Proxy-Server unter Linux installieren und betreiben

Vorwort zur Anleitung:

Diese Anleitung hat einen hohen Schwierigkeitsgrad. Installation von Linux und Kenntnisse in der Befehlszeile sind vorausgesetzt.

Was ist ein Proxy-Server?

Ein **Proxy-Server** ist ein Server, der als Vermittler zwischen einem Client und einem Zielserver fungiert. Er empfängt Anfragen vom Client, verarbeitet diese und leitet sie an den Zielserver weiter. Der Proxy-Server kann dabei verschiedene Funktionen erfüllen:

1. **Verbergen der Identität**
   - Der Proxy-Server kann die IP-Adresse des Clients verbergen, indem er Anfragen im Namen des Clients sendet. Dies schützt die Privatsphäre des Clients.

2. **Zugangskontrolle**
   - Proxies können verwendet werden, um den Zugang zu bestimmten Inhalten oder Websites zu blockieren, z.B. durch Netzwerksperren oder das Filtern von Webinhalten.

3. **Caching**
   - Ein Proxy-Server kann häufig angeforderte Inhalte speichern (cachen), um die Ladezeiten zu verkürzen und die Netzwerkbelastung zu verringern.

4. **Sicherheit**
   - Proxy-Server können als Schutzschild fungieren, um Angriffe zu blockieren oder schadhafter Datenverkehr zu erkennen.

5. **Verkehrsanalyse und -überwachung**
   - Sie können den Datenverkehr überwachen und analysieren, um z.B. unerwünschte Aktivitäten zu erkennen oder die Bandbreite zu optimieren.

Es gibt verschiedene Arten von Proxy-Servern:

- **Forward Proxy:** Ein Proxy, der Anfragen von Clients an externe Server weiterleitet.Häufig genutzt in Unternehmen, daher wird er in dieser Anleitung erläutert. Ein Forward-Proxy wird bei Datenverschleierung auch oft Elite-Proxy benannt.
- **Reverse Proxy:** Ein Proxy, der als Vermittler für Server fungiert, der eingehenden Datenverkehr von Clients bearbeitet, z.b. Verkehr zwischen Server und Web-Gui.
- **Transparent Proxy:** Ein Proxy, der den Datenverkehr ohne die direkte Zustimmung des Clients abfängt, oft für Caching oder Monitoring.

Zusammengefasst: Ein Proxy-Server dient als "Zwischenstelle" für Anfragen und Antworten im Internet und kann zur Verbesserung der Sicherheit, Anonymität, und Performance eingesetzt werden.

Vorbereitungen / Anforderungen:

Anforderungen • **Schwierigkeitsgrad-Installation:** Schwer • **Schwierigkeitsgrad-Anwendung:** Mittel • **Erforderliche Kenntnisse:** Linux-Befehlszeile **Systemvoraussetzungen:** • **Basis:** Debian • **Betriebssystem:** Ubuntu Server 22.04 (Headless) • **Festplattenspeicher:** Mindestens 10 GB • **CPU:** Mindestens 2 CPU • **RAM:** Mindestens 2 GB • **Harddrive**: schnelle SSD wird empfohlen * • **Internetverbindung:** Verfügbar * Bei einem Proxy-Server ist die Geschwindigkeit entscheidend, selbst wenn der Proxy auf einen virtuellen Host läuft!	Die Systemanforderungen wurden in einer virtualisierten Umgebung getestet. **Tipp:** Wenn du mit Virtualisierung, z.B. ProxMox, vertraut bist, kannst du deine eigenen Server effizient und stromsparend betreiben! **Getestet auf:** Ubuntu-Server 22.04 Ubuntu-Server 24.04

Installationsablauf: 1. Durchführung von System-Upgrade 2. Statische IP-Adresse festlegen und anwenden 3. (Optional) Hostnamen festlegen 4. Installation und Einstellungen 5. Systembedienung 6. Resümee 7. Störungsanalyse 8. Proxy-Server im Verbund mit Client-Systemen 9. Ausnahmefälle mit No-Proxy-Einstellung 10. Proxy-Einstellungen auf Desktop-Systemen 11. Proxy-Einstellungen auf Server-Systemen 12. Proxy-Einstellungen für APT-Updates 13. Schlussbemerkung	Optionale Schritte können weggelassen werden, da sie lediglich unterstützende Funktionen bieten, wie zum Beispiel die Zusammenfassung von Servern in einem Rechenzentrum. Für den Betrieb in einem Rechenzentrum sollten die optionalen Abläufe jedoch aus Gründen des beruflichen Stolzes in Betracht gezogen werden.

1. Durchführung von System-Upgrade

Vor jeder Installation ist ein System-Update erforderlich:

$> ssh DEINUSER@192.168.1.X	Stelle eine SSH-Verbindung zum Server her, um Remote-Operationen durchzuführen.
$> sudo apt update	Aktualisiere die Paketquellen, um sicherzustellen, dass du die neuesten Versionen der Pakete erhältst
$> sudo apt upgrade -y	Starte das System-Upgrade und verwende dabei die Option `--yes-to-all`, um alle Bestätigungsabfragen automatisch zu beantworten.
$> sudo apt autoclean	Entferne Pakete, die nicht mehr benötigt werden, um Speicherplatz zu sparen und das System zu optimieren.
$> sudo apt autoremove	Bereinige das System von überflüssigen Abhängigkeiten, die nach Paket-Deinstallationen übrig geblieben sind.

2. Statische IP-Adresse festlegen und anwenden

Ein Proxy-Server benötigt eine statische IP-Adresse. Alternativ kann auch eine per DHCP zugewiesene feste Adresse verwendet werden, jedoch sollte diese niemals geändert werden.

Es besteht die Möglichkeit, sich selbst vom Server auszuschließen, wenn die IP-Adresse nicht korrekt konfiguriert ist.

$> ip addr	Verwende den Befehl *ip addr*, um alle Netzwerk-Adapter und deren Namen anzuzeigen.
$> sudo su	Stelle sicher, dass du über Root-Rechte verfügst, um Änderungen vorzunehmen.
$> cd /etc/netplan	Gehe in das Verzeichnis, in dem sich die Netzwerk-Konfigurationsdateien befinden.
$> ls	Zeige mit dem Befehl ls alle relevanten YAML-Dateien im Verzeichnis an.
$> for i in $(ls); do mv $i $i.bak; done	Sichere alle bestehenden Konfigurationsdateien, bevor du Änderungen vornimmst.
$> touch /etc/netplan/01_static_ip.yaml	Erstelle eine neue Konfigurationsdatei
$> nano /etc/netplan/01_static_ip.yaml	Öffne die Datei mit einem Texteditor

network:
  version: 2
  renderer: networkd
  ethernets:
  ens18: #Edit this line according to your network interface name.
  dhcp4: no
  addresses:
  - 192.168.1.1/24
  gateway4: 192.168.1.1
  nameservers:
  addresses:
  - 8.8.8.8
  - 8.8.4.4
	(Copy-Paste)
	
	Ersetze **ens18** durch den Namen deines Netzwerk-Adapters (siehe Ausgabe von *ip addr*).
	
	Ändere die **IP-Adresse** und die **Netzmaske** nach Bedarf.
	
	Passen die **DNS-Adressen** an, falls erforderlich.
	
	Überprüfe und passe ggf. die **Routen (Gateway)** an.

	Speichern und Verlassen • Speichern der Datei: Ctrl + O • Editor verlassen: Ctrl + X
$> netplan generate && netplan apply	Wende die neuen Netzwerkeinstellungen an
--- Verbindungsunterbruch – IP wird neu gesetzt	

	Der Server wechselt nun zu einer neuen IP-Adresse, was dazu führt, dass deine aktuelle SSH-Sitzung unterbrochen wird.

3. (Optional) Hostnamen festlegen

Durch das Ändern des Hostnamens der Server-Station vergibst du einen eindeutigen Namen für das System.

In unserem Fall könnte der Hostname beispielsweise „S6-Proxy-Server“ lauten.

Die Hosts-Datei wird verwendet, um auf Server-Ebene IP-Adressen Namen zuzuordnen, die intern genutzt werden. Wenn du den Hostnamen änderst, muss auch der entsprechende Eintrag in der Hosts-Datei aktualisiert werden. Bei produktiven Systemen sollte diese Kleinigkeit unbedingt beachtet werden, da sie ein Zeichen von Berufsethos und Professionalität ist. Für Tests und Experimente ist dies jedoch nicht zwingend erforderlich.

Der neue Hostname muss zwingen der gleiche sein wie auch im Hosts-Files.

$> ssh DEINUSER@192.168.1.X	Stelle eine SSH-Verbindung zum Server über die neue IP-Adresse her.
$> sudo hostnamectl set-hostname DEINHOSTNAME	Setze einen neuen Hostnamen.
$> sudo nano /etc/hosts	Bearbeite die Datei /etc/hosts oder die entsprechende Datei, in der der Hostname definiert ist.
127.0.0.1 localhost 127.0.1.1 DEINHOSTNAME # The following lines are desirable for IPv6 capable hosts ::1 ip6-localhost ip6-loopback fe00::0 ip6-localnet ff00::0 ip6-mcastprefix ff02::1 ip6-allnodes ff02::2 ip6-allrouters	Ändere ausschließlich den markierten Bereich, der den Hostnamen enthält. Achte darauf, dass du keine anderen Einträge unbeabsichtigt veränderst. **Speichern und Verlassen** • Speichern der Datei: Ctrl + O • Editor verlassen: Ctrl + X
$> sudo reboot	Starte den Server neu, damit die Änderungen wirksam werden.

4. Installation und Einstellungen

$> ssh DEINUSER@192.168.1.X	Stelle eine SSH-Verbindung zum Server über die neue IP-Adresse her.
$> sudo apt install squid -y	Installiere den Squid-Daemon
$> sudo systemctl enable squid.service	Dienst für den Start aktiveren

Nun kommen wir zum Herzstück der Konfiguration – den Elite-Proxy, der sämtliche «Sicherheits»-Features bietet. Diese Konfiguration kann übernommen werden und ist für alle Netzwerke offen, jedoch empfehle ich, sich näher mit dem Thema auseinanderzusetzen: Was wird benötigt und was kann weggelassen werden? Hier findest du eine gute Sammlung von einem etabliertem Proxy-Server:

$> sudo su	Öffne die Root-Shell
$> > /etc/squid/squid.conf	Lösche die bisherige Konfiguration
$> nano /etc/squid/squid.conf	Öffne die Konfigurationsdatei und erstelle folgendes Beispiel:

```
#------------------------------------------------
# Include additional config files
#------------------------------------------------
include /etc/squid/conf.d/*.conf

#------------------------------------------------
# Network definitions
#------------------------------------------------
acl localnet src 192.168.0.0/16
acl localhost src 127.0.0.1

#------------------------------------------------
# Allowed destination ports (sinnvoll & sicher)
#------------------------------------------------
acl Safe_ports port 80          # HTTP
acl Safe_ports port 443          # HTTPS
acl Safe_ports port 21          # FTP (optional)

#------------------------------------------------
# Access rules
#------------------------------------------------
http_access deny !Safe_ports
http_access allow localnet
http_access allow localhost
http_access deny all

#------------------------------------------------
# Proxy listener
#------------------------------------------------
http_port 3128

#------------------------------------------------
# Cache settings (realistisch)
#------------------------------------------------
cache_mem 256 MB
maximum_object_size 16 MB
maximum_object_size_in_memory 512 KB

memory_replacement_policy lru
```

```
cache_replacement_policy lru

cache_dir ufs /var/spool/squid 100 16 256
coredump_dir /var/spool/squid

#----------------------------------------------------
# Refresh patterns
#----------------------------------------------------
refresh_pattern ^ftp:           1440 20% 10080
refresh_pattern -i (/cgi-bin/|\?) 0 0% 0
refresh_pattern .               0 20% 4320

#----------------------------------------------------
# Header handling
#----------------------------------------------------
#request_header_access All allow all
#reply_header_access All allow all
# Entferne alle potentiell personenbezogenen Header
request_header_access Referer deny all
#request_header_access Cookie deny all # Benötigt für Browsing
#request_header_access Authorization deny all # Bricht Logins, APIS und Downloads
request_header_access X-Forwarded-For deny all
request_header_access User-Agent allow all   # wir setzen unseren eigenen weiter unten!

# Erlaube nur die Header, die wir explizit benötigen
request_header_access Host allow all
request_header_access Accept allow all
request_header_access Accept-Language allow all
request_header_access Accept-Encoding allow all

# Setze einen generischen User-Agent (reduziert Fingerprinting)
#request_header_replace User-Agent "Mozilla/5.0 (compatible; Proxy/1.0)"

# Antwort-Header ebenfalls säubern
reply_header_access Server deny all
reply_header_access Via deny all
reply_header_access X-Powered-By deny all
reply_header_access All allow all
# falls du weitere Antwort-Header brauchst, gezielt freigeben
#----------------------------------------------------
# Proxy identification (unauffällig, aber ehrlich)
#----------------------------------------------------
via off
#forwarded_for transparent
forwarded_for delete

#----------------------------------------------------
# Optional – Header gezielt setzen
#----------------------------------------------------
#-Mozilla-Firefox
request_header_replace User-Agent "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.0.0 Safari/537.36"
```

$> exit	Verlasse die Root-Shell
$> sudo systemctl restart squid.service	Neue Einstellung laden

Kurze Beschreibung zu den Einstellungen:

memory_replacement_policy lru cache_replacement_policy lru
Diese Einstellungen betreffen die Handhabung des Caches: Welche Dateien sollen behalten und welche nicht? Für den Einstieg ist die Option «lru» eine gute Wahl
cache_dir ufs /var/spool/squid 100 16 256
Der Cache-Speicher möchte jetzt genau wissen: Wie groß soll der Cache sein? Wie viele First-Level- und Second-Level-Verzeichnisse sind vorgesehen? Was bedeutet das? Der superschnelle Cache, der die Anfragen organisiert, ist auf 100 MB verteilt und besteht aus 16 Verzeichnissen, die jeweils bis zu 256 Unterverzeichnisse enthalten können. Dies ist eine Fragmentierungsangabe: Je mehr Fragmente im Cache, desto langsamer wird er. Im Beispiel wird der Standardwert verwendet, mit dem mein Proxy einwandfrei läuft.
refresh_pattern ^ftp: 1440 20% 10080
Um dich richtig herauszufordern und die Anleitung als 'schwierig' einstufen zu können, fügen wir noch eine zusätzliche Ebene hinzu (leider muss das sein). Es geht darum, wie die Dateien im Cache behandelt werden. Hier kommt ein Regex ins Spiel, zum Beispiel `^ftp:`, der alles betrifft, was mit FTP zu tun hat und mit 'ftp:' am Anfang einer Zeile beginnt. Dieser Ausdruck bestimmt, welche Dateien im Cache betroffen sind – in diesem Fall also FTP-Verbindungen. Außerdem gibt es eine minimale Zeit (1440 Minuten), in der Dateien im Cache behalten werden, sowie eine maximale Zeit (10080 Minuten). In diesem Zeitraum gelten die Daten als 'frisch'. Der Prozentsatz berechnet das maximale Alter der Informationen. Hast du das verstanden? Ich selbst verstehe es auch nicht komplett, aber keine Sorge – mit den Standard-Einstellungen bist du auf jeden Fall auf der sicheren Seite: refresh_pattern ^ftp: 1440 20% 10080 refresh_pattern -i (/cgi-bin/|\?) 0 0% 0 refresh_pattern . 0 20% 4320
request_header_access All allow all request_header_access All deny all etc
Nun kommen wir zum eigentlichen, wichtigen Teil: Ein Proxy macht sich durch die Pakete, die er sendet, erkennbar. Bei einem Elite-Proxy möchte man die Außenwelt jedoch täuschen und den Eindruck erwecken, dass man nur ein einfacher, unbedeutender Computer ist – ganz sicher nicht jemand, der im World Wide Web auffällt. Daher will man bestimmte Header manipulieren oder ganz weglassen. An dieser Stelle führt kein Weg an eigener Recherche vorbei. Zu Beginn kann man den Request-Header entweder komplett zulassen oder alle Header blockieren. Wenn man einen Teil oder alle Header blockiert, spricht man von einem «Elite-Proxy».

5. Systembedienung

$> sudo systemctl start squid.service	Startet den Squid-Daemon, um die Zeit-Synchronisation zu aktivieren.
$> sudo systemctl restart squid.service	Starte den Squid-Daemon neu, um Änderungen an der Konfiguration wirksam werden zu lassen.
$> sudo systemctl stop squid.service	Stoppt den Squid-Daemon, wenn du ihn vorübergehend deaktivieren möchtest.
$> sudo nano /etc/squid/squid.conf	Bearbeitet die Squid-Konfigurationsdatei

$> squid -k parse $> /usr/sbin/squid -k parse	Prüft die Proxy-Einstellungen
$> sudo cat /var/log/squid/access.log $> sudo cat /var/log/squid/cache.log	Zeigt die Logs über den Zugriff und den Cache
$> sudo apt install squidview $> sudo squidview	Zeigt die Zugriff-Logs lesbar formatiert.
$> sudo apt install nmap -y $> nmap localhost	Listet alle offenen Ports auf, um sicherzustellen, dass der Squid-Daemon ordnungsgemäß kommunizieren kann.
$> man squid	Rufe das Handbuch oder die Hilfe für den Squid-Daemon auf, um detaillierte Informationen und Befehlsoptionen zu erhalten.
$> netstat -nat | grep :3128	Zeige verbundene Proxy-Clients an
$> curl -x http://192.168.1.2:3128 https://linux-schweiz.ch	Prüfe Verbindung durch den Proxy
$> squid -k shutdown $> rm -rf /var/spool/squid/ $> mkdir /var/spool/squid/ $> squid -z $> /etc/init.d/squid restart	Prozedur um einen Cache-Fehler zu beheben. Erzeugt einen neuen aber leeren Cache. Dies zu deiner Info, für den Notfall den du nie haben willst.Prozedur zur Behebung eines Cache-Fehlers: Sie erstellt einen neuen, jedoch leeren Cache. Dies dient nur zu deiner Information, für den Fall, den du hoffentlich nie erleben wirst.

7. Störungsanalyse

Statische IP-Adresse wird nicht gesetzt:

Beachte, dass die Netzwerk-Konfiguration im YAML-Stil erfolgt. Die „Incidents“ (Einschübe des Textes) sind essenziell.

Der Proxy-Server startet nach einem Reboot nicht eigenständig:

Aktiviere das Start-Flag für den Dienst mit dem Befehl systemctl enable squid.service, damit er beim Neustart des Systems automatisch gestartet wird.

Die Konfiguration gibt Fehler aus, und kann nicht gestartet werden

Mache ein Konfigurations-Test, welcher viele Fehler ausgibt oder Störungen anzeigt.

Einige CLI-Befehle benötigen ungewöhnlich lange…

Wenn du den optionalen Teil dieser Anleitung befolgt hast, überprüfe, ob der Hostname des Servers mit der Hosts-Datei übereinstimmt!

Wie kann ich die erweiterten Funktionen (des Elite-Proxy) überprüfen?

Einfach ist es nicht. Du kannst jedoch mit Curl prüfen, ob ein Proxy-Server funktioniert. Für eine detaillierte Analyse der Datenverschleierung musst du dich jedoch auf andere Quellen stützen. Eine mögliche Methode zur Analyse ist zum Beispiel die «Man-In-the-Middle»-Technik.

Kann ich für die Benutzung des Proxys ein Passwort festlegen?

Ja – für öffentliche Proxys sehr Sinnvoll. Das Anleitungs-Beispiel kommt aus einer lokalen Netzwerk-Umgebung, daher ist ein Passwort hierfür nicht nötig. Weitere Anleitungen findest du aber im Internet.

Kann ich ein Proxy-Server auf einen anderen Proxy-Server umleiten?

Ja, das ist eine Möglichkeit, um noch sicherer unterwegs zu sein. Allerdings geht dabei die Geschwindigkeit verloren. Wichtig: Es kann erforderlich sein, den Via-Header zuzulassen. Diese Einstellung betrifft jedoch nicht deine lokale Proxy-Konfiguration, sondern ist eine System-Einstellung.

8. Proxy-Server im Verbund mit Client-Systemen

Du hast erfolgreich einen Proxy-Server in Betrieb genommen – gut gemacht! Das ist schon eine Leistung, aber dieser Server bringt noch keinen direkten Nutzen für dich oder deine Nutzer.

Was du suchst, ist eine Möglichkeit, deine Clients automatisch mit dem Proxy zu verbinden, ohne dass deine Nutzer sich darum kümmern müssen. Es gibt vier Ansätze, dies zu erreichen, aber nur einer ist wirklich praktikabel. Um es dir zu erklären, gehe ich nur auf diesen einen ein:

Permanente Anbindung für alle Nutzer einer Arbeitsstation. Alle anderen Optionen sind keine vollwertigen Lösungen für den durchschnittlichen IT-Betrieb

Permanente Anbindung eines Computers für alle User – was heisst das?

Beim Start des Servers oder des Desktop-Systems werden die Umgebungsvariablen mit den Proxy-Einstellungen geladen. Alle Unterprogramme, wie zum Beispiel Browser, nutzen dann diesen zentralen Knotenpunkt. Für Systeme wie Ubuntu gibt es dafür vier grundlegende Methoden:

HTTP-Proxy → Stellt Verbindungen über den unsicheren Port 80 her.
HTTPS-Proxy → Verbindet sich über Port 443 und nutzt ein sicheres SSL-Zertifikat.
FTP-Proxy → Verbindet ausschließlich FTP-Verbindungen.
SOCKS-Proxy → Ein SOCKS-Proxy ist ein Netzwerkprotokoll, das die Kommunikation zwischen einem Client und einem Server in einem anderen Netzwerk ermöglicht.

Vielleicht wird dir jetzt eine scheinbar banale Frage bewusst, über die du dir bisher keine Gedanken gemacht hast: Kann man alle Internetverbindungen über den Proxy leiten? – Die kurze Antwort: Nein, nicht mit diesem Beispiel. Dazu muss eine andere Form der Netzwerkes aufgebaut werden.

Die Hauptanwendung, die dir am häufigsten begegnen wird, ist der HTTPS-Proxy. Der HTTP-Proxy folgt an zweiter Stelle, während der FTP-Proxy eher eine geringere Rolle spielen wird. Der SOCKS-Proxy wird vermutlich noch seltener verwendet. Es ist leicht vorstellbar, dass unter Servern, also «Headless-Systemen», die HTTPS-Komponente ebenfalls eher selten anzutreffen ist.

Lass dich aber nicht vom Thema ablenken: Über HTTPS wird heute viel kommuniziert, und wenn wir schon von Proxy-Verbindungen sprechen, sollten wir auch ein oft unbeachtetes Thema ansprechen, das indirekt einen großen Nutzen bringt.

9. Ausnahmefälle mit No-Proxy-Einstellung

Verwendet man einen Proxy-Server, wird das System versuchen, sämtlichen Datenverkehr über diesen Knoten zu leiten – sowohl HTTP als auch HTTPS. Doch genau dann kann es passieren, dass du dich zwangsläufig in deinem lokalen Netzwerk wiederfindest. In meinem Netzwerk gibt es viele Kommunikationsprozesse, die nicht funktionieren, wenn ich dem Proxy nicht klar machen kann, dass bestimmte Adressen – wie zum Beispiel der Druckerserver oder das WLAN-Management-Interface – nicht über den Proxy laufen sollen.

Mit der No-Proxy-Einstellung kannst du Ausnahmen definieren. Vielleicht fällt dir jetzt auf, dass eventuell auch Localhost-Kommunikation über den Proxy geleitet wird. Wichtige Systeme könnten dadurch ausfallen, und die Fehlermeldungen sind oft ganz exotisch! Eines ist klar: Du musst dich intensiv mit den No-Proxy-Einstellungen befassen!

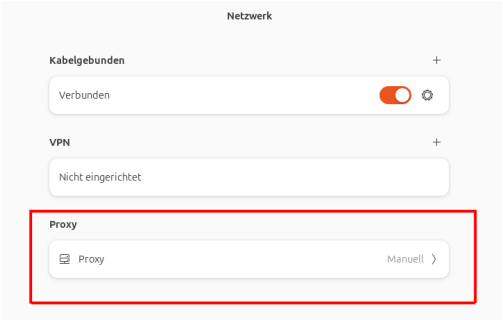
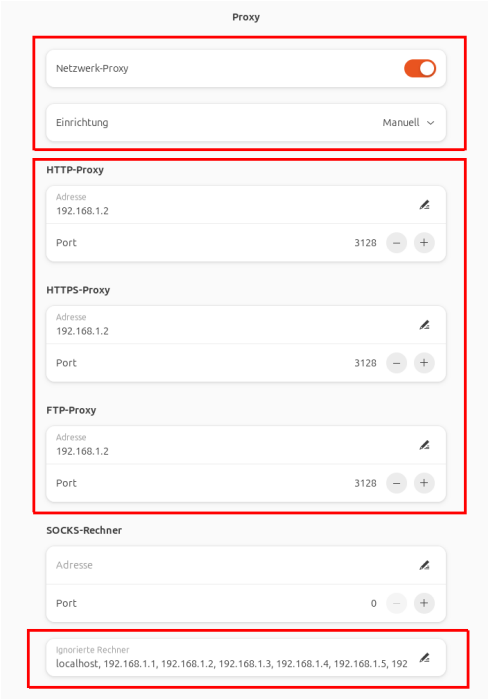
Meine Empfehlung: Definiere alle Adressen des lokalen Netzwerks und des Localhosts, sowohl für IPv4 als auch für IPv6 in der NO_PROXY-Einstellung.

Damit schaffst du eine Pufferzone, die den Proxy-Server klar zwischen „nach außen“ und „nach innen“ kommunizierenden Daten trennt. Dieses Feature ist neben dem Proxy-Server eines der

wichtigsten. Wenn alle Geräte und Server im Netzwerk dies berücksichtigen, entsteht ein Netzwerk mit einem definierten Datenfluss. Da haben wir sie also wieder – die Idee eines geordneten Datenflusses!

10. Proxy-Einstellungen auf Desktop-Systemen

Gut zu wissen: Diese Einstellung muss auf allen Desktop-Clients einer Station separat vorgenommen werden. Streng genommen hat dieser Abschnitt nichts mit der „permanenten Anbindung für alle User“ zu tun, jedoch lässt sich der Proxy-Server auf grafischen Systemen viel einfacher testen.

Proxy-Einstellungen unter Netzwerk	Manuelle Konfiguration
	

In den „ergänzenden Hinweisen“ findest du ein Beispiel, wie du die Verbindungen über den Proxy anzeigen kannst. Vergiss nicht, auch den Abschnitt „ignorierte Rechner“ (No_Proxy) zu konfigurieren.

11. Proxy-Einstellungen auf Server-Systemen

Falls du keinen Desktop hast und deinen Server „headless“ betreibst, kannst du die Anbindung über die Kommandozeile vornehmen. Dieses Beispiel zeigt, wie alle Nutzer ab dem Start der Station automatisch an den Proxy gebunden werden:

$> ssh DEINUSER@192.168.1.X	Stelle eine SSH-Verbindung zum Server her
$> sudo nano /etc/environment	Öffne die Datei für die Umgebung.
... export HTTP_PROXY=192.168.1.2:3128 export HTTPS_PROXY=192.168.1.2:3128 export FTP_PROXY=192.168.1.2:3128 export NO_PROXY=192.168.1.1,192.168.1.1,192.168.1.2,192.168.1.3,192.168.1.4,192.168.1.5,192.168.1.6,localhost,127.0.0.1,127.0.1.1,::1	Füge diese Zeilen an, und ändere anderer Einträge in dieser Datei NICHT! Wichtig: Achte darauf, das die Adressen des NO_PROXY keine Abstände haben!

$> sudo reboot	Diese Einstellungen können nur mit einem Reboot übernommen werden
$> ssh DEINUSER@192.168.1.X	Stelle eine SSH-Verbindung zum Server her
$> env | grep PROXY	Versichere dich, dass die Proxy-Einstellungen in die Umgebungsvariablen geladen wurden.

12. Proxy-Einstellungen für APT-Updates

Da wir nun auch wissen, dass es nicht einfach ist, alle Datenströme über den Proxy zu leiten (was bei näherem Nachdenken auch durchaus logisch erscheint), wollen wir dennoch so viele wie möglich über diesen Server führen – am besten diejenigen, die alle Server und Stationen gemeinsam haben. Dazu gehört auch das Update und Upgrade über Apt-Befehle.

$> ssh DEINUSER@192.168.1.X	Stelle eine SSH-Verbindung zum Server her
$> sudo touch /etc/apt/apt.conf.d/proxy.conf	Erstelle eine Datei unter apt.conf.d
$> sudo nano /etc/apt/apt.conf.d/proxy.conf	Öffne die Datei.
Acquire::http::Proxy "http://192.168.1.2:3128/"; Acquire::https::Proxy "https://192.168.1.2:3128/"; Acquire::ftp::Proxy "ftp://192.168.1.2:3128/";	Füge diese Zeilen an. (Hier gibt es übrigens keinen NO_PROXY-Einstellung)
$> sudo reboot	Diese Einstellungen können nur mit einem Reboot übernommen werden
$> ssh DEINUSER@192.168.1.X	Stelle eine SSH-Verbindung zum Server her
$> sudo apt update && sudo apt upgrade -y	Mache ein Update und beobachte den Proxy-Server.

Während auf einem grafischen System die HTTPS-Verbindungen deutlich zahlreicher sind, nehmen auf einem Server auch die HTTP- und FTP-Verbindungen durch Upgrades zu. Es macht daher viel Sinn, diese beiden Funktionen im lokalen Netzwerk zusammenzuführen.

13. Schlussbemerkung

Mit dem Einsatz eines Proxy-Servers in einem Netzwerk betritt man Neuland. Zum ersten Mal muss man sich intensiv mit Datenfluss und Datensicherheit im lokalen Netzwerk auseinandersetzen. Dadurch wird das Netzwerk automatisch etwas komplexer. Einige Rechenzentren wenden das Kubernetes-Prinzip auf ihre Server an, was die Wartung vereinfacht, jedoch den initialen Aufwand erhöht. Es ist sinnvoll, diesen Schritt sorgfältig zu planen und gründlich zu testen.