

Kompetenzportfolios (Thema: Aufbau einer DMZ)

Kompetenzportfolio - Andreas Biller:

Themen / Inhalte	Kenntnisse & Fertigkeiten			Ausblick / Fazit
	Vor dem Projekt	Während dem Projekt	Nach dem Projekt	
Linux (Basics): Command Line Interface, Navigation im Dateisystem, Benutzer und Gruppen, Man Pages, Berechtigungen, usw.	Im Betrieb wird mit Linux gearbeitet und ich bin u. a. für Installation und den technischen Support zuständig. Daher arbeite ich täglich auf der Kommandozeile im Linux Dateisystem, lege Benutzer an und vergebe und ändere Berechtigungen. Sich wiederholende Tätigkeiten werden mit Ansible-Playbooks oder Shell-Skripten erledigt, um Fehler zu vermeiden. In Man-Pages schaue ich regelmäßig, um benötigte Parameter für seltener benutzte Befehle herauszusuchen. Die Rechner im Betriebsnetz lassen sich remote über ssh administrieren. Mit grep, cut, pipes, xargs, wget und anderen Befehlen werden hier z. B. aus csv-Dateien urls aus Spalten ausgelesen, mit anderen Kriterien verglichen oder korrigiert. So können durch einfache Bash-Befehlsketten unter Linux sehr kreative Ergebnisse erzielt werden, für die andere schon mal Excel und dessen Sortierfunktionen benutzen müssen, um Daten aus csv-Dateien auswerten zu können.	Da ich im Betrieb jeden Tag auf der Linux-Kommandozeile arbeite gibt hier nur bedingt neues für mich zu lernen. Die Grundlagen werden eher durch Wiederholung und Anwendung aufgefrischt, wenn in einem Lab Befehle geübt werden, die im Arbeitsalltag nicht so häufig Verwendung finden. So habe ich z. B. dank der guten Erklärung des netacad-Labs die verschiedenen Eigenschaften bei gesetzten Sticky-Bits besser verstanden. Auch die Suche nach Dateien mit find fand ich erfrischend einfach, da ich im Betrieb selten nach Dateien an sich, sondern eher nach Text in Dateien mit Befehlen wie grep und Regulären Ausdrücken suche. Gerade das Arbeiten in der Bash schätze ich zunehmend dank dem schnellen Wiederholen von Befehlen durch die „Pfeil hoch“- bzw. „Pfeil runter“-Tasten oder dem interaktiven, rekursiven Suchen in der shell-History mit Strg+R.	Ich kenne mich dank vieler Aufgaben im Betrieb bereits gut im Linux-Dateisystem aus, deswegen wusste ich bereits, wo z. B. ausführbare Dateien liegen (bin), wie man die Path-Variable für neu installierte Befehle erweitert, wo Konfigurations- (etc) oder Log-Dateien (var) abgelegt werden. Ich kann dank den netacad-Labs und Chaptern vieles wiederholen und Skripte, Exit-Codes, verschiedene Outputvarianten mit Umgebungsvariablen, Pipes und Befehlen zu komplexen Anwendungen verknüpfen und weiß wo und wie ich Hilfe zu den benutzten Programmen finde.	Ich werde weiterhin auf Linux arbeiten und auch zuhause nutze ich in meiner Freizeit mindestens einen Rechner, auf dem Linux installiert ist. Grundwissen wird also immer wieder angewendet werden und sich festigen und weiterentwickeln, bis ich hoffentlich nicht mehr bei so vielen der seltener benutzten Befehlen in die Anleitung schauen muss.
Linux (Advanced): Shell-Skripte & Befehle, Umgebungsvariablen, Editoren (vi, nano, etc.), Konfigurationsdateien, usw.	Andere Aufgaben im Betrieb werden durch Skripte (.sh oder .rb) zu wiederholbaren Prozessen, CSV-Listen mit Befehlen wie grep, cut, xargs und dem Pipe-Operator ausgewertet. Auch Webseiten für Kunden erstelle ich dort über Commandline-Tools indem Markdown-Dateien mit jekyll und liquid templates zu html umgewandelt wird. Umgebungsvariablen benutzen wir in der Entwicklung und Produktion um die verwendeten Passwörter aus unserem Programmcode herauszuhalten. Zum Editieren benutze ich in der bash meist nano, ansonsten benutze ich der zusätzlichen Funktionalität wegen lieber einen graphischen Editor wie gedit oder sublime.	Hier war der Nutzen schon größer, da ich im Betrieb aufgrund der für unsere Plattform genutzten Programmiersprache meine Skripte hier meist in ruby schreibe und bash-Skripte nur dann verwende, wenn ich unbedingt einige der normalen Linux-Befehle für etwas Spezielles benötige. Aber auch hier ist vieles Wiederholung, gehört doch die Arbeit im CLI mit Umgebungsvariablen, Skripten, allgemeinen Befehlen wie git oder eher ruby-spezifischen wie rvm, bundle, rails, rubocop, etc. zu meinen täglichen Aufgaben. Zur Konfiguration editiere ich auch schon mal Dateien wie sudoers.d um Benutzern sudo-Rechte zu entziehen. Das Arbeiten in der shell fällt mir in der Arbeit dank um eigene Aliase erweiterter bash.rc einfacher als in einem unmodifizierten Debian wie dem genutzten	Ich habe viele Dinge wiederholt, die im Betrieb auch schon relevant waren, einige Sachen besser verstanden und gerade beim Skripten in der Shell Dinge angewendet, die im Betrieb wegen der Präferenz ruby zu benutzen häufig nicht in Shell-Skripten umgesetzt werden. Obwohl es mich sehr gereizt hätte, besser mit vi umgehen zu lernen, hat mir auch dieser kurze Versuch damit editieren zu wollen gezeigt, warum ich hier wenn möglich immer noch zu nano oder anderen, weniger komplexen Editoren greife.	Auch hoffe ich irgendwann die Zeit aufzubringen mich länger und intensiver mit vi oder vim zu beschäftigen. Da unser Betrieb stetig wächst und ich einen Linux-Rechner nach dem anderen konfiguriere, werde ich auch hier weitere Fertigkeiten entwickeln und mit dem gelernten Wissen Skripte schreiben und in Übung bleiben.
Netzwerk: Planung & Darstellung, Dienste, Routingtabellen, Statische Routen,	Netzwerke kenne ich hauptsächlich aus dem privaten Bereich, z. B. von der Konfiguration des Routers bzw. Netzwerkdruckers zuhause oder von kleinen Netzwerken über Hubs auf LAN-Parties. Mit ping habe ich bereits das eine oder andere mal die Funktionalität von LAN- und Internetverbindungen getestet, auch wenn ich nicht wußte, was dort alles genau passiert.	Dank der vorherigen Konfiguration von Netzwerken im ITS-Unterricht mit Packet-Tracer in der cisco-Syntax, auf Windows-Rechnern in deren Dialekt und nun dem Übersetzen bzw. Wiederholen der bereits bekannten Befehle nach Linux lerne ich hier gefühlt am meisten, da ich bei Befehlen wie ip route sehe, wie ähnlich sich die unterschiedlichen Systeme sind und dass sich meistens nur einige Schlüsselwörter oder die Schreibweise der Parameter	Ich habe ein besseres Verständnis davon erhalten, wie die Kommunikation in einem bzw. die Konfiguration eines Netzwerkes an sich funktioniert, egal auf welchem System man diese vornimmt. Dennoch fehlt mir noch einiges an Erfahrung, da ich im Fall von Problemen immer noch lange Suchen muß, bis ich das Problem eingrenzen kann.	Auch im Bereich Netzwerk kann ich das eine oder andere Gelernte aus diesem Projekt im Betrieb bestimmt noch praktisch umsetzen, da durch unseren momentanen Wachstum auch das verwendete Netzwerk immer wieder erweitert werden muss.

Kompetenzportfolios (Thema: Aufbau einer DMZ)

Testen (ping), usw.		unterscheiden, die zugrundeliegenden Prinzipien jedoch gleich sind.		
NAT: Webserver aufsetzen, Schnittstellen konfigurieren, Konfigurationen speichern, usw.	Ich habe schon eigene Webserver (apache2) konfiguriert und betrieben, um Webseiten lokal zu testen und mein altes Portfolio selbst zu hosten, bzw. Portforwarding im Router eingerichtet um online mit Freunden spielen zu können. Den Router sowie die Schnittstellen habe ich bisher über graphische Oberflächen (unter Windows) eingerichtet.	Da ich bereits eigene Erfahrungen mit dem Aufsetzen und Konfigurieren von apache2 sowohl unter Windows wie unter Linux habe, ist das miniwebserver Tool keine Quelle neuer Lernerfahrungen (wenn auch einfach zu benutzen). NAT und Portforwarding sind dank dem gezielten Suchen nach Informationen im Internet und dem in ITS Erlernten schnell in die benötigten Konfigurationsdateien (/proc/sys/net/ipv4/ip_forward) geschrieben und über den auch für die Firewall benutzten Befehl iptables in der FORWARD-Chain mit dem Parameter MASQUERADE eingerichtet.	Auch im Betrieb arbeite ich unter anderem mit einem lokalen apache2 Webserver, um zu bearbeitende Webseiten in html oder php über localhost anzeigen zu lassen. Auch jekyll beim Erstellen von Webseiten sowie das „ruby on rails“-Framework bietet neben einer CLI- auch eine über den Browser geservte Entwicklungsumgebung, in der einiges konfiguriert werden muss, bis alles läuft wie es soll. Das erlangte Wissen aus dem Projekt wird hier in der einen oder anderen Form sicher immer wieder auftauchen.	NAT muss ich betrieblich nicht wirklich einrichten, allerdings kann ich erlerntes Wissen bestimmt auf die eine oder andere Art bei der Arbeit an bzw. der Konfiguration von unserer bei Heroku gehosteten Webapp weiter verwenden.
Firewall: Skript erstellen bzw. anpassen, iptables verstehen & benutzen (Tables, Chains, Rules), usw.	Im Rahmen mehrerer Weiterbildungsmaßnahmen des Jobcenters durfte ich bereits vor der Arbeit im heutigen Betrieb als Aushilfe im Server- und IT-Bereich an der Rixdorfer Grundschule für einige Zeit Grundlagen in der PC- und Linux-Administration erlernen, unter anderem auch die prinzipielle Arbeitsweise und das Erstellen von Filterregeln für die einzelnen Chains bei einer Firewall, allerdings in einer graphischen Umgebung (webmin). Und leider musste ich diese Fähigkeiten dann nie wirklich praktisch anwenden und hatte das meiste davon bis heute wieder vergessen.	Da wir mit den Linux-Grundlagen und dem Erstellen des Netzwerkes die erste Hälfte des Projektes beschäftigt waren und auch in ITS erst zum zweiten Halbjahr etwas zu Access-Control-Lists gelernt haben, hatte ich nur dank dem beim NAT verwendeten Befehl iptables etwas zur Firewall gelernt, ohne dies jedoch zu wissen.	Nach dem Projekt habe ich die Bedeutung der einzelnen Chains für den Routing-Prozess verstanden und weiß wieder wie die iptable Regeln zusammen mit den Standard-Policies als Firewall verwendet werden können und wie ich diese mit Hilfe unseres Scriptes in der post-up bzw. pre-down Sequenz der interfaces speichern und so sowohl bei einem Neustart durch Befehle wie ifup/ifdown sowie beim Booten des Systems aktivieren bzw. auch wieder deaktivieren kann.	Das Wissen um die Funktionsweise von Firewall-Regeln und Grundlegende IT-Sicherheit sind wichtig, aber die Übung im Erstellen einer Dokumentation nach den IHK-Richtlinien für das bevorstehende Prüfungsprojekt sind dagegen nahezu unbezahlbar und werden mir von all dem Gelernten mit Sicherheit am ehesten noch unschätzbare Dienste leisten.

Kompetenzportfolio - Rico Krüger:

Themen / Inhalte	Kenntnisse & Fertigkeiten			Ausblick / Fazit
	Vor dem Projekt	Während dem Projekt	Nach dem Projekt	
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 nutzte 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 find / -name [name] sehr hilfreich. Anfangs wechsele 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 !#[Path][Shell] eingeleitet. Um ein Script auszuführen muss man die Berechtigung mit chmod +x [Path][Script] ändern. Zum Editieren wechsele ich zwischen nano und vi, welcher mich an das less Kommando erinnert. Aufgerufen wird ein Script über ./[Path][Script]. Kommandos, die man aufruft werden nicht gespeichert, solange die Änderung	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 .dotfiles, mithilfe man diese konfigurieren kann. Einige Befehle haben sich eingeprägt. Um mir Optionen anzeigen zu lassen benutze ich --help und für genauere Informationen man. 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).	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

Kompetenzportfolios (Thema: Aufbau einer DMZ)

		nicht in eine Datei geschrieben werden. Die Interfaces konfiguriere ich in /etc/interfaces. Hier kann man auch DSN, statische Routen und NAT konfigurieren und speichern. Diese Datei wird beim Booten von Linux geladen.	Dieses Verhalten lässt sich mittels [x]> ändern. Ich bevorzuge vi nano, da ich so auch gleich die Navigation mit less verinnerliche und ich beim Editieren die <i>home row</i> nicht verlassen muss. Die DNS-Server trägt man in der /etc/resolv.conf ein. Um das mittels eines Scripts zu lösen kann man >> benutzen. Für Routen und NAT erstellt man am besten ein Script.	und ausführen. Mir gefällt die Logik von vi und würde gerne die diversen Eingaben aus dem Effeß 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 genauen Inhalt und wie ich diese unter Linux konfiguriere.	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 echo "1" > /proc/sys/net/ipv4/ip_forward, für die statische Route ip route add -net [Netz] netmask [Netzmaske] gw [gw]. Um NAT zu konfigurieren für die Netzwerke nutze ich iptables -A FORWARD -o eth1 -s 172.16.9.0/24 --ctstate NEW -j ACCEPT. NAT konfiguriere ich mittels iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE. Danach muss ich ggf. die Interfaces über ifup / ifdown neustarten. Speicher ich diese Befehle in interfaces kann ich mit post-up / pre-down dafür sorgen, dass diese Konfiguration automatisch beim Booten geladen wird	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.
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 iptables -A 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 vertiefen 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.