NetWitness Network – encrypted traffic

NDR does not loose visibility

NetWitness Network is a module of the NetWitness Platform designed for deep visibility and analysis of network traffic to detect threats. It captures full packets or metadata from the network, automatically classifies and correlates them, and enables analysts to quickly identify anomalies, attacks, and malicious activity. The solution provides access to hundreds of metadata fields that support the detection of both potential attacks and attacks already in progress — allowing analysts to investigate and respond much faster than with traditional IDS/IPS or firewalls.

When monitoring encrypted traffic, the functionality of NDR solutions is limited. This is not due to limitations of the solutions themselves, but rather to the technical limitations of encryption. After all, we use encryption precisely to prevent unauthorized parties from accessing the payload.

NetWitness Network works with many **SSL/TLS decryptors**, such as F5 BIG-IP, Broadcom, Palo Alto, Cisco, Keysight (Ixia), and Gigamon. If traffic decryption is not possible or available, NetWitness Network will operate based on what is visible in the encrypted packets.

Encrypted traffic visibility

- TLS/SSL metadata: TLS version, cipher suites, ClientHello/ServerHello extensions. This enables fingerprinting (JA3 / JA3S/ JA4).
- JA3/JA3S/JA4 fingerprints — unique hashes from TLS handshake used for application/malware classification.
- SNI (Server Name Indication) and basic certificate information (e.g., issuer, subject, validity period) — when SNI is sent in clear text in Client Hello.
- Session telemetry: IP addresses, ports, session start/end times, packet/transfer sizes, number of sessions, traffic direction — useful for detecting behavioral anomalies.

Examples of threats that can be detected in encrypted traffic

Command & Control (C2) — TLS beaconing
Visibility: repetitive, periodic connections to the same host/IP (beacons), small data transfer, unusual JA3/JA3S fingerprints.
Metadata: client_ip, server_ip, duration, bytes_in/out, flow_count, ja3, sni.

Ransomware communication with C2 via HTTPS
Visibility: large outbound uploads (encrypted archives), short sessions to multiple domains, self-signed certificates/unusual issuers.
Metadata: bytes_out > bytes_in, cert_issuer, cert_self_signed, sni, ja3.

Data exfiltration over HTTPS (large uploads)
Visibility: unusual large outgoing sessions, long-lasting transfers, transfers outside normal hours.
Metadata: bytes_out, duration, time_of_day, server_country.

DGA / fast-flux C2 (malicious rotating domains)
Visibility: many different domains in a short time, low TTL, short sessions, JA3 fingerprint matching known malware.
Metadata: dns_qry, dns_ttl, server_ip diversity, ja3.

Malicious TLS cert anomalies (compromise / spoofing)
Visibility: recently generated certificates, self-signed, issuer not matching the organization, Let's Encrypt certificates used by unusual users.
Metadata: cert_subject, cert_issuer, cert_valid_from, cert_valid_to.

Domain fronting / CDN abuse
Visibility: SNI indicates a legitimate service, but tracked request body/paths (when decrypted) or anomalies in host distribution, links to suspicious backend IPs.
Metadata: sni, server_ip, http_host (if available), ja3.

TLS tunneling / SSL VPN abuse (e.g., covert channels)
Visibility: long, uniform sessions with high transfer in both directions, unusual TLS ports, unusual packet fragmentation.
Metadata: server_port, bytes_in/out, packet_size_stats, duration.

Malware using popular services (Dropbox/OneDrive/Google Drive) for exfiltration
Visibility: allowed hosts but unusual patterns (not used by the user), large amounts of uploads, new accounts/suspicious tokens (if available).
Metadata: sni, bytes_out, user_agent (if visible), session_owner.

Encrypted lateral movement (e.g., SMB over TLS, DoH/DoT abuse)
Visibility: unusual internal connections over TLS to hosts on the LAN, increasing volume from endpoint hosts.
Metadata: internal_ip_pairs, server_port (853, 443), bytes_in/out.
Filter example: where src_internal and dst_internal and tls=true and bytes_out > 1000000

Phishing / credential harvesting via HTTPS
Visibility: users connecting to sites with suspicious SNI/host, often a short series of POSTs, sudden spike in logins to external services.
Metadata: sni, http_method=POST (if decrypted), bytes_in small, referrer.

Certificate reuse / JA3 mismatches (application pretending to be a browser)
Visibility: JA3 fingerprint inconsistent with a typical client (e.g., JA3 indicates curl/libcurl but SNI/UA say "Chrome").
Metadata: ja3, user_agent, sni, cert_subject.

TLS-based exploit attempts (e.g., Heartbleed-style / malformed TLS handshakes)

Visibility: unusual TLS handshakes, repeated handshake failures, anomalies in cipher suites.
Metadata: tls_alerts, handshake_failures, cipher_list.

Use of unusual cipher suites / deprecated TLS versions
Visibility: client or server uses TLS1.0/SSLv3 or weak ciphers → potential exploit windows.
Metadata: tls_version, cipher.

Beaconing to compromised cloud services
Visibility: regular short connections to different hosts in the same cloud (e.g., AWS IP range), or use of cloud functions for C2.
Metadata: server_asn, server_ip_range, periodicity, ja3.

Suspicious client certificates (mutual TLS abuse)
Visibility: client presents an unusual certificate, new CN, certificates used outside the normal user profile.
Metadata: client_cert_subject, client_cert_issuer, client_cert_valid_from.

How to leverage the power of NetWitness Network with encrypted traffic?

Lists of known and potentially dangerous JA3/JA4 fingerprints are publicly available as feeds. NetWitness can use them to maintain a constantly updated database of dangerous fingerprints.

In accordance with the previously presented attack scenarios that can be detected in encrypted traffic, a range of metadata can be used as a basis for creating application and correlation rules.

Summary

The fact is that an increasing proportion of network traffic is encrypted in point-to-point mode. Without decryption, full payload analysis will not be possible. This partially reduces the ability of NDR-class systems to accurately detect threats, but it does not render them useless.

On the contrary, detecting threats in encrypted traffic is becoming a key element of modern security architecture. Detecting potentially dangerous connections or anomalies at the earliest possible stage of an attack remains crucial for infrastructure security. Regardless of whether the solution is able to look into the payload or not, the above examples show that NDRs are still effective in analyzing traffic. The only thing that is changing is the approach to monitoring, with a greater focus on metadata that can be normalized from encrypted traffic and basing detection mechanisms on this metadata.

About NetWitness

NetWitness provides comprehensive, yet flexible, and highly scalable threat detection, investigation and response capabilities for the largest organizations worldwide. The NetWitness Platform delivers complete visibility combined with applied threat intelligence and user behavior analytics to detect, prioritize, investigate threats, and automate response. This empowers security analysts to be more efficient and stay ahead of business-impacting threats.

Ready to learn more? Visit www.netwitness.com