Quando la crittografia non basta: come il protocollo STUN espone l'IP reale nelle chiamate Telegram

Maria Cattini | 14/05/2026 | Open source intelligence

La crittografia end-to-end protegge il contenuto di una chiamata. Non protegge il fatto che la chiamata avvenga, né da dove. Questa distinzione — tra protezione del payload e protezione dei metadati di connessione — è il punto in cui si apre una finestra di analisi che molti investigatori OSINT non considerano, e che molti obiettivi dimenticano di chiudere.

Il protocollo STUN (Session Traversal Utilities for NAT) è parte dell'infrastruttura che rende possibile le chiamate peer-to-peer nei messenger moderni. Non è una vulnerabilità software. Non richiede exploit. Funziona esattamente come progettato — ed è proprio questa normalità a renderlo rilevante in contesti investigativi.

Il problema strutturale: metadati oltre la crittografia

Messenger come Telegram, WhatsApp e Signal adottano crittografia end-to-end per proteggere il contenuto delle comunicazioni. Questo scudo è solido sul piano del payload. Il punto critico si trova altrove: nel momento in cui due dispositivi, spesso situati dietro i NAT dei rispettivi router, devono stabilire una connessione diretta per una chiamata vocale o video in modalità P2P, devono scambiarsi le proprie coordinate pubbliche su Internet — indirizzo IP e porta.

Questo scambio avviene tramite STUN. La richiesta al server STUN precede il flusso multimediale cifrato e, trattandosi di un pacchetto UDP, non è crittografata. Il pacchetto è visibile a chiunque monitori il traffico sull'interfaccia di rete attiva nel momento della chiamata.

Il dato che emerge non è il contenuto della conversazione. È l'indirizzo IP pubblico del dispositivo chiamante — che, in assenza di VPN, corrisponde alla connessione reale dell'utente.

Mappa del sistema

Layer di rete: il traffico STUN transita sull'interfaccia attiva (Wi-Fi o Ethernet) come pacchetto UDP non cifrato, prima dell'instaurazione del canale multimediale E2EE.

Layer applicativo: Telegram attiva il meccanismo P2P durante le chiamate vocali e video. I messaggi testuali non generano traffico STUN rilevante per questa tecnica.

Layer di analisi: il campo XOR-MAPPED-ADDRESS nei pacchetti STUN restituisce l'indirizzo IP pubblico e la porta del client richiedente. Il log di cattura contiene richieste di entrambi i partecipanti — è necessario distinguerle analizzando la struttura del pacchetto e la sequenza di scambio.

Layer strumentale: Wireshark è lo strumento primario di cattura e analisi. Alternative citate nel materiale sono STUNmon e Telegram get remote IP.

Cosa rimane fuori: questa tecnica non consente di accedere al contenuto della chiamata, non intercetta messaggi testuali, non bypassa E2EE. Restituisce esclusivamente metadati di connessione.

Metodologia operativa

Fase 1 — Preparazione dell'ambiente di cattura Installa Wireshark dall'installer ufficiale. Identifica l'interfaccia di rete attiva sul dispositivo in uso (Wi-Fi o Ethernet). Avvia la sessione di cattura su quella interfaccia prima di iniziare qualsiasi interazione con il target. La cattura deve essere attiva nel momento esatto in cui la chiamata viene accettata dall'interlocutore.

Fase 2 — Filtraggio del traffico Nella stringa di filtro di Wireshark, inserisci: stun. Questo esclude tutto il traffico non pertinente, isolando i pacchetti STUN. Il filtro va applicato prima che la chiamata venga stabilita, per non perdere i pacchetti iniziali dello scambio.

Fase 3 — Avvio della chiamata Effettua una chiamata vocale o video al destinatario su Telegram. La chiamata deve essere accettata: è nel momento di accettazione che i pacchetti STUN vengono scambiati per la negoziazione P2P. Un messaggio testuale non produce il traffico cercato.

Fase 4 — Estrazione dell'IP Nel log di Wireshark, usa la funzione di ricerca (Ctrl+F). Seleziona la modalità "Stringa" e cerca l'attributo XOR-MAPPED-ADDRESS. I pacchetti che contengono questo campo includono l'indirizzo IP pubblico e la porta del client remoto, restituite dal server STUN.

Logica di validazione: il log conterrà pacchetti generati da entrambi i partecipanti alla chiamata. La distinzione tra il proprio IP e quello dell'interlocutore richiede l'analisi della struttura del pacchetto (direzione sorgente/destinazione) e della sequenza logica dello scambio STUN. Un IP riconducibile alla propria connessione non è il dato cercato.

Strumenti alternativi: STUNmon e Telegram get remote IP sono citati come alternative a Wireshark per contesti operativi che richiedono un setup semplificato.

Rischi e limitazioni

VPN attiva sul target: se l'interlocutore opera tramite VPN, il campo XOR-MAPPED-ADDRESS restituirà l'IP del server VPN, non l'IP reale. Questa è la principale contromisura — e la sua assenza, quando verificata, è già un dato.

Ambiguità nel log: distinguere il proprio IP da quello del target richiede familiarità con la struttura dei pacchetti STUN. Errori di interpretazione producono falsi risultati.

Dipendenza dalla modalità P2P: la tecnica funziona quando Telegram stabilisce una connessione P2P diretta. In condizioni di rete particolari o per scelta dell'app, il traffico può essere instradato tramite server relay — in quel caso il pacchetto STUN non espone l'IP reale del target.

Finestra temporale stretta: i pacchetti STUN vengono scambiati durante la negoziazione iniziale della chiamata. La cattura deve essere attiva in quel momento preciso. Avviare Wireshark dopo l'inizio della chiamata può far perdere i pacchetti rilevanti.

Perimetro legale: il monitoraggio del traffico su reti altrui senza autorizzazione è illegale in quasi tutte le giurisdizioni. Questa tecnica è applicabile esclusivamente al traffico della propria rete, in contesti autorizzati, o a fini difensivi e di ricerca.

Layer analitico

Il caso STUN illustra una contraddizione strutturale presente in tutti i sistemi di comunicazione cifrata: la crittografia del contenuto non equivale all'anonimato della connessione. I due livelli sono separati per design. Un utente può credere di essere protetto dall'E2EE mentre espone la propria posizione di rete a chiunque stia monitorando il traffico locale.

Questa separazione non è una falla da correggere — è un vincolo architetturale. Affinché due dispositivi dietro NAT si connettano direttamente, devono scambiare indirizzi pubblici. STUN è il meccanismo standard per farlo. La scelta progettuale di non cifrare questo scambio (pacchetto UDP) deriva da considerazioni di latenza e compatibilità, non da negligenza.

Il pattern che emerge è sistematico: i metadati di connessione — chi chiama chi, quando, da dove — rimangono accessibili anche nei sistemi più robusti sul piano crittografico. Per l'analisi OSINT, questo significa che la superficie di osservazione non si chiude con la crittografia del payload.

La contromisura operativa è precisa: una VPN attiva sul dispositivo target reindirizza il traffico STUN attraverso il server VPN, rendendo il dato risultante inutilizzabile per l'identificazione geografica o dell'ISP reale. L'assenza di VPN, rilevabile indirettamente dall'IP restituito, è essa stessa un'informazione sulla postura di sicurezza dell'obiettivo.
La crittografia end-to-end protegge il contenuto di una chiamata. Non protegge il fatto che la chiamata avvenga, né da dove. Questa distinzione — tra protezione del payload e protezione dei metadati di connessione — è il punto in cui si apre una finestra di analisi che molti investigatori OSINT non considerano, e che molti obiettivi dimenticano di chiudere.

Il protocollo STUN (Session Traversal Utilities for NAT) è parte dell'infrastruttura che rende possibile le chiamate peer-to-peer nei messenger moderni. Non è una vulnerabilità software. Non richiede exploit. Funziona esattamente come progettato — ed è proprio questa normalità a renderlo rilevante in contesti investigativi.

Il problema strutturale: metadati oltre la crittografia

Messenger come Telegram, WhatsApp e Signal adottano crittografia end-to-end per proteggere il contenuto delle comunicazioni. Questo scudo è solido sul piano del payload. Il punto critico si trova altrove: nel momento in cui due dispositivi, spesso situati dietro i NAT dei rispettivi router, devono stabilire una connessione diretta per una chiamata vocale o video in modalità P2P, devono scambiarsi le proprie coordinate pubbliche su Internet — indirizzo IP e porta.

Questo scambio avviene tramite STUN. La richiesta al server STUN precede il flusso multimediale cifrato e, trattandosi di un pacchetto UDP, non è crittografata. Il pacchetto è visibile a chiunque monitori il traffico sull'interfaccia di rete attiva nel momento della chiamata.

Il dato che emerge non è il contenuto della conversazione. È l'indirizzo IP pubblico del dispositivo chiamante — che, in assenza di VPN, corrisponde alla connessione reale dell'utente.

Mappa del sistema

Layer di rete: il traffico STUN transita sull'interfaccia attiva (Wi-Fi o Ethernet) come pacchetto UDP non cifrato, prima dell'instaurazione del canale multimediale E2EE.

Layer applicativo: Telegram attiva il meccanismo P2P durante le chiamate vocali e video. I messaggi testuali non generano traffico STUN rilevante per questa tecnica.

Layer di analisi: il campo XOR-MAPPED-ADDRESS nei pacchetti STUN restituisce l'indirizzo IP pubblico e la porta del client richiedente. Il log di cattura contiene richieste di entrambi i partecipanti — è necessario distinguerle analizzando la struttura del pacchetto e la sequenza di scambio.

Layer strumentale: Wireshark è lo strumento primario di cattura e analisi. Alternative citate nel materiale sono STUNmon e Telegram get remote IP.

Cosa rimane fuori: questa tecnica non consente di accedere al contenuto della chiamata, non intercetta messaggi testuali, non bypassa E2EE. Restituisce esclusivamente metadati di connessione.

Metodologia operativa

Fase 1 — Preparazione dell'ambiente di cattura Installa Wireshark dall'installer ufficiale. Identifica l'interfaccia di rete attiva sul dispositivo in uso (Wi-Fi o Ethernet). Avvia la sessione di cattura su quella interfaccia prima di iniziare qualsiasi interazione con il target. La cattura deve essere attiva nel momento esatto in cui la chiamata viene accettata dall'interlocutore.

Fase 2 — Filtraggio del traffico Nella stringa di filtro di Wireshark, inserisci: stun. Questo esclude tutto il traffico non pertinente, isolando i pacchetti STUN. Il filtro va applicato prima che la chiamata venga stabilita, per non perdere i pacchetti iniziali dello scambio.

Fase 3 — Avvio della chiamata Effettua una chiamata vocale o video al destinatario su Telegram. La chiamata deve essere accettata: è nel momento di accettazione che i pacchetti STUN vengono scambiati per la negoziazione P2P. Un messaggio testuale non produce il traffico cercato.

Fase 4 — Estrazione dell'IP Nel log di Wireshark, usa la funzione di ricerca (Ctrl+F). Seleziona la modalità "Stringa" e cerca l'attributo XOR-MAPPED-ADDRESS. I pacchetti che contengono questo campo includono l'indirizzo IP pubblico e la porta del client remoto, restituite dal server STUN.

Logica di validazione: il log conterrà pacchetti generati da entrambi i partecipanti alla chiamata. La distinzione tra il proprio IP e quello dell'interlocutore richiede l'analisi della struttura del pacchetto (direzione sorgente/destinazione) e della sequenza logica dello scambio STUN. Un IP riconducibile alla propria connessione non è il dato cercato.

Strumenti alternativi: STUNmon e Telegram get remote IP sono citati come alternative a Wireshark per contesti operativi che richiedono un setup semplificato.

Rischi e limitazioni

VPN attiva sul target: se l'interlocutore opera tramite VPN, il campo XOR-MAPPED-ADDRESS restituirà l'IP del server VPN, non l'IP reale. Questa è la principale contromisura — e la sua assenza, quando verificata, è già un dato.

Ambiguità nel log: distinguere il proprio IP da quello del target richiede familiarità con la struttura dei pacchetti STUN. Errori di interpretazione producono falsi risultati.

Dipendenza dalla modalità P2P: la tecnica funziona quando Telegram stabilisce una connessione P2P diretta. In condizioni di rete particolari o per scelta dell'app, il traffico può essere instradato tramite server relay — in quel caso il pacchetto STUN non espone l'IP reale del target.

Finestra temporale stretta: i pacchetti STUN vengono scambiati durante la negoziazione iniziale della chiamata. La cattura deve essere attiva in quel momento preciso. Avviare Wireshark dopo l'inizio della chiamata può far perdere i pacchetti rilevanti.

Perimetro legale: il monitoraggio del traffico su reti altrui senza autorizzazione è illegale in quasi tutte le giurisdizioni. Questa tecnica è applicabile esclusivamente al traffico della propria rete, in contesti autorizzati, o a fini difensivi e di ricerca.

Layer analitico

Il caso STUN illustra una contraddizione strutturale presente in tutti i sistemi di comunicazione cifrata: la crittografia del contenuto non equivale all'anonimato della connessione. I due livelli sono separati per design. Un utente può credere di essere protetto dall'E2EE mentre espone la propria posizione di rete a chiunque stia monitorando il traffico locale.

Questa separazione non è una falla da correggere — è un vincolo architetturale. Affinché due dispositivi dietro NAT si connettano direttamente, devono scambiare indirizzi pubblici. STUN è il meccanismo standard per farlo. La scelta progettuale di non cifrare questo scambio (pacchetto UDP) deriva da considerazioni di latenza e compatibilità, non da negligenza.

Il pattern che emerge è sistematico: i metadati di connessione — chi chiama chi, quando, da dove — rimangono accessibili anche nei sistemi più robusti sul piano crittografico. Per l'analisi OSINT, questo significa che la superficie di osservazione non si chiude con la crittografia del payload.

La contromisura operativa è precisa: una VPN attiva sul dispositivo target reindirizza il traffico STUN attraverso il server VPN, rendendo il dato risultante inutilizzabile per l'identificazione geografica o

dell'ISP reale. L'assenza di VPN, rilevabile indirettamente dall'IP restituito, è essa stessa un'informazione sulla postura di sicurezza dell'obiettivo.