CARTE ALBĂ

v1.0.20

Warren

Un VPN no-log construit pe QUIC: identitate non-custodială, rezistență la cenzură și port forwarding-ul pe care îl credeam condamnat

DATA	14 iulie 2026
PERIMETRU	Motivații, arhitectura stivei, motorul WarrenGuard, rețea, identitate, plată, aplicații, model de amenințare, limite și traiectorie
STATUT	Public
VERSIUNE	v1.0.20
AUTOR	Echipa Warren

Rezumat

Warren este un VPN no-log pentru publicul larg, a cărui arhitectură se sprijină pe o decizie radicală: fundația întregii stive este protocolul QUIC, și nu o componentă de tunel dedicată. Din această alegere decurge restul. Traficul se confundă cu HTTP/3 obișnuit, ceea ce dă o rezistență la cenzură nativă, nu adăugată ulterior. Identitatea utilizatorului este o cheie publică derivată local dintr-o frază de recuperare, fără cont și fără adresă de e-mail: serverul nu deține niciun secret, iar un furt al bazei sale de date nu compromite niciun acces. Serverele de ieșire funcționează pe o imagine imuabilă, starea lor de execuție trăiește doar în memoria RAM și nu există nicio partiție de date: o confiscare fizică nu dă nimic.

Confidențialitatea și-a schimbat natura. Colectarea datelor a devenit o infrastructură permanentă și ieftină: ceea ce este captat este păstrat și rămâne exploatabil ani mai târziu, sub alte legi și de către alți actori decât cei de astăzi. În fața unei memorii de asemenea anvergură, prudența individuală nu mai protejează mare lucru atât timp cât datele continuă să existe undeva. Singura protecție care rezistă în timp este absența datelor, și acesta este principiul care guvernează Warren: să facă absente datele sensibile în loc să promită buna lor folosire.

Trei convingeri ghidează proiectul. Prima: no-log-ul are valoare doar dacă este verificabil, nu doar promis. A doua: port forwarding-ul, pe care principalii actori de pe piață l-au abandonat în 2023 sub presiunea abuzurilor, rămâne o funcționalitate legitimă al cărei mecanism trebuie întărit fără a-i restrânge folosirea. A treia: fiecare garanție de securitate trebuie să țină prin construcție, fără vreo căsuță de bifat pe care utilizatorul ar trebui să o găsească. Aceste protecții formează o ofertă unică, fără trepte cu plată: ceea ce variază de la o apărare la alta este starea ei implicită, reglată după costul în performanță.

Documentul de față descrie întreaga stivă, de la motorul criptografic până la planul de control, rămânând la nivel arhitectural. Arată prin ce mecanisme concrete ține fiecare garanție și unde se află granița a ceea ce Warren protejează.

Cuprins

Partea I. De ce Warren

Înainte de arhitectură, motivele. Cât valorează cu adevărat o promisiune no-log, ce spune despre această meserie sacrificarea port forwarding-ului în 2023 și de ce furtivitatea este o condiție de existență.

1. No-log-ul, și de ce vorba nu este de ajuns

Un VPN pentru publicul larg vinde o promisiune simplă: furnizorul dumneavoastră de acces, rețelele publice pe care le traversați și site-urile pe care le vizitați nu vă mai pot urmări. Promisiunea aceasta se sprijină în întregime pe o ipoteză rareori examinată: că furnizorul de VPN, la rândul lui, nu păstrează nicio urmă. Toată industria se sprijină pe acest cuvânt, „no-log", iar aproape toți actorii îl declină într-o politică de confidențialitate pe care nimeni nu o poate verifica.

Slăbiciunea este structurală. O politică este un text; un server este un sistem. Nimic nu îl obligă pe al doilea să îl respecte pe primul, iar utilizatorul nu are niciun mijloc de a observa ce se petrece în realitate în interiorul unui server de ieșire. Piața VPN este o piață a încrederii declarative, iar o piață a încrederii declarative tinde către nivelul cel mai de jos: clasamente sponsorizate, audituri de complezență, promisiuni neverificabile.

Singura ieșire din această capcană constă în a înlocui încrederea cu verificarea și, mai ales, în a face ca datele sensibile **să nu existe**. O informație care nu a fost niciodată scrisă nu poate fi nici confiscată, nici rechiziționată, nici scursă, nici vândută. Istoria recentă a sectorului oferă ilustrarea cea mai limpede: pe 18 aprilie 2023, cel puțin șase polițiști ai poliției naționale suedeze s-au prezentat la birourile Mullvad din Göteborg, cu un mandat de confiscare a mașinilor care conțineau date ale clienților. Aceste date nu existau. După ce Mullvad le-a arătat cum funcționează serviciul, și după consultarea procurorului, poliția a plecat fără să ia nimic. Minimizarea datelor funcționase în același timp ca apărare tehnică **și** juridică.

Mullvad este actorul care a defrișat cel mai serios acest teren: deciziile sale publice constituie cea mai bună documentație disponibilă despre capcanele meseriei, iar o parte din ceea ce construiește Warren constă în a trage învățămintele de acolo.

Warren duce această idee până la capătul consecințelor ei. Nicio identitate reutilizabilă pe partea de server. Niciun jurnal de activitate, nicio adresă IP sursă, nicio cerere DNS, nicio asociere durabilă între o plată și o utilizare. Acolo unde o urmă reziduală este inevitabilă tehnic, ea este mărginită în timp, criptată în repaus cu o cheie absentă din bază, iar durata ei de viață este fixată cu precizie. Un no-log credibil se măsoară prin ceea ce face imposibil, iar documentul de față descrie mecanismele care îl garantează.

2. Mai 2023: când port forwarding-ul a devenit imposibil de apărat

Port forwarding-ul permite unei mașini aflate în spatele VPN-ului să accepte conexiuni de intrare: partajare peer-to-peer în loc de simplă descărcare, găzduirea unui serviciu accesibil din exterior, obținerea unui NAT deschis pentru jocuri online. Este o funcționalitate pe care utilizatorii tehnici o cer și care, multă vreme, i-a deosebit pe furnizorii buni de ceilalți.

A fost abandonată. Pe 29 mai 2023, Mullvad anunța retragerea totală a port forwarding-ului, porturile existente încetând să funcționeze pe 1 iulie 2023. Justificarea, în termenii lor: *„unele persoane au folosit frecvent această funcționalitate pentru a găzdui conținut nedorit și servicii rău intenționate de pe porturile deschise pe serverele noastre. Acest lucru a dus la contacte din partea forțelor de ordine, la trecerea IP-urilor noastre pe liste negre și la rezilierea contractelor noastre de către furnizorii de găzduire."* O lună mai târziu, pe 29 iunie 2023, IVPN făcea același pas, cu retragere completă la 30 septembrie, observând că afluxul de clienți veniți de la un alt furnizor, în urma unei schimbări de politică similare, multiplicase riscul la ei. Abuzul se concentrează întotdeauna asupra ultimului furnizor care ține ușa deschisă.

Dilema este reală: costul abuzului unei funcționalități poate depăși valoarea ei, până la a deveni o chestiune de supraviețuire pentru infrastructură. Dar răspunsul pieței, suprimarea pur și simplu, pedepsește marea majoritate a utilizărilor legitime pentru a se feri de o minoritate dăunătoare. Warren face pariul invers, și îl documentează: restaurăm port forwarding-ul, **întărim mecanismul** împotriva abuzurilor tehnice și a scurgerilor de dezanonimizare, și **nu restrângem utilizarea**. Secțiunea 15 descrie mecanismul; secțiunea 32, limitele.

Un VPN trebuie să își servească utilizatorii mai degrabă decât să se protejeze de ei, iar răspunsul corect la abuz combină măsuri tehnice și juridice în loc să amputeze o funcționalitate.

3. Obfuscarea ca o condiție de existență

Obfuscarea este asociată cu cenzura de stat: Iranul, China, Rusia inspectează traficul în profunzime (DPI) pentru a detecta și bloca protocoalele de VPN. Este adevărat, dar este reducător. Blocarea UDP-ului, sau cea a unei semnături de protocol prea recognoscibile, apare cu mult înaintea aparatului de stat: în Wi-Fi-ul unei cafenele, al unui tren, al unui hotel, al unei companii. Un VPN al cărui protocol se lasă identificat de la primul pachet este un VPN care cade pe o parte importantă a rețelelor de zi cu zi.

Majoritatea protocoalelor de tunel expun o semnătură trivială. Handshake-ul WireGuard, de exemplu, începe într-un pachet de dimensiune fixă cu un octet de tip caracteristic: un echipament de filtrare îl recunoaște fără efort. A face furtiv un asemenea protocol obligă la altoirea unui strat de obfuscare deasupra, de întreținut separat, activat ulterior, deseori rezervat utilizatorilor care se gândesc să îl caute.

Am vrut ca furtivitatea să fie o proprietate a transportului însuși. Acesta este motivul profund al alegerii QUIC (secțiunile 5 până la 8): un tunel QUIC seamănă, pe fir, cu un navigator care încarcă un site în HTTP/3. Un cenzor care ar vrea să îl blocheze ar trebui să blocheze HTTP/3, adică o fracțiune majoră a webului modern, cu pagubele colaterale pe care asta le presupune. Obfuscarea de prim nivel, cea care șterge semnalele vizibile ale protocolului, este la Warren **activă implicit**, fără reglaj, pentru toată lumea. Apărările mai grele, cele care deformează însăși forma traficului, se plătesc în debit și în latență: ele există, sunt la dispoziția fiecăruia, dar se aprind doar dacă utilizatorul le cere. Această linie de despărțire, între ceea ce este permanent pentru că nu are contrapartidă și ceea ce este opțional pentru că are una, structurează întreaga noastră abordare a cenzurii (secțiunile 12 și 14).

NOTĂ · O LINIE DE DESPĂRȚIRE TEHNICĂ, NICIODATĂ COMERCIALĂ

Warren nu are nici trepte, nici opțiuni cu plată, nici funcționalități rezervate celor care ar plăti mai mult. Există o ofertă unică, iar tot ce descrie documentul de față face parte din ea, de la multi-hop la port forwarding, trecând prin apărarea împotriva analizei de trafic. Singurul lucru care se schimbă de la o funcționalitate la alta este starea ei implicită, iar singurul criteriu este arbitrajul între protecție și performanță. Ceea ce nu costă nimic în debit rulează permanent pentru toată lumea; ceea ce costă în performanță rămâne o alegere a utilizatorului și nu apasă niciodată pe factură.

4. Trei proprietăți, o convingere

Din aceste motivații decurg trei proprietăți structurante, care rezumă Warren înaintea oricărui detaliu tehnic.

Nicio identitate reutilizabilă pe partea de server. Utilizatorul este o cheie publică pe care o generează el însuși, niciodată un cont pe care îl deținem noi. Serverul nu cunoaște decât chei publice, inutile singure, și semnături de unică folosință. Un furt al bazei noastre de date nu dă acces la niciun cont, pentru că nu există nimic de furat care să fie un secret.

Niciun jurnal pe serverele de ieșire. Sesiunile active, alocările de porturi, tabelele de traducere a adreselor există doar în memoria RAM. O mașină oprită sau confiscată nu conține nimic despre utilizatorii ei. No-log-ul ține de sistemul însuși, conceput pentru a nu putea jurnaliza durabil.

Port forwarding-ul restaurat. Funcționalitate de prim-plan, cu un mecanism întărit și o utilizare nerestrânsă.

Convingerea de fond este că securitatea și confidențialitatea trebuie să fie **proprietăți de construcție**. Un kill-switch protejează pentru că un socket este mort, și nu pentru că o regulă de firewall a avut timp să se instaleze. Un no-log este garantat pentru că datele nu sunt scrise niciodată, și nu pentru că un script le curăță mai târziu. O identitate este anonimă pentru că protocolul nu transportă numele, și nu pentru că am promis să nu ne uităm la el. Tot restul documentului declină această exigență.

Partea II. Geneza stivei

Cum se ajunge la un VPN clădit pe QUIC: protocoalele înlăturate, ocolul care a costat scump și ce angajează această alegere.

5. De ce nu WireGuard

WireGuard este un protocol excelent: mic, auditat, rapid, integrat în nucleul Linux. A fost punctul de plecare al reflecției noastre.

Trei motive ne-au îndepărtat de el. Mai întâi, un tunel WireGuard gol este blocat sau limitat pe rețelele care filtrează UDP-ul sau îi recunosc semnătura, iar a-l face furtiv impune un strat de obfuscare suplimentar de întreținut. Apoi, WireGuard nu oferă nicio cale nativă către funcționalitățile care formează miezul Warren: multi-hop-ul criptat cap la cap, port forwarding-ul în bandă, apărarea împotriva analizei de trafic. Fiecare ar trebui altoită deasupra, cu propriul canal de control și propria întreținere, pentru că WireGuard nu oferă nici negociere de capabilități, nici canal de control aplicativ, nici extensibilitate a handshake-ului pe care să te sprijini. În fine, identitatea în WireGuard se bazează pe o cheie proprie tunelului, distinctă de identitatea criptografică a utilizatorului: două identități de sincronizat, două suprafețe de corelare, acolo unde noi voiam o singură cheie.

Aceste trei motive converg către aceeași concluzie: proprietățile cerute nu se altoiesc pe WireGuard, ele trebuie purtate de transportul însuși.

6. Ocolul prin Iroh, pivotul către QUIC

Raționamentul ne-a condus către QUIC, protocolul de transport modern pe care se sprijină HTTP/3. Prima tentativă s-a bazat pe Iroh, o stivă peer-to-peer clădită pe QUIC, seducătoare pentru că alinia identitatea nodului pe o cheie publică și promitea traversarea NAT-ului și multipath-ul. La utilizare, ea aducea mai ales greutate: mii de linii nefolosite pentru cazul nostru, o interfață care se schimba în fiecare săptămână, o clasă de bug-uri de traversare a NAT-ului nerezolvată și o coadă lungă de dependențe tranzitive.

În mai 2026 am operat un pivot net: abandonarea Iroh pentru Quinn, implementarea Rust de referință a QUIC, fără suprastratul peer-to-peer. Funcțiile ei de vârf, traversarea NAT-ului și multipath-ul, nu serveau la nimic în topologia Warren, unde un client se adresează unui server de ieșire cunoscut: era greutate moartă pentru acest produs. Migrarea a durat câteva zile și a produs rezultate măsurabile: variabilitate a debitului mult redusă, binare reduse la mai puțin de jumătate, arbore de dependențe adus de la optzeci de crate-uri la aproximativ douăzeci și cinci.

Regula pe care am tras-o de aici se aplică peste tot: de preferat o fundație modestă și înțeleasă în locul uneia ambițioase și opace. O dependență pe care nu o stăpânești este o datorie de securitate la fel de mult ca o datorie tehnică.

7. De ce am făcut fork la Mullvad

A construi un client VPN de calitate de producție pe trei sisteme de birou și două sisteme mobile reprezintă ani de muncă: gestiunea kill-switch-ului la nivelul firewall-ului, prevenirea scurgerilor DNS, rutarea după politici, interfață grafică multiplatformă, integrarea cu extensiile de rețea ale iOS și cu serviciul VPN al Android. Toate acestea există deja, maturizate de ani de producție, în aplicația open source a Mullvad.

Aplicația noastră de birou și mobilă este un fork al ei. Moștenim partea dificilă și verificată (pilotarea sistemului de operare, disciplina anti-scurgere) și înlocuim acolo componenta de tunel: acolo unde clientul de origine ridică un tunel WireGuard, al nostru ridică un tunel QUIC WarrenGuard. Nu rescriem ceea ce funcționează; efortul se concentrează pe valoarea noastră proprie, transportul și planul de control.

8. Pariul QUIC, și prețul lui

QUIC nu este doar un transport mai rapid. Este primitiva din care derivă întreaga stivă Warren, și aduce mai multe proprietăți pe care un VPN le poate exploata direct.

Handshake-ul lui combină stabilirea conexiunii și negocierea TLS 1.3 într-un singur dus-întors. Integrează nativ TLS 1.3, ceea ce dă acces la autentificarea mutuală prin chei publice brute (Raw Public Keys, RFC 7250) fără a fi nevoie să operezi o infrastructură cu chei publice. Transportă datagrame nesigure (RFC 9221), perfecte pentru a vehicula pachete IP care nu au nevoie să fie reordonate de transport. Știe să migreze o conexiune de la o rețea la alta, ceea ce permite trecerea de la Wi-Fi la rețeaua celulară fără ruperea sesiunii. Și, mai ales, se confundă pe fir cu HTTP/3.

QUIC singur nu este un VPN. Warren adaugă deasupra un strat aplicativ propriu: un format de handshake, identitatea prin wallet, criptarea multi-hop, apărarea împotriva analizei de trafic, obfuscarea. QUIC este fundația pe care se construiește tot restul.

Această alegere are un preț. O stivă QUIC în spațiul utilizator criptează acolo unde WireGuard se sprijină pe nucleu, iar QUIC adaugă criptarea numerelor de pachet, protecția antetului și mecanica confirmărilor de primire. Acesta este costul mimetismului HTTP/3, adică al proprietății care poartă tot restul stivei. Ingineria datapath-ului îl recuperează: tunelul susține mai mulți gigabiți pe secundă pe hardware adaptat (secțiunea 10).

DE REȚINUT · CINCI PROPRIETĂȚI PE CARE WARREN LE ȚINE DE LA TRANSPORT

Alegerea QUIC ține de cinci proprietăți de care Warren depinde direct și pe care un tunel WireGuard nu le poate purta fără să i se altoiască.

- **Furtivitate nativă.** Traficul se confundă cu HTTP/3 pe portul 443, iar această furtivitate este o proprietate permanentă a transportului. Un tunel WireGuard expune, dimpotrivă, o semnătură recognoscibilă din primul pachet și nu atinge aceeași discreție decât primind un strat de obfuscare separat, de întreținut.
- **Canal de control în bandă.** QUIC aduce fluxuri fiabile, o negociere de capabilități și un handshake extensibil. Multi-hop-ul sigilat, port forwarding-ul, apărarea împotriva analizei de trafic și jetoanele de sesiune călătoresc în aceeași conexiune. WireGuard nu oferă niciunul dintre aceste puncte de sprijin, iar fiecare dintre aceste funcții ar cere acolo un canal separat.
- **O singură identitate.** TLS 1.3 este integrat în transport: autentificarea mutuală prin chei publice brute, legarea de canal semnată, certificatul de acoperire și sigilarea post-cuantică decurg din el, toate ancorate în cheia utilizatorului. Nicio cheie proprie tunelului nu trebuie sincronizată în paralel.
- **Pachete IP în datagrame.** Datagramele nesigure (RFC 9221) transportă pachetele IP unul pentru unul, fără reordonare inutilă, în timp ce fluxurile fiabile poartă doar handshake-ul aplicativ.
- **Migrarea conexiunii.** O schimbare de rețea, de la Wi-Fi către Ethernet sau către rețeaua celulară, este detectată, iar conexiunea QUIC basculează pe noua cale fără o nouă strângere de mână, în timpul unui dus-întors de revalidare. Identificatorul de conexiune, care înlocuiește cvadruplul de adrese, este reînnoit pe noua cale, deci un observator nu leagă cele două căi prin acest identificator.

Partea III. WarrenGuard, motorul

Miezul tehnic: un singur datapath, în Rust, împărtășit de toate platformele, și mecanismele care îl fac furtiv, etanș și greu de prins în capcană.

9. WarrenGuard: un singur miez, toate platformele

WarrenGuard este motorul Warren: stiva software care implementează tunelul VPN pe QUIC, independent de tot ce este specific produsului Warren. Este o componentă open source (licență AGPL) concepută să fie generică: nu poartă nici politică de abonament, nici cheie de semnare, nici director Warren, și lasă aceste decizii pe seama celui care o desfășoară. Un terț ar putea să o folosească pentru a-și opera propriul VPN pe QUIC.

Această separare este o lege de arhitectură, aplicată strict. Stiva se citește în straturi, iar dependențele nu urcă niciodată înapoi.

DETALII TEHNICE · CELE PATRU STRATURI ȘI LEGEA DEPENDENȚEI

Motorul generic (WarrenGuard) este la temelie. Deasupra, două straturi independente unul de celălalt: **clientul Warren** (identitate, selecția serverelor, fațada aplicativă) și **serverul Warren** (planul de control, aplicarea abonamentului, cheile private). În vârf, **aplicația produs**. Regula: motorul nu cunoaște nimic despre Warren; clientul și serverul depind de motor, niciodată invers; clientul și serverul nu se cunosc niciodată direct; singura lor punte este rețeaua. Această direcție a dependenței este structurală: motorul trăiește în propriul depozit, se compilează singur și nu referențiază niciun cod Warren, astfel încât o dependență inversată nu s-ar compila. Ea garantează că acel cod sensibil, cel care deține cheile și aplică politica, nu poate să se scurgă în codul client.

Există **o singură implementare** a datapath-ului, în Rust, reutilizată peste tot. Aplicația principală (fork-ul Mullvad) și familia de SDK-uri (Rust, Dart, TypeScript) consumă același motor nativ și același fork QUIC. SDK-urile celorlalte limbaje învelesc acest miez nativ de datapath în loc să îl rescrie. O consecință directă pentru rezistența la cenzură: toate aplicațiile Warren, oricare ar fi originea lor, emit exact aceeași semnătură de handshake QUIC. Nu există o aplicație oficială recognoscibilă și aplicații terțe distingibile; există un singur comportament pe fir.

DETALII TEHNICE · QUIC CA TRANSPORT DE VPN: UZUL RĂSPÂNDIT, ȘI ALEGEREA WARREN

Trecerea unui VPN prin QUIC a devenit curentă, în două feluri distincte de cel al Warren. Primul este învelirea: tunelul rămâne WireGuard, învelit în QUIC pentru a se ascunde când rețeaua blochează. Transportul rămâne WireGuard, iar învelișul se plătește în funcții. Mullvad, care a adăugat această obfuscare QUIC în 2025, impune renunțarea la multi-hop și la apărarea împotriva analizei de trafic pentru a o activa.

Al doilea este releul legat de un cont. Apple iCloud Private Relay din 2021, Cloudflare WARP trecut la MASQUE implicit la sfârșitul lui 2024 și Google IP Protection în Chrome tunelează nativ pe QUIC, dar rămân relee legate de un cont de platformă, cu un singur salt pentru WARP, limitate la navigator pentru celelalte două, fără identitate prin cheie, fără port forwarding, fără apărare împotriva analizei de trafic. Handshake-ul lor MASQUE lasă numele de domeniu lizibil în primul pachet, lucru pe care fragmentarea Warren îl dejoacă (secțiunea 12).

Warren face din QUIC fundația unui VPN complet pe un singur datapath: identitate non-custodială prin cheie, multi-hop sigilat pe cheia de ieșire, port forwarding întărit și apărare împotriva analizei de trafic care se execută pe tunelul QUIC însuși.

Granița dintre deschis și închis urmează o linie simplă. Tot codul pe care îl execută utilizatorul, motorul și kitul client, este public și auditabil de către oricine. Rămân în proprietate planul de control, cheile de semnare și flota de servere de ieșire, adică operarea rețelei.

10. Datapath-ul QUIC

Transportul WarrenGuard se sprijină pe Quinn, implementarea Rust de referință a QUIC, din care Warren menține un fork țintit, detaliat în secțiunea 11. Secțiunea de față descrie datapath-ul pe care îl poartă: stabilirea sesiunii, structura traficului, controlul congestiei și gestiunea MTU-ului.

Handshake și autentificare. Tunelul folosește exclusiv TLS 1.3; TLS 1.2 este refuzat. 0-RTT este dezactivat de ambele părți, deliberat: reutilizarea unui secret de sesiune ar permite rejucarea de pachete IP tunelate, o breșă inacceptabilă pentru un VPN. Motorul generic se autentifică prin chei publice brute Ed25519, fără certificat X.509 și fără lanț PKI. Identitatea utilizatorului nu este prezentată în timpul handshake-ului TLS propriu-zis: începând cu versiunea 5 a protocolului, clientul semnează legătura criptografică de canal (o valoare derivată din sesiunea TLS, RFC 5705) pentru a dovedi posesia cheii sale, iar începând cu versiunea 6, serverul de ieșire face la fel în retur. Nu există, așadar, nicio cerere de certificat client pe fir, un semnal care ar trăda protocolul.

În producție, serverele de ieșire prezintă un certificat X.509 valid pentru un domeniu de acoperire anodin, ceea ce face handshake-ul indistinctibil de o conexiune HTTP/3 obișnuită pentru un observator. Un server configurat pentru o piață cenzurată refuză să pornească fără acest certificat, în loc să cadă în tăcere înapoi pe cheile brute: postura de furtivitate este un invariant de pornire.

Structura traficului. Fluxurile fiabile ale QUIC nu transportă decât handshake-ul aplicativ (negocierea adresei de tunel, a MTU-ului, a funcționalităților). Pachetele IP ale utilizatorului călătoresc în datagramele nesigure ale QUIC, unul pentru unul, fără reordonare inutilă. Codificarea mesajelor de control este strictă: un câmp necunoscut provoacă o respingere, nu există compatibilitate ascendentă aproximativă. Versiunile protocolului evoluează în pași neți, niciodată prin mutația unui format existent.

Controlul congestiei și MTU. Controlerul de congestie implicit este BBR, ales pentru că se descurcă mult mai bine decât algoritmii clasici în prezența pierderilor. Cu două procente de pierdere injectată pe un flux, un controler clasic se prăbușește la câțiva megabiți pe secundă, acolo unde BBR menține câteva sute. O gestiune activă a cozii de așteptare, aplicată pe flux asupra datagramelor tunelului, mărginește latența când urcă încărcarea: în loc să lase un tampon să se umfle și întârzierea să crească, pachetele în exces sunt înlăturate devreme, iar fereastra de emisie se dimensionează după debitul realmente disponibil. MTU-ul tunelului este inițializat prudent (1280 de octeți) pentru a evita găurile negre de fragmentare pe căile intercontinentale, apoi sondat în sus dinamic.

DETALII TEHNICE · ADAPTAREA DINAMICĂ A MTU-ULUI

Un tunel traversează căi de dimensiuni variate, iar unele rețele (Wi-Fi încapsulat, legătură prin satelit, rețea feroviară) impun un MTU redus sub valoarea obișnuită. Warren se adaptează pe două straturi, fără a redimensiona vreodată interfața tunel, care rămâne la 1280 de octeți.

Primul strat este descoperirea MTU-ului căii exterioare. Pornind de la pragul prudent de 1280 de octeți, ea sondează în sus când calea nativă o permite și nu coboară niciodată sub 1200 de octeți, minimul pe care QUIC îl garantează cap la cap. Pe replierea TCP, unde stiva poate fragmenta de la sine, această descoperire este dezactivată, iar bugetul intern este plafonat la 1100 de octeți, dimensiunea care traversează acest transport fără să cadă într-o gaură neagră.

Al doilea strat adaptează în direct spațiul util oferit pachetelor utilizatorului, pornind de la acest buget viu. Pentru fluxurile TCP, el rescrie opțiunea MSS a pachetelor de deschidere a conexiunii, astfel încât cele două capete să se dimensioneze singure pentru a încăpea în tunel, transparent și fără să depindă de ICMP. Aceeași ajustare este aplicată pe partea serverului de ieșire, ceea ce repară căile reduse chiar și pentru un client care nu a fost încă actualizat. Pentru celelalte fluxuri, un pachet prea mare este înlăturat, iar emițătorul lui primește un mesaj ICMP real („fragmentare necesară” în IPv4, „Packet Too Big” în IPv6) care poartă dimensiunea corectă, pentru ca propria lui descoperire de MTU să se ajusteze. În fine, aplicația semnalează, pur informativ, când MTU-ul efectiv coboară sub valoarea lui nominală; ajustarea MSS și mesajul ICMP sintetic fac treaba, indiferent dacă această indicație se afișează sau nu.

Performanță măsurată. Pe un server bare-metal 10 GbE (procesor AMD EPYC Zen4), între două mașini din același centru de date, un singur tunel susține stabil în jur de 5,5 până la 7 Gbit/s, adică 55 până la 70 la sută din debitul liniei, fără a satura procesorul de niciuna dintre cele două părți. Aceasta este capacitatea datapath-ului, și depășește cu mult ceea ce cere o utilizare reală. Pe planul încărcării, un server ține peste cinci mii de sesiuni simultane la mai puțin de zece la sută încărcare de procesor: factorul limitativ al unui server de ieșire este lățimea de bandă a legăturii lui, o alegere de dimensionare a rețelei mai degrabă decât o limită a stivei.

11. Fork-ul lui Quinn

QUIC este un protocol vast, iar a rescrie o stivă completă a lui ar fi deopotrivă o prăpastie de inginerie și o regresie de securitate. Warren pleacă deci de la Quinn, implementarea Rust de referință a QUIC, și o face să diverjeze printr-un număr mic de modificări țintite. Baza urmează fidel depozitul oficial: fork-ul este aliniat pe versiunea publicată curentă a Quinn și resincronizat când upstream-ul avansează, fiecare divergență fiind izolată într-un patch documentat care poate fi citit, aplicat sau retras separat.

DETALII TEHNICE · BARIERA ANTI-REGRESIE A FORK-ULUI

Trecerea tăcută înapoi de la fork la versiunea upstream ar face să se piardă câștigul de performanță și obfuscarea. Acest risc este neutralizat printr-o barieră de compilare: codul apelează reglaje care nu există decât în fork, astfel încât o întoarcere la versiunea upstream provoacă o eroare de compilare imediată în loc de o regresie discretă. Fork-ul nu poate fi abandonat din accident, ci doar prin decizie explicită. Fiecare nouă versiune a fork-ului este, de altfel, validată printr-un banc de test comparativ A/B înainte de a fi adoptată.

Divergențele se așază în trei familii.

Obfuscarea handshake-ului. Fork-ul adaugă două reglaje care guvernează punerea în formă a primului schimb: un prag minim de umplere pe pachetul inițial și un plafon pe dimensiunea primului fragment al negocierii TLS, astfel încât handshake-ul să se întindă pe cel puțin două datagrame UDP. Aceasta este ceea ce dejoacă extractorul de nume de domeniu al marelui firewall chinezesc (secțiunea 12), după aceeași idee ca protecția anti-osificare a navigatorului Chrome. Aceste reglaje sunt inerte implicit în upstream; motorul le activează.

Performanța datapath-ului. Trei modificări, fără efect asupra protocolului vizibil. Emisia în loturi este mărită, pentru a împinge mai multe datagrame pe apel de sistem. Tampoanele socket-urilor sunt dimensionate automat la creare, ceea ce evită pierderile peste gigabit pe care le provoacă dimensiunea implicită mică a Linux, și acoperă în trecere o lacună a upstream-ului pe Windows. O cale de emisie în loturi proprie platformelor Apple exploatează apelurile lor de sistem de tratare grupată.

Controlul congestiei și latența. Este familia cea mai substanțială, și corectează defecte reale. Două dintre ele erau moștenite din upstream în controlerul de congestie BBR: pe o conexiune puțin solicitată, ceea ce este cazul unui tunel în repaus, fereastra de congestie putea crește fără limită, observată la o jumătate de gigaoctet pe servere de ieșire de producție. Primul patch începe într-un singur termen de comparație, aliniat pe implementarea Chromium; al doilea restabilește regula de admitere a măsurătorilor de lățime de bandă. Peste ele, o gestiune activă a cozii de așteptare de tip FQ-CoDel înlocuiește coada adâncă implicită pe datagramele tunelului: ea mărginește latența sub încărcare și repartizează echitabil debitul între fluxurile interne multiplexate în tunel, cu prețul unei

fracțiuni din debit în condiții ideale. Un ultim reglaj aduce dimensiunea tamponului de emisie la produsul lățime de bandă-întârziere realmente măsurat al căii, în locul unei constante calculate pentru cazul cel mai defavorabil, ceea ce evită acumularea unor secunde de coadă pe o legătură lentă.

Nu totul este forkuit. Mai multe proprietăți pe care le-am putea crede specifice Warren țin de upstream-ul pur, motorul limitându-se să le configureze: neutralizarea spin bit-ului, dezactivarea 0-RTT, recuperarea după o gaură neagră de MTU. Motorul le alege reglajele fără să atingă codul.

Două dintre aceste patch-uri au vocația de a urca în upstream: fragmentarea pachetului inițial și repararea BBR, ambele conforme cu specificația și utile oricărui utilizator al Quinn. Sunt pregătite pentru o propunere către upstream. Restul ține de reglajul propriu cazului unui tunel VPN, pe care upstream-ul nu l-ar activa implicit, și rămâne așadar în fork.

12. Obfuscarea nativă

Obfuscarea de prim nivel a Warren șterge semnalele după care un echipament de inspecție ar recunoaște un protocol de VPN. Este activă implicit, pentru toată lumea, fără reglaj, iar costul ei în lățime de bandă este mic.

Mai multe semnale sunt neutralizate simultan. Protocolul aplicativ anunțat în handshake este exclusiv `h3`, cel al HTTP/3. Numele de server (SNI) și prima parte a mesajului de deschidere TLS sunt fragmentate pe cel puțin două datagrame UDP: este o apărare precisă împotriva unui extractor de SNI documentat al marelui firewall chinezesc (lucrări prezentate la USENIX Security 2025), care nu reasamblează un mesaj întins pe mai multe datagrame. Spin bit-ul QUIC, care ar putea servi drept fingerprint dacă ar fi constant, este neutralizat. Iar un sondor activ care ar încerca să provoace serverul pentru a-l identifica nu primește decât o închidere generică a transportului, niciodată un cod de eroare specific Warren: octeții proprii protocolului nu sunt emiși decât după un prim mesaj de stabilire valid, pe care un sondor este incapabil să îl producă.

Aceste proprietăți sunt invarianți testați automat: o modificare care le-ar strica face să eșueze integrarea continuă.

DETALII TEHNICE · RAȚIONAMENTUL DE INGINERIE, TRASAT ÎN DECIZIILE DE ARHITECTURĂ

Fiecare alegere de obfuscare este ancorată într-un document de decizie (ADR) care expune amenințarea și arbitrajul. Modelul de adversar reținut include explicit un sondor activ de clasă statală, nu doar o inspecție pasivă. O descoperire a fost structurantă: autentificarea mutuală TLS clasică lăsa să treacă o cerere de certificat client pe fir, invizibilă la inspecția pasivă dar detectabilă de un sondor; versiunea 5 a protocolului a suprimat-o în favoarea semnăturii în bandă. Alta: identitatea prin cheie publică brută a serverului era ea însăși un semnal detectabil, înaintea oricărei alte apărări, ceea ce a motivat trecerea la un certificat de acoperire X.509 în versiunea 6. Ștacheta vizată este precisă: indistinctibil de HTTP/3 în fața unui adversar realist care combină inspecția profundă, sondarea activă și fingerprinting-ul. Paritatea exactă în fața unui adversar global dotat cu învățare automată, care compară octet cu octet trafic de volumul unui torrent, este obiectul șantierului de fingerprint descris în secțiunea 33.

Obfuscarea nu se oprește la acest prim strat. Când tunelul este în repaus, în locul unei bătăi regulate de mesaje de menținere (o cadență pe care un observator o leagă ușor), Warren emite un trafic de acoperire cu intervale și dimensiuni variate, care înecă acest ritm. Iar certificatul de acoperire X.509 este însoțit, pe serverele de ieșire, de un serviciu HTTP de momeală: un sondor care interoghează serverul ca pe un site web primește un răspuns de server web obișnuit, în loc de o eroare care ar trăda un VPN. Aceste apărări rulează fără ca utilizatorul să fie nevoit să le caute.

Rămâne cazul rețelelor care nu filtrează doar o semnătură, ci blochează UDP-ul în întregime: Wi-Fi de companie, portal captiv, rețea încuiată. Un tunel pur QUIC pe UDP ar fi acolo de neatins. Warren basculează atunci **același** tunel QUIC pe o conexiune TCP protejată prin TLS 1.3, către **același** domeniu de acoperire și același port 443. Această repliere este armată implicit: clientul încearcă calea UDP și, dacă aceasta nu răspunde foarte repede, deschide în paralel calea de rezervă TCP, fără reglaj și fără așteptare lungă. Pe fir, conexiunea de rezervă seamănă cu o sesiune HTTPS obișnuită, exact ca și calea nominală. Acolo unde UDP-ul trece, el rămâne preferat, mai rapid; acolo unde este tăiat, tunelul ține oricum.

13. Multi-hop-ul și releul orb

Multi-hop-ul face traficul să treacă prin două servere succesive în loc de unul, astfel încât niciun punct unic să nu cunoască în același timp cine sunteți și ce faceți. Primul nod (releul) vede adresa dumneavoastră IP, dar nu destinația; al doilea (ieșirea) vede destinația, dar nu adresa dumneavoastră IP.

Am înlăturat soluția naivă, un tunel QUIC într-un alt tunel QUIC, pentru că ea suprapune două controale de congestie, două gestiuni de fereastră și două ajustări de MTU, cu o pierdere de performanță de ordinul jumătății. Warren folosește în locul ei două conexiuni QUIC independente, releul recopiind datagramele dintr-una în cealaltă fără să le decripteze.

Releul este **orb prin construcție**. Clientul sigilează conținutul traficului său către cheia publică a serverului de ieșire, conform standardului de criptare hibridă HPKE (RFC 9180). Releul nu deține niciodată cheia de decriptare a serverului de ieșire; el nu manipulează decât octeți opaci. O tentativă de deturnare a unei datagrame către un alt server de ieșire eșuează curat la decriptare, pentru că identificatorul serverului de destinație este legat criptografic de mesaj: nu există niciun „confused deputy” exploatabil.

DETALII TEHNICE · CHEI PER PACHET PE UN CANAL NESIGUR

Criptarea HPKE presupune de obicei un flux ordonat și fiabil. Or, datagramele QUIC se pot pierde și pot ajunge în dezordine. În loc să sufere desincronizarea contorului de secvență, Warren derivă o cheie unică per pachet din contextul HPKE, indexată după epocă și după numărul de secvență, și criptează cu un nonce fix: acest lucru este sigur tocmai pentru că cheia este unică pentru fiecare pachet, aceeași tehnică pe care o folosește QUIC intern. Datele asociate leagă fiecare datagramă de identificatorul serverului de ieșire vizat, ceea ce împiedică orice deturnare.

O infrastructură cu chei pe două niveluri (o rădăcină offline, o cheie operațională) semnează identitatea serverelor, cu o protecție împotriva întoarcerii la o versiune anterioară. Contextul de sigilare al unei sesiuni se rotește frecvent, la fiecare treizeci de minute, astfel încât fereastra în care aceeași cheie protejează trafic rămâne scurtă prin construcție.

Sigilarea însăși este **hibridă post-cuantică**. Schimbului de chei clasic X25519 i se adaugă un mecanism post-cuantic (ML-KEM), fără să îl înlocuiască: confidențialitatea nu poate fi niciodată mai slabă decât cu X25519 singur, chiar dacă unul dintre cele două mecanisme s-ar dovedi într-o zi slab. Această combinație răspunde adversarului care arhivează astăzi trafic criptat pentru a-l decripta în ziua în care un calculator cuantic o va permite. Alegerea post-cuantică este blocată printr-o semnătură, niciodată printr-un bit neautentificat pe care un echipament intermediar l-ar putea șterge pentru a forța o repliere, și acesta este comportamentul implicit.

14. Apărarea împotriva analizei de trafic

Criptarea conținutului nu ascunde **forma** traficului. Dimensiunile pachetelor, intervalele dintre ele, rafalele desenează un fingerprint pe care tehnici de învățare automată știu să îl lege de un site vizitat, chiar prin criptare. Este un unghi de atac împotriva căruia un simplu tunel criptat nu face nimic.

Warren se apără de el cu Maybenot, framework-ul ieșit din lucrările universității din Karlstad și sponsorizat de Mullvad, care îl folosește sub numele DAITA. Principiul: mașini cu stări probabiliste care adaugă padding (pachete false) și întârziere pentru a tulbura forma. Această apărare este cuplată pe fluxul de datagrame QUIC și negociată la fiecare sesiune. Warren înglobează un set de apărări venite din literatură (padding cu debit constant, zgomot aleatoriu și variante). Când un client o cere, serverul de ieșire trage una la sorți pentru acea conexiune, o anunță clientului, apoi o aplică pe traficul pe care îl trimite, în timp ce clientul aplică apărarea anunțată pe traficul pe care îl emite. Cum tragerea la sorți este independentă la fiecare conexiune, doi clienți ai aceluiași server de ieșire prezintă rareori aceeași formă de trafic, ceea ce lărgește evantaiul comportamentelor pe care le vede trecând un observator.

Această apărare se plătește în lățime de bandă: a tulbura forma traficului și a maximiza debitul sunt două obiective opuse. Warren face din ea, așadar, o alegere a utilizatorului, pe un datapath unic și într-o ofertă unică: un întrerupător, în aplicație, activează apărarea împotriva analizei de trafic cu prețul debitului. Stingerea ei păstrează toată obfuscarea de protocol din secțiunea 12, care, la rândul ei, nu costă aproape nimic și rulează permanent. Singurul criteriu de despărțire este arbitrajul între protecție și performanță.

DE REȚINUT · SEMNALUL DE NEGOCIERE NU MINTE

O apărare împotriva analizei de trafic care crede că rulează fără să ruleze este mai rea decât nicio apărare: utilizatorul își adaptează comportamentul la o protecție absentă. De aceea activarea nu este un simplu reglaj local. Clientul cere apărarea, serverul de ieșire răspunde anunțând mașina pe care o aplică, iar aplicația nu afișează „activă” decât la această confirmare. Dacă serverul nu o acordă, interfața o arată clar, atât pe pagina dedicată cât și pe ecranul de conexiune, în loc să lase impresia unei protecții care nu se execută.

15. Port forwarding-ul, întărit

Warren restaurează port forwarding-ul printr-un server NAT-PMP complet (RFC 6886) care rulează pe serverul de ieșire, accesibil doar din interiorul tunelului. Contractul este cel promis de secțiunea 2: utilizare nerestrânsă, mecanism întărit.

Întărirea împotriva abuzului tehnic este aplicată în cod. Fiecare adresă de tunel este limitată la cinci redirecționări simultane. Ritmul de alocare este plafonat la douăsprezece porturi noi pe fereastră glisantă de șaizeci de secunde. Un port eliberat trece printr-o carantină de cinci minute înainte ca un alt client să îl poată reutiliza. Durata de viață a unei redirecționări merge de la șaizeci de secunde la o oră, reînnoibilă. Toate aceste limite eșuează în mod închis: o cerere dincolo de cotă este refuzată explicit. Nicio tabelă persistentă nu leagă o cheie de utilizator de un port: corespondențele sunt efemere și nu există decât în memorie.

Întărirea împotriva dezanonimizării vizează un atac clasic, cunoscut sub numele de „Port Fail”. Dacă adresa de intrare și adresa de ieșire ale unui server sunt identice, un atacator abonat la același serviciu poate, printr-o redirecționare de port, să provoace o scurgere a adresei IP reale a unei victime al cărei trafic către acea adresă ocolește tunelul. Cauza rădăcină este că excluderea de rută către serverul VPN se bazează pe destinație.

DETALII TEHNICE · CORECTAT LA SOCKET, RĂMAS PE O SINGURĂ ADRESĂ IP

Mullvad era imunizat împotriva Port Fail încă din 2010, separând adresa de intrare de adresa de ieșire a serverelor sale. Warren atacă însă cauza rădăcină, nu simptomul: în loc să excludă din tunel *tot* ce merge către adresa serverului, ceea ce face să scape orice pachet către acea adresă, excepția este restrânsă **doar la socket-ul tunelului însuși**, prin primitive de sistem. Este abordarea WireGuard și recomandarea lucrărilor TunnelCrack (USENIX 2023); ea închide Port Fail și varianta TunnelCrack-ServerIP dintr-un singur gest, fără costul unei a doua adrese IP, iar Warren rămâne pe o singură adresă IP. Cum stăpânim integralitatea aplicațiilor noastre, un patch pe partea de client este suficient acolo unde un VPN generic ar trebui să corecteze pe partea de server, iar serverul de ieșire adaugă de altfel propria barieră, în apărare în adâncime.

macOS cere o grijă aparte, pentru că acolo coexistă adesea mai multe interfețe de rețea (Wi-Fi și Ethernet, de exemplu), iar legarea unui socket de una dintre ele poate, în anumite configurații, să îi facă să piardă orice rută de ieșire. Legarea socket-ului este deci dublată acolo de o supraveghere a ieșirii: dacă socket-ul legat încetează să emită cu adevărat, aplicația basculează singură pe o repliere prin rută dedicată, în loc să lase traficul într-o gaură neagră. Protecția se corectează singură în loc să depindă de o configurație de rețea ideală.

16. Kill-switch și fail-closed

Un kill-switch împiedică traficul să se scurgă în afara tunelului atunci când acesta cade. Lecția arhitecturală majoră a incidentelor documentate din sector (TunnelVision, TunnelCrack) este că nu trebuie niciodată să te încrezi în tabela de rutare ca mecanism de securitate: fail-closed-ul trebuie să se sprijine pe un firewall care blochează tot ce nu trece prin tunel.

Warren face din el o proprietate de construcție, nu o reacție. Atât timp cât tunelul poartă trafic, firewall-ul sistemului de operare este în refuz implicit: regula este pusă înaintea primului pachet, și nimic nu iese în afara tunelului pentru că nimic altceva nu este autorizat să iasă.

Defectarea software-ului însuși nu redeschide nimic. Regulile de blocare trăiesc în nucleu și supraviețuiesc procesului: o prăbușire lasă gazda mută în loc să lase să se scurgă, apoi serviciul repornește de la sine, se ridică în poziție blocată și restabilește tunelul fără intervenție. Și, pentru că o protecție care și-ar sechestra propria mașină ar fi o greșeală de inginerie, aplicația păstrează în orice împrejurare o ieșire explicită: un clic, sub elevarea nativă a sistemului, restabilește accesul la internet fără VPN, cu firewall și DNS restaurate.

Pentru cine are nevoie, **kill-switch-ul persistent** face pasul următor. Odată armat, blocarea supraviețuiește opririi demonului, prăbușirii lui și repornirii mașinii: oricât ar fi de totală defectarea, gazda rămâne mută până când utilizatorul dezarmează el însuși protecția. Warren nu îl impune implicit: o blocare persistentă impusă transformă cel mai mic incident software într-o mașină fără rețea, fără recurs, inclusiv pentru dezinstalare. Pe partea serverelor de ieșire, unde niciun om nu este acolo pentru a dezarma ceva, arbitrajul se inversează: binarele lor sunt compilate astfel încât o pană fatală **întrerupe** execuția fără a derula nici cea mai mică curățenie.

Pe sistemele de birou în mod tunel integral, firewall-ul sistemului de operare (nftables pe Linux, pf pe macOS, echivalentul pe Windows) este pus în refuz implicit, autorizând doar loopback-ul, dispozitivul tunel și traficul demonului către serverul de ieșire. Corecția Port Fail ține cu o treaptă mai jos, în rutare: socket-ul tunelului însuși este cel legat, în loc să se excepteze o destinație, ceea ce acoperă atât calea UDP nominală, cât și replierea TCP din secțiunea 12. DNS-ul este împins în tunel, fără cale către resolverul gazdei.

Nivelul de garanție diferă în funcție de mod, iar Warren le distinge pe cele două. Fail-closed-ul aplicat de firewall-ul sistemului de operare ține de modul tunel privilegiat. Modul proxy neprivilegiat oferă un fail-closed de altă natură, structural dar mărginit la aplicația configurată (secțiunea 30). O promisiune de tipul „zero pachet în afara tunelului” nu are sens decât sub un nivel de garanție precis: unele sisteme de operare se exceptează ele însele de la firewall-urile aplicative.

17. Siguranța memoriei și igiena secretelor

Motorul este scris în Rust, cu interdicția codului `unsafe` pusă la nivelul atelierului de compilare. Doar trei crate-uri relaxează această interdicție, fiecare dintr-un motiv precis și documentat: apelul de sistem de legare a socket-ului, interfața cu API-ul de rețea al Windows, deschiderea dispozitivului tunel privilegiat. Peste tot în rest, codul este fără `unsafe`. Această disciplină elimină prin construcție clase întregi de vulnerabilități de memorie.

Criptografia implicită este în întregime în Rust (biblioteca `ring`), fără lanț de compilare C în profilul curent. Materialele secrete (semințe, chei de semnare, fraze de recuperare) sunt șterse din memorie la eliberarea lor, iar niciun tip care poartă un secret nu expune o reprezentare de depanare care l-ar dezvălui: în cel mai bun caz adresa publică. Disciplina no-log este aplicată până la mesajele de eroare și la jurnale: niciodată o cheie publică completă, o adresă IP sursă, un nonce sau o sămânță în clar, de o parte și de alta a graniței dintre limbaje. Un identificator, dacă un fragment este cu adevărat necesar pentru diagnostic, este trunchiat.

18. Wire format-ul ca sursă de adevăr

Protocolul client al Warren este implementat o singură dată în Rust, apoi reexpus celorlalte limbaje prin SDK-uri. Pentru ca un kit TypeScript sau Dart să vorbească exact aceeași limbă ca serverele de producție, compatibilitatea binară trebuie să fie strictă, octet cu octet.

Garantăm acest lucru prin **golden vectors**: un ansamblu de fișiere de referință partajate care îngheață fiecare format (derivarea identității, adresa, semnătura cererii, cadrele de handshake, cadrul multi-hop, directorul semnat) la nivel de bit. Fiecare SDK rejoacă aceleași fișiere. Un format este ancorat prin octeți exacți în loc să fie descris printr-o proză supusă interpretării. A modifica un vector înseamnă a schimba wire format-ul, ceea ce impune urcarea unei versiuni de schemă, niciodată mutarea celei existente. Nu se modifică niciodată un vector ca să treacă un test: o divergență de vector este o regresie reală de protocol. Această rigoare este ceea ce permite mai multor implementări să coexiste fără să derive vreodată, și deci tuturor aplicațiilor să prezinte aceeași semnătură pe fir.

Versionarea protocolului și negocierea capabilităților urmează aceeași logică. Funcționalitățile sunt anunțate printr-o mască de biți, versiunile coabitează curat pe durata unei migrări, iar un nod care nu înțelege o versiune nouă o refuză net în loc să încerce o interpretare aproximativă.

Partea IV. Rețeaua

Ce rulează de partea cealaltă: servere fără nimic de confiscat, un releu orb și un plan de control care știe cât mai puțin cu putință.

19. Serverele de ieșire: nimic de confiscat

Un server de ieșire este locul cel mai sensibil din tot sistemul: acolo traficul redevine clar și acolo coexistă, pe durata unei sesiuni, adresa IP reală a utilizatorului și destinația lui. Toată concepția urmărește ca o confiscare a acestei mașini să nu dea nimic.

Fiecare server de ieșire de producție rulează pe o imagine Alpine Linux imuabilă, instalată de o unealtă care preia controlul unui server gol și îl reconstruiește într-un sistem în doar-citire. Sistemul de fișiere rădăcină este o imagine comprimată în doar-citire (squashfs), suprapusă cu un strat de scriere (un overlay) care nu există decât în memoria RAM. Sistemul aflat în execuție este de unică folosință: o repornire îl șterge integral.

DE REȚINUT · CE PERSISTĂ, ȘI CE NU PERSISTĂ

Nu există **nicio partiție de date**. Singurul lucru care trebuie să supraviețuiască unei reporniri este identitatea nodului (cheia lui de semnare), iar ea supraviețuiește nu printr-un disc inscriptibil, ci pentru că este gravată în doar-citire în imagine și regravată identic la fiecare actualizare. Tot restul (sesiuni active, alocări de porturi, cache DNS, jurnale) trăiește în stratul din memorie și dispare la repornire. O mașină oprită și apoi confiscată dă sistemul de operare în doar-citire și propria cheie de identitate, și nimic despre utilizatori. O mașină pornită și confiscată dă, în cel mai rău caz, sesiunile vii prezente în memorie în clipa confiscării, niciodată un istoric. Nucleul nu scrie niciun core dump: la oprirea unui proces din eroare, sistemul salvează în mod normal toată memoria acestuia într-un fișier, care ar conține aici adresele utilizatorilor și sesiunile lor; această scriere este dezactivată. Jurnalele serviciilor, la rândul lor, trăiesc în memoria RAM și nu ating niciodată discul.

Desfășurarea actualizărilor urmează aceeași logică. Sistemul folosește două sloturi rădăcină (A și B). O actualizare se face fără întrerupere și fără repornire, în doi timpi: mai întâi înlocuirea la cald a binarului pentru procesul în curs, apoi reconstrucția slotului inactiv pentru ca el să servească aceeași versiune la următoarea pornire. Un mecanism de rezervă din initramfs basculează automat pe slotul vechi dacă unul nou nu reușește să pornească. Această imuabilitate are un corolar prețios pentru verificabilitate: ceea ce rulează este exact ceea ce a fost construit, iar orice alterare presupune reconstruirea unei imagini.

20. Anatomia unui server de ieșire

Planul VPN al unui server de producție începe pe un singur port, **443**: socket-ul UDP care poartă QUIC și socket-ul TCP care termină replierea din secțiunea 12. În spatele lor rulează un repartizor unificat care este în același timp releul multi-hop și terminatorul de ieșire. Fiecare conexiune QUIC de intrare poartă, în clar, o etichetă de rutare care indică serverul de ieșire vizat: dacă acesta este nodul însuși, el termină conexiunea local; altfel, o retransmite orbește către un alt nod. Această concepție pliază saltul simplu și saltul dublu pe același datapath, în toate modurile. Pe acest port, un server de ieșire seamănă cu un server HTTP/3: portul 443, protocolul `h3`, un nume de server plauzibil.

Mai multe mecanisme de securitate țin de detalii care fac sau desfac un VPN no-log.

Traducere de adrese anti-corelare. Serverul maschează adresele utilizatorilor în spatele adresei lui, și o face cu o alocare de port sursă **complet aleatorie**. Nucleul nu păstrează portul sursă efemer ales de client în cvintuplul de ieșire: un client care ar reutiliza un port sursă fix nu ar putea face să apară un port de ieșire stabil pe care un observator să îl lege de sesiunile lui.

Resolver DNS fără jurnal. Serverul de ieșire nu jurnalizează nicio cerere DNS, prin construcție. Un resolver recursiv rulează ca serviciu separat, rezolvând el însuși numele cu minimizarea cererii (niciun terț nu vede cererea completă), cu cache-ul fixat în memoria RAM astfel încât numele rezolvate să nu atingă niciodată discul, fără modul de jurnalizare, cu core dump-urile dezactivate.

Model de privilegii. Serverul de ieșire rulează ca utilizator neprivilegiat. Capabilitățile de sistem de care are nevoie (administrarea rețelei, legarea la portul 443) îi sunt acordate ca un set de capabilități ambientale (*ambient capabilities*) după abandonarea privilegiilor de root, niciodată ca niște capabilități atașate fișierului binar.

DETALII TEHNICE · DE CE CAPABILITĂȚI AMBIENTALE, NU CAPABILITĂȚI DE FIȘIER

Nucleul șterge setul de capabilități ambientale la execuția unui binar care poartă capabilități de fișier; iar pe sistemul de fișiere suprapus în doar-citire, capabilitățile de fișier se aplică în mod puțin fiabil. Atașarea de capabilități la binar strica deci legarea la portul 443 în mod intermitent. Modelul capabilităților ambientale, transmis prin abandonarea privilegiilor, este în același timp mai sigur (binarul nu poartă nicio capabilitate în repaus) și mai robust pe acest tip de imagine.

21. Releul orb și directorul semnat

Rolul de releu (primul salt al multi-hop-ului) constă în a transmite text criptat opac de la client către serverul de ieșire ales, fără a deține vreodată cheia de decriptare a acestuia din urmă. Ce poate vedea releul: adresa IP sursă a clientului (el îi este pereche în rețea), eticheta de rutare în clar și octeți opaci. Ce nu poate vedea: textul clar dintre client și serverul de ieșire, destinația sau traficul decriptat. Sigiliul HPKE este adevărata graniță de securitate.

Un nod află de existența celorlalte noduri ale flotei printr-un **director semnat**, recuperat de la un punct de acces autentificat rezervat serverelor de ieșire înregistrate. Fiecare intrare din director este verificată față de cheia operațională înainte ca un releu să accepte să compună către ea: un plan de control compromis nu poate nici să injecteze un server de ieșire rău intenționat, nici să lase o simplă cheie de abonat să enumere adresele serverelor de ieșire.

Lista serverelor publicată aplicațiilor urmează același principiu al minimizării. Este semnată și nu dezvăluie decât ceea ce îi trebuie unei aplicații pentru a alege un server și a compune către el: identificatorul nodului, localizarea lui (țară și oraș, ceea ce vede utilizatorul în listă), o pondere de selecție, punctele lui de intrare și capabilitățile pe care le anunță. A fost slăbită deliberat pentru a nu mai conține nici rolurile nodurilor, nici **adresa de ieșire**, tot atâtea pârghii pe care un cenzor le-ar folosi pentru a enumera și a bloca flota. O aplicație află pe unde se intră, niciodată pe unde se iese.

22. Planul de control

Planul de control este serviciul HTTP unic cu care vorbesc serverele de ieșire și aplicațiile. El gestionează abonamentele, voucherele anonime, plățile, registrul serverelor de ieșire și heartbeat-urile lor, directorul multi-hop, emiterea jetoanelor de sesiune anonime. Nu este niciodată expus direct: este protejat de un reverse proxy.

Asimetria de cunoaștere dintre planul de control și serverele de ieșire este miezul modelului de confidențialitate.

Planul de control știe: care cheie publică este abonată și până când. Este un fapt de autentificare inevitabil, dar fără niciun istoric de activitate. El cunoaște referințele contabile ale plăților, registrul serverelor de ieșire, statistici agregate. Nu știe ce server de ieșire a folosit un abonat, nici destinațiile lui.

Un server de ieșire știe: sesiunile vii din memorie, alocările de adrese de tunel, traficul pe care îl scoate. Nimic din toate acestea nu este scris pe disc.

Releul, în fine, nu vede decât adresa IP a clientului și octeți opaci.

Nicio componentă, singură, nu deține lanțul complet care leagă o identitate de o activitate. Aceasta este proprietatea pe care secțiunea 29 o duce la concluzia ei cu credențialele de sesiune anonime.

23. No-log-ul, verificat de mașină

Pe partea de server, no-log-ul este un ansamblu de controale automate.

Fiecare serviciu poartă un test care parcurge propriul cod sursă și **face să eșueze integrarea continuă** dacă o linie de jurnalizare ar ajunge să interpoleze o cheie publică completă, o adresă IP de client, un nonce sau un secret. No-log-ul este astfel păzit de unelte, nu de vigilența umană.

Reverse proxy-ul își filtrează jurnalele de acces pentru a scoate din ele adresa IP a clientului și ansamblul antetelor, păstrând doar metoda, URL-ul, statutul și durata; el dezactivează complet jurnalizarea pentru URL-urile susceptibile să conțină un secret. Acolo unde un serviciu nu are nevoie de adresa clientului, antetul de adresă transmisă este fixat la o valoare nulă, astfel încât nicio componentă din aval să nu o afle; acolo unde API-ul are nevoie de ea pentru a plafona debitul, o păstrează în memorie, niciodată jurnalizată.

Ceea ce este stocat în baza de date este inventariat într-o matrice de retenție, ea însăși păzită de un test de integrare continuă: o nouă tabelă durabilă nu poate fi livrată fără o linie care documentează ce conține, de ce și pentru cât timp. Datele deliberat niciodată stocate sunt explicite: trafic, DNS, conexiuni, adrese IP sursă, lățime de bandă per utilizator.

NOTĂ · DOUĂ TABELE, ZERO LINII

Schema conține două tabele capabile să jurnalizeze activitate: migrările le creează, deci sunt prezente în baza de producție. Scrierea în ele este în spatele unui întrerupător dezactivat, iar acest întrerupător nu este armat. Un auditor care ar deschide baza ar vedea două tabele goale. Este o proprietate verificabilă, și este forma cea mai puternică pe care o poate lua un no-log: nu o informație pe care promitem să nu o exploatăm, ci o informație care nu este scrisă niciodată.

24. Operarea flotei fără a strica no-log-ul

Actualizarea unei flote de servere de ieșire fără întrerupere și fără fereastră de vulnerabilitate este o problemă de inginerie de sine stătătoare. Warren o rezolvă printr-un plan de control al desfășurării a cărui proprietate de securitate ține de lanțul de semnătură: catalogul versiunilor este semnat offline, cu o protecție împotriva întoarcerii la o versiune anterioară (un contor monoton) și împotriva rejucării (o dată de expirare).

Desfășurarea urmează o mașină cu stări per nod (drenare, basculare, verificare, repunere în serviciu) și o regulă strictă: un singur server canary, cel mai puțin încărcat, trece primul; starea lui de sănătate este comparată cu restul flotei lăsat intact, iar analizorul care dă verdictul este separat de actuatorul care aplică. Abia după validarea canary-ului urmează restul. Drenarea este un veritabil „make-before-break”: serverul aflat în retragere își semnalează starea în bandă, aplicațiile îl exclud din selecția lor, iar supervizorul aplicației stabilește o sesiune nouă înainte de a o închide pe cea veche. O actualizare nu îi întrerupe pe utilizatorii conectați la ea.

25. Blocarea DNS: opt-in, uniformă, publică, fără deturnare

Warren propune o blocare de conținut la nivelul DNS (reclame, trackere, programe rău intenționate și categorii opționale). Concepția ei o ține la distanță de tot ce ar face dintr-un filtru o unealtă de supraveghere sau de cenzură.

Este **opt-in** și dezactivată implicit. Utilizatorul alege categoriile, iar alegerea lui călătorește codificată în adresa resolverului pe care îl folosește în tunel, fără ca serverul să fie nevoit să țină un profil per utilizator. Este **uniformă și publică**: listele de blocare sunt identice pentru toți și publicate într-un depozit deschis, iar nicio țintire per utilizator nu este măcar exprimabilă în sistem, în lipsa oricărei dimensiuni de utilizator în resolver. O schimbare de listă impusă sub constrângere ar apărea în istoricul public al depozitului. Este **fără deturnare**: un domeniu blocat primește un răspuns „acest domeniu nu există” (NXDOMAIN), și nu o redirecționare tăcută către un server care ar observa tentativa. Și este **fără jurnal**, ca tot restul.

Partea V. Identitate, plată și aplicații

Cum îți dovedești un abonament fără cont, cum plătești fără să te legi de propria utilizare și ce garantează cu adevărat aplicațiile pe fiecare sistem.

26. Identitatea: un wallet, nu un cont

Identitatea unui utilizator Warren este o pereche de chei Ed25519 pe care o generează el însuși pe dispozitivul lui, pornind de la o frază de recuperare de douăsprezece cuvinte în standardul BIP39 (*mnemonic*-ul, 128 de biți de entropie). Cheia publică, codificată într-o adresă lizibilă care începe cu `wb`, este identificatorul lui. Cheia privată nu părăsește niciodată dispozitivul, unde este păstrată criptată în repaus de magazinul de secrete al sistemului de operare: keychain pe macOS și iOS, fără sincronizare către cloud; DPAPI pe Windows; magazin de chei hardware pe Android; cheie sigilată la mașină pe Linux.

Nu există nici cont, nici adresă de e-mail, nici parolă. Acest model este **non-custodial** în sensul criptografic al termenului, o proprietate pe care nici Mullvad, nici IVPN, nici Proton nu o oferă până astăzi.

PE SCURT · CE ÎNSEAMNĂ CU ADEVĂRAT „NON-CUSTODIAL”

Luați numărul de cont din șaisprezece cifre al Mullvad, care este totuși ce face piața mai bun în materie de cont fără e-mail: el este generat de server, stocat de el și comparat la fiecare conectare. Este, criptografic, un bearer token pe care îl deține furnizorul, iar un furt al bazei lui ar expune toate conturile în clar. La Warren, perechea de chei este generată local, cheia privată nu părăsește niciodată dispozitivul, iar serverul nu vede decât chei publice (inutile singure) și semnături de unică folosință. Un furt al infrastructurii noastre nu dă acces la niciun cont, o semnătură nu poate fi rejucată, iar utilizatorul își dovedește identitatea fără să își dezvăluie vreodată secretul. Contrapartida este că nu există nicio recuperare pe partea de server: a pierde cele douăsprezece cuvinte înseamnă a pierde abonamentul. Nimeni altcineva nu le deține, nici măcar noi.

Adresa `wb` împrumută un format de codificare din lumea lanțurilor de blocuri (formatul SS58), dar nu există **niciun lanț de blocuri** în Warren, niciun contract inteligent, niciun abonament înscris într-un registru distribuit. Componenta pe care o numim „contract” este un contract în sens software, cel al formatului de schimb dintre client și server; este o bibliotecă, nu un jeton. Abonamentele trăiesc într-o bază de date clasică, în afara lanțului.

27. Dovada abonamentului fără cont

Din moment ce nu există nici sesiune, nici bearer token, cum îi dovedește un client serverului că are dreptul să acționeze? Printr-o semnătură. Fiecare cerere către planul de control poartă patru antete: cheia publică, o semnătură Ed25519, o marcă de timp și un nonce. Mesajul semnat acoperă metoda, calea, marca de timp, nonce-ul și o amprentă a corpului; el nu conține cheia publică însăși. Serverul verifică semnătura, refuză o marcă de timp aflată în afara unei ferestre de șaizeci de secunde și respinge un nonce deja văzut grație unei tabele anti-rejucare.

Un furt al bazei de date dă, în cel mai bun caz, ansamblul cheilor publice abonate și data lor de expirare: nicio adresă de e-mail, niciun nume, nicio adresă IP, niciun jurnal, nicio serie temporală de activitate per cont. Nimic nu permite reconstruirea unei chei private, de vreme ce niciuna nu trece vreodată prin server.

Controlul numărului de dispozitive nu este o legătură durabilă, ci o limită de simultaneitate: un plafon de dispozitive **conectate în același timp** per abonament, aplicat printr-un registru de rezervări cu durată scurtă de viață, care se regenerează singure. Nimic nu este înregistrat durabil; reinstalarea nu costă un loc. Un identificator de dispozitiv aleatoriu, doar în memorie, este tras la fiecare execuție.

28. Plata, decorelată de utilizare

Plata este punctul în care anonimatul unui VPN se câștigă sau se pierde, pentru că a plăti presupune adesea dezvăluirea unei identități reale (un card bancar, un e-mail). Warren decuplează plata de utilizare printr-o ecluză.

Toată verificarea plăților este **auto-găzduită**: verificăm noi înșine semnătura fiecărui eveniment de plată, fără să trecem printr-un intermediar terț care ar vedea în același timp plata și identitatea. Fiecare furnizor (card, lanțuri de blocuri, magazine mobile) este redus la un eveniment neutru care nu poartă decât o sumă, o monedă și o referință opacă: este un tip de date a cărui lipsă de date personale este verificată la compilare.

Mecanismul ecluzei se sprijină pe un voucher anonim. Plata produce un voucher din care se stochează doar hash-ul; acest voucher este apoi schimbat contra unei chei publice, schimbul însuși necerând nicio autentificare, de vreme ce voucherul este singura dovadă. În momentul plății, aplicația trage un identificator de achiziție aleatoriu; furnizorul de plată nu vede decât acest identificator, niciodată cheia publică a utilizatorului; site-ul de plată nu vede niciodată cheia publică; iar planul de control nu află cheia publică decât în clipa schimbului voucherului.

DETALII TEHNICE · JONCȚIUNEA ÎN REPAUS, MĂRGINITĂ ȘI ÎNTĂRITĂ

O joncțiune în repaus leagă o referință de plată de o cheie publică, pentru ca o rambursare sau o contestație să poată tăia efectiv accesul (fără această legătură, un voucher rambursat ar rămâne utilizabil). Ea este întărită în trei feluri. Este criptată în repaus sub o cheie care **nu se află în baza de date**, astfel încât un furt brut al bazei nu permite citirea ei. Linia este ștearsă după douăzeci de zile (nouăzeci pentru plățile mobile). Mărcile de timp sunt trunchiate la zi, iar expirarea rotunjită la miezul nopții următoare, astfel încât nicio durată să nu dezvăluie secunda unui schimb. Combinată cu absența totală a jurnalului de adrese IP, această joncțiune nu dă loc niciunei corelări exploatabile în condiții normale de exploatare.

Cardul bancar este în serviciu, iar canalele cu decorelare mai puternică, Bitcoin, Lightning și Monero, se sprijină pe același voucher anonim: ecluza, și nu mijlocul de plată, este cea care poartă proprietatea de anonimat. Reînnoirea automată este pilotată de client, cardul rămânând la furnizorul de plată și niciodată la noi.

29. Credențialele de sesiune anonime

Ecluza de plată împiedică legarea unei identități reale de o cheie publică. Rămâne o ultimă joncțiune de rupt: cea dintre cheia publică abonată și utilizarea tunelului. Atât timp cât serverul de ieșire vede cheia publică a utilizatorului în momentul conectării, o legătură există, chiar dacă nu este jurnalizată.

Warren o rupe cu credențiale de sesiune anonime bazate pe Privacy Pass (jetoane cu semnătură oarbă RSA, standardele RFC 9578 și RFC 9474). Principiul: planul de control emite jetoane către o cheie publică abonată, dar aceste jetoane sunt orbite, astfel încât jetonul finalizat pe care îl cheltuiește utilizatorul este criptografic imposibil de legat de emitere. Cheile de emitere se schimbă în fiecare oră, derivate dintr-o sămânță, ceea ce garantează că un jeton nu este cheltuibil decât în ora pentru care a fost semnat.

Nicio componentă a Warren nu poate produce o înregistrare care să lege o cheie publică abonată de un server de ieșire, de o sesiune sau de o destinație. Planul de control nu vede decât emiterea, autentificată de wallet. Serverul de ieșire nu vede decât un jeton anonim și un număr de serie de unică folosință. Releul nu vede decât adresa IP și text criptat.

Acest mecanism este în producție pe toate platformele: emitentul de jetoane rulează, serverele de ieșire acceptă protocolul, iar aplicațiile de birou ca și cele mobile își deschid sesiunile cu jetoane anonime. Jetoanele sunt bătute în avans, în fundal și la cadență fixă, niciodată în momentul conectării. Aplicația întreține astfel o rezervă locală de jetoane valide pentru orele care vin; cheile de emitere schimbându-se în fiecare oră, ea păstrează câteva pentru fiecare tranșă orară. Dacă orarul emiterii l-ar urma pe cel al conectărilor, ar dezvălui o legătură între abonament și utilizare; de aceea cele două sunt decuplate. Iar dacă rezerva se găsește epuizată, de exemplu după o perioadă lungă offline, sesiunea se deschide prin semnătura wallet-ului (secțiunea 27) în loc să eșueze: conexiunea reușește întotdeauna, disponibilitatea nu depinde niciodată de stocul de jetoane.

30. Aplicațiile: două căi, aceeași exigență de adevăr

Warren se execută pe macOS, Windows și Linux (aplicație de birou, fork Mullvad), pe Android și iOS (aplicații native), în linie de comandă pentru serverele fără interfață și ca extensie de navigator. Sub toate aceste suprafețe, un singur miez nativ.

Două căi de date coexistă, cu proprietăți de securitate diferite.

Prima este un **datapath proxy neprivilegiat**. O stivă TCP/IP în spațiul utilizator, expusă local ca proxy SOCKS5 sau HTTP CONNECT, termină fluxurile aplicației locale și le împinge ca datagrame QUIC către serverul de ieșire. Nu cere niciun privilegiu pe niciun sistem. Proprietatea lui de fail-closed este structurală, dar de o natură aparte.

DE REȚINUT · FAIL-CLOSED-UL CA PROPRIETATE A CICLULUI DE VIAȚĂ

Portul proxy-ului local nu există **decât atât timp cât tunelul este ridicat**. Aplicația este îndreptată către acest port. Dacă tunelul cade, portul moare, iar conexiunile aplicației eșuează în loc să cadă înapoi pe ruta directă. Nicio regulă de firewall de instalat la timp: este o proprietate a ciclului de viață, fără tunel, fără port, fără scurgere. Această garanție este valabilă per aplicație, pentru aplicația configurată: o aplicație neconfigurată, sau o componentă care ar ocoli proxy-ul, poate să scurgă. Este un nivel de garanție distinct de fail-closed-ul integral al firewall-ului.

A doua este **calea tunelului de sistem**, privilegiată, care captează tot traficul dispozitivului printr-un dispozitiv tunel real, cu rutare implicită scindată, împingerea DNS-ului și kill-switch la nivelul firewall-ului sistemului de operare. Este calea aplicației de birou și a aplicațiilor mobile. Garanția ei de fail-closed este cea a firewall-ului (secțiunea 16), aplicată de sistem.

DNS-ul nu se scurge în niciunul dintre cele două moduri: în modul proxy, rezolvările se fac la serverul de ieșire, în tunel, fără să treacă prin resolverul gazdei; în modul tunel, politica firewall-ului nu autorizează portul 53 decât către resolverul tunelului, a cărui adresă nu este rutabilă decât prin tunel.

„Conectat" nu înseamnă „protejat". Distincția vine dintr-un caz concret: un server de ieșire aflat în retragere poate continua să răspundă pachetelor de menținere a tunelului, astfel încât conexiunea QUIC pare perfect vie deși el nu mai lasă nimic să treacă. Interfața ar afișa „conectat" în timp ce utilizatorul nu mai are niciun acces la internet.

Supravegherea transportului nu este deci suficientă; trebuie exersat datapath-ul cap la cap. Aplicația deschide periodic o conexiune reală **prin** tunel, către o țintă exterioară: ea nu poate reuși decât dacă datapath-ul transportă cu adevărat octeți dintr-un capăt în celălalt. Când eșuează de mai multe ori la rând, sesiunea este declarată moartă, iar aplicația se reconectează în altă parte.

Este un principiu care guvernează toată concepția aplicațiilor Warren: starea afișată decurge din traficul care trece cu adevărat, măsurat, și nu dintr-un fanion pe care software-ul se mulțumește să îl ridice.

31. Forumul, fără e-mail și fără adresă IP

Forumul comunitar al Warren duce aceeași logică până la autentificare. Acolo îți dovedești identitatea printr-o semnătură a wallet-ului, exact aceeași cerere canonică precum peste tot în altă parte, navigatorul neatingând niciodată cheia. Forumul primește un identificator împerecheat, un e-mail sintetic, și niciodată o parolă, un e-mail real sau o adresă IP de client. Numele de utilizator public este un șir pronunțabil derivat dintr-o amprentă a cheii, determinist pentru ca suportul să poată recalcula legătura, dar neinversabil și necorelabil cu adresa publică. Coloana care conținea cheia publică în clar a fost de altfel retrasă din schemă.

Partea VI. Securitate și limite

Ce protejează Warren, ce nu protejează și cine controlează ce.

32. Model de amenințare sintetic

Un sistem de securitate se judecă după ce protejează, după ce protejează parțial și după ce nu protejează. Iată sinteza pentru Warren.

Garanții dobândite. Furnizorul dumneavoastră de acces și rețelele pe care le traversați nu văd decât un trafic confundat cu HTTP/3 către un server de ieșire, niciodată destinațiile dumneavoastră. Site-urile pe care le vizitați văd adresa serverului de ieșire, nu pe a dumneavoastră. Un furt al bazei de date a planului de control nu dă niciun acces și niciun istoric. O confiscare fizică a unui server de ieșire nu dă nimic despre utilizatori. Un releu multi-hop este criptografic orb la conținut. Sesiunile se deschid cu jetoane anonime, pe toate platformele: nicio componentă nu poate lega abonamentul dumneavoastră de utilizarea dumneavoastră. No-log-ul pe partea de server este păzit de teste care blochează integrarea continuă.

Garanții parțiale sau condiționate. Obfuscarea învinge inspecția profundă și sondarea activă realiste, dar nu încă paritatea exactă de fingerprint în fața unui adversar care compară octet cu octet traficul cu cel al unui navigator real (secțiunea 33). Apărarea împotriva analizei de trafic protejează forma traficului atunci când utilizatorul o activează; stinsă, protecția se referă la conținut și la recognoscibilitatea protocolului, nu la formă. No-log-ul este garantat de arhitectură, de codul deschis și de teste care păzesc fiecare serviciu.

În afara razei de acțiune, prin natură. Warren nu protejează împotriva unui adversar pasiv global capabil să observe ambele extremități și să coreleze prin timp. Nu protejează împotriva unui program rău intenționat de pe dispozitivul dumneavoastră. Nu protejează împotriva fingerprinting-ului navigatorului dumneavoastră (VPN-ul maschează adresa IP, nu amprenta navigatorului). Nu protejează împotriva unei coaliții care ar controla în același timp releul și serverul de ieșire ale circuitului dumneavoastră.

Rămâne întrebarea cine controlează ce. Sunt **controlate central**: planul de control, cheia de semnare a versiunilor (păstrată offline) și certificatul de acoperire. Sunt **cu adevărat descentralizate**: identitatea utilizatorului (non-custodială, pe dispozitivul lui), identitatea fiecărui server de ieșire și datapath-ul din memoria RAM a serverelor de ieșire.

DE REȚINUT · MODELUL DE AMENINȚARE ÎN TREI FRAZE

Warren garantează că datele sensibile nu există acolo unde ar putea fi confiscate și că traficul nu se lasă identificat sau blocat în mod trivial. Nu garantează anonimatul absolut în fața unui adversar care observă în același timp accesul dumneavoastră la internet și ieșirea rețelei și le suprapune, nici în fața unui program care vă controlează dispozitivul. Niciun VPN nu o poate face, iar un VPN care pretinde asta minte.

33. Traiectorie

Warren este în producție, iar arhitectura lui a fost croită pentru a absorbi componente noi fără să fie rescrisă. O stivă de confidențialitate care încetează să avanseze este o stivă care se lasă ajunsă din urmă: iată unde bate efortul.

Paritatea de fingerprint. Serverele de ieșire prezintă un certificat de acoperire obișnuit, handshake-ul este fragmentat pentru a dejuca extractoarele de nume de domeniu, iar amprenta tunelului este supravegheată în integrare continuă, la fiecare commit, pentru a interzice orice derivă. Obiectivul următor este cel mai exigent din domeniu: a face amprenta QUIC a Warren **identică octet cu octet** cu cea a unui navigator de larg consum. Niciun furnizor nu a atins-o, iar uneltele necesare nu există încă în Rust: le construim.

Atestarea imaginii serverelor. Imaginea serverelor de ieșire este imuabilă (secțiunea 19). Lucrăm la a face din ea o dovadă opozabilă: o semnătură a amprentei ei, controlată la pornire, ar face să fie refuzată orice imagine neoficială. Ea este cea care va condiționa deschiderea rețelei către operatori terți: fără atestare, „descentralizat” ar însemna „servere necunoscute pe care nu le putem verifica”, un recul deghizat în progres. Preferăm să descentralizăm atunci când va fi verificabil.

Concluzie

Warren este rezultatul unei suite de convingeri duse până la consecințele lor tehnice. No-log-ul este o proprietate de construcție: de acolo vin identitatea non-custodială, serverele de ieșire în memoria RAM, credențialele de sesiune anonime, no-log-ul păzit de unelte. Rezistența la cenzură este nativă, purtată de transportul însuși: de acolo vin alegerea QUIC ca fundație și obfuscarea permanentă. A-l servi pe utilizator primează asupra protejării de el: de acolo vine port forwarding-ul restaurat și întărit, acolo unde Mullvad și IVPN l-au suprimat în 2023.

Aceste proprietăți au un cost. Mimetismul HTTP/3 cere o stivă QUIC în spațiul utilizator, și am scris-o. Minimizarea datelor cere o infrastructură concepută să nu rețină nimic, și am clădit-o. Ceea ce rămâne în față, paritatea perfectă de fingerprint, arhitectura o așteaptă fără să trebuiască refăcută.

Codul face dovada. Motorul și kitul client sunt deschise și auditabile, iar proprietățile descrise aici sunt verificabile direct acolo. Acolo se judecă ceea ce afirmă documentul de față.

Anexa A. Glosar

BBR: algoritm de control al congestiei bazat pe estimarea lățimii de bandă și a întârzierii, robust la pierderi.

BIP39: standard de reprezentare a unei semințe criptografice sub forma unei liste de cuvinte memorabile.

DAITA: apărare împotriva analizei de trafic prin adăugare de padding și de întârziere, care tulbură forma traficului.

Datapath: calea pe care o urmează pachetele utilizatorului, de la dispozitivul tunel până la serverul de ieșire. De deosebit de planul de control, care gestionează abonamentele și descoperirea serverelor.

DPI (Deep Packet Inspection): inspectarea conținutului pachetelor de către un echipament de rețea, pentru a identifica sau a bloca un protocol.

Ed25519: schemă de semnătură cu curbă eliptică, rapidă și sigură, folosită pentru identitatea Warren.

Fail-closed: comportamentul unui sistem care, în caz de defectare, blochează în loc să lase să treacă.

Fingerprint: semnătură observabilă a unui protocol sau a unui software (dimensiuni de pachete, ordinea extensiilor TLS, cadență), care permite identificarea lui chiar și criptat.

Golden vectors: fișiere de referință care îngheață un format binar la nivel de bit, rejucate de fiecare implementare pentru a garanta că toate vorbesc exact același protocol.

Handshake: schimbul inițial care stabilește o conexiune, negociază criptarea și autentifică părțile.

Heartbeat: mesaj periodic prin care un server semnalează planului de control că este viu.

HPKE (RFC 9180): criptare hibridă cu cheie publică, folosită pentru a sigila traficul multi-hop cap la cap.

Kill-switch: mecanism care împiedică orice scurgere de trafic în afara tunelului atunci când acesta cade.

Momeală (*decoy*): serviciu fals (aici un veritabil server HTTP/3) prezentat unui sondor pentru ca acesta să nu deosebească VPN-ul de un site obișnuit.

MTU: dimensiunea maximă a unui pachet pe o cale de rețea dată.

NAT-PMP (RFC 6886): protocol de deschidere de port printr-o traducere de adrese, folosit pentru port forwarding.

No-log: absența jurnalizării activității utilizatorilor.

Non-custodial: model în care secretul de identitate este deținut de utilizator singur, niciodată de furnizor.

Overlay: strat de scriere suprapus peste un sistem de fișiere în doar-citire. La Warren el nu există decât în memoria RAM, deci nimic nu persistă la repornire.

Padding: octeți de umplere adăugați unui flux pentru a masca dimensiunea reală a datelor.

Privacy Pass (RFC 9578, RFC 9474): jetoane cu semnătură oarbă care permit dovedirea unui drept fără a fi identificabil.

QUIC: protocol de transport modern peste UDP, fundația HTTP/3, care integrează TLS 1.3.

Quinn: implementarea Rust de referință a QUIC, din care Warren menține un fork.

Raw Public Key (RFC 7250): autentificare TLS prin cheie publică brută, fără certificat X.509 și fără PKI.

Reverse proxy: server plasat în fața unui serviciu pentru a primi traficul în locul lui (aici, el filtrează jurnalele de acces și nu expune niciodată serviciul direct).

Slot A/B: cele două amplasamente rădăcină ale unui server de ieșire. Se actualizează amplasamentul inactiv, se basculează la pornirea următoare și se revine la cel vechi dacă cel nou eșuează.

Spin bit: bit de antet QUIC destinat măsurării rețelei; Warren îl neutralizează pentru că valoarea lui ar putea servi drept fingerprint.

Squashfs: sistem de fișiere comprimat în doar-citire, folosit ca rădăcină imuabilă a serverelor de ieșire.

SS58: format de codificare a adreselor împrumutat din ecosistemul Substrate, folosit aici ca simplă codificare, fără lanț de blocuri.

Voucher: bon anonim produs de o plată, din care se stochează doar hash-ul și care se schimbă apoi contra unui abonament fără a dezvălui identitatea plătitorului.

WarrenGuard: motorul VPN pe QUIC al Warren, generic și open source.

Wire format: definiția binară exactă a unui mesaj schimbat în rețea. Este contractul dintre implementări, înghețat de golden vectors.

Anexa B. Referințe

Surse primare ale retragerii port forwarding-ului:

- Retragerea port forwarding-ului la Mullvad, 29 mai 2023, porturi tăiate pe 1 iulie 2023. `https://mullvad.net/en/blog/removing-the-support-for-forwarded-ports`
- Retragerea progresivă a port forwarding-ului la IVPN, 29 iunie 2023, retragere completă pe 30 septembrie 2023. `https://www.ivpn.net/blog/gradual-removal-of-port-forwarding/`
- Percheziție cu mandat la sediul Mullvad, 18 aprilie 2023, plecare fără a confisca nimic. `https://mullvad.net/en/blog/mullvad-vpn-was-subject-to-a-search-warrant-customer-data-not-compromised`

Standarde:

- RFC 6886, NAT Port Mapping Protocol (NAT-PMP)
- RFC 7250, Raw Public Keys in TLS
- RFC 5705, Keying Material Exporters for TLS
- RFC 9000 și RFC 9221, QUIC și datagramele nesigure
- RFC 9180, Hybrid Public Key Encryption (HPKE)
- RFC 9312, proprietăți operaționale ale QUIC (spin bit)
- RFC 9474 și RFC 9578, semnături oarbe RSA și jetoane Privacy Pass

Lucrări academice și industriale:

- Analiza extractorului de SNI QUIC al marelui firewall chinezesc, USENIX Security 2025
- TunnelCrack, dezanonimizare și ocolire de tunel VPN, USENIX Security 2023
- TunnelVision (CVE-2024-3661), deturnare de rută prin DHCP
- Maybenot, framework de apărare împotriva analizei de trafic, universitatea din Karlstad
- Literatură despre fingerprinting-ul site-urilor web (PETS, CCS)

Peisajul QUIC (transport de VPN):

- Mullvad, obfuscare QUIC (MASQUE) pentru WireGuard, 2025. `https://mullvad.net/en/blog/introducing-quic-obfuscation-for-wireguard`
- Cloudflare, WARP peste MASQUE ca transport implicit, 2024. `https://blog.cloudflare.com/zero-trust-warp-with-a-masque/`
- Apple iCloud Private Relay, tunel HTTP/3 și MASQUE, 2021. `https://www.apple.com/icloud/docs/iCloud_Private_Relay_Overview_Dec2021.pdf`
- Google IP Protection (MASQUE și jetoane Privacy Pass) în Chrome. `https://github.com/GoogleChrome/ip-protection`