Albstadt-Sigmaringen University

Netzwerk- und Systemsicherheit

Praktikum

Version 2.0 WiSe 2021

C. Henrich, P. Knittel

Inhaltsverzeichnis

Vorwort

Herzlich Willkommen zum Praktikums des Moduls *Netzwerk- und Systemsicherheit.*

Im Wintersemester 2021/2022 habe ich dieses Dokument, das Sie in diesem Praktikum begleitet, in ein neues Format gesetzt.

Über Fragen, Verbesserungsvorschläge und Anregungen freue ich mich. Schicken Sie diese bitte per E-Mail an henrich@hs-albsig.de.

Changelog

Hier finden Sie die wichtigsten Änderungen, die in einer Version hinzugekommen sind.

Version	Änderungen
2.0	neues Format,Kontrollfragen im Dokument

Kapitel 1

Einführung

Als Basis für die Implementierung der Aufgabe dient eine rein virtuelle Umgebung, bestehend aus virtualisierten Maschinen (VMs) und Netzwerken. Jedes Team erhält Root-Zugriff auf drei VMs (zwei Desktop- und eine Router-Installation) in dieser Umgebung. Das Betriebssystem der VMs ist eine Linux-Distribution, die per SSH-Zugang der Aufgabe entsprechend angepasst werden muss. Die Bearbeitung erfolgt in der Regel in Zweierteams.

1.1 VMs

Kurzbeschreibung der vorbereiteten VMs

1.1.1 Desktop-Installation

Die Desktop-Installation (im folgenden als *Desktop* bezeichnet) umfasst

- Betriebssystem: Devuan 2.0 ASCII (64bit) mit angepasster Mate Desktop-Umgebung
- Virtuelle Hardware: 2 Prozessorkerne , 1 GB Ram , 5 GB HD , 1 Ethernetkarte
- Anpassungen
  - IP Konfiguration per DHCP
  - Hostnamen `labXY` (`XY`: `01` bis `45`, ohne die durch 3 teilbaren Nummern).
  - IPv6 mit Kernel Parameter deaktiviert
  - verschiedene Dienste deaktiviert
  - kein grafischer Netzwerkmanager installiert
  - Bootloader (GRUB, GRand Unified Bootloader) zu Praktikumszwecken umkonfiguriert
  - Installations-Benutzer wurde entfernt, direkter Root-Zugang freigeschaltet
  - `sudo` Technik wird im Praktikum nur in einer Aufgabe verwendet, ist sonst allerdings empfehlenswert
  - Benutzer `lab99` für die Auswertungen eingerichtet (vgl. Abschnitt 2.1).
  - Installation eines `x2Go`-Servers (komprimiertes/zwischengespeichertes X11 Protokoll über SSH)

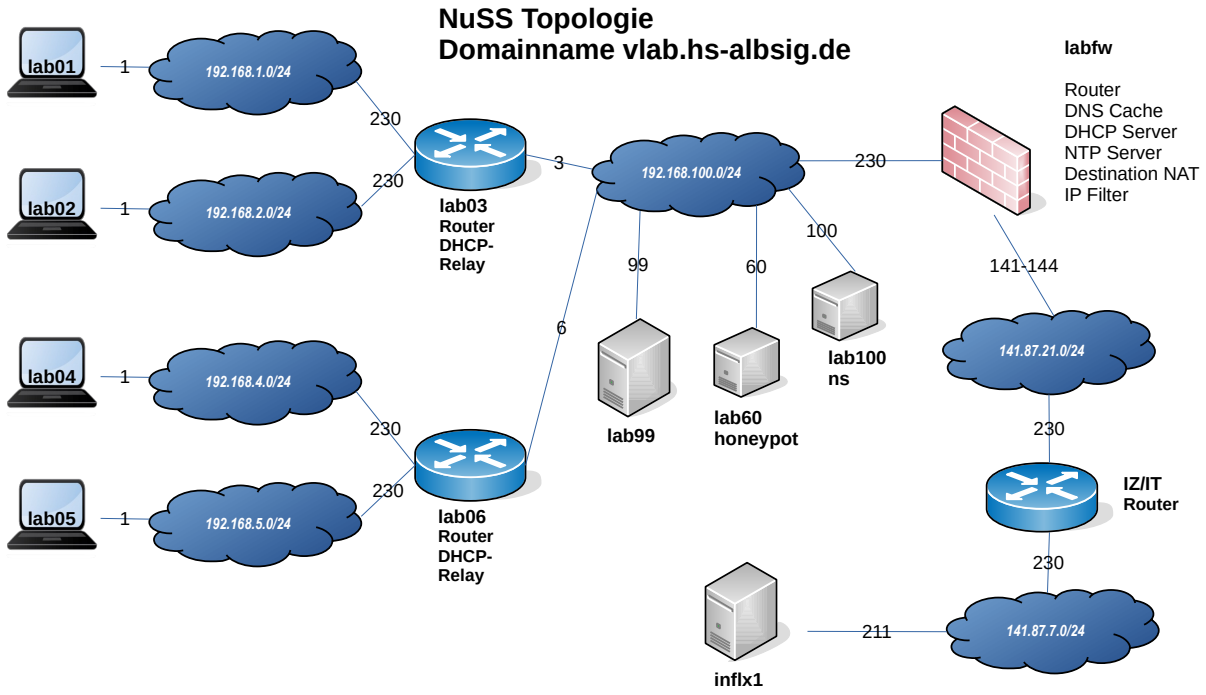

Abbildung 1.1: Topologie

Hinweis

Die `sudo` Technik wird im Praktikum nur in einer Aufgabe verwendet, ist sonst allerdings empfehlenswert.
Im folgenden Dokument werden die Befehle entweder mit vorangestelltem $ oder # angegeben. Befehle mit `$` sollten Sie als Benutzer ausführen, Befehle mit `#` entweder als `root` oder mit `sudo`.

1.1.2 Router-Installation

Die Router-Installation (im folgenden als *Router* bezeichnet) umfasst:

- Betriebssystem: Devuan 2.0 ASCII (64bit)
- Virtuelle Hardware: 2 Prozessorkerne , 512 MB Ram , 5 GB HD , 3 Ethernetkarten
- Anpassungen
  - IP Konfiguration von `eth0`, `eth1` und `eth2` statisch konfiguriert, statisches Routing eingerichtet
  - Hostnamen `labXY` (`XY`: alle durch 3 teilbaren Nummern von `01` bis `45`).
  - verschiedene Dienste deaktiviert
  - Bootloader (GRUB, GRand Unified Bootloader) zu Praktikumszwecken umkonfiguriert

- Installations-Benutzer wurde entfernt, direkter Root-Zugang freigeschaltet
- `sudo` Technik wird im Praktikum nur in einer Aufgabe verwendet, ist sonst allerdings empfehlenswert
- Installation eines DHCP-Relay Dienstes für die zugehörigen Desktops

1.2 Virtuelle Netze und Zugang

Kurzbeschreibung der Virtuellen Netze und des Zugangs (vgl. Abbildung 1.1)

Jede VM liegt in einem Class-C Netzwerk im Bereich `192.168.0.0/16` (RFC 1597 Private Internets) und wird per Network Address Translation (NAT) mit dem Hochschulnetz/Internet verbunden. Die Desktop VMs erhalten per DHCP über das jeweilige DHCP-Relay (Router `lab03/lab06/...`) von einem DHCP-Server (`labfw`) immer die HOST-IP `1` in ihrem Class-C Netzwerk sowie das Standard-Gateway.

Die IP Konfiguration der Router VMs ist statisch konfiguriert, zusätzlich ist ein DHCP-Relay-Dienst installiert, der die Anfragen der DHCP Clients an den DHCP Server (`labfw`) weiterleitet.

Als Nameserver wird die VM `labfw` (Zugangsrouter, interne IP-Adresse 192.168.100.230) verwendet, die auch für die notwendigen NAT und Filterregeln zuständig ist. Diese verwaltet alle lokalen Namen der Domain `vlab.hs-albsig.de`. Anfragen über nicht lokale Namen werden nicht weitergeleitet. Die VM `labfw` lässt nur `ssh` (bei uns `22001-22045`) von `inflx1/2/3/4` nach innen (VM-Netze) zu, und stellt zusätzlich noch einen NTP-Server zur Verfügung, der von den `vlab.hs-albsig.de` VMs zur Zeitsynchronisation genutzt werden kann. Des weiteren dient `labfw` als DHCP-Server für alle Desktops VMs.

Der Zugangsrouter `labfw` hat folgende externe IP Adressen über die alle Zugriffe von außen (`inflx1/2/3/4`) auf das interne Netz laufen.

Gruppe 1:	`141.87.21.141`	Gruppe 3:	`141.87.21.143`
Gruppe 2:	`141.87.21.142`	Gruppe 4:	`141.87.21.144`

Alle eingehenden Verbindungen werden mittels der jeweilig auftretenden Portnummer (`220XY`) zur entsprechenden VM `labXY` im internen Netz auf Port 22 weitergeleitet. Beispielhaft der Zugang von `inflx1/2/3/4` aus, als Benutzer `root` auf die VM `lab45` der Gruppe 2 in einem Terminal:

```
$ ssh -p 22045 -l root 141.87.21.142
```

Wie in 1.1 kurz erwähnt wurde auf Ihren Desktop VMs ein `x2Go`-Server installiert, der es Ihnen ermöglicht auch komplette X11 Sitzungen mit Desktop-Umgebungen, wie z. B. Mate/XFCE4, innerhalb eines Fensters auszuführen. Dazu kommen wir in Abschnitt **??**.

1.3 Vorbereitung

1.3.1 Erster Login

Aufgabe

Aufgabe 1.1 Richten Sie Ihre Praktikumsumgebung ein.

1. Stellen Sie auf `inflx1/2/3/4` in einem Terminal eine Verbindung per `ssh` zu Ihren jeweiligen VMs (alle drei) als Benutzer `root` her.
2. Nach erfolgreichem Anmelden mit dem Erstpasswort machen Sie sich mit den Maschinen und der Umgebung vertraut:
   - versuchen Sie die VMs der anderen Teilnehmer mit dem Befehl `ping` zu erreichen, danach `labfw` sowie eine Maschine außerhalb von `vlab.hs-albsig.de` (z. B. `inflx1.ad.fh-albsig.de`).
   - Stellen Sie fest, wie viel freien Festplattenplatz Sie noch haben (`df`, insbesondere die Root-Partition '/'), wie viele und welche Prozesse/Threads laufen (`ps -ef` oder `ps axu`), und welche TCP/UDP Ports sich auf Ihrer Maschine im *Listen* Modus (sogenannte *offenen Ports*) befinden (`lsof -i -P -n` und `netstat -a -A inet -n -p` bzw. `netstat -a -A inet6 -n -p`).
   - Mit dem Befehl `ifconfig` können Sie sich u.a. die Konfiguration der Ethernetkarten anzeigen lassen.
   - Aktuelle Routing Einträge, insbesondere auf Ihrem Router, können Sie z. B. mit den Befehlen `netstat -r -n` oder `route -n` einsehen.

   Die Ergebnisse der Befehle müssen nicht dokumentiert werden, diese dienen lediglich zu Ihrer Orientierung.
3. Ändern Sie mit dem Befehl `passwd` die Passwörter des Benutzers `root` auf Ihren VMs.

Achtung

Stellen Sie sicher, dass Sie die neuen `root`-Passwörter nicht vergessen.

1.3.2 Paketquellen umstellen

Um die folgenden Aufgaben besser umsetzten zu können werden nun die sogenannten *Paketquellen* Ihres Systems auf eine lokale Quelle umgestellt. Normalerweise wird weitere Software und Updates für Ihr System aus dem Internet heruntergeladen und installiert. Im Rahmen dieses Praktikums nutzen wir besser einen eigenen Server, der als sogenannter *Mirror* eingerichtet ist. Hierzu dient die VM `lab99.vlab.hs-albsig.de`

Aufgabe

Aufgabe 1.2 Stellen Sie die System-Paketquellen auf `lab99.vlab.hs-albsig.de` um. Um den Mirror zu nutzen, ändern Sie die Datei `/etc/apt/sources.list`. Der Befehl `# cat /etc/apt/sources.list` zeigt die aktuellen System-Paketquellen an (in unserem Fall leere Datei).
Da wir mit dem `x2Go`-Server auf den Desktop VMs ein Paket eines zusätzlichen Softwareanbieter installiert haben, benötigen wir dessen PGP Key (ID E1F9 5838 5BFE 2B6E) zur Paketverifizierung, und fügen die jetzt lokale Paketquelle der Einfachheit halber in `sources.list` hinzu. Die angepasste Dateien `sources.list` und der `x2Go Public PGP Key` sind auf der VM `lab99` abgelegt. Mit dem Kommandozeilen Programm `wget` lassen sich u.a. bequem einzelne Dateien per `http` von einer entfernten Maschine laden. Mit den folgenden Befehlen lässt sich die Umstellung auf allen VMs z. B. realisieren:

Desktop VMs	`# wget http://lab99.vlab.hs-albsig.de/lab/public/x2Go.gpg`
Desktop VMs	`# apt-key add x2Go.gpg`
alle VMs	`# wget http://lab99.vlab.hs-albsig.de/lab/public/sources.list`
Router	Zeile mit `...x2go...` in der gerade abgeholten `sources.list` entfernen
alle VMs	`# cp sources.list /etc/apt/sources.list`
alle VMs	`# apt-get update`

1.3.3 Systemupdate durchführen

Aufgabe

Aufgabe 1.3 Installieren Sie die neusten Updates/Patches für Ihren VMs als Benutzer `root`. Am besten zuerst auf Ihrem Router, danach auf Ihren Desktops.
`# apt-get update` (falls nicht schon in 1.4 ausgeführt)
`# apt-get dist-upgrade`

Vermutlich wird Ihnen in unserer Umgebung zuerst ein *Changelog* angezeigt, Diesen können Sie mit `q` beenden.

Hinweis

Im Normalfall ist es allerdings kein Fehler diesen zu lesen, da diese Auflistung von Änderungen sehr wichtig sein.

Achtung

Sollte ein neues Paket namens `GRUB` (vgl. Seite 1 Anpassungen) installiert werden, und ein Auswahl-Dialog erscheinen, dann wählen Sie bitte *Aktuell installierte Version beibehalten* aus.
Sollte ein neues Paket namens `openssh-server` (vgl. Seite 1 Anpassungen) installiert werden, und ein Auswahl-Dialog erscheinen, dann wählen Sie bitte *Aktuell installierte Version beibehalten* aus (Damit ist die Konfigurationsdatei `/etc/ssh/sshd_config` gemeint).

Mit den Befehlen `apt-get clean` und `apt-get autoremove` können Sie sich etwas mehr freien Festplattenplatz schaffen.

Da vermutlich u.a. eine neuer Kernel dabei ist, müssen Sie Ihre VM neu Starten (`# reboot`). Beachten Sie hierbei die Aufgabe des Routers.

1.3.4 Beliebigen Benutzer anlegen

Aufgabe

Aufgabe 1.4 Legen Sie auf Ihren VMs einen neuen Benutzer mit eigener Gruppe an.

Benutzer/Gruppen-Namen sowie die zugehörigen IDs (Empfehlung >= 1000) sind beliebig. (Außer lab99 und 8000, vgl 1.1, bzw 2.1)

```
# groupadd -g 1234 mygroup
# useradd -m -s /bin/bash -g mygroup -u 1234 myname
# passwd myname
# chmod 700 /home/myname
```

Versuchen Sie sich nun als Benutzer *myname* per `ssh` an allen VMs in separaten Terminals anzumelden.

Dieser Benutzer dient u.a. für Ihre `x2Go`-Logins (vgl. 1.7) auf Ihren Desktop VMs.

Hinweis

Im allgemeinen sollte man auf den Systemen als Benutzer `root` nur arbeiten, wenn dies unbedingt erforderlich ist. Eine komplette X11-Sitzung als Benutzer `root` ist immer zu vermeiden (bzw. ist verboten).

1.3.5 Zugang mit Grafik-Anwendungen

Der Zugang mit Benutzung von Grafik-Anwendungen (X11 Zugang, dieser ist nur zu den Desktop VMs möglich)

X-11 über SSH-Tunnel

Wenn die SSH-Verbindung mit dem Parameter `-X` aufgebaut wird, erstellt diese zusätzlich einen sogenannten X11-Tunnel, der es Ihnen ermöglicht auch graphische Anwendungen auf Ihrer VM zu starten (X-Client), und diese bei sich darstellen und bedienen zu lassen (X-Server).

Bauen Sie eine neue Verbindung von der `as1` zu Ihrer VM wie folgt auf:

```
$ ssh -X -p 220XY -l myname 141.87.21.14Z
```

`XY` entsprechend Ihrer Ziel VM, `Z` entsprechend Ihrem Zugangsrouter.

Starten Sie verschiedene graphische Anwendungen wie z. B. `caja`.

```
$ caja &
```

Nach dem Schließen der Anwendung und der Eingabe von `exit`, kann es sein, dass die SSH Verbindung sich nicht richtig beendet. Geben Sie dann `<STRG-C>` ein.

x2Go-Client Konfiguration

Wie in 1.1 kurz erwähnt wurde auf Ihren Desktop VMs ein `x2Go`-Server installiert, der es Ihnen ermöglicht auch komplette X11 Sitzungen mit Desktop-Umgebungen wie z. B. `Mate`, innerhalb eines Fensters auszuführen. Folgendes ist beispielhaft für eine Desktop VM.

1. Starten Sie den `x2Go` Client (unter Anwendungen/Internet) und konfigurieren Sie eine neue `x2Go` Verbindung wie folgt
   - Sitzung => Neue Sitzung
     - Sitzung
       - Sitzungsname: *beliebiger Name*
       - Host: `141.87.21.14Z`
       - Login: `myname`
       - SSH-Port: `220XY`
       - Sitzungsart: `Mate`
     - Verbindung
       - Verbindungsgeschwindigkeit: LAN
     - Ein/Ausgabe
       - Display: z. B. `800x600` (Das `x2Go`-Sitzungsfenster kann unter Linux beliebig vergrößert/verkleinert werden)
     - Medien
       - Audio deaktivieren
       - client-seitige Druckunterstützung deaktivieren
     - Freigegebene Ordner
       - Keine Ordner freigeben
       - SSH Tunnel deaktivieren
     - OK (nicht vergessen ;-)
2. Danach kann die Verbindung gestartet werden. Sie gelangen an ein Login-Fenster mit dem Sie sich dann als `myname` mit entsprechendem Passwort an Ihrer VM anmelden können.
3. Beenden Sie die `x2Go`-Sitzungen über ein normales Abmelden vom System Ihrer VM.

1.4 Hinweise

Auswertung

Den aktuellen Status Ihrer Aufgabenbearbeitung können Sie mit einem Browser von Ihren Desktop VMs aus über die URL `http://lab99.vlab.hs-albsig.de/lab` abfragen. Diese HTML-Seiten werden allerdings nur zyklisch aktualisiert. Eine Aktualisierung kann bis zu 10 Minuten dauern.

man pages

Zu fast allen Befehlen und Konfigurationsdateien existieren sogenannte *manpages* (Manuals), die mit `man <Befehl>` gelesen werden können (z. B. `$ man chown`). Mit der Tastenkombination `Shift+7` (`/`) können Sie suchen. Beendet wird mit `q`.

root

Die meisten Änderungen und Einstellungen Ihres Systems müssen als Benutzer `root` (User-ID 0) vorgenommen werden.

Änderung von Konfigurationsdateien

In einem Terminal mit `su -` Ihre ID zu der des Benutzers `root` wechseln, und dann die entsprechende Datei mit `pico` oder `vi` (ist etwas schwieriger zu bedienen, wird aber empfohlen, da u.a. auf jedem Linux/BSD/Unix/MacOSX-System verfügbar) editieren.

root mit graphischer Oberfläche

Wenn Sie über einen X11 Zugang verfügen können Sie u.a. mit `sux -` oder `gksu` und mit z. B. `gedit` die entsprechende Datei öffnen und editieren. Mit `sux/gksu` erreichen Sie (wie mit `su -`) einen Wechsel zu User-ID 0 und zusätzlich die Möglichkeit den X-Server benutzen zu können. Inzwischen ist es bei den meisten aktuellen Distributionen allerdings nicht mehr notwendig, Sie können auch bei uns nach einem normalen `su-` direkt eine grafische Anwendung starten.

Dienste

Über die Scripte in `/etc/init.d` (System V Init System, SysVInit) können Sie die meisten Dienste starten oder stoppen (z.B `service apache2 stop`), und auch das automatische Starten des Dienstes beim Hochfahren des Systems unterbinden (z. B. `update-rc.d -f apache2 remove`). Achtung, bei `update-rc.d` gibt es teilweise keine Fehlermeldung wenn der Dienst nicht gefunden wurde.

Die meisten Linux Distributionen setzen inzwischen `systemd` ein, der u. a. `SysVInit` vollständig ersetzt.

Logdateien

Es ist immer zu empfehlen eine Blick in die Logdateien des Systems bzw. der Dienste zu werfen. Die meisten Logdateien befinden sich unter `/var/log` und können z. B. mit `tail -f /var/log/syslog` in einem Terminal mehr oder weniger in Echtzeit mitgelesen werden (beenden mit Ctrl-C).

weitere Hilfe

Im Anhang I finden Sie eine Liste von Befehlen (ohne Parameter), die allerdings mit Sicherheit noch nicht vollständig ist.

1.5 Kontrollfragen

Frage

Frage 1.1 Beim ersten Login auf ihren neuen Maschinen über SSH wurden Sie mit einer (ja/nein) Frage begrüßt. Welchen Sinn hatte diese Frage und was geschieht, wenn Sie diese mit Ja beantworten.

Frage

Frage 1.2 Welche Funktion hat die Datei `sources.list` und in welchem Zusammenhang steht sie zur Sicherheit eines Systems?

Frage

Frage 1.3 Was steht in der Datei `/.ssh/authorized_keys` und was bedeutet es, wenn man einen Eintrag in ihr erstellt?

Frage

Frage 1.4 Warum war es nötig, die Benutzerrechte des Home Verzeichnisses von `lab99` zu ändern und welche Kommandos stehen dazu zur Verfügung?

Frage

Frage 1.5 Welche Rolle spielt die Zeitsynchronisation bei der Sicherheit eines Netzwerkes? Nennen Sie Dinge, die schief gehen können, wenn keine gemeinsame Systemzeit zwischen den Systemen vorliegt.

Kapitel 2

Vorbereitung der Auswertung

2.1 Benutzer `lab99`

Aufgabe

Aufgabe 2.1 Legen Sie einen neuen Benutzer auf Ihrem Router mit folgenden Daten an:

- Name `lab99`
- UserID `8000`
- Primary Group `lab99` mit GroupID `8000`
- ! Supplementary Groups (also zusätzliche Gruppenmitgliedschaft): nur die Gruppe, die benötigt wird um `/var/log/syslog` lesen zu können! (Zur Erinnerung: bei `useradd` ist das der Parameter der mit `-G` eingeleitet wird)
- Login Shell `/bin/bash`
- Home Directory `/home/lab99` mit den Eigenschaften
  - Owner `lab99`
  - Group `lab99`
  - Permissions `rwx---`
- kein gültiges Passwort, in `/etc/shadow` sollte statt dem verschlüsseltem Passwort ein Ausrufezeichen stehen (derzeit bei `useradd` Standard).

Für die Überprüfung Ihrer Lösungen der folgenden Aufgaben benötigen wir eine Login-Möglichkeit auf Ihren Systemen. Hierzu dient der in **??** angelegte Benutzer `lab99`, der auf Ihren Desktop VMs schon fertig angelegt ist. Für die Authentifizierung nutzen wir hierbei das *Public Key* Verfahren der SSH. Das Prinzip dieses Verfahrens ist vereinfacht ausgedrückt wie folgt:

1. Generierung eines Schlüsselpaares bestehend aus *Public-* und *Private-Key*
2. Ablages des *Public-Key* auf dem Rechner, an dem man sich anmelden möchte
3. Authentifizierung mit dem *Private-Key* der auf dem Rechner von dem aus die Verbindung aufgebaut wird verfügbar sein muss

Sie müssen im Rahmen dieser Aufgabe zwar keine Keys erstellen, trotzdem der Hinweis, dass dies mit `ssh-keygen` bewerkstelligt werden kann.

Erstellen Sie das Verzeichnis `/home/lab99/.ssh` (bitte führenden Punkt bei `.ssh` beachten) mit folgenden Eigenschaften

- Owner/Group `lab99`
- Permissions `rwx---`

Von Ihren VMs aus finden Sie unter der URL
`http://lab99.vlab.hs-albsig.de/lab/public/lab99.pub`
den Public-Key für den Benutzer `lab99`.

Kopieren Sie `lab99.pub` nach `/home/lab99/.ssh` mit dem Dateinamen authorized_keys auf Ihrem Router. Ändern Sie die Rechte dieser Datei auf `rw----`, mit Owner/Group `lab99`. Beachten Sie, dass der 'Public-Key' nur eine Zeile hat, falls Sie mit 'copy+paste' arbeiten sollten.

Tipp : Recht einfach ist obiges z. B. mit den Befehlen `su - lab99` und `wget` umzusetzen.

2.2 Installation und Konfiguration eine NTP-Dienstes

Um den NTP-Server auf `labfw` (vgl. 1.2) für die Zeitsynchronisation zu nutzen, installieren Sie auf allen VMs den NTP-Dienst `openntpd` mit

```
# apt-get install openntpd
```

und passen die zugehörigen Konfigurationsdateien

`/etc/default/openntpd`
`/etc/openntpd/ntpd.conf`

wie folgt an:

- Beim Start des Dienstes sollen auch Zeitunterschiede von mehr als 180 Sekunden akzeptiert werden. (vgl. Bemerkung in der Konfigurationsdatei `/etc/default/openntpd`).
- Der Zeitabgleich soll nur mit der Maschine `labfw.vlab.hs-albsig.de` erfolgen. Setzen Sie eine Kommentarzeichen vor die vermutlich schon eingetragenen Server.

Mit dem Befehl `# service openntpd restart` sollte dann die erste Synchronisation ausgeführt werden. Die eventuelle Meldung über eine leere „Drift“ Datei kann hierbei ignoriert werden. Der Dienst läuft im übrigen unter dem Namen `ntpd` nicht `openntpd`. Die Kontrolle der aktuellen Systemzeit kann z. B. mit `date` durchgeführt werden.

2.3 Public-Key-Infrastruktur

2.3.1 Erstellen und Anwenden einer Public-Key-Infrastruktur

Am Beispiel OpenSSL soll eine Public-Key-Infrastruktur aufgebaut werden, die unter anderem das Problem der sicheren Authentifizierung löst. Dabei erstellt jedes Team eine sogenannte Zertifizierungsstelle (CA Certification Authority) erstellt, mit der dann verschiedene sogenannte Zertifizierungsanfragen (CSR Certificate Signing Request) digital unterschrieben werden und so ein

gültiges Zertifikat (CER) entsteht. Diese Zertifikat besteht aus einem öffentlichen Teil (mit Public-Key) und einem geheimen Teil (mit Private-Key). Dies verhält sich mit der eigentlichen CA im übrigen genauso, diese unterschreibt mit Ihrem Private-Key. Diese Private-Keys können/sollten noch mit einer Passphrase geschützt werden, dies setzen wir im Rahmen dieses Praktikums allerdings nicht ein. In diesem Praktikum wählen wir das Dateinamensende `.cer` für den Public-Key und `.key` für den Private-Key. Die mit OpenSSL erstellten Zertifikate nennt man SSL-Zertifikate und entsprechen dem X.509 Standard (ITU-T).

Ziel dieser Aufgabe ist, zwei Zertifikate für zwei HTTP-Server zu erstellen und die HTTP-Server dann SSL-fähig zu machen. Für einen Test sowie zur besseren Veranschaulichung sollen dann noch zwei HTTP-Clients das fehlende Zertifikat der CA importieren, damit eine SSL Verbindung ohne Warnungen wie `Diesem Zertifikat wird nicht vertraut...` zu den zwei HTTP-Servern aufgebaut werden kann.

Die CA erstellen Sie bitte nur auf Ihrer ersten Desktop VM (`lab01, lab04, lab07,...`). Erstellen Sie dann dort auch beide Zertifikate für die zwei HTTP-Server. Die HTTP-Server sind nur auf Ihren Desktop VMs zu installieren. Wir benutzen als HTTP-Server die Software `apache2`. Als HTTP-Client benutzen Sie am besten die dort schon installierte Software *Firefox*.

Die Art der Lösung dieser Aufgabe ist Ihnen grundsätzlich freigestellt, wir benötigen für die Auswertung lediglich den Public-Key ihrer CA, und lauffähige HTTP-Server auf Ihren Desktop VMs die sowohl auf dem TCP-Port 80 (http) als auch auf dem TCP-Port 443 (https) Verbindungen entgegen nehmen. Aufgrund der Komplexität empfehlen wir allerdings die folgende Vorgehensweise.

Achtung

`XY` mit führender Null

CA Erstellen

Bauen Sie zunächst die erforderliche Verzeichnisstruktur auf der ersten Desktop VM auf (bitte als normaler Benutzer, allerdings nicht als `lab99`):

```
$ cd ~
$ mkdir openssl
$ cd openssl
$ mkdir myCA
$ mkdir myCA/private
$ chmod 700 myCA/private
$ mkdir myCA/newcerts
$ touch myCA/index.txt
$ echo 10 > myCA/serial
```

Original Konfigurationsdatei kopieren und bearbeiten:

```
$ cp /etc/ssl/openssl.cnf .
$ pico openssl.cnf
```

Stellen Sie sicher, dass die folgenden Einträge in der Konfigurationsdatei enthalten sind. Ändern Sie vorhandene Einträge ab und fügen Sie gegebenenfalls neue Einträge hinzu.

Unter `[ CA_defaul ]`:
⇒ `dir = ./myCA`
⇒ `certificate = $dir/myroot.cer`
⇒ `private_key = $dir/private/myroot.key`

Unter `[ req_distinguished_name ]`:
⇒ `countryName_default = DE`
⇒ `stateOrProvinceName_default = BW`
⇒ `localityName_default = Albstadt-Sigmaringen`
⇒ `0.organizationName_default = HS Albstadt-Sigmaringen`
⇒ `organizationalUnitName_default = INF`

Generieren Sie jetzt die CA (Zertifikatsstelle):

```
$ openssl req -config openssl.cnf -newkey rsa:2048 -days 3650 -x509 -nodes ↩
  -out myCA/myroot.cer -keyout myCA/private/myroot.key
```

⇒ als Common Name: `NuSS Praktikum Team Z` (`Z` ist XY des Routers geteilt durch 3, also bei den VMs `lab25/26/27` ⇒ `Z = 9`)
⇒ Mailadresse: `root@localhost`

Erstes Zertifikat Erstellen

Generieren Sie das Certification Request (Zertifikatsanfrage):

```
$ openssl req -config openssl.cnf -newkey rsa:1024 -nodes -out labXY.csr ↩
  -keyout labXY.key
```

⇒ als CN `labXY.vlab.hs-albsig.de` (hier unbedingt den richtigen FQDN angeben)
⇒ Mailadresse `root@localhost`
⇒ keine weiteren Angaben bei *challenge* und *optional company name*

Stellen Sie jetzt das Zertifikat aus (Generate Certificate):

```
$ openssl ca -config openssl.cnf -batch -notext -in labXY.csr -out labXY.cer
```

Sie können jetzt noch das Zertifikat überprüfen (Test Certificate):

```
$ openssl x509 -noout -text -in labXY.cer
```

Zweites Zertifikat Erstellen

Erstellen Sie jetzt ein zweites Zertifikat für die zweite Desktop VM. Das Vorgehen ist analog zur Erstellung des ersten Zertifikats (Request, Generate, Test), Sie müssen lediglich `XY` entsprechend ändern.

2.3.2 HTTPS Demonstration

Apache auf beiden Desktop VMs installieren `# apt-get install apache2`

Im Rahmen dieses Praktikums kümmern wir uns nicht um eine sichere Konfiguration des HTTP-Servers. Er dient lediglich zur Demonstration des Einsatzes von SSL-Zertifikaten.

Verzeichnisstruktur auf beiden Desktop VMs anlegen und Zertifikate kopieren. In den unteren Befehlen gehen wir davon aus dass der Benutzer `myname` die SSL-Zertifikate nach obigen Anleitung erstellt hat.

Das Zertifikat für die zweite Desktop VM kopieren Sie am besten mit dem Befehlt `scp` (einfacher Dateitransfer über `ssh`) auf beiden Desktop VMs.

1. auf beiden Desktop VMs
   `# mkdir /etc/apache2/ssl`

2. auf der ersten Desktop VM (XY der 1. Desktop VM)
   `# cp /home/myname/openssl/labXY.{cer,key} /etc/apache2/ssl`

3. auf der ersten Desktop VM (XY der 2. Desktop VM)
   `# scp /home/myname/openssl/labXY.{cer,key} root@labXY:/etc/apache2/ssl`

4. auf beiden Desktop VMs
   `# chmod 600 /etc/apache2/ssl/labXY.key`

Bearbeiten Sie jetzt die Apache Konfigurationsdatei auf beiden Desktop VMs:
`# pico /etc/apache2/sites-available/default-ssl.conf`

Ändern Sie folgende Einträge in der Konfigurationsdatei (`XY` entsprechend der jeweiligen Desktop VM)
⇒ `SSLCertificateFile /etc/apache2/ssl/labXY.cer`
⇒ `SSLCertificateKeyFile /etc/apache2/ssl/labXY.key`

Laden Sie jetzt das Apache SSL-Modul und schalten Sie die gerade bearbeitete `site` ein.
`# a2enmod ssl`

`# a2ensite default-ssl`

Lassen Sie die Konfigurationsdateien neu einlesen. `# service apache2 reload`

Versuchen Sie nun mit Ihren HTTP-Clients der Desktop VMs, zuerst eine normale `http` und dann eine `https` Verbindung zu Ihren HTTP-Servern aufzubauen. Benutzen Sie hierbei als URL `http://labXY.vlab.hs-albsig.de` bzw. `https://labXY.vlab.hs-albsig.de`.

Bei der SSL-Verbindung sollte Sie Ihr HTTP-Client warnen, dass nicht überprüft werden kann ob die Verbindung sicher ist. Dies liegt an dem fehlenden Zertifikat der CA. Brechen Sie die Verbindung nun ab. Importieren Sie jetzt Ihr Zertifikat `myroot.cer` in den HTTP-Client. Bei z. B. Firefox 78 über

⇒ *Bearbeiten → Einstellungen → Datenschutz & Sicherheit → Zertifikate anzeigen → Zertifizierungsstellen → Importieren*

Dabei bitte `Dieser CA vertrauen um Websites zu identifizieren` auswählen. Hierbei können Sie sich auch gleich Ihr CA Zertifikat einmal anzeigen lassen.

Nach dem Import wiederholen Sie die SSL-Verbindung, jetzt sollte Ihr HTTP-Client keine Warnung mehr anzeigen.

Für die spätere Auswertung kopieren Sie bitte das Zertifikat Ihrer CA `myroot.cer` nach `/home/lab99/rootXY.cer` auf der ersten Desktop VM. !! XY !! ist die Nummer Ihrer ersten Desktop VM mit eventuell führender Null, lesbar für den Benutzer `lab99`.

Hinweis zur Auswertung

Ausgewertet wird derzeit wie folgt:

- Apache (2.3) : Ob auf Ihren Desktop VMs ein Dienst auf Port 80 TCP läuft.
- SSL (2.4) : Ob auf Ihren Desktop VMs ein Dienst auf Port 443 TCP läuft, und eine korrekte SSL-Verbindung aufgebaut werden kann. Hierzu wird jedes mal das von Ihnen bereitgestellte Zertifikat `/home/lab99/rootXY.cer` kopiert.

2.4 Kontrollfragen

Frage

Frage 2.1 Welche Aufgabe erfüllt eine CA?

Frage

Frage 2.2 Was ist der Unterschied zwischen den Dateien `root.cer` und `root.key` und wofür werden sie jeweils verwendet? Wie sind sie aus Sicherheitsgesichtspunkten zu behandeln?

Frage

Frage 2.3 Was verbirgt sich im Praktikum hinter Dateien mit dem Namen `*.csr`. Wofür werden sie verwendet?

Frage

Frage 2.4 Warum beklagt sich der Firefox Browser beim Aufruf der neuen Website über https? Welche Möglichkeiten gibt es, diese Meldung zu beseitigen und welche Sicherheitsauswirkungen haben diese jeweils?

Kapitel 3

Firewall einrichten

Ziel ist die Einrichtung einer Firewall auf Ihrem Router. Diese Firewall dient allerdings lediglich zur Filterung von IP-Verbindungen. Die Filter basieren auf der Software `Netfilter`, die bei Linux seit längerem Standard ist. Bedient und konfiguriert wird diese Software mit dem Paket `iptables`. Darauf aufsetzend findet man dann weitere Software wie z. B. `ufw` bei Ubuntu Distributionen.

Da wir im Rahmen dieses Praktikums `ipv6` nicht berücksichtigen können sollte dieser IP-Stack für dieses Praktikum auf dem Router generell abgeschaltet werden. Gehen Sie hierzu bitte wie folgt vor:

1. In der Datei `/etc/default/grub` die Zeile
   `GRUB_CMDLINE_LINUX_DEFAULT="quiet ipv6.disable=0"` zu
   `GRUB_CMDLINE_LINUX_DEFAULT="quiet ipv6.disable=1"` ergänzen.

2. Danach mit
   `# update-grub`
   `# reboot`
   aktivieren. Der Neustart ist hierbei erforderlich.

Auf der VM `lab99` liegt ein vorbereitetes Debian-Paket (Debian-Package) zur Installation auf Ihrem Router bereit. Dieses Paket (`tinyfw.deb`) ist stark vereinfacht und implementiert nicht alle Funktionalitäten und Anforderungen von Debian-Paketen. Es wurde erstellt um die Komplexität von `Netfilter/iptables` zu reduzieren, und um auch für mehr als 2 Netzwerkkarten die Filterregeln möglichst einfach gestalten zu können.

Hier die Vorgehensweise zur Installation von `tinyfw`:

1. Debian-Paket und Prüfsummen Datei abholen
   `# wget http://lab99.vlab.hs-albsig.de/lab/public/tinyfw.deb`
   `# wget http://lab99.vlab.hs-albsig.de/lab/public/tinyfw.md5`

2. Mit folgendem Befehl können Sie abprüfen ob die Datei korrekt übertragen wurde, oder eventuell geändert wurde. (Sicherer wäre natürlich die Prüfsumme `tinyfw.md5` nicht auf dem gleichen Server liegen zu haben ;-)
   `$ md5sum -c tinyfw.md5`

3. Paket installieren mit
   `# dpkg -i tinyfw.deb`

Die derzeitige Version 0.08 besteht aus 4 Dateien.

- `/usr/sbin/tinyfw.sh` (Shell Script)
- `/etc/init.d/tinyfw` (System V Startupscript)
- `/etc/default/tinyfw` (Konfigurationsdatei zur generellen 'Sperren/Entsperren' der Filteraktivierung)
- `/etc/tinyfw/tinyfw.conf` (Konfigurationsdatei für die Regeln)

Sie bearbeiten allerdings lediglich die Konfigurationsdateien, können sich aber auch einmal die zwei anderen Dateien ansehen. Es sind schon verschiedene Regeln in `/etc/tinyfw/tinyfw.conf` definiert, die folgendes erlauben :

- SSH Zugang auf den Router aus dem Netz `141.87.0.0/16`
- DNS (Namensauflösung) und NTP (Zeitabgleich) für den Router zu `labfw.vlab.hs-albsig.de`
- DHCP für die Desktop VMs

ICMP (z. B. `ping`) für 'alle' zu 'allen' ist in `/usr/sbin/tinyfw.sh` schon fest codiert erlaubt, alles andere ist nicht zugelassen.

Wenn Sie `tinyfw` aktivieren sollten Sie deshalb lediglich Verbindungen von `inflx1/2/3/4` zu Ihrem Router haben, nicht zu den Desktop VMs. Jede Regel die Sie hinzufügen ist eine weitere zugelassene Verbindung ('White-List').

Nach der Aktivierung werden Sie feststellen, das vieles nicht mehr wie gewohnt funktioniert, unter anderem werden auch verschiedene Auswertungen die Sie bisher auf 'PASSED' hatten wieder 'FAILED' sein, sobald Sie sich überhaupt wieder Zugang zu Ihren Desktop VMs ermöglicht haben, um die Auswertung per grafischem HTTP-Client (Browser) zu erreichen.

Um `tinyfw` aktivieren zu können muss zuerst der Eintrag

`TINYFW_ENABLED="Yes"`

in der Datei

`/etc/default/tinyfw`

gesetzt werden. Dadurch wird `tinyfw` auch automatisch beim Bootvorgang aktiviert (Deaktivieren für den Bootvorgang mit != "Yes").

Manuell aktivieren können Sie `tinyfw` mit dem Befehl

```
# service tinyfw start
```

Diesen Befehl müssen Sie auch nach jeder Änderung der Regeln in `tinyfw.conf` ausführen. Dadurch werden die *alten* Regeln gelöscht und die *neuen* aktiviert.

Alle Regeln deaktivieren können Sie mit

```
# service tinyfw stop
```

Allerdings sind die Regeln nach einem Neustart wieder aktiv, solange TINYFW_ENABLED in `/etc/default/tinyfw` auf `Yes` gesetzt ist.

Mit

```
# service tinyfw status
```

können Sie sich die aktiven Regeln im 'Iptables-Output-Format' anzeigen lassen.

Konfiguriert werden die Regeln, wie schon erwähnt, in der Datei :

`/etc/tinyfw/tinyfw.conf`

Dort finden Sie 3 so genannte CHAINS (Ketten, diese sind hier auch nur Variablennamen) die für folgendes zuständig sind :

- `INPUT_CHAIN`: alles was zu Ihrem Router von Außen erlaubt ist
- `OUTPUT_CHAIN`: alles was Ihrem Router nach Außen erlaubt ist
- `FORWARD_CHAIN`: alles was Ihr Router weiterleiten soll

INPUT/OUTPUT_CHAIN betrifft also nur Ihren Router direkt, FORWARD_CHAIN dann z. B. alle Verbindungen zu und von Ihren Desktop VMs die über Ihren Router laufen.

In der Konfigurationsdatei können Sie folgende vordefinierte Variablen verwenden :

- Portnummer `1` bis `1023`, können nur von `root` verwendet werden : `LOWPORTS`
- Portnummer `1024` bis `65535`, können von jedem verwendet werden : `HIGHPORTS`
- Portnummer `1` bis `65535`, also alle möglichen Ports : `ANYPORTS`
- Jede beliebige IP Adresse: `ANY`

Achtung

Vergessen Sie hierbei nicht, das $ Zeichen voran zu stellend.

Jede CHAIN beginnt mit doppeltem Hochkommata und endet mit diesem. Dazwischen finden Sie pro Zeile die Angaben für eine Regel wie folgt:
`Protokoll, Absender-IP-Adresse, Absender-Portnummer, Ziel-IP-Adresse, Ziel-Portnummer`

Als Protokoll kommen in diesem Praktikum nur `tcp` oder `udp` in Frage. Die Absender Portnummer wird meistens per Bereich angegeben (z. B. mit `$HIGHPORTS`), da diese nur bei einzelnen Anwendungen im Voraus bekannt sind. Meistens ist `$HIGHPORTS` ausreichend. Da Ihr Router mehrere IP-Adressen hat, ist auch in den INPUT/OUTPUT Einträgen eine Ziel-IP-Adresse bzw. Absender-IP-Adresse einzutragen.

Die einzelnen Regeln können Leerzeichen und Kommentare enthalten. Das Kommentarzeichen `#` kann am Anfang einer Regel/Zeile oder z. B. am Ende einer Regel stehen.

Ergänzen Sie die benötigten Regeln in der Konfigurationsdatei um folgende Mindestanforderungen zu erfüllen: (TIP : am besten in unterer Reihenfolge, Schritt für Schritt, und nicht alles auf einmal)

1. Nameserverabfragen für die Desktop VMs zu `192.168.100.230`
2. Zeitabgleich für die Desktop VMs zu `192.168.100.230`
3. SSH Zugang zu den Desktop VMs von `inflx1/2/3/4` aus
4. HTTP Zugang für alle VMs zu `lab99.vlab.hs-albsig.de` (Updates, Software Installationen, Dokumente, usw.)
5. SSH Zugang Ihrer VMs untereinander
6. entsprechende Regeln, damit die Auswertungen von Aufgabe 2.1 bis Aufgabe 2.4 wieder auf *PASSED* wechseln. . .

Lassen Sie für diese Aufgabe in einem zweiten Terminal als Benutzer `root` am besten den Befehl `tail -f /var/log/syslog` mitlaufen. Dort werden Sie jetzt verschiedene Logeinträge von `tinyfw` bemerken (`INPUT : DROP`, `OUTPUT : DROP`, `FORWARD : DROP`). Darunter befinden sich u.a Abkürzungen wie `SRC/DST/PROTO/SPT/DPT` die Ihnen Hinweise geben könnten, warum die ein oder andere Auswertung noch auf `'FAILED'` steht. Allerdings sind vermutlich viele DROPs zu Recht, dass ist schließlich die Aufgabe einer Firewall. . .

Die Auswertungen der 3. Aufgabe sind derzeit wie folgt :

- Filter (3.0) Sind Filter für Ihren Router und Ihren Desktop VMs aktiv.
- DNSClient (3.1) Können Router und Desktop VMs erfolgreich den Nameserver `192.168.100.230` abfragen.
- NTPClient (3.2) Können Router und Desktop VMs erfolgreich den Zeitabgleich mit `192.168.100.230` vornehmen.
- NumRules (3.3) Ob Ihre Firewall wirklich nach den Vorgaben eingestellt wurde, ist relativ schwierig auszuwerten. So kann es gut sein dass Sie ihre Firewall 'offener' als notwendig konfiguriert haben und trotzdem alles auf `PASSED` ist.

Sie benötigen ca. 35 Regeln, diese Schätzung berücksichtigt die Regeln für spätere Aufgaben.

Damit Sie etwas genauer wissen ob Sie die Aufgabe in etwa richtig bearbeitet haben, wurde diese Auswertung eingeführt. Diese zählt schlicht die Anzahl udp/tcp-Regeln in `tinyfw.conf`, und sollte mindestens 27 und höchstens 43 betragen. Besser wäre sicher eine Auswertung in der Form `iptables -L -n | grep ACCEPT | wc -l`, dazu bräuchte der Benutzer `lab99` aber Root-Rechte.

Wenn Sie mehr als 43 oder weniger als 27 Regeln benötigen um die Anforderungen zu erfüllen, melden sie sich bitte.

Alle vorhergehenden Auswertungen sollten ebenfalls wieder auf `PASSED` sein, da sonst z. B. 3.1/3.2/3.3 nicht erfolgreich sein können solange die Auswertung 2.1/2.2 nicht möglich ist (SSH-Login des Benutzer `lab99`).

3.1 Kontrollfragen

Frage

Frage 3.1 Worin unterscheiden sich die `INPUT`, `FORWARD` und `OUTPUT` Chain?

Frage

Frage 3.2 Wie können Sie eine Regel angeben, die nur für einen einzigen Host gilt?

Frage

Frage 3.3 Wie können Sie eine Regel angeben, die für ein ganzes Netzwerk gilt?

Frage

Frage 3.4 Welche Informationen werden benötigt, um eine Regel eindeutig zu bestimmen (5-Tupel)?

Frage

Frage 3.5 Wie stellen Sie fest, ob ihre Regeln unerwünscht viel Verkehr blocken und welcher Verkehr dies ist? Wie haben Sie geprüft, ob ihre Clients ihren Nameserver erreichen können?

Frage

Frage 3.6 Warum funktioniert der `ping` Befehl, ohne dass Sie jemals eine Regel dafür angelegt hatten?

Frage

Frage 3.7 Weshalb ist es ausreichend, eine Regel für den Verbindungsaufbau von Ihren VMs zum Webserver auf `lab99` zu erstellen? Weshalb ist keine Regel für die Rückrichtung nötig?

Kapitel 4

VPN einrichten

Ziel dieser Aufgabe ist eine sichere VPN (Virtual Private Network) Verbindung zu erstellen mit Verschlüsselung und Authentifizierung durch X509 Zertifikate sowie Benutzerkennung zu erstellen. Hierzu wird die Software *OpenVPN* verwendet. Die OpenVPN Implementierung ist SSL basierend (auf UDP oder TCP), und kann auch mit X509 Zertifikaten umgehen.Die Verbindung (der VPN-Tunnel) wird im Praktikum über UDP realisiert.

Derzeitiger Default-Port ist 1194. (Berücksichtigen Sie dies bei der Konfiguration Ihrer Filterregeln auf dem Router). Die Software wird auf den beiden Desktop VMs installiert und so konfiguriert, dass ein VPN-Tunnel von einer Desktop VM zu anderen über den Router hinweg entsteht, der es z. B. ermöglicht alle TCP/UDP Dienste zwischen den 2 Desktop VMs wieder zu nutzen.

Die Vorgehensweise ist hierbei wie folgt gedacht:

1. VPN mit statischem Schlüssel
2. VPN mit X509 Zertifikaten
3. VPN mit X509 Zertifikaten und Benutzerkennung/Anmeldung

Installieren Sie auf den beiden Desktop VMs (`lab01,lab02,lab04,lab05,...`) die VPN Software `openvpn`.

```
# apt-get install openvpn
```

Falls der Dienst schon gestartet ist mit

```
# service openvpn stop
```

wieder beenden. Damit nicht bei jedem Reboot die (noch zu konfigurierenden) VPN-Tunnel erstellt werden bitte noch

```
# update-rc.d -f openvpn remove
```

ausführen. Für dieses Praktikum werden wir nicht den eigentlich vorgesehenen `System V Init-Script` Mechanismus (`service ...`) verwenden. *Es ist wichtig das der Dienst nicht automatisch gestartet wird , da Ihre VM sonst beim Hochfahren hängen bleiben könnte.*

Man unterscheidet bei OpenVPN immer Server und Client. Den Server betreiben Sie bitte immer auf Ihrer ersten Desktop VM (`lab01,lab04,...`), den Client immer auf der zweiten Desktop VM (`lab02,lab05,...`). Im folgenden werden beide mit VPN-Server und VPN-Client benannt.

Als Konfigurationsverzeichnis nutzen wir: `/etc/openvpn`

4.1 VPN mit statischem Schlüssel

In einfachen Fällen kann auch mit einem statischen Schlüssel der generiert und verteilt wird, gearbeitet werden. Für einen ersten Test können Sie den Ansatz mit dem statischen Schlüssel verwenden.

Dieser kann mit

```
# openvpn --genkey --secret /etc/openvpn/static.key
```

beispielsweise auf dem VPN-Server erstellt werde. Kopieren Sie die Datei zu dem VPN-Client in das gleiche Verzeichnis

```
# scp /etc/openvpn/static.key root@labXY:/etc/openvpn
```

Nun werden noch für beide je eine minimale Konfigurationsdatei benötigt. Sie können diese beliebig benennen, hier im folgenden mit dem Dateinamen `static.conf`.

Inhalt der Server Konfigurationsdatei `/etc/openvpn/static.conf`:

```
####################################################################
verb 3
dev tun
ifconfig 10.0.1.1 10.0.1.2
secret /etc/openvpn/static.key
####################################################################
```

Inhalt der Client Konfigurationsdatei `/etc/openvpn/static.conf`:

```
####################################################################
verb 3
dev tun
ifconfig 10.0.1.2 10.0.1.1
secret /etc/openvpn/static.key
remote labXY.vlab.hs-albsig.de
####################################################################
```

`XY` entspricht hier Ihrer ersten Desktop VM. Der Eintrag `verb` (verbose) ist nicht unbedingt notwendig, erhöht aber die Ausgabe von Informationen seitens `openvpn`.

Starten Sie `openvpn` auf dem VPN-Server in einem Terminal wie folgt:

```
# openvpn /etc/openvpn/static.conf
```

Danach sollte auf dem Server ein neues Interface mit Namen `tun0` zur Verfügung stehen. (`# ifconfig`) Dann wird auf der anderen Desktop VM mit

```
# openvpn /etc/openvpn/static.conf
```

der Client gestartet. Verschiedene Meldungen erscheinen, und nach ein paar Sekunden sollte der VPN Tunnel zur Verfügung stehen. Öffnen Sie weitere Terminals auf Ihren Desktop VMs um den Tunnel zu testen. Sie müssten den VPN-Server unter `10.0.1.1` und den VPN-Client unter `10.0.1.2`, z. B. mit ping oder ssh jeweils von der *Gegenstelle* aus erreichen können.

Über diesen Tunnel lassen sich jetzt z. B. beliebige UDP/TCP basierende Protokolle benutzen, Sie können dies auch einmal mit `HTTP/HTTPS` wie in 2.3/2.4 ausprobieren. Als URL dann je nach Richtung `http://10.0.1.1` oder `http://10.0.1.2`. Was passiert bei einer HTTPS - Verbindung?

Beenden können Sie sowohl VPN-Server als auch VPN-Client mit `<CTRL>+<C>`.

4.2 VPN mit X.509 Zertifikaten

Nun soll die Lösung mit dem statischen Key durch eine mit X.509 Zertifikaten abgelöst werden. Wir benutzen hierzu die schon erstellten Zertifikate Ihrer Desktop VMs (aus Aufgabe 2.3/2.4 `openssl`).

Auf dem VPN-Server muss noch eine neue Datei erstellt werden, die für den Diffie-Hellmann Schlüsselaustausch benötigt wird (auch zur zyklischen Schlüsselregenerierung).

```
# openssl dhparam -outform PEM -out /etc/openvpn/dh1024.pem 1024
```

Danach kopieren Sie Ihre entsprechenden X.509 Zertifikate/Keys und Ihre Root-CA in die Konfigurationsverzeichnisse. Damit kann dann der Server den Client und der Client den Server verifizieren.

VPN-Server auf z. B. lab01 : benötigt `lab01.cer, lab01.key, myroot.cer` VPN-Client auf z. B. lab02 : benötigt `lab02.cer, lab02.key, myroot.cer`

Beachten Sie hierbei insbesondere die Dateiattribute der `*.key` Dateien. Diese sollten nur für den Benutzer `root` lesbar sein. Wie in Aufgabe 4.1 werden noch für beide je eine minimale Konfigurationsdatei benötigt. Hier sind diese mit `x509.conf` benannt:

Inhalt der Server Konfigurationsdatei `/etc/openvpn/x509.conf`:

```
#####################################################################
verb 3
dev tun
server 10.0.1.0 255.255.255.0
ca /etc/openvpn/myroot.cer
cert /etc/openvpn/labXY.cer
key /etc/openvpn/labXY.key
dh /etc/openvpn/dh1024.pem
#####################################################################
```

Inhalt der Client Konfigurationsdatei `/etc/openvpn/x509.conf`:

```
#####################################################################
verb 3
dev tun
remote labXY.vlab.hs-albsig.de
client
ca /etc/openvpn/myroot.cer
cert /etc/openvpn/labXY.cer
key /etc/openvpn/labXY.key
#####################################################################
```

XY entsprechen hier natürlich zum Teil unterschiedlichen Werten.

Überprüfen Sie ob event. noch alte `openvpn` Prozesse aktiv sind und beenden Sie diese gegebenenfalls (z. B. mit `# killall openvpn`). Dann zuerst auf dem VPN-Server und danach auf dem VPN-Client `openvpn` starten, und hierbei als Parameter die X509 Konfigurationsdateien angeben.

Testen können Sie den Tunnel mit den IP-Adressen `10.0.1.1` (Server) und `10.0.1.6` (Client). (`10.0.1.2` und `10.0.1.5` werden intern benötigt und sind nicht erreichbar)

4.3 VPN mit X.509 Zertifikaten und Benutzerkennung

Nun wird Aufgabe 4.2 noch um die Benutzerkennung erweitert. Am besten kopieren Sie die vorhandenen Konfigurationsdateien `/etc/openvpn/x509.conf` z. B. nach `/etc/openvpn/x509login.conf` und ergänzen Sie diese um folgende Einträge:

Auf dem VPN-Server:

```
######################################################################
plugin /usr/lib/openvpn/openvpn-plugin-auth-pam.so login
######################################################################
```

Auf dem VPN-Client:

```
######################################################################
auth-user-pass
######################################################################
```

Überprüfen Sie, ob eventuell noch alte `openvpn` Prozesse aktiv sind und beenden Sie diese gegebenenfalls. Dann zuerst auf dem VPN-Server und danach auf dem VPN-Client `openvpn` starten, und hierbei als Parameter die gerade erstellten Konfigurationsdateien angeben.

Der VPN-Client fordert Sie dann zur Eingabe eines Benutzernamens und eines Passwortes auf. Hier wählen Sie als Benutzer nicht `lab99`, da dieser kein reguläres Passwort hat. Verwenden Sie z. B. den Benutzer den Sie in 1.6 auf der ersten Desktop VM (*VPN-Server*) angelegt haben.

Nach der Authentifizierung mit einem *legalen* Benutzer des VPN-Servers durch den Client kann der Tunnel wieder mit den IP-Adressen `10.0.1.1` und `10.0.1.6` getestet werden.

Um die Auswertung der Aufgabe durchführen zu können ergänzen Sie die Konfigurationsdatei des VPN-Servers jetzt noch bitte um diese eine Zeile:

```
######################################################################
syslog
######################################################################
```

und starten VPN-Server und VPN-Client neu. Die Meldungen des VPN-Servers sollten nun in der Log-Datei `/var/log/syslog` erscheinen.

Anmerkung: Der Dateinamen entspricht nur zufällig der Konfigurationsanweisung. Diese Anweisung bewirkt, dass `openvpn` seine Meldungen zu dem Dienst `Syslog` sendet, und `Syslog` verteilt diese dann, z. B. in unterschiedlichen Dateien, je nach Syslog-Konfiguration. Damit der Tunnel für die Auswertung auch noch aktiv ist, wenn Sie sich an den beiden Desktop VMs abgemeldet haben, gehen Sie am besten wie folgt vor:

Auf dem VPN-Server mit Konfigurationsdatei `x509login.conf`:

```
# killall openvpn
```

```
# nohup /usr/sbin/openvpn /etc/openvpn/x509login.conf &
```

Auf dem VPN-Client mit Konfigurationsdatei `x509login.conf`:

```
# killall openvpn
```

```
# nohup /usr/sbin/openvpn /etc/openvpn/x509login.conf
```

⇒ Aufgrund der Username/Password Abfrage ohne `&`
Nach der Eingabe von Username/Password dann mit `<CTRL>+<Z>` anhalten und mit `bg` in den Hintergrund schicken

`<CTRL>+<Z>`

```
# bg
```

Kontrollieren Sie den VPN-Tunnel wie gewohnt mit `ifconfig,ping,ssh,...`.

Melden sie sich jetzt bei beiden Desktop VMs ab. Dann melden Sie sich bitte wieder an und kontrollieren ob der aufgebaute VPN-Tunnel noch vorhanden ist.

Die Auswertungen der Aufgabe 4 sind derzeit wie folgt:

- VPN (4.3) `labXY` (*VPN-Server*) Ist die Gegenseite erreichbar und wurde ein SSL Verbindung mit Benutzerkennung initiiert. Sollte die Auswertung von `PASSED` wieder auf `FAILED` zurückfallen, auch wenn Ihr VPN-Tunnel steht, bauen Sie den Tunnel bitte erneut auf.

- VPN (4.3) `labXY` (*VPN-Client*) Ist die Gegenseite erreichbar.

4.4 Kontrollfragen

Frage

Frage 4.1 Welcher Modus wird durch `openVPN` in der Aufgabe verwendet und was bedeutet dies für Pakete, die als Broadcast im Subnet des Servers (`192.168.X.0/24`) gesendet werden?

Frage

Frage 4.2 Betrachten Sie die Ausgabe von `ifconfig` und schauen Sie sich das Device `tun0` an. Wo kommt dieses her und warum hat es eine solch seltsame Ausgabe?

Frage

Frage 4.3 In Ihrem Browser mussten Sie in einer der vorigen Aufgaben das Zertifikat ihrer CA (`myroot.cer`) importieren, damit die gesicherte Verbindung über X.509 Zertifikate möglich war. Wie wurde dies für `openVPN` gelöst?

Frage

Frage 4.4 Weshalb verwendet `openVPN` ein neues virtuelles Netzwerk? Könnte der Client nicht direkt die Netzwerkadressen des Servers (`192.168.X.Y`) auf dem tun Interface (anstatt des 10er Netzwerks) verwenden?

Frage

Frage 4.5 Was bewirken die Schlüsselwörter CA, CERT und KEY in der openVPN Konfiguration?

Frage

Frage 4.6 Was ist PAM und was bewirkt `/usr/lib/openvpn/openvpnauthpam.so`?

Frage

Frage 4.7 Beschreiben Sie den Zweck und die Funktionsweise der Software rkhunter.

Frage

Frage 4.8 Welche Funktion haben die Werkzeuge `strace`, `ltrace` und `ldd`?

Frage

Frage 4.9 Was tut das Werkzeug `ssldump`? Wie wird es eingesetzt?

Frage

Frage 4.10 Welche Informationen landen in den in `/var/log` gespeicherten Log Dateien.

- `syslog`
- `auth.log`
- `dmesg`

Frage

Frage 4.11 Was findet sich in den Dateien `.bash_history` und `.viminfo`? Weshalb ist der Inhalt auch aus Sicherheitssicht interessant?

Frage

Frage 4.12 Wie und warum würden Sie sudo und die Datei /etc/sudoers in einem System mit mehreren Benutzern verwenden? Wie stehen die Dateien im Bezug zur Systemsicherheit?

Kapitel 5

Systemüberwachung Teil 1

In den folgenden Aufgabe werden Sie sich etwas intensiver mit Ihren Systemen beschäftigen, und hierbei verschiedene Programme benutzen, die z.T. noch installiert werden müssen.

Des weiteren werden Sie uns einen direkten Root-Zugang nach bestimmten Vorgaben einrichten.

Installieren Sie folgende Software Pakete auf ihren VMs, falls diese nicht schon installiert sind:

`postfix`	auf allen VMs
`mailutils`	auf allen VMs
`rkhunter`	auf allen VMs
`nmap`	auf allen VMs
`dnsutils`	auf allen VMs
`ssldump`	auf allen VMs
`wireshark-gtk`	auf den beiden Desktop VMs

Kurzbeschreibungen der obigen Paketen:

`postfix` Ist ein eMail System
Wählen sie bei der Installation bitte "Nur Lokal" aus. Ignorieren Sie vorläufig die Warnung `WARNING: /etc/aliases exists, but does not have a root alias.`

`mailutils` Werkzeuge für die eMail Bearbeitung

`rkhunter` Ist ein Überwachungssystem

`nmap` Ist ein Portscanner.
Und hier gleich der wichtige Hinweis:
Wenden Sie diesen nicht auf fremde Systeme ohne Zustimmung der dort zuständigen entsprechenden Systembetreuer an. Auch der Weg zu diesen Systemen sollte hierbei berücksichtigt werden. Denken sie in diesem Zusammenhang auch an *Industrie 4.0*.
Hier im Praktikum ist es Ihnen allerdings erlaubt die anderen VMs zu *scannen*.

`dnsutils` Werkzeuge für DNS Abfragen

`ssldump` Werkzeug zum Mitschneiden/Auswerten von SSL Daten

`wireshark-gtk` Netzwerk Analyse Tool, die GTK+ Version
Es erscheint eine kurze Erklärung und die Frage ob noch anderen Benutzern die Paket-Aufzeichnung erlaubt werden soll. Beantworten sie diese Frage mit *Ja*. Dann sollten Sie den Benutzer den sie angelegt haben zusätzlich in die Gruppe `wireshark` mit aufnehmen. Am besten mit
`# usermod -a -G wireshark myname` .

Sollten Sie aktuell als `myname` auf Ihren Desktop VMs eingeloggt sein, müssen Sie sich abmelden und neu anmelden, sonst greift die Erweiterung der Gruppenzugehörigkeit nicht. Wenn Sie bei der Installation keine Frage gestellt bekamen oder diese falsche beantwortet haben, können Sie mit

```
# dpkg-reconfigure wireshark-common
```

die notwendige Konfiguration nachholen bzw. korrigieren.

Gleiches gilt, falls das Paket schon installiert sein sollte. Beschäftigen sie sich nun etwas intensiver mit folgenden Befehlen und Dateien

- `ps (axu)`
- `top`
- `netstat (-a -A inet -p -n)`
- `lsof (-i -n -P)`
- `nmap`
- `df`
- `tcpdump`
- `strace`
- `ltrace`
- `ldd`

Logdateien unterhalb von /var/log insbesondere

- `syslog`
- `auth.log`
- `dmesg`

Benutzerbezogene Daten in `$HOME`

- `.bash_history`
- `.viminfo`

Später kommen dann vermutlich noch weitere Befehle hinzu :

- `host`
- `dig`
- `nslookup`
- `ssldump`
- `openssl s_client`
- `openssl s_server`

5.1 Root Zugang einrichten

Für zukünftige Aufgaben benötigen wir Root Zugang zu Ihren Systemen. Diesen realisieren Sie in diesem Fall mit dem sudo-System. Machen Sie sich etwas damit vertraut.

`$ man sudo` `$ man sudoers`

Wir haben folgende Anforderung:

Der Benutzer lab99 soll ohne jegliche Passworteingabe alle Befehle auf allen Systemen als Benutzer root ausführen dürfen.

Dies ist selbstverständlich sonst keinesfalls zu empfehlen, wird aber für unser Praktikum jetzt notwendig.

Am besten realisieren Sie dies in einer separaten 'sudoers' Datei mit dem Befehl :

```
# visudo -f /etc/sudoers.d/lab99
```

Testen Sie das Ergebniss z. B. wie folgt:

```
# su - lab99
$ id
uid=8000(lab99) gid=8000(lab99) Gruppen=8000(lab99),4(adm)
$ sudo su -
# id
uid=0(root) gid=0(root) Gruppen=0(root)
# exit
$ exit
```

5.2 Umstellen auf einen neuen Nameserver

Konfigurieren Sie alle 3 VMs so, dass diese zukünftig den Nameserver `192.168.100.100` verwenden. Damit dies reibungslos funktioniert muss die IP Konfiguration der zwei Desktop VMs von `dhcp` auf `static` umgestellt werden.

Achtung

Beachten Sie dabei, dass Sie nicht nur die neuen IP-Adressen eintragen müssen, sondern auch andere Einstellungen vornehmen müssen.

Editieren Sie hierzu zuerst die Datei `/etc/network/interfaces` auf den Desktop VMs entsprechend, und starten Sie die Desktop VMs neu. Es hat sich bewährt, diese Umstellung zunächst an einer VM vorzunehmen und auszutesten, bevor die zweite Desktop-VM umgestellt wird.

Achtung

Berücksichtigen Sie diese Umstellungen bei den Filterregeln der Firewall auf Ihrem Router.

Der neue Nameserver wird auf allen drei VMs über die Datei `/etc/resolv.conf` konfiguriert.

Ändern Sie den Eintrag `nameserver 192.168.100.230` in `nameserver 192.168.100.100` um.

Wenn Ihre Desktop VMs nun statisch konfiguriert und getestet sind, entfernen Sie den DHCP-Relay Dienst auf dem Router wie folgt:

```
# apt-get purge isc-dhcp-relay && apt-get autoremove
```

Beschäftigen sie sich nun noch etwas intensiver mit folgenden Befehlen : `host`, `dig`, `nslookup`

5.3 Systemüberwachung Teil 2

Datei für die Lösungen vorbereiten

Erstellen Sie eine Datei mit Namen `sysinfo.txt` im Verzeichnis `/home/lab99` auf allen Ihren VMs : z. B. als Benutzer `root` mit

```
# touch /home/lab99/sysinfo.txt
```

Stellen Sie sicher das der Benutzer `lab99` die Datei lesen kann. z. B. als Benutzer `root` mit

```
# chgrp lab99 /home/lab99/sysinfo.txt
```

```
# chmod 640 /home/lab99/sysinfo.txt
```

In dieser Datei werde Sie dann zukünftig die Lösungen verschiedener noch folgender Aufgaben ablegen. Immer in der folgenden Zeilenform :

```
# Tag = Value
```

Sie können hierbei Leerzeichen und Tabulatoren verwenden.

Beispiel:

```
ptr =  lab60.vlab.hs-albsig.de
cipher=0x1234
 johnpw =  ganzGeheim
```

5.4 Nameserver Abfragen

Da Sie bisher das Domain Name System vermutlich hauptsächlich über die Art Anfragen

Wie lautet die IP Adresse des Rechners `www.hs-albsig.de` (A Resource Record)

kennen, werden im folgenden Aufgaben gestellt die noch weitere Aufgaben/Eigenschaften eines Nameservers aufzeigen sollen.

Insbesondere die Resource Records

PTR, MX, CNAME, TXT,SOA

und den Zonentransfer (AXFR).

Sie können für die Lösungen der Aufgaben die Befehle host, dig, nslookup benutzen. Hierzu ist der Nameserver `192.168.100.100` sehr offen konfiguriert und lässt sehr viel zu.

5.4.1 FQDN

Wie lautet der zugeordnete FQDN (Fully Qualified Domain Name) der IP Adresse `192.168.100.100`? Ihre Antwort bitte mit dem Tag

`ptr=`

in die Datei `sysinfo.txt` auf dem Router hinzufügen. FQDN bitte ohne abschließenden Punkt eintragen!

5.4.2 Mail Exchanger

Welcher FQDN ist als sogenannter Mail Exchanger für die Domain `vlab.hs-albsig.de` eingetragen ?

Ihre Antwort bitte mit dem Tag

`mx=`

in die Datei `sysinfo.txt` auf dem Router hinzufügen. FQDN bitte ohne abschließenden Punkt eintragen! (Anmerkung ; Ist nur eine MX Eintrag, wir haben keinen richtigen Mailserver ;-))

5.4.3 Alias Eintrag

Der FQDN `honeypot.vlab.hs-albsig.de` ist ein sogenannter Alias Eintrag, Wie lautet der referenzierte FQDN ?

Ihre Antwort bitte mit dem Tag

`cname=`

in die Datei `sysinfo.txt` auf dem Router hinzufügen. FQDN bitte ohne abschließenden Punkt eintragen!

5.4.4 TXT Eintrag

Welchen TXT Eintrag hat die Zone `vlab.hs-albsig.de`?

Ihre Antwort bitte mit dem Tag

`txt=`

in die Datei `sysinfo.txt` auf dem Router hinzufügen. Bitte ohne Anführungszeichen. (Anmerkung : der TXT Eintrag wird sonst oft für SPF (Sender Policy Framework) oder DKIM (Domain Keys Identified Mail) Einträge verwendet)).

5.4.5 Seriennummer

Welche Seriennummer (serial) hat die Zonendatei `vlab.hs-albsig.de` ?

Innerhalb des SOA Records der Zonendatei finden Sie eine 10 Stellige Seriennummer (serial). Benutzen Sie hierzu vielleicht den Befehl `dig` (Tipp: `+multiline`).

Ihre Antwort bitte mit dem Tag

`serial=`

in die Datei `sysinfo.txt` auf dem Router hinzufügen.

5.4.6 Anzahl Resource Record Type A

Wie viele *Resource Record Typ A* Einträge gibt es in der Zonendatei `vlab.hs-albsig.de` ?

Für die Lösung dieser Aufgabe nehmen Sie am besten einen *Zonentransfer* vor. Dann sollten Sie alle *Forward* Einträge (RR Typ A) der Domain `vlab.hs-albsig.de` zu sehen bekommen.

Sie werden eventuell überrascht sein wie viel Informationen der Nameserver preisgibt, wenn man weiß wie man Fragen muss und der Nameserver dies auch erlaubt. Benutzen Sie hierzu den Befehl `dig` mit den entsprechenden Parametern.

Beachten sie hierbei Ihre Firewall auf dem Router, diese muss vermutlich noch angepasst werden.
Ihre Antwort bitte mit dem Tag

`arecords=`

in die Datei `sysinfo.txt` auf dem Router hinzufügen.

5.5 Netzwerkauswertungen

In den folgenden Aufgaben werden Sie offene Ports des TCP Protokolls (UDP kommt später) auf Ihren und fremden Systemen bestimmen, und auch Logauswertungen bezüglich Zugriffsversuchen auf Ihre VMs vornehmen.
Ihre Ergebnisse werden in den Dateien '/home/lab99/sysinfo.txt mit den entsprechenden Tags aufgenommen.

5.5.1 Portnummern

Bestimmen Sie die Portnummern der offenen Ports auf Ihren Desktop VMs für das Protokoll TCP die von außen erreichbar sind.
Geben Sie die Ports hierbei

- numerisch aufsteigen sortiert,
- mit Komma getrennt an,
- nicht mit den eventuell vorhanden Servicenamen (vgl. `/etc/services`)

an.
Benutzen Sie hierzu z. B. die Befehle `netstat` und/oder `lsof` mit den entsprechenden Parametern.
Ihre Antwort bitte mit dem Tag

`openports=`

in die Datei `sysinfo.txt` auf beiden DesktopVMs hinzufügen.

5.5.2 Scan

Ihr Router wird zyklisch von der VM lab99 gescannt, Hierbei sind dann 6 TCP Verbindungsversuche innerhalb von einer Sekunde auf Ihrem Router feststellbar. Der Bereich der betroffenen Ports liegt zwischen 1000 und 7000. Dieser Bereich wurde gewählt um z. B. eine Verwechslung mit den gescannten Ports beim Firewall Test zu vermeiden (21 23 25 445 995 8080).
Geben Sie die Ports hierbei

- numerisch aufsteigen sortiert
- mit Komma getrennt an,
- nicht mit den eventuell vorhanden Servicenamen (vgl. `/etc/services`)

an.
Benutzen Sie hierzu z. B. den Befehl `tcpdump` mit den entsprechenden Parametern und/oder eine der Logdateien.
Ihre Antwort bitte mit dem Tag

`scannedports=`

in die Datei `sysinfo.txt` auf dem Router hinzufügen.

5.5.3 Offene Ports

Finden Sie die offenen TCP Ports auf der VM `honeypot.vlab.hs-albsig.de` heraus.

Geben Sie die Ports hierbei

- numerisch aufsteigen sortiert
- mit Komma getrennt an,
- nicht mit den eventuell vorhanden Servicenamen (vgl. `/etc/services`)

an.

Benutzen Sie hierzu z. B. den Befehl `nmap` mit den entsprechenden Parametern.

Ihre Antwort bitte mit dem Tag

`foundports=`

in die Datei `sysinfo.txt` auf dem Router hinzufügen.

5.6 Systemüberwachung Teil 3

Voraus gesetzt werden u. a.

- Aufgabe 3.0 (Firewall einrichten), nicht zwingend aber wichtig.
- Aufgabe 5.1 (Root Zugang einrichten), zwingend.

Auf Ihren Systemen sind von uns verschiedene Veränderungen vorgenommen wurden. Diese betreffen

- die Benutzerverwaltung,
- das Dateisystem und
- die laufenden Prozesse und TCP Sockets.

5.6.1 Benutzerverwaltung

Es wurden „heimlich“ zwei neue Benutzer mit unterschiedlichen Rechten auf Ihrer ersten Desktop VM angelegt.

Zuerst müssen Sie die Benutzernamen, Gruppennamen, UID, GID herausfinden. Danach die zugeordneten Home Verzeichnisse. Sie werden feststellen das beide Home Verzeichnisse für normale Benutzer unüblich sind. Untersuchen Sie diese. Stellen Sie fest ob reguläre Passwörter für diese Accounts gesetzt sind, wenn nicht, wie sich diese Benutzer am System anmelden könnten.

Sie können diese Accounts auch mit `# su - benutzername` selbst ausprobieren.

!!! Seien Sie hier aber vorsichtig. Im realen Fall muss man hier anders vorgehen !!!

Aufgaben :

SSH-Zugang

Entscheiden Sie, welcher Benutzer deutlich mehr Rechte hat, und verbieten Sie den Zugang per SSH.

Ändern Sie hierzu die Konfigurationsdatei Ihres des SSH Servers `/etc/ssh/sshd_config`, `$ man sshd_config` entsprechend. Es gibt hierbei verschiedene Möglichkeiten. Geprüft wird lediglich ob ein Anmelden per SSH noch möglich ist.

Testen Sie es selbst z. B. von Ihrem Router aus. Vergessen Sie nicht, nach jeder Konfigurationsänderung den SSH-Dienst neu zu starten.

Für diese Tests können Sie temporär ein anderes Passwort setzen. Überlegen sie vorher wie Sie das Passwort wieder auf den ursprünglichen Zustand zurücksetzen.

Passwort herausfinden

Finden Sie das Passwort des *anderen* Benutzer heraus.

Installieren Sie hierzu das Paket `john` und lesen Sie die zugehörige Manpage. Das gesuchte Passwort ist recht einfach gehalten, und `john` sollte zum dechiffrieren nicht lange benötigen.

Ihre Antwort, das Passwort, bitte mit dem Tag

`johnpw=`

in die Datei `sysinfo.txt` auf Ihrer ersten Desktop VM hinzufügen.

SSH Change Root Umgebung einrichten

Nun soll der Benutzer aus 5.6.1.2 in einer Change Root Umgebung für weitere Untersuchungen ;-) quasi eingefangen werden. Die heutigen SSH Server bieten hierzu ein einfache Möglichkeit einen Benutzer in einer solchen Umgebung landen zu lassen. Lesen sie hierzu bitte in der Man Page von `sshd_config` unter dem Stichwort `ChrootDirectory` nach.

In unserem Szenario und für unsere Zwecke reicht hierzu eine zweizeilige Ergänzung in der Datei `/etc/ssh/sshd_config` aus. Als Change Root Verzeichnis benutzen wir `/var/lib/chroot`.

```
Match User Benutzername
ChrootDirectory /var/lib/chroot
```

Achtung: Bitte an das Ende der Datei einfügen, da der `Match` Block für alle folgenden Einstellungen bis zum nächsten „Match“ Block bzw. Dateiende gilt ! Aufwendiger wird das Vorbereiten dieser Umgebung. Für die Lösung der Aufgabe werden Sie u.a. folgende Befehle verwenden :

mknod, ldd, ltrace, strace, mount

Sollte z. B. strace noch nicht auf Ihrem System vorhanden sein können Sie dies mit 'apt-get install strace' installieren.

Wir empfehlen folgende Vorgehensweise für unsere Anforderungen:

1. Anlegen der notwendigsten Verzeichnisse unterhalb `var/lib/chroot`: `bin, dev, etc, home, lib, lib64, proc, usr`:
2. Anlegen der notwendigen Gerätedateien mit `mknod`: `null, random, tty, zero` Übernehmen sie hierbei die Typ, Major, Minor Bezeichnungen von den Gerätedateien in `/dev`.
3. Mounten des `proc` Dateisystem (Ist nicht immer unbedingt notwendig).

4. Kopieren der Programme, benötigten *Shared Libraries* und event. zugehörige andere Dateien.
5. Welche *Shared Libraries* welches Programm benutzt erfahren Sie z. B. mit dem Befehl `ldd`
6. Welche Dateien ein Programm zu öffnen versucht erfahren Sie z. B. mit `strace -e open ping localhost`.

Im folgenden ein Beispiel mit dem Befehl `cat`:

```
# ldd $(which cat)
linux-vdso.so.1 => (0x00007ffdc45e5000)
libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007f1a95075000)
/lib64/ld-linux-x86-64.so.2 (0x00007f1a95433000)
```

Die erste Angabe `linux-vdso.so.1` (Linux Virtual Dynamic Shared Object) ist Sache des Kernels und braucht nicht kopiert werden, besser kann nicht kopiert werden, da nicht als Datei vorhanden :-). `# cp -p /lib/x86_64-linux-gnu/libc.so.6 /var/lib/chroot/lib/x86_64-linux-gnu/`
`# cp -p /lib64/ld-linux-x86-64.so.2 /var/lib/chroot/lib64/`

Etwaige Unterverzeichnisse von `# /var/lib/chroot/{lib,lib64}` müssen Sie dabei natürlich schon erstellt haben.

Folgende Anforderungen der Chroot Umgebung werden gestellt:

- Die Befehle `bash, cat, id, ls, ps, strace` sollen verfügbar sein, und auch wie gewohnt funktionieren.
- Der Befehl `ping` soll verfügbar sein, funktionieren und auch mit Hostnamen zurecht kommen.
- Die Chroot Umgebung darf keine 'Links' enthalten und soll nicht größer als 10MB sein.

Hinweise :

- Kopieren Sie das Home Verzeichnis des Benutzer am besten 1 zu 1 mit `$ cp -a` in die Chroot Umgebung unter `/var/lib/chroot/home`. Der Benutzer sollte im ersten Augenblick nicht bemerken das er in einer Chroot Umgebung untergebracht ist.
- Das Virtuelle Dateisystem `proc` lässt sich einfacher mounten als es zunächst aussieht: z. B. `# mount -t proc proc <wohin>` Denken Sie daran dass dieser `mount` nach einem Reboot wieder fehlt. Anmerkung: Es besteht auch die Möglichkeit dass Benutzer nur die eigenen Prozesse sehen wenn sie z. B. `$ ps axu` ausführen.
  z. B. `# mount -o remount,hidepid=2 /var/lib/chroot/proc`
  Wir setzen dies im Rahmen diese Praktikums nicht ein, Sie können es aber gerne temporär ausprobieren.
- Wenn Sie mit `strace` arbeiten lassen Sie sich nicht irritieren durch die event. vielen Fehlermeldungen mit `(No such file or directory)`. Oft (!nicht immer!) werden nur die wenigsten Dateien wirklich gebraucht.
- Überlegen Sie wie sie die Meldung `I have no Name!` wegbekommen könnten. (Kopieren Sie hierbei event. nicht die ganzen Dateiinhalte...).

- Testen Sie Ihre Konfiguration Remote mit `$ ssh -l benutzername labXY` oder Lokal mit `$ ssh -l benutzername localhost` .
- Beachten Sie, dass bestimmte Dateiattribute beim normalen Kopiervorgang event. nicht übernommen werden (z. B. `# cp $(which ping)` ;-).
- Die Aufgabe *ping mit Hostnamen* ist nicht einfach. Gehen Sie diese am besten mit `strace` an, z. B. `$ strace -e open ping -w 1 -c 1 lab99` .

5.6.2 Dateisystem

In dem Dateisystem Ihrer ersten Desktop VM wurden zwei Veränderungen vorgenommen.

Eine ausführbare Datei ist gespeichert worden bei der das SUID Bit gesetzt ist.

Finden sie diese Datei z. B. mit Hilfe des Befehls `find` und dem Parameter `-perm` und Abgleich mit folgender MD5 Summe der gesuchten Datei :

`3c7ad7d23b74c93b838ef55035d07784`

Ihre Antwort, nur der Dateiname, bitte mit dem Tag

`newfile=`

in die Datei `sysinfo.txt` auf Ihrer ersten Desktop VM hinzufügen.

Die zweite Veränderung ist nicht so leicht zu finden, hat aber mit obiger Veränderung zu tun und ist in 5.6.3 gefragt.

5.6.3 Laufende Prozesse und TCP Sockets

Auf Ihrem System ist ein von außen erreichbarer TCP Dienst installiert worden. Finden Sie das zugehörige Programm und untersuchen Sie den Zusammenhang mit der gefunden Datei aus 5.6.2.

Beachten Sie auch die Aufgabe 5.5.1, abhängig davon wann Sie 5.5.1 bearbeitet haben, müssen Sie den gefundenen Port eventuell noch hinzufügen.

Finden Sie die Portnummer heraus auf die dieser Dienst einen `bind` gemacht hat und unter welcher UID dieser läuft..

Ihre Antworten, beides numerisch, bitte mit den Tags

`newport=`
`newuid=`

in die Datei `sysinfo.txt` auf Ihrer ersten Desktop VM hinzufügen.

5.6.4 System Nacharbeiten und Verbesserungen

In 5.0 wurden drei Pakete auf Ihren VMs installiert die von Ihnen bisher vermutlich nicht wirklich genutzt wurden:

`postfix, mailutils, rkhunter`

Die ersten zwei Pakete beinhalten MTA,MDA,MUA (Mail Transfer Agent, Mail Delivery Agent, Mail User Agent) sind also Grundbestandteil des eMail Systems. Für unserer Zwecke brauchen diese nicht weiter konfiguriert werden, sondern sollten schon funktionieren.

Dies ist im realen Einsatz keinesfalls so, sondern bedarf diverser Nacharbeiten (Stichworte : Sicherheit, SPAM, Trojaner/Viren...)

Beschäftigen sie sich jetzt mit folgenden Manpages (Sie brauchen nicht alle Parameter durchlesen ;-) :

`cron crontab mail rkhunter`

Dies sollte ausreichen im Ihnen einen Einstieg in `rkhunter` und das von diesem benutzte Mailsystems und Cronsystems zu geben. Anmerkung : Das Cronsystem ist eigentlich immer in allen Unixoiden Basisinstallationen vorhanden, wie ursprünglich das Mailsystem auch.

Schauen Sie sich folgende zwei Konfigurationsdateien an:

`/etc/rkhunter.conf` und `/etc/default/rkhunter`

In der ersten Datei brauchen Sie in Rahmen dieser Aufgabe nicht zu ändern. In der zweiten suchen Sie den Eintrag der `rkhunter` über das Cronsystem zu einem täglichen Durchlauf inklusive eMail Versand an den Benutzer `root` aktiviert. Ergebnisse dieser Durchläufe finden sie dann zukünftig in

`/var/log/rkhunter.log`

oder in der eMail des Benutzers `root`. Diese können Sie mit dem Programm `mail` als Benutzer `root` lesen. Ein 'alias' auf einen 'normalen' Benutzer wäre eventuell aus Sicherheitsgründen besser, werden wir aber im Rahmen dieses Praktikums nicht anlegen.

Wäre eine Aktivierung vor Bearbeitung der Aufgaben 5.6 hilfreich gewesen ? ;-) und überlegen Sie was man an dem `rkhunter` System verbessern bzw. anders konfigurieren könnte/sollte.

5.7 Kontrollfragen

Frage

Frage 5.1 Bei vielen Anwendungen spricht man von Server und Client. Wie heißen die jeweiligen Pendants im Domain Name System?

Frage

Frage 5.2 Wo wird der Client unter Linux konfiguriert?

Frage

Frage 5.3 Welche Funktionen hat `openssl`? (Hinweis: Es sind viel mehr als es auf den ersten Blick erscheint.)

Frage

Frage 5.4 Welche Arten von Ressource Records kennen Sie und was für Rückschlüsse lassen sich anhand dieser auf das Gesamtsystem ziehen?

Frage

Frage 5.5 Was ist ein Zonentransfer und weshalb kann er zur Erkundung eines Systems interessant sein (Thema reconnaissance beim Pentest)?

Frage

Frage 5.6 Wie läuft ein TCP Scan bei `nmap` technisch ab? Wo liegt der Unterschied beim TCP-Connect-Scan und beim TCP-Syn-Scan? Kennen Sie weitere Arten mit `nmap` zu scannen?

Frage

Frage 5.7 Im Praktikum haben Sie sich vor allem mit TCP beschäftigt. Viele der Netzwerkinfrastrukturdienste sind jedoch keine TCP Dienste. Worauf sollten Sie bei der Erkundung eines Systems ebenso achten?

Kapitel 6

TLS Sicherheit

Bei allen folgenden Aufgaben denken Sie auch noch an die Log Dateien in `/var/log`. Insbesondere auch die, des im folgenden mehrmals umkonfigurierten HTTP Servers `apache2`. Diese befinden sich in `/var/log/apache2`. Die Einträge die dort zu finden sind, können insbesondere bei der Fehlersuche sehr behilflich sein.

6.1 TLS Handshake untersuchen

Untersuchen Sie einen TLS Verbindungsaufbau zwischen Ihren Desktop VMs mit Wireshark (Hinweis : event. mit dem Filter SSL) . Verwenden Sie für den Verbindungsaufbau den Browser Firefox (am besten mit dem Parameter `--no-remote`) auf der zweiten Desktop VM und die HTTPS Webseite auf der ersten Desktop VM. Betrachten Sie die aufeinander folgenden Pakete und deren Inhalte. Beantworten Sie für sich folgende Fragen, wobei nur die mit Tag Angabe ausgewertet werden :

- Wie viele IP Pakete werden hin und her gesendet um den kompletten HTTPS Request incl. Response zu übertragen?
- Ab dem wievielten Paket können tatsächlich Anwendungsdaten vom Client zum Server und umgekehrt verschickt werden?
- Wie viele Cipher Suites bietet der Client (Firefox) an? (Tag ist `numcipher=`, eine Dezimalzahl) Zählen sie auch diejenigen mit die 'Wireshark' eventuell als 'unknown' bezeichnet.
- Welche asymmetrischen und symmetrischen Crypto Algorithmen sind in diesen Cipher Suites enthalten?
- Welche Version von TLS wird ausgehandelt? (Tag ist `tlsversion=`, hexadezimal mit führendem `0x`, z.B. `0x0401`) Wie haben in unserer Umgebung schon beobachtet das es zwischen 2 Versionen wechseln kann. Geben Sie bitte die höhere verwendete TLS Version in den 'Application Data' Paketen an.
- Verwendet die Verbindung Kompression? Welche Nachricht gibt den Ausschlag hierfür? Wie ist hier der Wert (1Byte) des Kompressionsverfahrens? (Tag ist `comp=`, hexadezimal mit führendem `0x`, z.B. `0x01`)

Ihre Antworten bitte mit den Tags

```
numcipher=
tlsversion=
comp=
```

in die Datei `sysinfo.txt` auf Ihrer ersten Desktop VM hinzufügen.

6.2 Cipher Suites deaktivieren

Machen Sie sich etwas mit der Konfiguration von Firefox über den Dialog `about:config` (in die URL Zeile eingeben) vertraut. Finden Sie heraus wo die von SSL verwendeten Cipher Suites verwaltet werden (Hinweise : Suche nach SSL) . Deaktivieren Sie alle Cipher Suites die nicht auf RSA basieren, hier alle die mit DHE oder ECDHE beginnen. Dies ist zwar sicherheitstechnisch nicht sinnvoll (die Diffie-Hellman basierte Schlüsselaushandlung ist der RSA basierten Schlüsselaushandlung vorzuziehen), jedoch zeigt die Übung wo Sie im Ernstfall unsichere Cipher Suites clientseitig abschalten können. Beobachten Sie nun den Handshake in Wireshark erneut. Beobachten Sie, wie sich die Menge der angebotenen Cipher Suites im Client Hello verändert.

Aufgabe

Aufgabe 6.1 Welche Cipher Suite wählt der Server jetzt?
Ihre Antwort, hexadezimal mit führendem `0x` und, falls vorkommend, kleinen Buchstaben, bitte mit dem Tag
`cipher=`
in die Datei `sysinfo.txt` auf Ihrer ersten Desktop VM hinzufügen.

Danach aktivieren Sie die DHE und ECDHE Cipher Suites bitte wieder, sonst könnte es vielleicht noch zu Problemen mit Folgeaufgaben in diesem Praktikum kommen.

6.3 Umleitung von HTTP auf HTTPS

Traditionelle Umleitung von HTTP auf HTTPS einrichten (! Auf der Ersten Desktop VM !) Als Betreiber einer Website sollten Sie an der Sicherheit der Kommunikation Ihrer Besucher Interesse haben. Heute gibt es praktisch keine Browser mehr, die kein HTTPS unterstützen. Daher können Clients meist guten Gewissens auf die sichere Version von HTTP umgeleitet werden. Richten Sie eine Weiterleitung von HTTP auf HTTPS ein, indem Sie einen Redirect in die Konfiguration Ihrer HTTP Website eintragen. Dies erreichen Sie indem Sie die Datei
`/etc/apache2/sites-available/000-default.conf`
bearbeiten. Hier ist die Konfiguration für ihre nicht gesicherte Website zu finden. Eine dauerhafte Umleitung können Sie folgendermaßen erreichen: Tragen Sie in den Virtual Host Teil die Zeile

```
ServerName labXY.vlab.hs-albsig.de
Redirect permanent / https://labXY.vlab.hs-albsig.de/
```

ein. Beachten Sie hierbei den abschließenden `/`. Dieser ist, zumindest in unserer Umgebung, wichtig.

Nach einem Neustart von Apache mit `# service apache2 restart` sollten Sie nach dem Aufruf von

`http://labXX.vlab.hs-albsig.de`

im Firefox Browser direkt nach `https://labXX.vlab.hs-albsig.de` weitergeleitet werden.

Beobachten Sie nun den Aufruf von `http://labXX.vlab.hs-albsig.de` und die resultierende Umleitung in Wireshark. Welche TCP Verbindungen werden dabei aufgebaut? Welche Teile davon

sind gesichert und welche sind ungesichert? Betrachten Sie jeweils die HTTP Nutzlast und die HTTP Header und machen Sie sich mit deren Inhalten vertraut. Löschen sie hierbei den Browser Cache öfter mit <CTRL>+<SHIFT>+<DEL>.

Zum Abschluss testen Sie die Umleitung bitte noch mit `wget` wie folgt:

```
# wget -S http://labXY.vlab.hs-albsig.de/index.html --ca-certificate=/path/to/myroot.cer
```

(`/path/to/myroot.cer` ist das Zertifikat Ihrer CA aus 2.3/2.4)

oder

`# wget -S http://labXY.vlab.hs-albsig.de/index.html -no-check-certificate` (ohne Zertifikatsüberprüfung)

Der Paramter `# -S` bewirkt das auch der vom HTTP Server mitgeschickte HTTP Header angezeigt wird. Die Dateiangabe `# index.html` kann auch weggelassen werden. ⇒ Ausgewertet wird lediglich ob Ihr HTTP Server auf der ersten Desktop VM mit Redirect konfiguriert ist und funktioniert.

6.4 HTTPS erzwingen

Aufruf von HTTPS durch HTTP Strict Transport Security (HSTS) erzwingen (! Auf der Zweiten Desktop VM !). Es ist oft nicht ausreichend, eine einfache Umleitung auf eine HTTPS Seite einzurichten. HSTS erzwingt zukünftig eine gesicherte HTTPS Verbindung sobald ein Client/Browser zum ersten mal die Seite aufgerufen hat. HSTS wird über die HTTP Header an den Client kommuniziert. Es kann in der Konfiguration Ihrer HTTPS Seite unter

`/etc/apache2/sites-available/default-ssl.conf`

aktiviert werden. Fügen Sie dazu den Eintrag

```
Header always set Strict-Transport-Security "max-age=31536000; includeSubDomains"
```

zu ihrem virtuellen Host für HTTPS hinzu. Lesen Sie unter `https://tools.ietf.org/html/rfc6797` nach was die Teile dieses Aufrufs bedeuten. Um die Option Header mit Apache verwenden zu können muss noch das Modul `headers` aktiviert werden. Sie erreichen dies mit

```
# a2enmod headers
```

Nach einem Neustart von Apache sollte nun die Header Information zu HSTS in die HTTP Antwort einfließen. Prüfen Sie dies indem Sie Ihre Website von der jeweils anderen Desktop VM auf der Kommandozeile z.B. mit `wget` wie in Aufgabe 6.3, allerdings mit der URL

`https://labXY.vlab.hs-albsig.de`.

Im Ergebnis sollten Sie jetzt die folgende Zeile sehen:

```
Strict-Transport-Security: max-age=31536000; includeSubDomains
```

Damit der Benutzer auch mit der URL `http://labXY.vlab.hs-albsig.de` eine, dann mit HSTS konfigurierte, SSL Verbindung bekommt, wenden Sie jetzt noch den Redirect von Aufgabe 6.3 an.

Beobachten Sie nun einen mehrfachen Verbindungsaufbau von Ihrem Firefox Browser zu http://labXX.vlab.h
albsig.de (zweite Desktop VM). Worin unterscheidet er sich nun vom Verbindungsaufbau den Sie in der vorigen Teilaufgabe (erste Desktop VM) betrachtet haben. Welche Nachrichten werden verschlüsselt und welche werden unverschlüsselt gesendet. Löschen sie hierbei den Browser Cache öfter mit '<CTRL>+<SHIFT>+<DEL>'.

⇒ Ausgewertet wird lediglich ob Ihr HTTP Server auf der zweiten Desktop VM mit Redirect und HSTS konfiguriert ist und funktioniert.

6.5 Certificate Pinning

Certificate Pinning einrichten (! Auf der Ersten Desktop VM !).

In dieser Teilaufgaben werden Sie SPKI (Subject Public Key Information) Certificate Pinning für Ihre Website einrichten. Aufgrund der im Praktikum verwendeten lokal selbst erstellten Zertifikate verhält sich das Certificate Pinning jedoch etwas unterschiedlich zum Certificate Pinning mit regulär von bekannten CAs ausgestellten Zertifikaten im Internet. Dies liegt daran, dass die großen Browser Hersteller das Certificate Pinning bzw. die resultierende Verweigerung des Seitenaufrufs (bei nicht zum Pinning passendem Zertifikat) bei Zertifikaten von nicht von vornherein in den Browser eingebauten CAs nicht durchführen. Durch bestimmte Einstellungen des Browsers Firefox lässt sich dieses Verhalten in unterschiedlichen Stufen ändern.

6.5.1 Vorbereitungen zum Certificate Pinning

Machen Sie sich zuerst nochmals mit dem Konzept des Certificate Pinnings vertraut. Sie finden weitere Informationen hier:
https://developer.mozilla.org/de/docs/Web/Security/Public_Key_Pinning
https://tools.ietf.org/html/rfc6797

Wir werden das Certificate Pinning nur auf ihrer ersten Desktop VM durchführen. Um dies vorzubereiten müssen Sie insgesamt drei Zertifikate für diese VM besitzen (Ein Zertifikat als Haupt-Pinning-Zertifikat, eines als Backup Zertifikat und eines als ungepinntes Zertifikat mit dem wir einen Angriff simulieren). Ein Zertifikat besitzen Sie bereits. Es sind also noch zwei weitere Zertifikate wie in Teilaufgabe 2.3/2.4 beschrieben zu erstellen.

Verwenden Sie bei allen Zertifikaten stets den DNS Namen Ihrer ersten Desktop VM als Common Name (CN), also `labXY.vlab.hs-albsig.de` mit eventuell führender Null. Die anderen Parameter des Zertifikats sollten Sie, in diesem Praktikum, wie vorbelegt beibehalten. Damit dies funktioniert müssen Sie in der Datei openssl.cnf noch das Kommentarzeichen vor `unique_subject = no` entfernen. Sollte sich in Ihrem Verzeichnis `myCA` eine Datei namens `index.txt.attr` befinden, müssen Sie dort auch noch `unique_subject = yes` auf `unique_subject = no` abändern, wenn es diesen Eintrag dort gibt. Als Datei Namen wählen Sie entsprechend

- `labXYbackup.{cer,key}`
- `labXYattack.{cer,key}`

Kopieren Sie die Zertifikate und Private Keys nach `/etc/apache2/ssl` wie in Aufgabe 2.3/2.4 beschrieben, sodass Sie diese später einfach verwenden können. Berechnen Sie nun für die zwei ersten Zertifikate (Orignal und Backup) die nötigen Pins. Dies kann mit dem folgenden Befehl geschehen:

```
# openssl x509 -in labXY.cer -pubkey -noout | openssl rsa -pubin -outform der |
openssl dgst -sha256 -binary | openssl enc -base64
```

```
# openssl x509 -in labXYbackup.cer -pubkey -noout | openssl rsa -pubin -outform
der | openssl dgst -sha256 -binary | openssl enc -base64
```

Setzen Sie sich kurz mit dem Befehl auseinander. Was bedeuten die Teile jeweils? Was tun sie? Die folgenden Manpages sind dafür hilfreich: `openssl, x509, rsa, dgst, enc`

Das Format der beiden Pinning Hashes (Original SSL Cert und Backup Cert) sollten so ähnlich aussehen wie das der folgenden Pinning Hashes:

```
C/1eCc3AsiCeF6TlmtI4YKaa4EOTWtb7dZuVpIcTs48=
26CCxHLrxwZsek1Fp0m2LhqoKyuinKuzXDU53NvhZhE=
```

Diese Hashes müssen Sie nun in den HTTP Headern an die Browser ausliefern. Hierzu editieren Sie die Konfigurationsdatei Ihrer SSL Website:

`/etc/apache2/sites-available/default-ssl.conf`

Fügen Sie dem Virtual Host für SSL folgende Zeile, natürlich mit Ihren Pinning Hashes, hinzu (alles in einer Zeile):

```
Header always set Public-Key-Pins:

"pin-sha256=\"C/1eCc3AsiCeF6TlmtI4YKaa4EOTWtb7dZuVpIcTs48=\";
pin-sha256=\"26CCxHLrxwZsek1Fp0m2LhqoKyuinKuzXDU53NvhZhE=\";
max-age=15768000; includeSubDomains"
```

Machen Sie sich mit den Teilen dieser Direktive vertraut. Im zugehörigen RFC sind die Parameter ebenfalls erklärt. Aktivieren Sie das Apache Modul `headers` (vgl. 6.4) und starten Sie nun Ihren Apache Webserver neu.

6.5.2 Certificate Pinning prüfen

Überprüfen Sie nun mit `wget` (vgl. Aufgabe 6.3.) ob die Pinning Information richtig übermittelt wird. Es sollte nun eine ähnliche Ausgabe wie:

```
Public-Key-Pins:

pin-sha256="C/1eCc3AsiCeF6TlmtI4YKaa4EOTWtb7dZuVpIcTs48=";
pin-sha256="26CCxHLrxwZsek1Fp0m2LhqoKyuinKuzXDU53NvhZhE=";
max-age=15768000; includeSubDomains
```

in den HTTP Headern zu sehen sein.

1. Starten sie nun auf Ihrer zweiten Desktop VM den Browser Firefox.
2. Löschen sie folgende Bestandteile der Chronik des Browsers: Cache, Webseite-Einstellungen. Am besten über `<CTRL>+<SHIFT>+<DEL>`
3. Öffnen Sie die Erweiterte Konfiguration des Browser mit der Eingabe von `about:config` in der URL Eingabezeile.
4. Bejahen sie die Warnung und suchen dann in der oberen Eingabezeile nach `security`
5. Sie sollten jetzt u. a. drei Einstellmöglichkeiten die das Pinning betreffen sehen und diese wie folgt ändern
   - `security.cert_pinning.enforcement_level` auf den Wert `2`
   - `security.cert_pinning.max_max_age_seconds` auf den Wert `31536000`
   - `security.cert_pinning.process_headers_from_non_builtin_roots` auf `true`
6. Öffnen Sie nun einen neuen Tab und mit `<CTRL>+<SHIFT>+<I>` die Entwicklerwerkzeuge zur besseren Kontrolle. Dort von Interesse sind jetzt besonders die Tab's *Konsole* und *Netzwerk*.

7. Öffnen Sie jetzt die HTTPS Webseite ihrer ersten VM im aktuellen leeren Tab. Beobachten Sie dabei die Ausgabe des Browsers im Tab *Netzwerk*. Klicken Sie dort auf den letzten `GET` und dann rechts auf *Sicherheit*. Dort sollte u.a. der Eintrag `Public-Key-Pinning aktiviert` erscheinen.

Das Pinning funktioniert nur wenn Sie im Browser Ihre Root-CA importiert haben. Wenn der Zielrechner über 'Ausnahme hinzufügen' (Server Zertifikat) importiert wurde, löschen Sie dessen Zertifikat wieder, importieren Ihre Root-CA (wenn noch nicht geschehen), und rufen die Startseite des Zielrechners erneut auf. !!!

6.5.3 Certificate Pinning Angriff simulieren

Nun ist es an der Zeit zu prüfen was geschieht, wenn Sie die gepinnten Zertifikate durch ein anderes Zertifikat austauschen. Hierzu tauschen Sie nun ihr erstes SSL Zertifikat (`*.key` und `*.cer`) gegen das dritte (ungepinnte) Zertifikat `labXYattack.{cer,key}` aus.

In der Konfiguration ihrer Website in `/etc/apache2/sites-available/default-ssl.conf` haben Sie u.a. angegeben wo die Zertifikats Dateien liegen. Bei uns im Praktikum in `/etc/apache2/ssl`. Kopieren Sie nun

- `labXYattack.cer` nach `/etc/apache2/ssl/labXY.cer`
- `labXYattack.key` nach `/etc/apache2/ssl/labXY.key`

Danach starten Sie Apache nochmals neu.

Öffnen Sie nun ihre HTTPS Website erneut im Browser. Überprüfen Sie im unteren 'Netzwerk' Tab das diese nicht aus dem Cache geholt wurde.

Nun sollte im Browser Fenster in etwa folgende Meldung erscheinen

```
Fehler: Gesicherte Verbindung fehlgeschlagen
An error occurred during a connection to labXY.vlab.hs-albsig.de. The server uses
key pinning (HPKP) but no trusted certificate chain could be constructed that matches
the pinset. Key pinning violations cannot be overridden.
Error code: MOZILLA_PKIX_ERROR_KEY_PINNING_FAILURE
```

Wenn Sie nun mit `<CTRL>+<SHIFT>+<DEL>` Ihre Chronik inkl. *Website-Einstellungen* wieder löschen und die Seite erneut laden wird diese allerdings wieder angezeigt, jedoch mit Deaktiviertem *Public-Key-Pinning*.

6.6 Kontrollfragen

Frage

Frage 6.1 Woher hat der Benutzer mit den „vielen“ Rechten diese? Welchem anderen Benutzer entspricht dieser Benutzer?

Frage

Frage 6.2 Weshalb war das Finden des Passworts mit `john` so schnell möglich? Wie könnte man diese Zeit deutlich verlängern?

Frage

Frage 6.3 Was befindet sich in `/proc` und wofür werden die Inhalte benötigt?

Frage

Frage 6.4 Warum sehen Sie nach der Einrichtung der `chroot` Umgebung die Meldung `I have no Name!`? Wie konnte man diese Meldung beseitigen?

Frage

Frage 6.5 Wie kann es sein, dass der Neue Prozess aus Aufgabe 5.6.3 mit einer hohen (nicht privilegierten) UID läuft und trotzdem Port 21 verwendet? Wie kann man so etwas bewerkstelligen?

Frage

Frage 6.6 Wäre eine Aktivierung von `rkhunter` gleich nach der Installation hilfreich gewesen? Wie hätte Ihnen dies das Leben erleichtert?

Frage

Frage 6.7 `rkhunter` findet Veränderungen und zeigt sie an. Wie könnte ein Benutzer mit entsprechenden Rechten `rkhunter` austricksen?

Kapitel 7

Systemüberwachung Teil 4

7.1 Das Syslog System

Im Rahmen dieses Praktikums können wir ihnen das System zum Erfassen von Systemereignissen nur kurz vorstellen. Unter Unix Systeme wird hier der Dienst `syslog` eingesetzt, der derzeit auf vielen aktuellen Unix Systemen mit `rsyslogd` (*reliable and extended syslogd*) realisiert wird.

Grob vereinfacht nimmt dieser Dienst Meldungen von anderen Prozessen entgegen und verteilt diese dann je nach Konfiguration auf z. B. Logdateien und Gerätedateien, oder gibt Sie an andere Syslog Server weiter (*Loghost*). Zur Steuerung was mit den einzelnen Meldungen zu geschehen hat senden die Prozesse zusätzlich zur eigentlichen Meldung u.a. noch sogenannte *facilities* und *levels*.

Auszug aus einer Manpage (`logger`):

`Valid facility names are: auth, authpriv (for security information of a sensitive nature), cron, daemon, ftp, kern (can't be generated from user process), lpr, mail, news, security (deprecated synonym for auth), sys- log, user, uucp, and local0 to local7, inclusive.`

`Valid level names are: alert, crit, debug, emerg, err, error (deprecated synonym for err), info, notice, panic (deprecated synonym for emerg), warning, warn (deprecated synonym for warning). For the priority order and intended purposes of these levels, see syslog(3).`

Verschiedene aktive Netzwerkkomponenten wie Switches/Router und auch Drucker unterstützen das in `RFC 3164` (https://www.ietf.org/rfc/rfc3164.txt) beschriebene Protokoll, welches sicherlich verschiedene Schwachstellen/Sicherheitsprobleme hat. Es gibt auch Implementierungen für andere Desktop.- und Server Betriebssysteme.

Viele Dienste wie z. B. der Apache HTTP Server können so konfiguriert werden das Sie auch das Syslog System verwenden. In unserem Praktikum gehen die meisten Meldungen des Apache Servers am Syslog System vorbei direkt nach `/var/log/apache2`. Dies könnte umkonfiguriert werden, ist aber nicht Ziel dieser Aufgabe.

Die Konfigurationsdateien des Syslog Systems finden Sie hier:
`/etc/rsyslog.conf`
`/etc/rsyslog.d/`

Die Datei `rsyslog.conf` steuert das grundsätzliche Verhalten und die Funktionen des Dienstes, Wird diese geändert muss der Dienst mit

```
# service rsyslog restart
```

neu gestartet werden.

Im Verzeichnis `/etc/rsyslog.d` befinden sich dann die Konfigurationen der für das bearbeiten/verteilen der Systemmeldungen. Die meisten Änderungen dort, können mit

```
# service rsyslog reload
```

aktiviert werden.

Anmerkungen:

- Bei `reload` wird dem Prozess `rsyslog` das Signal HUP geschickt welches diesen dazu veranlasst seine Konfigurationsdateien neu einzulesen.
- Wenn Sie sich nicht sicher sind ob ein `reload` ausreicht, verwenden Sie einfach einen `restart`.
- Mit dem Progamm `logger` lassen sich sehr einfach Nachrichten über das Syslog System verschicken. Benutzen Sie es eventuell für Tests.

7.2 Loghost einrichten

Ziel dieser Aufgabe ist es, das zusätzlich zur aktuellen Konfiguration, alle Systemmeldungen Ihrer VMs in einer Datei auf Ihrem Router landen. Benennen Sie diese Datei, wie im folgenden beschrieben, mit `mylogs` abgelegt im üblichen Logdateien Verzeichnis `/var/log`. Dies ist wichtig für die Auswertungen. Normalerweise sind die Dateinamen mehr oder weniger frei wählbar, solange sie nicht mit anderen kollidieren. Beachten Sie auch die angegeben Dateiattribute. Damit bestimmen Sie u.a. wer auf dem System was lesen darf. Für die Auswertung muss die Gruppe `adm` lesen können.

Sicherlich müsste auch noch darüber nachgedacht werden ob eventuell Meldungen die ein Datenschutz.- und/oder Sicherheitsprobleme darstellen/beinhalten können nicht im Klartext über das Netz gehen dürfen, und es ja auch nicht gesichert ist wo diese wirklich landen. In unserem Praktikum lassen wird dies unberücksichtigt. Die Aufgabestellung kann mit den folgenden Schritten bearbeitet werden.

Auf dem Router:

- Konfiguration das alle Meldungen (alle `facilites` und deren `levels`) in eine Datei geschrieben werden.
- Diese `Logdatei` mit den richtigen Attributen anlegen.
- Konfiguration des `rsyslogd` das dieser Meldungen über das Netzwerk annimmt.
- Anpassen der Firewall auf dem Router

Auf den Desktops :

- Konfiguration das alle Meldungen (alle `facilites` und deren `levels`) zum Loghost (Router) geschickt werden.

Gehen Sie im einzelnen am besten wie folgt vor:

Auf der Router VM

1. Erstellen Sie eine neue Datei `/etc/rsyslog.d/mylogs.conf` mit folgendem einzeiligen Inhalt
   `*.* /var/log/mylogs`
2. !!! Erstellen Sie die Logdatei '/var/log/mylogs' mit den Attributen : Owner root. Group adm, Mode 640 !!!
3. Führen Sie einen 'restart' des 'rsyslog' Dienstes durch und beobachten Sie ob Meldungen in der neue Logdatei ankommen.
4. Editieren Sie die Datei `/etc/rsyslog.conf`: Entfernen Sie die Kommentarzeichen der 2 Zeilen (module(load=ïmudp"), input(type=ïmudp"port="514")) unterhalb `# provides UDP syslog reception`
5. Führen sie einen 'restart' des 'rsyslog' Dienstes durch
6. Kontrollieren Sie ob dieser Dienst jetzt auf dem UDP Port 514 (Standart Syslog Port) empfangsbereit ist.
7. Passen Sie Ihre Firewall auf dem Router an, sodass Ihre 2 Desktop VMs Meldungen per Syslog Protokoll an Ihren Router senden können.

Auf beiden Desktop VMs

1. Erstellen Sie eine neue Datei `/etc/rsyslog.d/mylogs.conf` mit folgendem einzeiligen Inhalt, wobei `Z` die Host IP ihre Routers im 100er Segment ist.
   `*.* @192.168.100.Z`
2. Sie können sicherlich auch die IP Adressen des Routers jeweils aus 'Sicht' der Desktop VMs wählen (192.168.?.230).
3. Führen Sie einen 'restart' des 'rsyslog' Dienstes durch und beobachten Sie ob Meldungen auf Ihrem 'Loghost' ankommen.

7.3 Logrotate auf dem Loghost konfigurieren

Damit die Logdateien eines System nicht beliebig wachsen und ewig bestehen bleiben, gibt es z. B. das Programm 'logrotate' mit dem sehr einfach und trotzdem sehr genau bestimmt werden kann, wie mit den Logdateien zu verfahren ist. Normalerweise wird `logrotate` über das Cron System ausgeführt, man kann es allerdings auch explizit aufrufen, oder auch für andere Zwecke als das Rotieren von Logdateien benutzen.

Da mit der neuen Logdatei `mylogs` sehr viel Daten zu erwarten sind ist es angebracht diese Datei in die Logrotate Konfiguration mit aufzunehmen. Konfiguriert wird `logrotate`, ähnlich wie `rsyslogd` in `/etc/logrotate.conf` und `/etc/logrotate.d/`. Wir brauchen nur die Datei `/etc/logrotate.d/rsyslog` z. B. wie folgt zu ergänzen (vgl. Eintrag für `/var/log/syslog`):

```
/var/log/mylogs { create 0640 root adm
rotate 7
daily
missingok
notifempty
delaycompress
compress
```

```
postrotate
invoke-rc.d rsyslog rotate > /dev/null
endscript
}
```

Mit der hierbei hinzugefügten Zeile `create 0640 root adm` können gleich noch die Dateiattribute explizit gesetzt werden. Kontrollieren Sie ob nach dem rotieren die Zugriffsrechte der Orginal und der rotierten Datei(en) korrekt sind.

Schauen Sie sich die obigen Einträge ruhig einmal genauer an. Es gibt noch weitere Einstellungen wie z. B. maximale Größe. Im `/var/log` Verzeichnis sollten Sie schon rotierte' Logdateien (`*.Nummer`, `*.gz`) vorfinden. Gleiche Ausprägungen Ihrer `mylogs` Datei sollten dann 'ab morgen' dort zu finden sein. Sie sollten das Rotieren aber sofort selbst testen, dabei werden verschiedene andere Dateien auch 'rotiert', was bei uns aber keine Rolle spielt. Der Befehl lautet: `# logrotate -f /etc/logrotate.d/rsyslog`

7.4 Kontrollfragen

Frage

Frage 7.1 Nach welchem Mechanismus wählt der Server die Cipher Suite für eine Verbindung aus den vom Client angebotenen Cipher Suites aus?

Frage

Frage 7.2 Wie sieht die Kommunikation zwischen Client und Server aus, wenn der Client Port 80 anfragt und auf Port 443 umgeleitet wird?

Frage

Frage 7.3 Was bewirken die Parameter `maxage=31536000; includeSubDomains` bei HSTS?

Frage

Frage 7.4 Welche Vorteile hat es aus Sicht der Sicherheit, wenn Sie die Logs mehrerer Systeme zentral sammeln können?

Frage

Frage 7.5 Sehen Sie Gefahren bei der Umsetzung eines Logservers, so wie er im Praktikum verwendet wurde?

Teil I

Anhang

Verweise

Vorlesungen

- Vorlesung Einführung Informatik Semester 1, Praktikum Einführung Informatik Semester 1,
- Vorlesung Betriebssysteme Semester 2, Praktikum Betriebssysteme Semester 2,
- Vorlesung Netzwerke Semester 3, Praktikum Netzwerke Semester 3,
- Vorlesung Netzwerk- und Systemsicherheit Semester 4,

jeweils mit den zugehörigen Dokumenten.

Befehle

Hier noch eine Liste von Befehlen (ohne etwaige Parameter) die Sie im Rahmen dieses Praktikums vielleicht verwenden werden:

- `man info`
- `cd pwd mkdir rmdir ln ls cp mv rm`
- `chown chgrp`
- `df du`
- `more less tail cat echo`
- `kill killall nohup bg fg`
- `ps top uptime date`
- `reboot shutdown halt`
- `su sux id w who last sudo`
- `ifconfig netstat lsof route host dig nslookup tcpdump ssldump nmap`
- `useradd userdel usermod groupadd groupdel groupmod passwd chsh chfn`
- `ping ssh scp tracepath wget`
- `vi pico gedit firefox thunar thunderbird`

- `logger, logrotate`
- `strace lstrace ldd`
- `apt-get apt-cache apt-key dpkg`
- `ssh-keygen md5sum sha256sum base64`
- `service update-rc.d`
- `iptables openssl openvpn`

Links

Folgende Links waren sehr hilfreich :

- https://help.ubuntu.com
- http://www.openssl.org/docs
- http://www.openssh.com/manual.html
- http://www.bind9.net/manuals
- http://www.netfilter.org/documentation/index.html
- http://openvpn.net/index.php/open-source.html