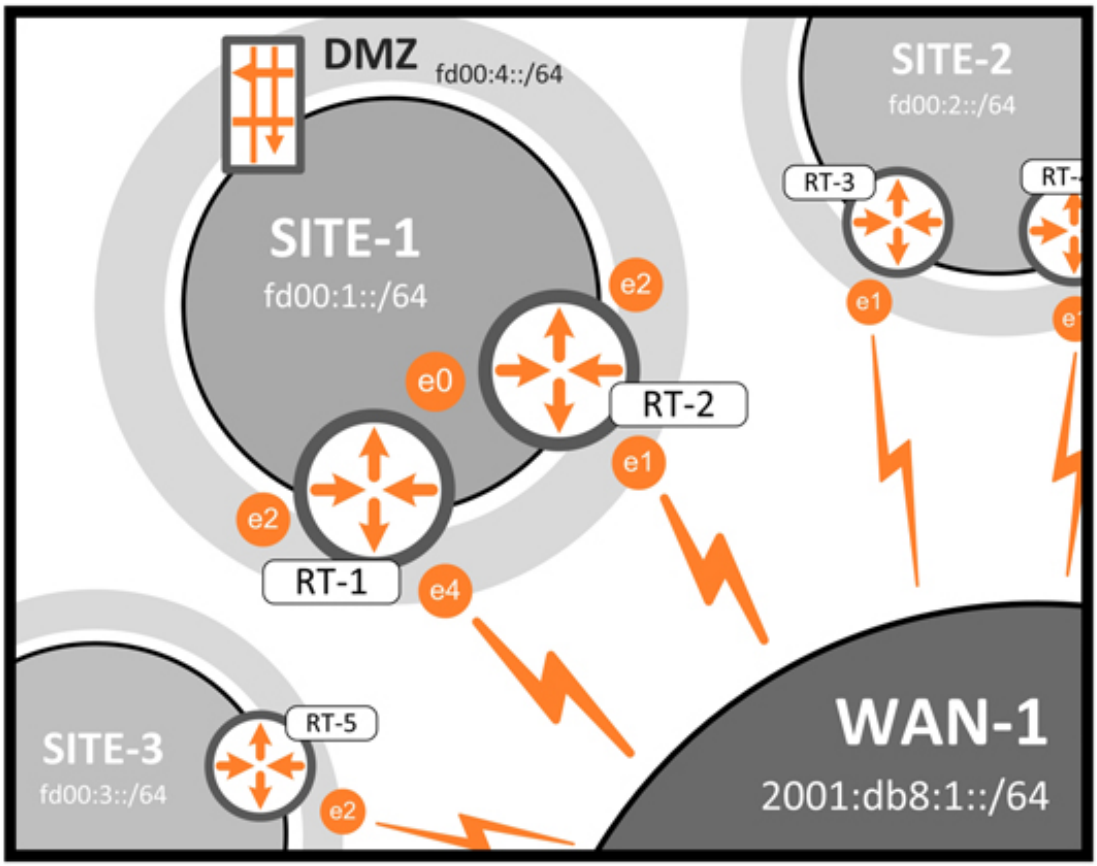

Der VyOS-Praktiker

Enterprise-Routing mit Open-Source

Markus Stubbig

Inhaltsverzeichnis

Vorwort

VyOS? Nie gehört!
Das ist die übliche Reaktion. Dennoch halten Sie dieses Buch in der Hand. Recherche? Schnäppchen?

Oder einfach nur neugierig? Wenn VyOS so toll ist, dass es ein ganzes Buch füllen kann, warum wird es dann nicht überall eingesetzt und in den Fachzeitschriften besungen?

Berechtigte Frage, denn grundsätzlich kann VyOS alles! Die kleinen Fehler tauchen erst im Lauf der Kapitel auf. Wenn Sie dieses Buch tatsächlich ganz oder teilweise durcharbeiten, werden Sie merken, ob VyOS zu Ihnen passt und in Ihrem Umfeld Aufgaben übernehmen darf.

Allzu häufig gehört der persönliche Geschmack mit zu den Entscheidungskriterien. Wer kauft schon Cisco, wenn er für diese Firma nichts übrig hat? VyOS macht da keinen Unterschied, also bitte: Gehen Sie neutral an die Sache heran. Und wenn Sie nur Aufgrund von Neugierde durchblättern, haben Sie die besten Voraussetzungen.

VyOS kam mir bei einer Recherche unter die Finger: Ich brauchte ein offenes Router-Betriebssystem, dass ich um *Dynamic Multipoint–VPN* erweitern wollte. Kandidaten gab es viele und mit fast allen war dieses Vorhaben möglich, wenn auch mit unterschiedlich hohem Aufwand. Die alphabetische Vorgehensweise brachte VyOS ans Ende der Liste. Die Evaluierung ist immer gleich: Distro installieren, Paketsystem untersuchen, Software kompilieren, Konfiguration in vorhandene GUI/CLI einbauen. Bei VyOS kam die Überraschung: DMVPN war bereits integriert und vollständig nutzbar. Recherche beendet.

In meinen Fingern lebt die Kommandostruktur eines Cisco-Routers, daher war die Bedienung von VyOS erst mal anders

und blöd. Aber der Rest hat mich überzeugt. Und hier liegt diese Überzeugung gebunden oder als E-Book vor Ihnen.

Viel Spaß beim Ausprobieren, Staunen und Fluchen.

Übersicht

Teil 1, *Für Einsteiger,* beginnt mit dem Aufbau der Netzwerk-Umgebung mit physikalischen Geräten oder auf einer virtuellen Plattform. Die erstellten Maschinen erhalten ihr Betriebssystem und eine erste Konfiguration. Anschließend gesellen sich die grundlegenden Funktionen Routing und Logging dazu.

In Teil 2, *Für Fortgeschrittene,* bekommen die Router ernsthafte Aufgaben, die in jedem Netzwerk unabhängig von seiner Größe, erfüllt sein müssen. Neben dem Einsatz als Firewall und Adressumsetzer, zeigt VyOS seine Fähigkeiten bei IPv6 und als Webproxy.

Teil 3, *Für Experten,* taucht in Enterprise-Themen ein und baut standortverbindende VPN-Tunnel und Router-Cluster zur Verfügbarkeitssteigerung. Ein Kompatibilitätstest zeigt die Zusammenarbeit mit Routern von Cisco Systems.

Softwareentwickler kommen in Teil 4, *Für Entwickler,* auf ihre Kosten. VyOS bringt viele Methoden mit, um eigene Kommandos zu bauen oder Konfigurationsabläufe zu automatisieren. Ein einfaches Managementframework ermöglicht Massenänderungen auf Basis von Python und SSH.

Außerhalb der Laborumgebung macht VyOS in Teil 5, *Für Praktiker,* eine gute Figur als DSL-Router und Lastverteiler für mehrere Internetleitungen.

Teil 6, *Für Trickser,* zeigt Möglichkeiten zur Leistungssteigerung und viele kleine Handgriffe, die die tägliche Arbeit mit VyOS reibungsfreier gestalten. Zuletzt wandert die Konfigurationsdatei in die Cloud und landet revisionssicher bei Dropbox oder Google-Drive.

Ressourcen

`https://vyos.io`
Die Homepage von VyOS liefert einen guten Einstieg ins Thema und verlinkt zum Wiki, Forum, Download-Bereich und zu Video-Anleitungen.

`http://blog.vyos.net`
Hier verkünden die Entwickler Neuigkeiten oder machen Ankündigungen für Änderungen und Aufrufe an die Community.

`http://www.vyatta4people.org`
Inoffizielle Verbesserungen, kleine Helper-Tools aus der Community und eigene Softwareentwicklungen sammeln sich auf dieser Webseite. Viele der Angaben beziehen sich auf Vyatta, funktionieren aber auch unter VyOS.

`https://dl.networklinx.com/vyatta/`
`http://www.osvx.net/downloads/docs/vyatta/`
Die Handbücher von Vyatta sind hervorragende Nachschlagewerke zur Syntax und Verwendung von einzelnen Kommandos.

Schriftkonventionen

`Nichtproportionalschrift` zeigt die erzeugte Ausgabe eines Kommandos.

Schreibmaschinenschrift wird für Konfigurationen und Schlüsselwörter benutzt, die buchstabengetreu eingetippt werden müssen.

`Nichtproportionalschrift Fett` zeigt Befehle, die eine Ausgabe erwarten.

`Hervorhebungen` weisen auf besondere Wörter oder Zeilen innerhalb von Kommandos oder Bildschirmausgaben hin.

```
ein-sehr-langer-kommando-aufruf --mit --sehr \ --vielen "Optionen"
```

Kommandos mit vielen Argumenten können länger als eine Zeile sein. Für die bessere Übersicht werden diese Kommandos mehrzeilig abgedruckt und um zwei Zeichen eingerückt. Am Ende jeder Zeile steht der Backslash als Hinweis darauf, dass es in der nächsten Zeile weitergeht.

Rechtliches

Einleitung

VyOS ist ein quelloffenes Netzwerk-Betriebssystem für Router und Firewalls. Es basiert auf Debian-Linux und vereint Applikationen wie Quagga, Netfilter, StrongSwan und OpenVPN unter einem einheitlichen Kommandozeileninterface. VyOS läuft als physikalische Hardware, virtuelle Maschine oder in der Cloud.

Zu den bekannten Namen gehört VyOS noch nicht. Eher unbekannt punktet es in den Bereichen Funktionalität und Bedienung. Denn VyOS hat das Herz von Linux und das Aussehen von Juniper mit der Budgetanforderung eines Freibiers.

VyOS ist:

Unvollkommen. Und das ist positiv gemeint. Da ist noch genug Raum zum Wachsen. Auch die Implementierung von Features ist teilweise eigenartig: Das moderne, und auf besondere Anwendungsfälle spezialisierte VXLAN ist dabei. Bei IPv6 fehlt allerdings noch sehr viel.

Open Source. Der Vorteil einer quelloffenen Lösung ist nicht immer sein Preis. Denn wirklich umsonst ist Open-Source–Software auch nicht! Lizenzgebühren fallen nicht an, aber die Zeit der IT-Abteilung zum Einrichten einer wenig dokumentierten Software ohne Herstellersupport darf nicht unterschätzt werden.

Bis heute stehen die unbewiesenen Vermutungen im Raum, dass der US-Geheimdienst NSA Hintertüren in die Sicherheitssoftware von namhaften Herstellern einbauen lässt. Als Endkunde lässt sich das nicht überprüfen, aber es bleibt eine Spur von Zweifel, wenn diese Geräte im eigenen Netz zum Einsatz kommen.

In Open-Source–Produkten können sich Sicherheitsexperten austoben und haben eine realistische Chance, den Schadcode zu finden. Andersherum ist es für Hersteller auch deutlich schwieriger eine Hintertür im Quellcode zu verstecken, wenn dieser für jedermann offen zugänglich ist.

Try before Buy. Wie bei Shareware-Programmen kann (und sollte) VyOS vor dem Einsatz getestet werden, bevor irgendwelche Investitionen in die Infrastruktur beginnen. Und wer freut sich über eingeschränkten Funktionsumfang, eine Evaluierungslizenz oder einen 30-Tage-Zeitraum? In diesem Zusammenhang steht *Try* für Ausprobieren mit Beispielszenarien und *Buy* für den Einsatz in der eigenen Umgebung.

Hardware-frei. VyOS ist Software. Diese Software braucht Hardware. Aber die Wahl der Hardware oder einer virtuellen Umgebung bleibt offen. Das macht eine sichere Kaufentscheidung schwierig. Welche Komponenten sind notwendig, um beispielsweise eine 34 Mbit/s–Leitung mit einem VPN-Tunnel und starker Verschlüsselung zu sättigen?

In der Vergangenheit gab es viele limitierende Gründe, warum eine softwarebasierte Lösung für Netzwerkinfrastruktur nicht an die Leistung der physikalischen Geräte herankam. Der Hauptgrund war das suboptimale Zusammenspiel von Software und Treiber mit der darunterliegenden Hardware. Bei der immens großen

Auswahl von Netzwerkkarten, Mainboards, Prozessoren und Memory ist es für eine Software schwierig auf jede Kombination der Komponenten optimal vorbereitet zu sein.

Heutzutage sind normale Server oder eingebettete Systeme überraschend performant, sodass auch eine nicht-optimierte Software bei kleiner Paketgröße Bandbreiten jenseits der 100 Mbit/s durchbrechen kann.

Die Hardware-Frage klärt das Unternehmen *Ubiquiti,* die ihre *EdgeRouter* mit einem modifizierten VyOS betreiben. Anpassung, Optimierung und Weiterentwicklung machen daraus mittlerweile das unabhängige *EdgeOS.* Es gibt immer noch viele Parallelen zwischen EdgeOS und VyOS. EdgeOS kommt in vielen Kapiteln unter die Lupe und zeigt seine Stärken und Schwächen.

Linux. Unter VyOS läuft ein angepasstes Debian. Der Zugriff aufs Betriebssystem ist nicht gesperrt oder passwortgeschützt. Mit einem einfachen sudo bash liegt der Zugang offen.

Das bringt Möglichkeiten zum Anpassen, Verbessern und Nachinstallieren von Tools. Dagegen steht die Gefahr, dass die eigene Änderung ungewollte Instabilität mitbringt.

Best Of. VyOS erfindet das Rad nicht neu und bedient sich für seine Features an den vertrauten Linux-Diensten, die nach Jahren der Entwicklung eine hohe Stabilität erreicht haben. Die Daemons der Routingprotokolle stammen von Quagga. Der SSH-Server gehört zu OpenSSH und für die Umsetzung der Firewallregeln helfen netfilter bzw. iptables.

Diebstahl? Keineswegs! Eher ein Nachweis, dass Open-Source funktioniert. Solange Lizenzbedingungen eingehalten werden, darf Fremdsoftware beigemischt werden. Gerade im Security-Umfeld ist es höchst erwünscht, dass Anwendungsentwickler keine eigenen Implementierungen stricken, sondern sich an den freien und stabilen Bibliotheken bedienen.

Geschichte

Die Historie von VyOS ist eng verbunden mit Vyatta und Brocade. Angefangen hat es mit der *Vyatta Corporation,* die 2006 als Abspaltung von Debian eine eigene Firewalldistribution sein wollte. Das war auch relativ erfolgreich, denn in den darauffolgenden Jahren bis 2010 wurde eine Version nach der anderen veröffentlicht.

Mitte 2011 hat Ubiquiti aus der Vyatta-Version 6.3 ein eigenes EdgeOS gebaut und für seine Produkte *EdgeRouter* optimiert.

Ende 2012 kam der Paradigmenwechsel: Die Firma hinter der kostenfreien und quelloffenen Software verkaufte an den Netzwerkausrüster Brocade. Wenig überraschend erschien kurz darauf das letzte freie Release 6.6 Anfang 2013. Vyatta war kommerzialisiert.

Eine kleine Projektgruppe wollte Vyatta weiterführen und begann mit VyOS als eine Abspaltung von Vyatta. Das im Dezember 2013 veröffentlichte VyOS 1.0 ist also nichts anderes als Vyatta 6.6 mit Änderungen im Namen und beim Logo. Die Entwicklung von VyOS beginnt.

2014 erschien ein kurzlebiger Fork von VyOS unter dem Namen VX/OS. Die Versprechungen der Roadmap waren vielseitig, aber das Projekt ist kurz darauf eingeschlafen.

Mitte 2014 macht Brocade Ernst und entfernt jede Webseite bezogen auf Vyatta aus seinem Angebot. Vyatta heißt jetzt *Brocade vRouter 5400.* Die Bedienung und Versionierung ist unverändert, aber für die Nutzung werden Lizenzgebühren fällig.

Parallel dazu geht die Entwicklung an VyOS weiter. Ende 2014 gibts bereits VyOS 1.1.0 und Anfang 2016 kommt die aktuelle Version 1.1.7 auf den Markt.

EdgeRouter

Den kommerziellen Routern von Ubiquiti [1] kommt eine besondere Bedeutung zu, denn sie nutzen ein weiterentwickeltes Vyatta auf performanter und gleichzeitig günstiger Hardware. Das Ergebnis ist ein VyOS-ähnliches Betriebssystem, bei dem die Hardware-Frage bereits beantwortet ist.

Andersherum gibt es das Betriebssystem der EdgeRouter nicht als virtuelle Variante. Das liegt hauptsächlich an der Produktpolitik von Ubiquiti. Außerdem ist es schwierig, das Betriebssystem für die MIPS64-Architektur so zu optimieren, dass es auf der eigenen Hardware *und* in einer virtuellen Umgebung leistungsstark arbeitet. Beides ist möglich, bedeutet aber Entwicklungsaufwand seitens des Herstellers.

Zur Begriffsklärung: Ubiquiti ist der Hersteller. Die Produktserie rund um Routing und Switching heißt *EdgeMAX.* Als *EdgeRouter* bietet Ubiquiti verschiedene Routermodelle mit drei bis acht Ethernet-Ports an. Das angepasste Vyatta läuft unter der Bezeichnung *EdgeOS* und ist das Betriebssystem der EdgeRouter. Ubiquiti hat zum Entwicklungsbeginn bei Versionsnummer 1.0 gestartet und ist bisher (2017) bei Version 1.9.1 angekommen.

Für eine Webrecherche ist das Schlagwort „edgerouter“ am aussagestärksten.

In den meisten Kapiteln wird der Vergleich von VyOS zu den EdgeRoutern von Ubiquiti gezogen. Ausnahmen bilden Features, die es unter EdgeOS schlichtweg nicht gibt, weil Vyatta sie nicht implementiert hat oder weil sie unter der Leitung von Ubiquiti nicht als relevant eingestuft sind.

Teil I

Für Einsteiger

Kapitel 1

Das Labornetzwerk

Ein einzelner VyOS-Router ohne umgebendes Netzwerk ist wenig beeindruckend. Für den praxisnahen Einstieg erwacht VyOS in einem konstruierten Labornetz zum Leben. In dieser Umgebung kann VyOS Kapitel für Kapitel mit seinen Fähigkeiten glänzen. Vor dem Einstieg in den Umgang mit VyOS steht der Aufbau des Labornetzwerks.

Alle Themen der Kapitel haben einen praktischen Hintergrund. Theoretische Grundlagen werden nur am Anfang eines Kapitels angesprochen um Verständnis aufzubauen oder angestaubtes Wissen aufzufrischen. Die Beispiele und Übungen sind zum Nachspielen konzipiert.

Die Kapitel basieren alle auf demselben Netzaufbau. Es stellt ein kleines Firmennetz mit drei Standorten und redundanten WAN-Verbindungen dar. Je nach Komplexität eines Themas reicht ein Teil des Labornetzwerks aus, um die Kernaussage zu beschreiben.

Wenn ein Kapitel einen gesonderten Aufbau benötigt oder ein weiteres Gerät untersucht werden soll, gibts am Anfang der Lektion einen entsprechenden Hinweis mit Erklärung.

Ressourcen

Der stets unveränderte Aufbau des Labornetzes hat den charmanten Vorteil, dass zwischen den Kapiteln nicht

umgebaut werden muss. Kein Umverkabeln der Geräte oder Umkonfigurieren der virtuellen Umgebung. Das spart Zeit und verhindert Fehler. Und nach ein paar Kapiteln wird das Labornetz zum vertrauten Begleiter, denn die Namen der Router, Clients, Netzschnittstellen und IP-Adressen bleiben gleich.

Das vollständige Labornetz ist als Netzdiagramm in Abbildung 1.1 dargestellt. In den folgenden Kapiteln werden meist nur Teile dieses Netzwerks zur Untersuchung benutzt.

Da ein händischer Eingriff nach dem ersten Aufbau nicht mehr notwendig ist, kann das Lab auch „aus der Ferne“ betrieben werden - Remotezugriff vorausgesetzt.

Die offiziellen Angaben für Arbeitsspeicher und Festplattengröße sind nicht die Mindestausstattungen – aber nah dran. Damit ist es möglich, das Lab auf dem eigenen Laptop zu starten oder preisgünstig in Hardware nachzubauen. Beispielsweise nutzt ein VyOS-Router gerade mal 256 MB Arbeitsspeicher mit einer 2 Gigabytes großen Festplatte.

Manche Kapitel arbeiten isoliert; andere benötigen Internetzugriff. Der Zugang zum Internet läuft stets über den Core-Router, der hinter seiner Netzkarte *eth0* das Internet erwartet. Ganz praktisch passiert das in einer virtuellen Umgebung über eine NAT-Schnittstelle. In Hardware reicht ein Uplink zum DSL-Router. Hierbei ist alles möglich, was letztendlich ins Internet führt.

Virtualisierung

Alle Geräte im Lab können vollständig virtualisiert werden. Der einzige Hardwarerouter wird dann durch einen VyOS-Router ersetzt, weil EdgeOS keine virtuelle Plattform

unterstützt. Auch ein Mischbetrieb mit physikalischen Geräten ist möglich.

Jeder Router im Labornetz ist dann eine eigene virtuelle Maschine (VM) mit virtuellen Netzwerkkabeln zu den benachbarten VMs. Die Verbindungsnetze zwischen den VMs sind VMnetX (bei VMware) und vboxnetX (bei VirtualBox). Eine physikalische Netzwerkkarte im Hostsystem ist nötig, wenn mit echter Hardware gemischt wird.

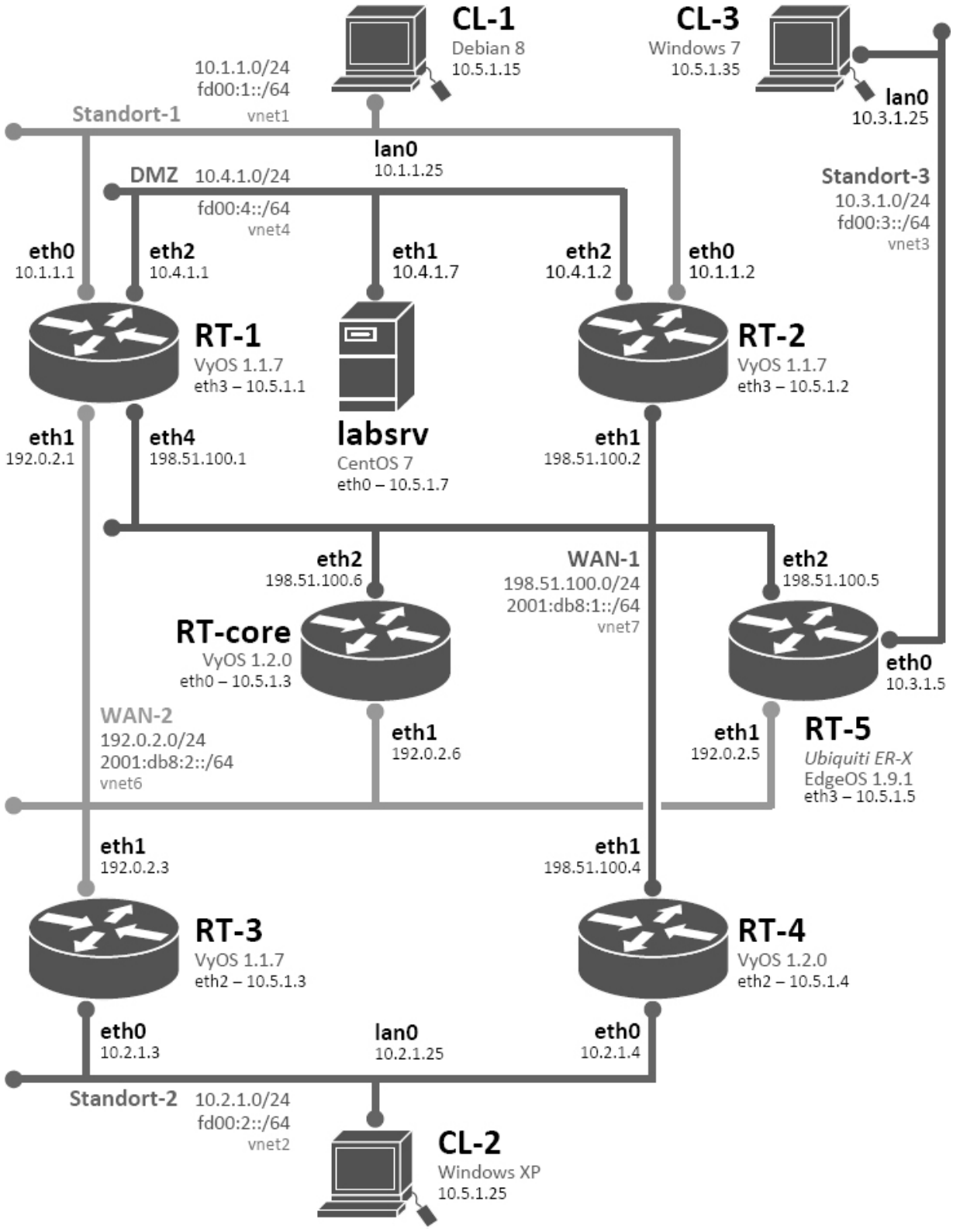

Abbildung 1.1: Das Labornetzwerk als Vorlage für alle Kapitel

Welches Routerinterface in welchem virtuellen Netz zuhause ist, zeigt Tabelle 1.1.

Router	**Interface**	**VMnet/vboxnet**	**IPv4**	**IPv6**
RT-1	eth0 eth1 eth2 eth3 eth4	VMnet1 VMnet6 VMnet4 Management VMnet7	10.1.1.1 192.0.2.1 10.4.1.1 10.5.1.1 198.51.100.1	fd00:1::1 2001:db8:2::1 fd00:4::1 fd00:5::1 2001:db8:1::1
RT-2	eth0 eth1 eth2 eth3	VMnet1 VMnet7 VMnet4 Management	10.1.1.2 198.51.100.2 10.4.1.2 10.5.1.2	fd00:1::2 2001:db8:1::2 fd00:4::2 fd00:5::2
RT-3	eth0 eth1 eth2	VMnet2 VMnet6 Management	10.2.1.3 192.0.2.3 10.5.1.3	fd00:2::3 2001:db8:2::3 fd00:5::3
RT-4	eth0 eth1 eth2	VMnet2 VMnet7 Management	10.2.1.4 198.51.100.4 10.5.1.4	fd00:2::4 2001:db8:1::4 fd00:5::4
RT-5	eth0 eth1 eth2 eth3	VMnet3 VMnet6 VMnet7 Management	10.3.1.5 192.0.2.5 198.51.100.5 10.5.1.5	fd00:3::5 2001:db8:2::5 2001:db8:1::5 fd00:5::5

RT-core	eth0 eth1 eth2	Management VMnet6 VMnet7	10.5.1.6 192.0.2.6 198.51.100.6	fd00:5::6 2001:db8:2::6 2001:db8:1::6
labsrv	eth0 eth1	Management VMnet4	10.5.1.7 10.4.1.7	fd00:5::7 fd00:4::7

Tabelle 1.1: Alle Router mit Interface und VMnet/vboxnet

Technisch nicht erforderlich, aber hilfreich zum Auswerten: Die Netzwerkkarten der VMs verwenden vordefinierte MAC-Adressen. Damit sind alle Geräte in den Kommandoausgaben eindeutig erkennbar und mit den Beispielen im Buch vergleichbar.

Getestet und geprüft sind die Labs mit VMware Workstation 10, VMware ESXi 6 und VirtualBox 5.0.

Hardware

VyOS läuft grundsätzlich auf Geräten mit i386- oder x86_64-Prozessor. Auch der Typ der Netzwerkkarte ist unwichtig, da das Labor-Netz Verständnis bieten soll und nicht Höchstleistung. Bei Unsicherheit über passende Hardware lohnt sich ein Blick in die Kompatibilitätsmatrix von Debian [2].

Die Netze zwischen den Routern basieren auf Ethernet. Jedes Teilnetz ist eine eigene Broadcast-Domäne. Bei der Verkabelung ist es also wichtig, dass sich die Kabel verschiedener Netzsegmente nicht vermischen. Für die korrekte Trennung gibt es zwei gängige Methoden:

Trennung mit Switchen Jedes Netzsegment hat seinen eigenen Switch oder Hub. Die Switche sind untereinander

nicht verbunden. Da die Subnetze eher klein sind, reichen 5-Port-Geräte aus. Ein beliebiger Switch ist dafür passend.

Trennung mit VLANs Alle Kabel führen zum selben Switch. Kabel bzw. Switchports, die zum selben Netzsegment gehören, landen in einem gemeinsamen virtuellen LAN (VLAN). So erhalten beispielsweise alle Switchports zum/vom hellgrauen Kernnetz die Zuordnung zu VLAN 6.

Da alle Router mit allen Anschlüssen mit diesem Switch verkabelt sind, muss es ein Modell mit ausreichend vielen Ports sein. Der Switch muss kein Routing zwischen den VLANs beherrschen. Ein VLAN-fähiger Layer-2 Switch ist ausreichend.

Ein Mischbetrieb ist ebenfalls möglich: Beispielsweise terminieren die WAN-Segmente auf einen Switch und die Standort-Netze auf einen anderen Switch. Die Anforderung an die Geräte entspricht der Methode *Trennung mit VLANs.*

Router

Die VyOS-Router verwenden die aktuelle stabile VyOS-Version 1.1.7 und teilweise zum Reinschnuppern und Vergleichen die Vorab-Version 1.2.0 als 64-bit-Image. Der einzige Ubiquiti-Router läuft auf der aktuellen EdgeOS-Version 1.9.1. Wenn zusätzliche Versionen oder Geräte anderer Hersteller mitspielen, wird das entsprechende Gerät ersetzt oder der Laboraufbau ergänzt.

Jeder Router hat eine zusätzliche Netzwerkkarte für den Konsolenzugriff. Darüber erreicht der SSH-Client sein Ziel, wenn eine Konfigurationsänderung mal schiefgeht. Dieses Management-Interface kann auch weggelassen werden, wenn die Hardware nicht genug Schnittstellen bietet.

Die Labor-Router sind von eins bis sechs durchnummeriert. Diese Router-Nummer findet sich in den

IPv4-, IPv6- und MAC-Adressen wieder. Damit sind Adressen in einer Kommandoausgabe leichter dem passenden Gerät zuzuordnen.

Der Name der Netzkarte ist stets am Routersymbol angeschlagen. Die vollständige IPv4-Adresse ist unterhalb davon abgedruckt. Die Angaben zum IPv4-Netz und IPv6-Präfix stehen unweit davon an der Subnetz-Linie.

Der Core-Router benutzt schon VyOS in der neueren Betaversion. Das hat weniger mit Mut zu tun, als mit der Verfügbarkeit von Features: Erst VyOS 1.2.0-beta1 bietet einen PPPoE-Server, der im Laboraufbau von Kapitel 12 benötigt wird.

Die Kompatibilität zu EdgeOS wird an Router 5 getestet: Ein kleiner Ubiquiti EdgeRouter-X nimmt im Lab Platz und bietet fünf Gigabit-Ports. EdgeRouter gibts nur in Hardware. Wenn EdgeOS uninteressant ist, kann dieser Teil des Netzwerks ausgelassen oder durch VyOS ersetzt werden.

Adressierung

Die Netze der imaginären Außenstellen bauen auf private IPv4-Adressen bzw. Unique-Local IPv6-Adressen. In jedem Standort gibt es einen symbolischen Client, der nur zum Prüfen von Features oder zum Erzeugen von Datenverkehr benutzt wird. Mehr als ping, traceroute, netstat oder einen Webbrowser wird nicht gefordert. Das Betriebssystem ist relativ egal; Im Demo-Lab finden aus Popularitätsgründen Debian und Windows 7 Verwendung.

Die zwei standortverbindenden Netze stellen das zentrale Kernnetz dar. Zur besseren Unterscheidung der IP-Adressen bedienen sich die Geräte aus den Adressblöcken für Dokumentation (RFC 5737): 192.0.2.0/24 und 198.51.100.0/24. Die IPv6-Adressen stammen ebenfalls aus

unterschiedlichen Bereichen, um eine Unterscheidung optisch zu vereinfachen; die Standort-LANs benutzen fd00::/16 und das Kernnetz bedient sich aus dem Präfix 2001:db8::/32.

Die Adressen sind genau dafür vorgesehen und kollidieren nicht mit einem öffentlichen Bereich. Weiterhin ist die Adressierung bewusst einfach gehalten: Die Adressbereiche sind einheitlich strukturiert und haben nur „normale“ Netzmasken von /24 (IPv4) oder /64 (IPv6). Erst die Kapitel rund um Routing und OSPF führen zu Trickserei mit Masken, Sub- und Supernetting.

Zusammengefasst zeigt Tabelle 1.2 die IPv4- und IPv6-Bereiche, die sich hinter den VMnet-Netzen verstecken. Zusätzlich benötigte Adressen (z. B. für PPPoE, Tunnel, VRRP) stammen aus denselben Bereichen.

VM/vboxnet	**Funktion**	**IPv4**	**IPv6**
VMnet1	Standort 1	10.1.1.0/24	fd00:1::/64
VMnet2	Standort 2	10.2.1.0/24	fd00:2::/64
VMnet3	Standort 3	10.3.1.0/24	fd00:3::/64
VMnet4	DMZ	10.4.1.0/24	fd00:4::/64
VMnet5	Management	10.5.1.0/24	fd00:5::/64
VMnet6	VPN	10.6.0.0/16	fd00:6::/64
VMnet7	WAN hellgrau	192.0.2.0/24	2001:db8:2::/64
	WAN dunkelgrau	198.51.100.0/24	2001:db8:1::/64
	VPN/OSPF	203.0.113.0/24	2001:db8:3::/64

Tabelle 1.2: Alle virtuellen Netze mit IP-Bereichen

Labor-Server

Alle zentralen Funktionen übernimmt der Labor-Server, der ebenfalls physikalisch oder virtuell integriert wird. Wenn der VyOS-Router auf ein Client-Server-Protokoll getestet wird, übernimmt der Labserver stets die Rolle des Gegenstücks. Er akzeptiert von den Routern Anfragen zu NTP, DNS, Syslog, FTP/TFTP, NetFlow und HTTP.

Der eingesetzte Labserver setzt auf CentOS 7, um das Lab etwas weniger Debian-lastig zu gestalten.

Verwendung

Jedes Kapitel verwendet nur einen Teil des Labornetzwerks. Weniger Geräte ermöglichen eine bessere Kontrolle, wenn es an die Beispiele und Kommandoausgaben geht. Diese Limitierung dient nur der Übersicht – gerne dürfen weitere Router zugeschaltet werden, um Features intensiver zu testen.

Die IP-Adressen bleiben stets dieselben, wenn auch mit anderer Bedeutung.

Kapitel 2

Plattform

Im nächsten Schritt geht es an die Verwirklichung des Labors. Es beginnt mit der Erstellung oder Beschaffung der Geräte, gefolgt von der Installation und zuletzt mit der Vernetzung.

Wie in Kapitel 1 schon angedeutet, kann das Lab auf physikalischer Hardware laufen oder komplett in einer virtuellen Umgebung sein Zuhause finden. Für den Aufbau macht das einen großen Unterschied – für die Beispielszenarien der folgenden Kapitel ist es unentscheidend.

Die Vorgehensweise bei allen Methoden ist einheitlich: Es beginnt mit dem Anlegen der virtuellen Netze, deren Trennung entweder mit einem virtuellen Switch oder einer Portgruppe erfolgt. Danach gehts ans Erstellen der virtuellen Maschinen (VM) und zuletzt erhalten die neuen VMs ihre Netzadapter in den beheimateten VM-Netzen.

Die Wahl der Virtualisierungssoftware hängt von den persönlichen Vorzügen ab. Die folgenden Erklärungen beziehen sich auf VMware ESXi, Workstation und Player, sowie auf VirtualBox.

Dieses Kapitel kann kein Fachbuch über VMware oder VirtualBox ersetzen! Die Installation der VMs setzt Grundwissen in den jeweiligen Produkten voraus. Die Beschreibungen behandeln nur den Aufbau der neuen VM