

Themen / Inhalte /	Kenntnisse und Fertigkeiten		Nach dem Projekt, der Lernsituation o.ä.	Ausblick / eigene Ziele, Wünsche
	Vor	Während		
Linux (Shell: Navigation, Befehle, Scripte, Editor)	Auf der Arbeit oder zu Hause arbeite ich selten in der Konsole oder mit Linux. Über die Verzeichnisstruktur und wichtige Dateien weiß ich kaum etwas. Die meisten Befehle und deren Optionen sind mir nicht geläufig. Ich habe in Linux noch kein Script geschrieben. Ich nutze bisher stets einen grafischen Editor.	Ich nutze anfangs viel Google um nach Befehlen oder Dateien zu suchen. Die Befehle werden geläufiger und ich versuche mich an die man pages für Kommandos zu gewöhnen. Um Dateien schnell zu finden ist <code>find / -name [name]</code> sehr hilfreich. Anfangs wechsle ich noch in den Ordner um Dateien zu öffnen. Das ist nicht nötig. Alle Dateien lassen sich von überall her ansprechen. Das ist bei grafischen Oberflächen nicht gegeben. Ein Script wird stets mit <code>#!/[Path][Shell]</code> eingeleitet. Um ein Script auszuführen muss man die Berechtigung mit <code>chmod +x [Path][Script]</code> ändern. Zum Editieren wechsle ich zwischen nano und vi, welcher mich an das <code>less</code> Kommando erinnert. Aufgerufen wird ein Script über <code>./[Path][Script]</code> . Kommandos, die man aufruft werden nicht gespeichert, solange die Änderung nicht in eine Datei geschrieben werden. Die Interfaces konfiguriere ich in <code>/etc/interfaces</code> . Hier kann man auch DSN, statische Routen und NAT konfigurieren und speichern. Diese Datei wird beim Booten von Linux geladen.	Ich fühle mich in der Shell wesentlich wohler und finde mich in der Verzeichnisstruktur zurecht. Das Arbeiten in der Konsole bringt viele Vorteile mit sich. Getätigte Befehle lassen schnell wiederholen und man muss sich nicht lange durch irgendwelche Fenster und Verzeichnisse navigieren. Viele Programme haben <code>.dotfiles</code> , mithilfe man diese konfigurieren kann. Einige Befehle haben sich eingepreßt. Um mir Optionen anzeigen zu lassen benutze ich <code>--help</code> und für genauere Informationen <code>man</code> . Nichtsdestotrotz greife ich noch, vor allem bei mir noch unbekannten Befehlen, auf Google zurück. Bevor man ein Script mit mehreren Befehlen schreibt, kann man die einzelnen Befehle erstmal problemlos in der Shell testen und hier auch nach Hilfe suchen. Alle Befehle haben Standardkanäle für Ausgabe(0), Eingabe(1) und Fehler(2). Dieses Verhalten lässt sich mittels <code>[x]></code> ändern. Ich bevorzuge vi nano, da ich so auch gleich die Navigation mit <code>less</code> verinnerliche und ich beim Editieren die <code>home row</code> nicht verlassen muss. Die DNS-Server trägt man in der <code>/etc/resolv.conf</code> ein. Um das mittels eines Scripts zu lösen kann man <code>>></code> benutzen. Für Routen und NAT erstellt man am besten ein Script.	Ich würde gerne weiter mit Linux arbeiten um meine Fähigkeiten im Scripting und Nutzen des Shell zu verbessern und so auch meine Produktivität zu steigern. Zudem ist Linux „sauberer“ als Windows, wo schon eine schier unendliche Anzahl an Diensten, Programmen und Bibliotheken vorinstalliert ist und man von vornherein jeglichen Überblick verloren hat, welche Dienste und Programme schon vorinstalliert sind und was diese eigentlich schon alles definieren und ausführen. Mir gefällt die Logik von vi und würde gerne die diversen Eingaben aus dem Effeff beherrschen. So könnte ich wesentlich schneller arbeiten ohne auch nur die Tastatur verlassen zu müssen.
Netzwerk (NAT, Portforwarding, Statische Routen)	Ein Netzwerk zu konfigurieren haben ich bisher nur unter Windows mithilfe einer GUI gemacht. Wofür Statische Routen, NAT usw. gebraucht werden, wusste ich zwar jedoch kannte ich nicht den	Das wichtigste ist erstmal, dass ich lerne dass ich ohne weiteres ein Linux-System als Router konfigurieren kann. Um IP-Forwarding zu aktivieren nutze ich <code>echo "1" > /proc/sys/net/ipv4/ip_forward</code> , für die statische Route <code>ip route add -net [Netz] netmask [Netzmaske] gw [gw]</code> . Um NAT zu konfigurieren für die Netzwerke nutze ich <code>iptables -A FORWARD -o eth1 -s 172.16.9.0/24 --ctstate NEW -j ACCEPT</code> . NAT konfiguriere ich	Ich habe diese Befehle in mein Firewall.sh Script geschrieben und bin jetzt in der Lage, diese Befehle automatisiert ausführen zu lassen. Mithilfe von NAT werden interne Adressen in eine öffentliche Adresse des Routers gewandelt. Der Router speichert diese Zuordnung in eine Tabelle. Dort werden die Anfragen mit Hilfe von Portnummern gespeichert um diese wieder dem Client und dem zugehörigen Dienst zuordnen zu können.	Ich habe mein Wissen definitiv gesteigert und werde es auch noch weiter. Ich finde es hat Spaß gemacht und hilft mir auch auf der Arbeit. Ich werde mir mal privat ein kleines Netzwerk erstellen und dafür ein raspberry pi nutzen.

	genauen Inhalt und wie ich diese unter Linux konfiguriere.	mittels <code>iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE</code> . Danach muss ich ggf. die Interfaces über <code>ifup / ifdown</code> neustarten. Speicher ich diese Befehle in <code>interfaces</code> kann ich mit <code>post-up / pre-down</code> dafür sorgen, dass diese Konfiguration automatisch beim Booten geladen wird		
Firewall	Wofür eine Firewall gut ist, war mir schon bewusst, jedoch nicht nach welchem Prinzip sie arbeitet. Auch fand jegliche Konfiguration stets über eine GUI statt.	Ich lerne, dass solche Befehle in einer Liste abgearbeitet werden. Diese werden mit <code>iptables -A</code> angehängt. Dabei unterscheidet der Router in der INSIDE, OUTSIDE und FORWARD-chain. Man kann hier die Ports, das Protokoll sowie die Quell- und Zieladressen definieren. Dabei spielt es immer eine Rolle aus welcher Richtung die Anfrage kommt und an wen sie gerichtet ist. Diese Regeln werden von oben nach unten abgearbeitet.	Ich habe ein Firewall.sh Script. In diesen sich jetzt die NAT-Regeln und ein paar weitere Berechtigungen für DNS, HTTP/S und SSH. Ich denke, ich habe das Prinzip der Abarbeitung der ACLs verstanden und kann die verschiedenen chains auseinanderhalten.	Mein Überblick hat sich erhöht und ich kann das Prinzip gut nachvollziehen. Inwieweit ich diese Thematik jedoch noch vertiefe kann ich zum jetzigen Zeitpunkt noch nicht sagen, könnte mir aber vorstellen mal für zu Hause mir eigene Firewall aufzusetzen.
VMs	Ich habe vorher schon mit VMware und HyperV gearbeitet.	Ich habe VMWare Player genommen um Linux auszuführen.	Ich habe über Virtualisierung nichts Neues gelernt.	Ich werde wohl mehr mit HyperV, aufgrund der kostenlosen Lizenz arbeiten. So kann ich auch ein komplettes virtuelles Netzwerk errichten.