Incident Report

Incidentbericht: Verteilte HTTPS-Überlastung des Apache-Reverse-Proxys und OPNsense-WebUI-Störung

> **Zeitraum der beobachteten Ereignisse:** 13.–17. November 2026 **Dokumentation erstellt:** 18. November 2026 **Systeme:** Apache-Reverse-Proxy `www` (`192.168.242.44`), vorgeschaltete OPNsense, GitLab, Cloud/Nextcloud und weitere HTTPS-Dienste **Status am Ende der vorliegenden Untersuchung:** HTTPS-Überlastung **nicht nachweislich beendet**; OPNsense-WebUI nach Paketaktualisierung **wieder funktionsfähig**; lokaler Root-Konsolenzugang **zuletzt ungeklärt/nicht möglich**. **Belegbasis:** ausschließlich die in der Unterhaltung genannten Konfigurationen, Kommandoausgaben, Logauszüge und übermittelten OPNsense-Regel-/Statusangaben. Keine ergänzten externen Messdaten.

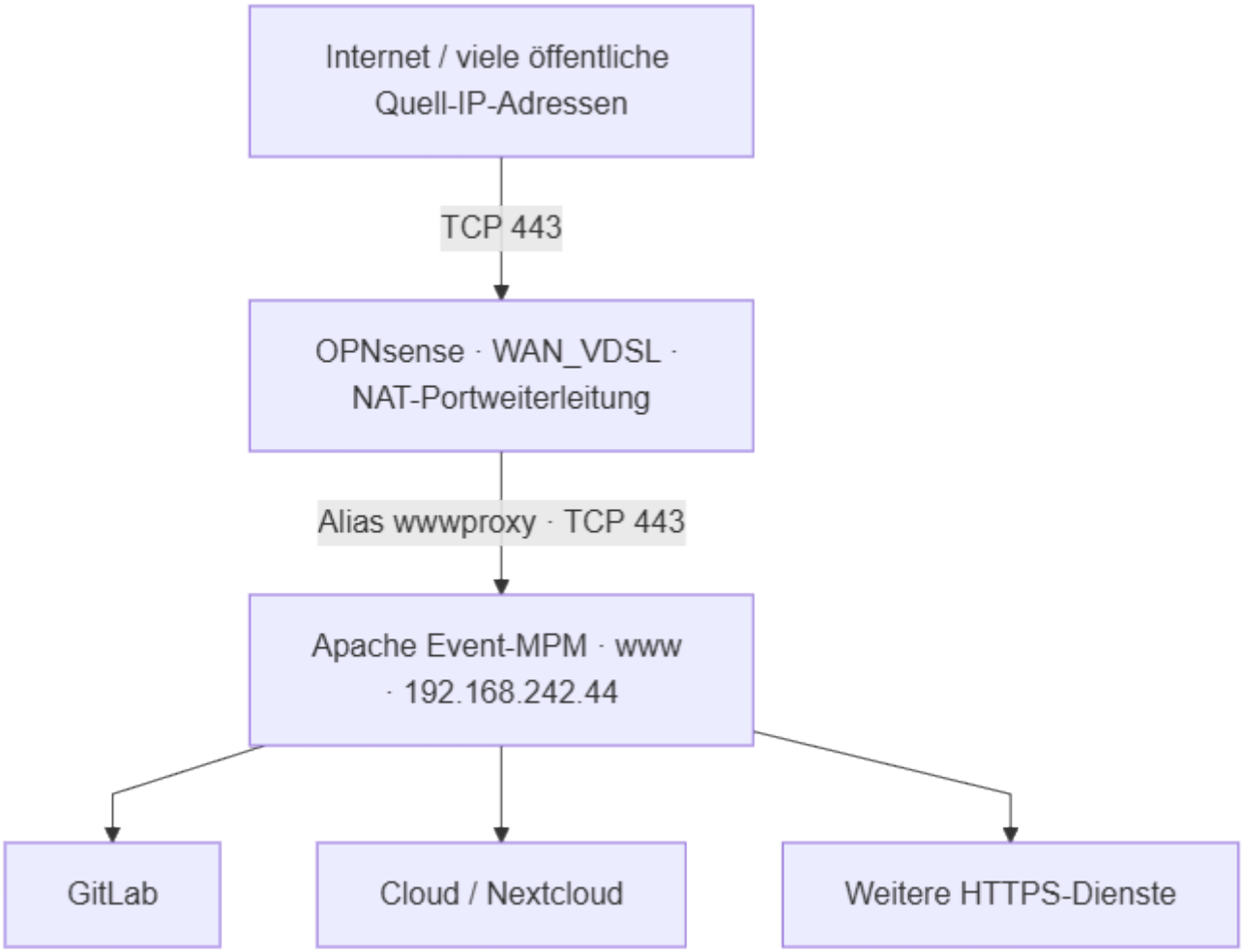
1. Kurzbeschreibung

Auf dem produktiven Apache-Reverse-Proxy kam es zu einer anhaltenden Überlastung des HTTPS-Zugangs. Zahlreiche TCP-Verbindungen wurden angenommen, ohne dass rechtzeitig vollständige HTTP-Header vorlagen. Apache belegte zeitweise alle verfügbaren Worker, während die TCP-Accept-Warteschlange an ihre Grenze stieß und der Kernel fortlaufend Listen-Überläufe registrierte. Zeitgleich stieg das Aufkommen von HTTP-408-Antworten mit leerem Request deutlich an; die GitLab-Anfragen gingen in einem beobachteten Zeitfenster stark zurück.

Technisch stärkste Hypothese: breit verteilte Slow-HTTP-Header-/Slowloris-artige Überlastung mit einzelnen zusätzlichen unvollständigen TLS-Handshakes. Die Messungen stützen diesen Mechanismus, beweisen aber **nicht** die Absicht oder Koordination der Quellsysteme und identifizieren keinen Urheber.

Während der Arbeiten kam es nach einer OPNsense-Aktualisierung zusätzlich zu einem HTTP-500-Fehler der Firewall-Weboberfläche. PHP-CGI-Prozesse stürzten nachweislich ab. Die Paketinstallation war inkonsistent; nach Abschluss der ausstehenden Paketaktualisierung war die Weboberfläche wieder erreichbar. Die Ursache eines späteren Problems bei der lokalen Root-Anmeldung blieb offen.

2. Infrastruktur und Ausgangszustand



Topologie, durch die Angaben bestätigt: OPNsense übernimmt die Portweiterleitung von `WAN_VDSL:443` auf den Alias `wwwproxy`, TCP 443. Der Reverse-Proxy hat auf `ens18` ausschließlich `192.168.242.44/24` und als Standardgateway `192.168.242.1`. In einzelnen Logs kommen auch `192.168.242.1` und `192.168.242.2` vor; die genaue Rolle von `.2` wurde nicht ermittelt. Die OPNsense-WebUI ist über Port `8888` erreichbar.

Komponente / Parameter	Vor dem Eingriff bekannter Wert
Apache	2.4.52, Event-MPM
`ThreadsPerChild`	25
`MaxRequestWorkers`	konfiguriert 256; **effektiv 250** (Abrundung auf Vielfaches vo
`ServerLimit`	zunächst kein expliziter Wert in der gefundenen Datei
`Timeout`	300 s
`KeepAliveTimeout`	5 s
`ExtendedStatus`	`On`
`RequestReadTimeout`	`header=20-40,minrate=500` und `body=10,minrate=500`
RAM / Swap (17.11.)	1,9 GiB RAM; 341 MiB benutzt; 1,4 GiB verfügbar; Swap 2,(

Komponente / Parameter	Vor dem Eingriff bekannter Wert
Apache-Kindprozesse	überwiegend ca. 17–19 MiB RSS und 27 Threads je Kindpr
OPNsense-Version vor Paket-Reparatur	`25.1.3`

Mess- und Loghinweise

- Der HTTPS-Default-VHost war `cloud.opendynamic.de:443`. HTTP 408 mit leerem Request wurden diesem VHost zugeordnet; das ist **kein Nachweis**, dass Cloud das Ziel sämtlicher Verbindungen war.
- Access-Log: `/var/log/apache2/other_vhosts_access.log` einschließlich rotierter Dateien. Die historischen Vergleiche nutzten auch Logrotationen vom 13.–16.11.
- Apache-Status: `/server-status?auto`, insbesondere `BusyWorkers`, `IdleWorkers` und Scoreboard-Zustand `R` = *Reading Request*.
- Kernelzähler `TcpExtListenOverflows` und `TcpExtListenDrops` sind kumulativ; nur die **Differenz innerhalb eines bekannten Zeitintervalls** ist als Rate interpretierbar.

3. Chronologie des Incidents

Zeitangaben in **MEZ (UTC+1)**. Bei Arbeitsschritten ohne protokollierte Uhrzeit wird bewusst keine genaue Minute angegeben.

Datum / Zeit	Ereignis / Beobachtung	Einordnung
13.11., ca. 17:03	Früher auffälliger Anstieg von HTTP-408-Ereignissen.	Erstmals in der Untersuchung genannt
14.11., bis nachmittags	Über längere Zeit ca. **8.000–11.000 HT**	Andauernde außergewöhnliche Timeou
16.11., 17:10–17:14	**259** HTTP 408 in fünf Minuten.	Vergleichsfenster.
16.11., 17:15–17:19	**2.103** HTTP 408 in fünf Minuten; **1.999** untersuchten 408 **100 % Request** `"-"`	Anstieg um ca. **8,1×** im Vergleich zum v Fünfminutenfenster.
16.11., zeitgleich	GitLab-HTTPS-Requests pro fünf Minuten **11.271** zu **4.513** (ca. **260 %**)	Zeitlich zugewiesene I stiegen GitLab-Anfragen zeitweise zugl
17.11., ca. 11:58	Listen-Überläufe **+1.888 in etwa 10 s** (**510/511**.	Verbindungseingang stark belastet.
17.11., ca. 12:01	Apache **246 Busy / 4 Idle**, überwiegend `R`, mehrere Threads im Kernel-`poll`	Worker primär beim Empfang gebunde CPU-Last.
17.11., 12:13:49	`RequestReadTimeout handshake=15` aktiviert; Apache-Reload.	Erste Handshake-Grenze, keine anhalt
17.11., ca. 12:15	Vollständiger Apache-Neustart.	Auslastung blieb nahe Worker-Grenze.
17.11., ca. 12:27	Erweitertes `reqtimeout:info`-Loggin **Header- / 1 Handshake-Timeout**; in d **, 0 Body-Timeouts**.	Unvollständige HTTP-Header dominier Stichprobe.

Datum / Zeit	Ereignis / Beobachtung	Einordnung
17.11., danach	Header-Limit auf `header=10-20,minrate=500` gekürzt.	Momentaufnahme **222 Busy / 3 Idle**, Q keine gesicherte Entlastung des Zugan
17.11., danach	Listen-Überläufe **+1.472 / 10 s** (**147,2/s**	Weiterhin erheblicher Rückstau.
17.11., danach	`MaxRequestWorkers 400`, `ServerLimit 16` gesetzt, Apache neu gestartet.	Apache erreicht **400 Busy / 0 Idle**; Que
17.11., ca. 12:41	Weitere zehn `Request header read` öffentlichen IPs.	Header-Timeouts bestehen fort.
17.11., danach	Notfall-Härtung `handshake=10 header=5-10,minrate=500`	Danach **198 Busy / 2 Idle**, davon **198 512/511**.
17.11., 12:44–12:45	Wiederholt `AH03490: scoreboard is ServerLimit.`	Separater Scoreboard-Engpass nachge
17.11., danach	Prozessliste: **9** Kindprozesse mit je **27** Threads,	Auch weitere Momentaufnahmen zeigen belegte Scoreboard-Slots.
17.11., danach	`ServerLimit 16 → 32`, `MaxRequestW`	Momentaufnahme **394 Busy / 6 Idle**, T
17.11., später	Listen-Überläufe **+848 / 10 s** (**84,8/s**);	**400 Busy / 0 Idle**, Queue **46/511** und volle Worker-Belegung.
17.11., später	Letzte 1.000 Header-Timeouts: **985** un	Sehr breit verteilte Quelladressen; Einz Hauptabwehr wenig aussichtsreich.
17.11., nach OPNsense-Update	WebUI auf Port 8888 liefert **HTTP 500**; `cgi`-Abstürze mit **SIGSEGV** und **SIGILL**.	Zweite Störung getrennt vom Apache-
17.11., ca. 15:33–15:50	`php-cgi.core` (27.181.056 Byte) vorh	Absturz durch Core Dump und Kernel-L auslösende Bibliothek unbekannt.
17.11., Reparaturphase	`pkg check -d -a`: `net-snmp` vermisst	Inkonsistenter Paketstand nachgewies Repository **8.3.23**; `opnsense` installiert **25.1.3**, Repository **25.1.12**.
17.11., Reparaturphase	`pkg upgrade -n`: **31 Upgrades**, **3 Ne bestehenden Pakets**, **1 Entfernung**. K `upgrade` ausgeführt.	WebUI anschließend nach Nutzerangal kein Neustart angefordert.
17.11., anschließend	Root-Passwort per GUI nicht übernommen, gemeldet; später scheitern Anmeldung mi	**Offenes Problem**: lokaler Root-Konto wiederhergestellt.

4. Messwerte und Grafiken

4.1 HTTP-408-Sprung am 16. November

Fenster	HTTP 408	Veränderung
17:10–17:14	259	Referenz
17:15–17:19	2.103	**8,1-fach** bzw. **+1.844**

Alle untersuchten 408-Ereignisse im Spitzenfenster enthielten als Request nur `"-"`; wegen des Default-VHosts ist die Cloud-Zuordnung kein Nachweis des angegriffenen Hostnamens.

4.2 Zeitgleiche GitLab-Anfragen

Die Zahl beobachteter GitLab-Requests fiel von **11.271 auf 4.513** (ca. **?60 %**). Die Gleichzeitigkeit deutet auf eine Beeinträchtigung hin, belegt aber allein keinen ursächlichen Zusammenhang. Der andersartige Verlauf am 13.11. spricht gegen eine pauschale Gleichsetzung von HTTP-408-Anstieg und GitLab-Rückgang.

4.3 Timeout-Typen: Stichprobe der letzten 1.000 Meldungen

Timeout-Typ	Anzahl	Anteil
HTTP-Header	950	**95 %**
TLS-Handshake	50	5 %
HTTP-Body	0	0 %

Geltungsbereich: die zuletzt betrachteten **1.000** **`reqtimeout:info`-Einträge**. Dies ist keine vollständige Verteilung aller eingegangenen TCP-Verbindungen.

4.4 TCP-Accept-Warteschlange – drei Momentaufnahmen

Zeitpunkt / Zustand	`Recv-Q`	`Send-Q`
Vor Scoreboard-Anpassung	512	511
Nach `ServerLimit 32`	18	511
Spätere Kontrolle	46	511

Die Queue-Werte sind **Einzelmessungen**, keine Mittelwerte. Trotz eines zeitweise deutlich niedrigeren `Recv-Q` blieb Apache später wieder mit **400/400** Workern ausgelastet.

4.5 Listen-Überläufe – Vergleich von 10-Sekunden-Messungen

Messung	Differenz	Zeitfenster	Berechnete Rate
Frühe Messung am 17.11.	+1.888	ca. 10 s	ca. **188,8/s**

Messung	Differenz	Zeitfenster	Berechnete Rate
Nach erster Header-Verschä	+1.472	10 s	**147,2/s**
Nach `ServerLimit 32`	+848	10 s	**84,8/s**

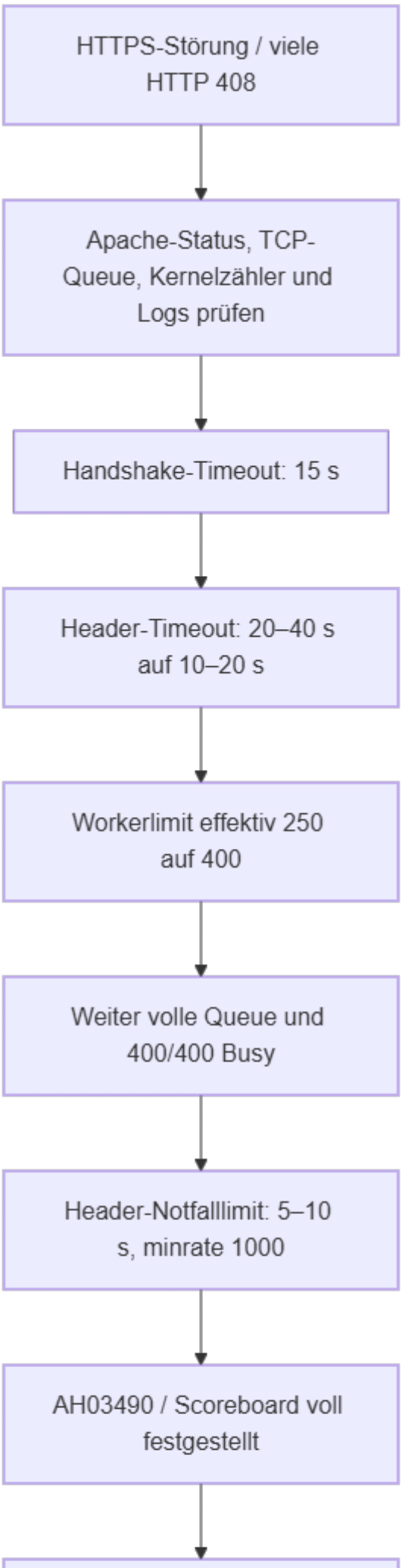
Interpretation: Die gemessene Rate nahm über diese Beobachtungen ab. Weil Last und Messzeitpunkte unterschiedlich waren, ist die Verbesserung nicht eindeutig einer einzelnen Konfigurationsänderung zuzurechnen. Ein weiterer Zähleranstieg um **1.880** wurde zwischen zwei Messwerten ohne bekanntes Zeitintervall erfasst und daher **nicht** in eine Sekundenrate umgerechnet.

4.6 Breite Quell-IP-Verteilung

```
Letzte 1.000 HTTP-Header-Timeouts
├─ 985 unterschiedliche IP-Adressen
│  ├─ 978 IPs mit genau einem Ereignis
│  └─   7 IPs mit mehreren Ereignissen
└─ 1.000 Timeout-Ereignisse insgesamt
```

Der Anteil einmalig auffälliger IPs betrug **978/985 = 99,3 % der unterschiedlichen IPs**. Einzel-IP-Sperren wären unter diesem beobachteten Muster kein wirksamer Hauptschutz. Zwei interne Adressen (`192.168.242.1`, `192.168.242.2`) kamen ebenfalls vor; sie dürfen nicht ohne Prüfung blockiert werden.

5. Ablauf der Maßnahmen


HTTPS-Störung / viele HTTP 408
Apache-Status, TCP-Queue, Kernelzähler und Logs prüfen
Handshake-Timeout: 15 s
Header-Timeout: 20–40 s auf 10–20 s
Workerlimit effektiv 250 auf 400
Weiter volle Queue und 400/400 Busy
Header-Notfalllimit: 5–10 s, minrate 1000
AH03490 / Scoreboard voll festgestellt

Maßnahme	Beobachtung danach	Bewertung
`handshake=15`	Weiter fast alle Worker beschäftigt	Allein nicht ausreichend
`header=10-20,minrate=500`	222 Busy / 3 Idle, aber Queue 512/511	Keine ausreichende Wiederherstellung
`MaxRequestWorkers=400`, `ServerLimit=16`	400/400 Busy, Queue ca. 510/511	Mehr Kapazität wurde ebenfalls ausgel
`handshake=10 header=5-10,minrate`	198 Busy / 2 Idle, Queue 512/511	Kurzzeitiger BusyWorkers-Rückgang, a Generationen beachten
`ServerLimit=32` bei `MaxRequestWorkers=400`	394 Busy / 6 Idle, Queue 18/511; später 400 Busy / 0 Idle, Queue 46/511	Scoreboard-Engpass vermutlich entsch beseitigt
OPNsense SYN-Cookie-Modus	`adaptive` mit Start **60 %**, Ende **30 %** wurde besprochen	**Nicht belegt, dass diese Einstellung** Verbindungen kein hinreichender Schu

Zuletzt im Chat genannte Apache-Konfiguration:

```
# Event-MPM
ServerLimit 32
ThreadsPerChild 25
MaxRequestWorkers 400

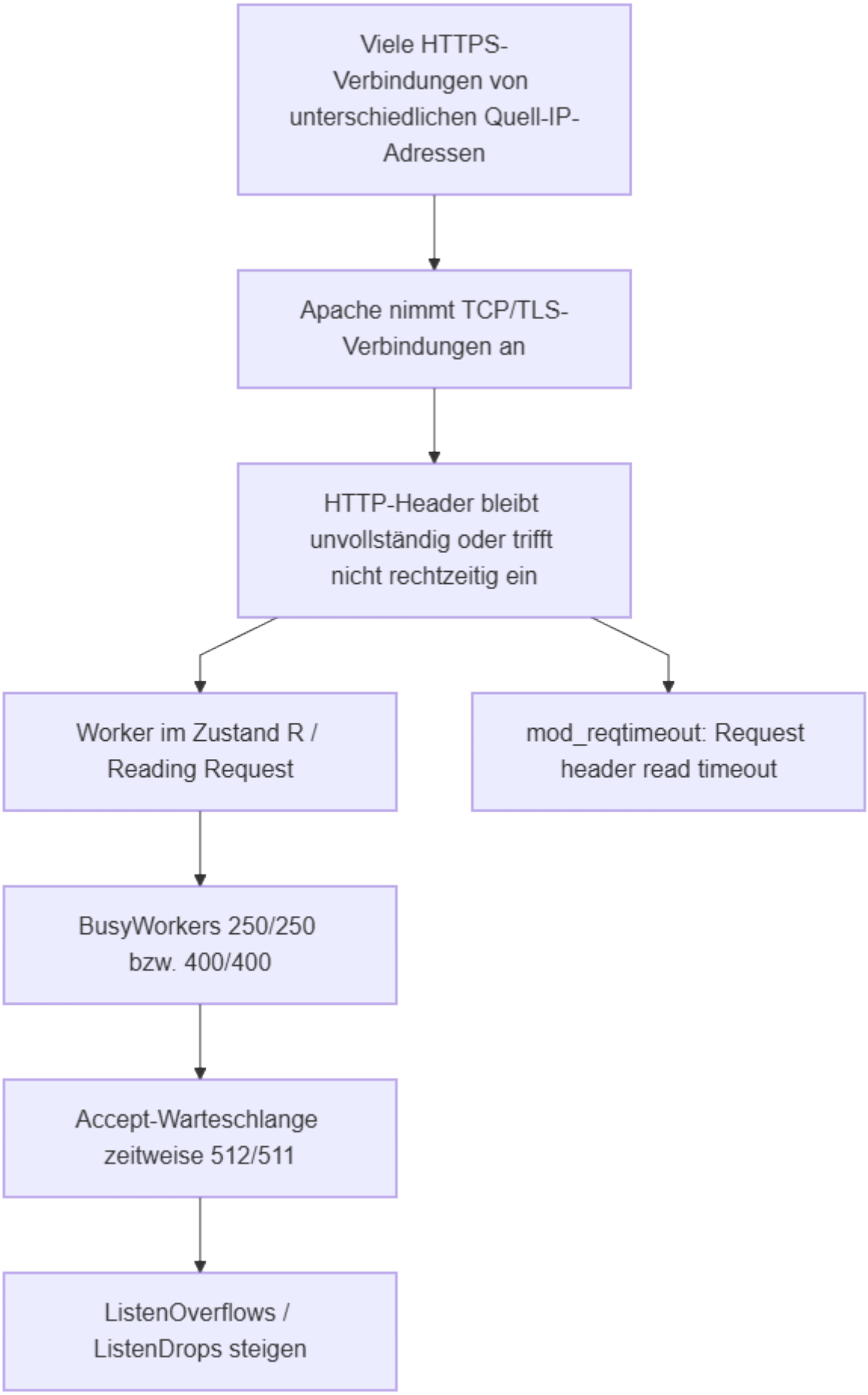
# mod_reqtimeout
RequestReadTimeout handshake=10 header=5-10,minrate=1000 body=10,minrate=500

# Diagnose-Logging (aktiviert)
LogLevel warn reqtimeout:info
```

Diese Werte wurden während der Störung als Notfallkonfiguration eingesetzt. Eine spätere Rücknahme, weitere Änderung oder heutige Wirksamkeit ist durch die vorliegenden Daten nicht belegt. `reqtimeout:info` kann zusätzliches Logvolumen verursachen.

6. Technische Ursachenanalyse

6.1 Primärer Engpass: Anfrageempfang


Viele HTTPS-Verbindungen von unterschiedlichen Quell-IP-Adressen
Apache nimmt TCP/TLS-Verbindungen an
HTTP-Header bleibt unvollständig oder trifft nicht rechtzeitig ein
Worker im Zustand R / Reading Request
mod_reqtimeout: Request header read timeout
BusyWorkers 250/250 bzw. 400/400
Accept-Warteschlange zeitweise 512/511
ListenOverflows / ListenDrops steigen

Evidenz für die Hypothese:

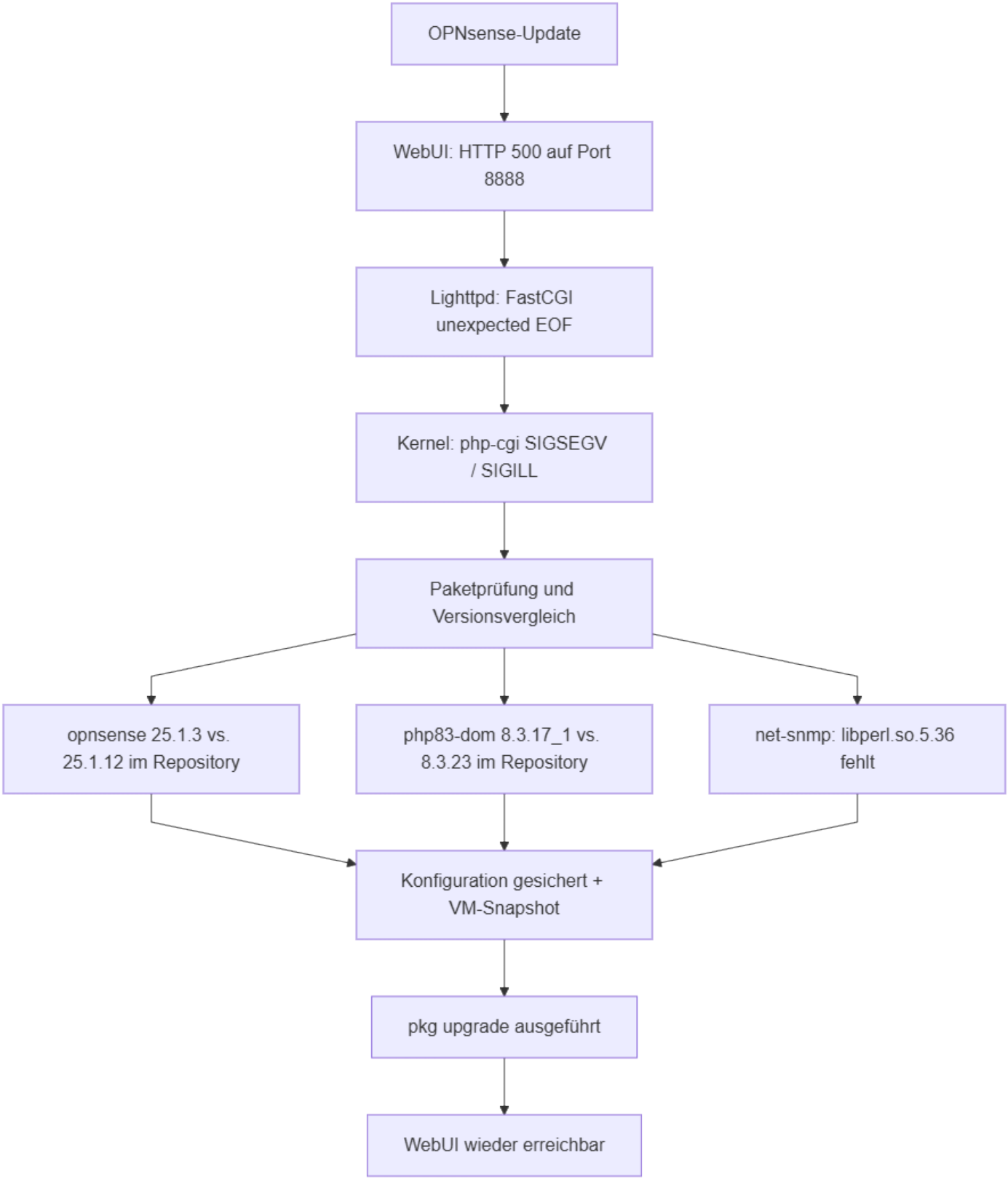
1. Apache-Status: überwiegend `R`, u. a. **379** `R`, **18** `W`, **3** `_` bei 400 Worker-Slots in einer Momentaufnahme.
2. `reqtimeout:info`: **950/1.000** Timeouts beim Lesen von HTTP-Headern; **50/1.000** beim TLS-Handshake.
3. TCP-Listener zeitweise vollständig angestaut (z. B. **512/511**), Kernel-Listen-Überläufe bis ca. **188,8/s** in einem 10-Sekunden-Fenster.
4. Extrem breite Quell-IP-Verteilung: **985 IPs** auf **1.000** Header-Timeouts.
5. Apache-Threads warteten vielfach in `poll()`; hohe CPU-Auslastung war nach den vorliegenden Beobachtungen nicht der erkennbare Engpass.

Grenze der Beweisführung: Diese Indikatoren belegen eine Überlastung durch langsam bzw. unvollständig eingehende Anfragen, **nicht** die vorsätzliche Steuerung aller Clients. Auch die tatsächliche Dauer und vollständige TLS-/HTTP-Sequenz jeder Verbindung wurde nicht nachgewiesen. Der 408-Default-VHost ist kein belastbarer Ziel-VHost-Beweis.

6.2 Sekundärer Engpass: Apache-Scoreboard

Die wiederholte Meldung `AH03490: scoreboard is full, not at MaxRequestWorkers. Increase ServerLimit.` dokumentiert einen **separaten Engpass**. Die Prozessliste zeigte mehrere alte Kindprozesse nach Reloads (sieben Prozesse mit drei Threads) neben neuen Kindprozessen (neun Prozesse mit 27 Threads). Ein `ServerLimit` von **16** bot bei **400** erlaubten Workern (`16 × 25`) keinen zusätzlichen Prozess-Slot-Spielraum. Nach der Erhöhung auf **32** fiel `Recv-Q` zeitweise auf **18** und später **46** statt **512**. Das ist mit der Entschärfung des Scoreboard-Problems vereinbar, beweist aber für sich allein keine dauerhafte Stabilisierung.

6.3 OPNsense: getrennte Störung durch Paketstand / WebGUI



Diagnosebefunde: `php -v`, `php -m`, ein einfacher Phalcon-Test (`5.9.3`), `php-cgi -v` sowie der direkte Aufruf der OPNsense-WebUI-Einstiegsdatei funktionierten. `pkg check -s -a` meldete keinen Prüfsummenfehler; `pkg check -d -a` fand jedoch die fehlende Abhängigkeit bei `net-snmp`. Ein Core Dump von `php-cgi` (SIGSEGV) wurde untersucht; die Absturzadresse lag in einem `rwx`-Speicherbereich ohne eindeutige Bibliothekszuordnung. **Eine bestimmte PHP-Erweiterung als Verursacher ist nicht bewiesen.**

Reparaturwirkung: Nach `pkg upgrade` war das Webinterface laut Rückmeldung wieder erreichbar und ein Neustart nicht erforderlich. Der Zusammenhang mit dem inkonsistenten Paketstand ist stark, jedoch liegt keine isolierende Bestätigung vor, welches einzelne Paket den PHP-Absturz auslöste.

6.4 Separates Folgeproblem: Root-Anmeldung

Eine Passwortänderung über die WebUI wurde nicht zuverlässig übernommen. Über das OPNsense-Konsolenmenü wurde ein Passwort-Reset zunächst als erfolgreich gemeldet; später endeten **sowohl die Anmeldung mit dem neuen als auch mit dem alten Passwort** am lokalen `login:` mit `Login incorrect`. Das Tastaturlayout wurde als korrekt geprüft. **Endstatus: Konsolenanmeldung nicht wiederhergestellt.** Der aktuelle Root-Anmeldestatus der WebUI wurde anschließend nicht abschließend bestätigt. Dies ist ein offener Betriebs- und Wiederherstellungsaspekt, nicht automatisch ein Indiz für den ursprünglichen HTTPS-Angriff.

7. Auswirkungen und Grenzen

Beobachtete Auswirkungen:

- Hohe HTTP-408-Last, mehrere Tausend Ereignisse in kurzen Intervallen und zeitweise 8.000–11.000 pro Stunde.
- Hohe Apache-Worker-Belegung bis **400/400**, phasenweise keinerlei freie Worker.
- Volle beziehungsweise zeitweise fast volle TCP-Accept-Warteschlange und fortlaufend steigende Listen-Überlauf-/Drop-Zähler.
- GitLab-Anfragezahl im dokumentierten Fünfminutenvergleich um ca. **60 %** geringer.
- Temporärer Verlust der OPNsense-Webverwaltung durch HTTP 500 nach Update, später behoben.
- Root-Konsolenzugang zum zuletzt dokumentierten Zeitpunkt nicht möglich.

Nicht aus den vorliegenden Daten ableitbar: konkrete Zahl betroffener Benutzer, Dauer vollständiger Nichterreichbarkeit, gemessene externe Ende-zu-Ende-Erreichbarkeit, Bandbreitenauslastung der VDSL-Leitung, belastbare Täterzuordnung, tatsächliche Aktivierung von SYN-Cookies, dauerhaftes Ende der HTTPS-Überlastung.

8. Empfehlungen und offene Maßnahmen

Priorität	Maßnahme	Begründung / Prüfkriterium	Status nach Chat
P1	Vorgeschaltete DDoS-Mitigation beim Anschluss-/Hostingprovider anfragen, ausdrücklich für verteilte TCP-/TLS-/HTTP-Connection-Exhaustion	Der an Telekom pro-IP-Sperren daher wenig erfolgversprechend	

Priorität	Maßnahme	Begründung / Prüfkriterium	Status nach Chat
P1	Reverse-Proxy von außen und intern erneut messen: Busy/Idle, `R`, Accept-Queue Listen-Überläufe pro festem Zeitfenster, legitime Requests	Letzter Stand noch 42/400 B und 0 Overflows/s	**Geschlossen**
P1	Lokalen OPNsense-Root-Konsolenzugang kontrolliert wiederherstellen; bestehenden funktionierenden Zugang nicht verlieren	Neues Passwort gesetzt	**Geschlossen**
P2	Vorläufige Apache-Timeout-Härtung auf Fehlerraten legitimer Clie prüfen und später nachjustieren	`header=5-10,minrate=100` langsame legitime Clients beeinträchtigen	Wirkung auf legitime Nutzer **nicht merkbar**
P2	Apache-`AH03490` nach dem Neustart gezielt erneut kontrollieren	Rückgang der TCP-Queue legt Entschärfung nahe	**Geschlossen**
P2	OPNsense-Paket-/Abhängigkeitsprüfung nach Reparatur wiederholen	Vorher `net-snmp`-Abhängigkeitsfeh und Versionskonflikt	**Geschlossen**

Erfolgskriterien für den Abschluss des HTTPS-Incidents

Ein Abschluss sollte **nicht** allein aus einer kurzzeitig gesunkenen Accept-Queue oder BusyWorkers-Zahl abgeleitet werden. Erforderlich wären mehrere aufeinanderfolgende, zeitlich dokumentierte Messungen mit **freien Apache-Workern**, **nicht dauerhaft gefüllter Accept-Warteschlange**, **geringer beziehungsweise ausbleibender Listen-Überlaufrate** und

erfolgreichen externen Zugriffen auf die produktiven Dienste. Entsprechende Abschlussmessungen liegen im Chat nicht vor.

9. Fazit

Die vorliegenden Messungen sprechen stark für eine **breit verteilte Überlastung des Apache-Reverse-Proxys durch unvollständige bzw. zu langsam eingehende HTTP-Header**, mit einem kleineren Anteil von TLS-Handshake-Timeouts. Zusätzliche Apache-Worker allein reichten nicht aus; zusätzlich entstand nach wiederholten Reloads ein **nachgewiesener Scoreboard-Engpass**, der durch `ServerLimit 32` plausibel entschärft wurde. Die verbleibende Worker-Sättigung und Listen-Überläufe zeigen: **Der HTTPS-Incident war zum letzten dokumentierten Messzeitpunkt nicht behoben.**

Die zeitlich nachfolgende **OPNsense-WebUI-Störung** war ein gesondertes Betriebsproblem nach einem nicht konsistent abgeschlossenen Paketupdate. Sie wurde durch das Nachziehen der ausstehenden Pakete praktisch behoben.

Revision #3
Created 2026-11-18 16:25:40 CET by Felix Diekmann
Updated 2026-11-19 08:16:22 CET by Felix Diekmann