Welches Zugriffsprotokoll wird für HTTPS-Datenverkehr angemeldet?

Inhalt

Unterstützt von Kei Ozaki und Siddharth Rajpathak, Cisco TAC Engineers.

Frage:

Welches Zugriffsprotokoll wird für HTTPS-Datenverkehr angemeldet?

Umgebung: Cisco Web Security Appliance (WSA) mit AsyncOS 7.1.x und höher, HTTPS-Proxy aktiviert

Die Art und Weise, wie die Cisco Web Security Appliance (WSA) HTTPS-Datenverkehr protokolliert, unterscheidet sich von normalem HTTP-Datenverkehr. HTTPS-Einträge, die in Accesslogs aufgezeichnet werden, sehen je nach Behandlung der Anforderung unterschiedlich aus. Im Allgemeinen hat es andere Eigenschaften als normaler HTTP-Datenverkehr.

Welche Daten protokolliert werden, hängt davon ab, welchen Bereitstellungsmodus Sie verwenden (expliziter Weiterleitungsmodus oder transparenter Modus).

Sehen wir uns zunächst einige Schlüsselwörter an, die Ihnen helfen, Zugriffsprotokolle einfach zu lesen.

TCP_CONNECT - Zeigt an, dass der Datenverkehr transparent empfangen wurde (über WCCP oder L4-Umleitung ...usw.)
CONNECT - Zeigt an, dass der Datenverkehr explizit empfangen wurde.
DECRYPT_WBRS - Dies zeigt, wie die WSA beschlossen hat, den Datenverkehr aufgrund der WBRS-Bewertung zu entschlüsseln.
PASSTHRU_WBRS - Dies zeigt, wie die WSA beschlossen hat, den Datenverkehr aufgrund der WBRS-Bewertung zu passieren.
DROP_WBRS - Hier sehen Sie, wie die WSA beschlossen hat, den Datenverkehr aufgrund der WBRS-Bewertung zu verwerfen.

- Wenn **HTTPS-**Datenverkehr entschlüsselt wird, protokolliert die WSA zwei Einträge.
- **TCP_CONNECT** oder **CONNECT** je nach Art der Anfrage und **"GET https://"** mit der entschlüsselten URL.
- Die vollständige **URL** wird nur angezeigt, wenn die WSA den Datenverkehr entschlüsselt.

Beachten Sie außerdem:

- Im transparenten Modus wird der WSA zunächst nur die Ziel-IP-Adresse angezeigt.
- Im expliziten Modus wird der Zielhostname für die WSA angezeigt.

Im Folgenden finden Sie einige Beispiele, die Sie in den Accesslogs sehen würden:

Transparent - Entschlüsseln
1252543170.769 386 192.168.30.103 TCP_MISS_SSL/200 0 TCP_CONNECT tunnel://192.168.34.32:443/ - DIRECT/192.168.34.32 - DECRYPT_WBRS-DefaultGroup-test.id-NONE-NONE-DefaultRouting <Sear,5.0,-,-,-,-,-,-,-,-,-,-,-,-,-,-,-,-,-,-> - 1252543171.166 395 192.168.30.103 TCP_MISS_SSL/200 2061 GET https://www.example.com:443/sample.gif - DIRECT/192.168.34.3 2 image/gif DEFAULT_CASE-test.policy-test.id-NONE-NONE <Sear,5.0,0,-,-,-,-,-,-,-,-,-,-,-,-,-,-,-> -
Transparent - Passthrough
1252543337.373 690 192.168.30.103 TCP_MISS/200 2044 TCP_CONNECT tunnel://192.168.34.32:443/ - DIRECT/192.168.34.32 - PF ASSTHRU_WBRS-DefaultGroup-test.id-NONE-NONE-DefaultRouting <Sear,9.0,-,-,-,-,-,-,-,-,-,-,-,-,-,-> -
Transparent - Verwerfen
1252543418.175 430 192.168.30.103 TCP_DENIED/403 0 TCP_CONNECT tunnel://192.168.34.32:443/ - DIRECT/192.168.34.32 - P_WBRS DefaultGroup-test.id-NONE-NONE-DefaultRouting <Sear,-9.1.0,-> -
Explicit - Entschlüsseln
252543558.405 385 10,66.71.105 TCP_CLIENT_REFRESH_MISS_SSL/200 40 CONNECT tunnel://www.example.com:443/ - DIRECT/www.example - DECRYPT_WBRS-DefaultGroup-test.id-NONE ONE-DefaultRouting <Sear,5.0,-,-,-,-,-,-,-,-,-,-,-,-,-,-> 1252543559.535 1127 10,66.71.105 TCP_MISS_SSL/200 2061 GET https://www.example.com:443/sample.gif - DIRECT/www.example.com image/gif DEFAULT_CASE-test.policy-test.id.id KEINE-KEINE <Sear,5.0,0,-,-,-,-,-,-,-,-,-,-,-,-,-,-,-> -
Explicit - Durchlassen
1252543491.302 568 10,66.71.105 TCP_CLIENT_REFRESH_MISS/200 2256 CONNECT tunnel://www.example.com:443/ - DIRECT/www.example - PASSTHRU_WBRS-DefaultGroup-test.id NONE-NONE-DefaultRouting <Sear,9.0,-,-,-,-,-,-,-,-,-,-,-,-,-,-> -
Explicit - Drop
1252543668.375 1 10.66.71.105 TCP_DENIED/403 1578 CONNECT tunnel://www.example.com:443/ - NONE/- - DROP_WBRS-DefaultGroup-test.id-NONE-NONE-NONE <Sear, 1,-,-,-,-,-,-,-,-,-,-,