

VPN - Router Firewall Sicherheit

Eine gute VPN-Router-Firewall braucht eine strikte Default-Deny-Strategie, klare Ausnahmen nur für das Nötige und regelmäßige Updates. Hier ist ein praxisnaher Leitfaden für IPv4 und IPv6.

Wichtige Grundregeln

- Firmware aktuell halten (Sicherheitslücken schließen).
- Fernzugriff auf das Router-Interface nur über VPN, falls überhaupt erlaubt; wenn nötig, nur aus bestimmten Netzen zulassen.
- Standard-Login des Routers ändern, starke Passwörter verwenden, Admin-Zugriff per SSH/Telnet absichern bzw. deaktivieren.
- UPnP/Port-Forwarding deaktivieren, bzw. nur gezielt zulassen.
- VPN-Server-Port seriell absichern: nur notwendige Ports offen, oft nur von bestimmten Quellen zulassen (IP-Whitelist).
- DNS-Sicherheit: DNS-Lecks vermeiden (DNS über VPN, DNSSEC wenn möglich, ggf. DNS-over-TLS/HTTPS unterstützen).

Allgemeine Firewall-Topologie

- Segmente: WAN (Internet), LAN (dein Heimnetz), ggf. VPN-Interface (wenn der Router selbst als VPN-Server dient).
- Default-Drop-Doktrin: Alle eingehenden Verbindungen von WAN ins LAN standardmäßig ablehnen.
- Erlauben von ausgehenden Verbindungen aus dem LAN ins Internet (mit Zustandsprüfung).
- VPN-Verkehr gezielt freigeben: Nur die Ports/Protokolle, die dein VPN-Server braucht (z. B. WireGuard: UDP 51820; OpenVPN: UDP/TCP 1194), von WAN zum Router zulassen, falls der Router als VPN-Server fungiert.
- Logging/Alerts aktivieren, aber sensible Logs ggf. lokal halten.

Empfohlene konkrete Einstellungen (IPv4 und IPv6 separat, aber ähnliche Logik)

1. Allgemeine Einstellungen (egal ob IPv4/IPv6)

- Remote-Management deaktivieren oder auf explizite, sichere Quellen beschränken.
- Admin-Passwort stark machen; SSH/Remote-Management deaktivieren oder auf Scoped-ACL begrenzen.
- UPnP/NAT-P-PUN deaktivieren.
- Feld "NAT/Firewall" aktivieren.
- Firmware automatisch prüfen oder regelmäßige Manuelle Checks.
- DNS-Optionen prüfen: kein DNS-Mpoisoning, DNS-Leak-Schutz aktiv, ideal DNS über VPN verwenden.

2. Standard-Rule-Set (Firewall-Grundregeln)

- Implicit deny: Abweisung aller eingehenden Verbindungen vom WAN ins LAN, außer explizit erlaubt.
- Abgeleitete Verbindungen (established/related) zulassen in beide Richtungen, damit Antworten funktionieren.

3. IPv4-spezifische Regeln (Beispiele)

- Eingehend vom WAN auf VPN-Port zulassen (falls Router als VPN-Server):
 - TCP/UDP Port frei geben (z. B. UDP 51820 für WireGuard; UDP/TCP 1194 für OpenVPN) von WAN zu Router-VPN-Schnittstelle.
 - Einschränkung: Quelle auf bestimmte öffentliche Adressen/WAN-Interface, falls möglich.
- Eingehend andere Ports: standardmäßig ablehnen.
- Ausgehend aus LAN ins WAN: zulassen, aber mit Zustandsprüfung.
- ICMPv4: Eingehend vom WAN blocken (außer Drucker/Administration, meist besser auf drop). In vielen Heimumfällen genügt es, nur Ping eines langen diagnostic zu blockieren; viele Router erlauben eine begrenzte ICMP-Fehlermeldung. Wenn du aktiv Ping vom WAN brauchst, beschränke ihn zeitlich oder auf bestimmte Quellnetze.

4. IPv6-spezifische Regeln (Beachte: IPv6 entwirft grundsätzlich mehr inbound-Verbindungen)

- Default-Deny für IPv6-Eingänge vom WAN ins LAN.
- Wichtige ICMPv6-Typen freigeben (Neighbor Discovery Traffic, Router Solicitation/Router Advertisement, Redirects, Echo-Requests/Replies je nach Bedarf). Wichtig: ICMPv6 ist essenziell für Stabilität von IPv6, blockiere ihn nicht komplett.
- VPN-Portfreigaben analog zu IPv4, aber über IPv6-Adressen/Netzwerke:
 - Wenn du den Router als VPN-Server nutzt, öffne die entsprechenden UDP/TCP-Ports auch für IPv6.
 - Falls VPN nur über IPv4 erreichbar ist, können IPv6-Ports trotzdem geschlossen bleiben, aber prüfe, ob VPN-Dienste auch IPv6 unterstützen (und entsprechend freigeben).
- IPv6-RA/Router-Advertisement: lasse nur legitime Router Announcements zu (typischerweise vom eigenen Router oder vom Provider), blockiere ggf. unnütze Router-Ankündigungen.

5. VPN-spezifische Sicherheitspraktiken

- Kill-Switch: Alle VPN-Verbindungen zum Internet müssen durch den VPN-Client oder Router gezwungen werden; keine unverschlüsselten Leaks, wenn VPN ausfällt.
- Split-Tunneling nur gezielt, falls nötig; ansonsten alle Verkehr durch VPN leiten.
- DNS-Leaks verhindern: DNS-Anfragen über VPN senden, kein Tropfen ins LAN-DNS.

- Zugriffsschutz des VPN-Servers: starke Authentifizierung (z. B. Zertifikate, Pre-Shared Keys mit ausreichender Länge), begrenze Zugriffe auf bestimmte IPs.
- Log- und Monitoringsicht: regelmäßig Check der Verbindungslogs; Alarme bei verdächtigen Zugriffen.

Beispielhafte konkrete Regeln (grafisch nicht, nur Text)

- WAN -> Router-VPN-Port: erlauben (UDP 51820) von WAN IP-Liste oder beliebig, je nach Setup.
- LAN -> WAN: erlauben, mit etablierte/related nur (und DNS-Verbindung falls gewünscht).
- WAN -> LAN (alle anderen Ports): abweisen (Default deny).
- IPv6 WAN -> LAN: nur die relevanten Ports / Protokolle freigeben (z. B. OpenVPN-Ports, falls IPv6 unterstützt; ansonsten nur Fall durchgehende IPv6-Firewall befolgen).
- LAN -> VPN-Clients: erlauben, falls Remote-Verbindungen vom LAN zum VPN-Server gebraucht werden.
- ICMPv6: erlaubt nur essential types (MSNI/Router Solicitation, Diff Zeiten). Für Heimanwendungen oft sinnvoll, sonst blockieren.

Wichtige Prüfungspunkte

- Teste von außen, ob der VPN-Server wirklich nur über VPN erreichbar ist; versuche, offene Ports von außen zu scannen, und prüfe, ob du gezielt zugreifen kannst.
- Prüfe regelmäßig die Logs (Anmeldeversuche, Verbindungsabbrüche, unerwartete Ports).
- Prüfe, ob IPv6 wirklich durch den Router in dein LAN fließt oder von deinem Provider geblockt wird (je nach Provider). Falls dein Netzwerk kein IPv6 nutzt, schalte IPv6 im Router aus oder schränke es sauber ein.