il pulsante **"Aggiungi app"** (`+`) e inserire l'URL del repository:

   ```
   https://github.com/CyberPingU/Kryfo
   ```

3. **configurazione del filtro delle versioni (raccomandato)**: nelle opzioni avanzate della scheda di configurazione, impostare il filtro per intercettare le build Dreadful:
   - *includi tag che corrispondono al filtro*: `*dreadful*`;
4. **download e installazione iniziale**: premere **"Aggiungi"**. Obtainium contatterà le API di GitHub, individuerà l'ultima release disponibile (ad esempio `v0.6.3-dreadful`), scaricherà il pacchetto `app-release.apk` verificandone l'integrità e avvierà l'installazione di sistema;
5. **aggiornamenti automatici in background**: nelle impostazioni di Obtainium è consigliabile attivare il controllo periodico degli aggiornamenti in background. Non appena la pipeline compila e pubblica una nuova versione con patch di sicurezza o allineamenti upstream, l'applicazione notificherà la disponibilità dell'aggiornamento con un clic.

Configurazione raccomandata su Android: risveglio affidabile in background

I sistemi operativi Android moderni integrano meccanismi di risparmio energetico molto aggressivi (inclusa la modalità Doze e la sospensione automatica delle applicazioni inattive), che possono congelare i socket di rete e impedire ai listener Tor e UnifiedPush di ricevere le notifiche a schermo spento.

Per garantire la massima tempestività nella ricezione dei messaggi cifrati senza interruzioni:

1. aprire le **Impostazioni** di sistema di Android e selezionare la voce **Applicazioni**;
2. individuare **Kryfo** (`work.dreadful.kryfo`) e accedere alla sezione **Batteria**;
3. impostare l'utilizzo della batteria su **"Senza restrizioni"** (*unrestricted*).

Questa configurazione colloca il pacchetto nella whitelist di sistema di Android, esentandolo dalle restrizioni di Doze e concedendogli il diritto di attivare immediatamente la connessione di rete e il wakelock non appena arriva una segnalazione dal relay.

Specifiche della pipeline di compilazione automatizzata

Tutte le build di Kryfo Dreadful sono riproducibili, trasparenti e tracciabili. Il repository include un flusso GitHub Actions (`.github/workflows/sync-and-build.yml`) che opera nel seguente modo:

1. **sincronizzazione upstream**: ogni 6 ore un processo automatico interroga il repository originale di `GnosiWorks/Kryfo` per rilevare nuovi tag stabili;
2. **applicazione dinamica delle patch**: l'ambiente di compilazione esegue il checkout del sorgente upstream pulito e applica in sequenza le 5 patch Dreadful:
   - `0001-vault-onboarding-plausible-absence.patch`: genera il vault casuale a 16 cifre in caso di skip per uniformità forense;

    - `0002-push-gateway-relay-dreadful.patch`: instrada le notifiche push sul gateway `r.dreadful.work`;
    - `0003-gpush-logging.patch`: abilita la diagnostica avanzata del flusso di ricezione push;
    - `0004-remove-devchat-and-marios.patch`: sanifica l'ambiente rimuovendo contatti hardcoded e link esterni;
    - `0005-protection-on-restore.patch`: protegge l'onboarding con pulsante "Ignora", impone il blocco applicazione anche nel ripristino profilo e localizza l'archivio isolato in 17 lingue;
3. **compilazione nativa e packaging**: Flutter compila i moduli nativi in C++/Rust (il runtime Tor `libhalo.so` e le estensioni SQLCipher crittografate per tutte le architetture Android);
4. **firma crittografica della release**: l'APK finale viene firmato con il keystore ufficiale di release e reso disponibile per il download immediato.

Considerazioni finali

Nel contesto delle comunicazioni difensive, la sicurezza non può dipendere dall'inconsapevolezza o da assunti ottimistici. Se un software implementa uno spazio cifrato "nascosto", la sua stessa architettura deve garantire che la sua presenza non possa essere né provata né smentita con strumenti di perizia tecnica.

Con l'introduzione della **uniformità forense di stato**, del pulsante di bypass **"Ignora"** con auto-sigillo a 16 cifre, del push relay gateway dedicato `r.dreadful.work` e della distribuzione sicura via **Obtainium**, **Kryfo Dreadful** trasforma una piattaforma di per sé eccezionale in uno strumento operativo a prova di coercizione fisica e sequestro forense.

Il codice sorgente completo, le patch e i binari verificabili sono disponibili pubblicamente sul repository GitHub di CyberPingU/Kryfo (https://github.com/CyberPingU/Kryfo).

Fluxer Icon

Canale di discussione e supporto

Canale di discussione e supporto. Per confrontarti su configurazioni o altro entra su fluxer.dreadful.work (https://fluxer.dreadful.work/invite/dreadful).