Verteilte Systeme und Kommunikationsnetze

Firewall mit IPTables

Praktikum 9

Fachhochschule Bielefeld
Campus Minden
Studiengang Informatik

Beteiligte Personen:

Name	Matrikelnummer
Michael Nickel	1120888
Jan Augstein	1119581

Aufgaben:

Aufgabe	Gelöst
Aufgabe 1	alles außer c) und g)

---

23. Januar 2018

Inhaltsverzeichnis

1 Aufgabe 1 - Firewall mit IPTables 3
1.1 Teilaufgabe a) . . . 3
Frage- bzw. Aufgabenstellung . . . 3
Lösung . . . 3
Ergebnis . . . 3
1.2 Teilaufgabe b) . . . 3
Frage- bzw. Aufgabenstellung . . . 3
Lösung . . . 3
Ergebnis . . . 4
1.3 Teilaufgabe d) . . . 4
Frage- bzw. Aufgabenstellung . . . 4
Lösung . . . 4
Ergebnis . . . 5
1.4 Teilaufgabe e) . . . 5
Frage- bzw. Aufgabenstellung . . . 5
Lösung . . . 5
Ergebnis . . . 6
1.5 Teilaufgabe f) . . . 6
Frage- bzw. Aufgabenstellung . . . 6
Lösung . . . 6
Ergebnis . . . 7
1.6 Teilaufgabe g) . . . 7
Frage- bzw. Aufgabenstellung . . . 7
Lösung . . . 7

2 Quellen 7

1 Aufgabe 1 - Firewall mit IPTables

1.1 Teilaufgabe a)

Frage- bzw. Aufgabenstellung

Welche Ports verwenden die Programme zur Kommunikation? Wie ändert man die Standardeinstellungen? Ist dazu ein Neustart des Systems nötig?

Lösung

Bei Linux Betriebssystemen kann man sich die verwendeten Ports mit dem Befehl *netstat -tulpn* anzeigen lassen. netstat ist ein häufig benutztes Diagnose-Werkzeug um verschiedene Informationen über die Netzwerkschnittstellen anzeigen zu lassen. Die Portnummer sieht man dann in der Spalte *Local Address* nach dem letzten Doppelpunkt. Dies sollte dann so aussehen:

```
michael@michael-VirtualBox:~$ sudo netstat -tulpn
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.1.1:53            0.0.0.0:*               LISTEN      864/dnsmasq
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      7881/cupsd
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN      7122/mysqld
tcp6       0      0 :::80                   :::*                    LISTEN      1442/apache2
tcp6       0      0 ::1:631                 :::*                    LISTEN      7881/cupsd
udp        0      0 0.0.0.0:5353            0.0.0.0:*                           697/avahi-daemon: r
udp        0      0 0.0.0.0:40446           0.0.0.0:*                           864/dnsmasq
udp        0      0 0.0.0.0:33310           0.0.0.0:*                           864/dnsmasq
udp        0      0 0.0.0.0:631             0.0.0.0:*                           7882/cups-browsed
udp        0      0 0.0.0.0:46895           0.0.0.0:*                           697/avahi-daemon: r
udp        0      0 127.0.1.1:53            0.0.0.0:*                           864/dnsmasq
udp        0      0 0.0.0.0:68              0.0.0.0:*                           838/dhclient
udp        0      0 0.0.0.0:68              0.0.0.0:*                           842/dhclient
udp6       0      0 :::5353                 :::*                                697/avahi-daemon: r
udp6       0      0 :::42854                :::*                                697/avahi-daemon: r
```

Abbildung 1: Portnummer anzeigen lassen

Ergebnis

Wir können die verschiedenen Ports der Programme auslesen.

1.2 Teilaufgabe b)

1, **2**

Frage- bzw. Aufgabenstellung

Richten Sie eine restriktive Firewall ein (prohibitive Sicherheitspolitik). Es soll sichergestellt werden, dass ausschließlich der Port des Webservers von außen zugänglich ist. Die Firewall soll automatisch beim Start des Systems geladen werden. Nutzen Sie für die Lösung der Aufgabe ausschließlich die Kommandozeile und das Kommando iptables.

Lösung

Da der Webserver den Port 80 (localhost) besitzt, müssen wir dafür sorgen, dass nur eingehende Pakete für diesen Port angenommen werden und sonst keine. Dafür benutzen wir den Befehl *iptables -A INPUT -i lo -j ACCEPT*. Das -A hängt eine Regel an die

Chain an, dass -i ist für die Netzwerkschnittstelle zuständig über die Pakete eingehen (in diesem Fall localhost) und das -j legt die Aktion fest. Anschließend benutzen wir noch den Befehl *iptables -A INPUT -j DROP* damit keine weiteren Pakete angenommen werden. Daher auch der Command *DROP*. Diese Befehle schreiben wir in eine neue Datei im init.d Ordner über den Command
touch /etc/init.d/firewall && chmod 755 /etc/init.d/firewall && nano /etc/init.d/firewall.
Damit das Skript bei jedem Neustart ausgeführt wird, schreiben wir folgenden Code in die *rc.local*-Datei:

```
1 if [ -e '/etc/init.d/firewall' ]
2 then
3     /bin/sh '/etc/init.d/firewall'
4 fi
```

Listing 1: rc.local

Ergebnis

Wir können mit *iptables* eine Firewall einstellen und sorgen dafür, dass die Firewall bei jedem Systemstart aktiv ist.

1.3 Teilaufgabe d)

3

Frage- bzw. Aufgabenstellung

Nun öffnen Sie mit netcat auf dem Rechner A (eine Ihrer VMs) ein oder zwei Ports zwischen 0 und 100. Nutzen Sie dafür den Befehl: nc -l [Portnummer].

nmap ist ein Portscanner, mit dem man u.a. die eigene Firewall auf Sicherheitslücken testen kann. Mit
nmap Zieladresse -p [Anfangsport]-[Endport]
kann nach offenen Ports in einem bestimmten Portbereich gescannt werden. Führen Sie nun einen Portscan auf VM A von VM B aus. Welches Ergebnis sehen Sie?

Lösung

Zunächst öffnen wir den Port 48 mit dem Befehl *nc -l 48*. Anschließend scannen wir den Port mit dem Befehl *nmap localhost -p 48*. Wie man in Abbildung 2 erkennen kann, bekommt man die Ausgabe dass der Port offen ist (STATE -> open). Wenn ich nun von einer anderen VM einen Portscan ausführe, wird auch hier erkannt dass der Port mit der Nummer 48 auf der VM A offen ist (siehe Abbildung 3).

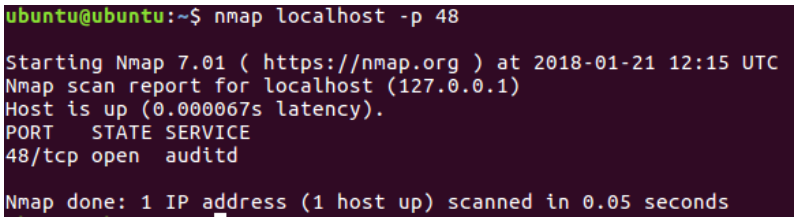

Abbildung 2: Portscan mit nmap

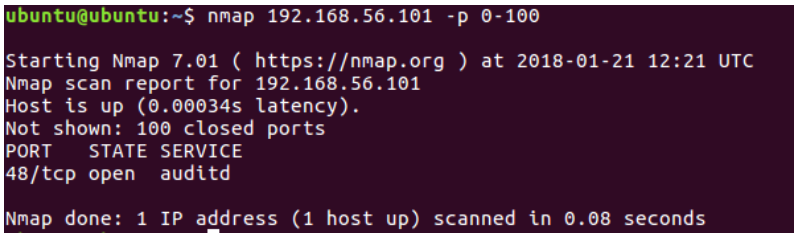

Abbildung 3: Portscan von anderer VM

Ergebnis

Wir können mit dem Befehl netcat verschiedene Ports öffnen und nmap anwenden um einen Portscan durchzuführen.

1.4 Teilaufgabe e)

4

Frage- bzw. Aufgabenstellung

Definieren Sie nun Regeln mit iptables, die einen solchen Scan blocken. Dazu bietet es sich an, alle Pakete zu verwerfen, die ungewöhnlich schnell Verbindungsanfragen senden.
Sehen Sie sich die folgende Regel an und versuchen Sie zu verstehen, welchen Sinn einzelne Befehle erfüllen. Sie werden sie im Verlauf dieser Übung noch benötigen. Wenden Sie diese Regel an. Die zweite Regel soll nun bei genau solchen Paketen in der gleichen Liste nachsehen, ob die Adresse eines Paketes bereits eingetragen ist. Trifft dies zu, soll die letzte Zeitmarke upgedatet werden. Wenn nun ein und dieselbe Adresse versucht 20 oder mehr Verbindungsanfragen innerhalb von 10 Sekunden zu senden, sollen all diese Pakete verworfen werden. Wie sieht die entsprechenede Regel aus? Testen Sie die Funktionalität der Regeln mit nmap.

Lösung

Die Regel *iptables -A INPUT -p tcp -i eth1 -m state –state NEW -m recent –set* sagt zunächst am Anfang, dass es sich um eine Regel handelt die eine neue Chain vom Typ INPUT anhängt (-A INPUT). Des weiteren wird mit dem Befehl *-p* gesagt, dass nur Pakete vom Protokoll tcp überprüft werden und die *-i* Option sagt aus, dass nur Pakete

von der Netzwerkschnittstelle *eth1* überprüft werden. Weiterhin wird noch gesagt, dass nur Pakete mit dem Status *NEW* überprüft werden. *recent* erlaubt es dynamisch Listen mit IP-Adressen zu erstellen und dann an Hand der Listen Pakete zu filtern. Die zweite Regel würde dann so aussehen:

```
1 iptables -A INPUT -p tcp -i eth1 -m state --state ESTABLISHED -m recent
    --update --seconds 10 --hitcount 20 -j REJECT --reject-with tcp-reset
```

Listing 2: iptables mit "recent"

Mit dem Befehl *–update* fügen wir einen weiteren Zeitstempel ein, welcher dann mit *–seconds* überprüft ob in den letzten 10 Sekunden 20 weitere Zeitstempel vorliegen (*–hitcount*). Wenn das der Fall ist, wird die Verbindung zurückgesetzt (*-j REJECT –reject-with tcp-reset*.

Ergebnis

Wir können Regeln mit iptables aufstellen, die dafür sorgen, dass gewisse Scans blockiert werden.

1.5 Teilaufgabe f)

5

Frage- bzw. Aufgabenstellung

Nun sollen die Portscans mithilfe der LOG-Funktion protokolliert werden. Dazu kann zuerst eine neue Kette für Portscans mit Namen PORTSCAN o.ä. erstellt werden. Als zweites muss eine Regel definiert werden, die alle Pakete in dieser Kette protokolliert. Als Log-Prefix soll "PORTSCAN erkannt –"verwendet werden. iptables loggt i.d.R. in der Datei /var/log/syslog. Kontrollieren Sie, ob Ihre Ausgabe in der Logdatei erscheint.

Lösung

Zunächst erstellen wir eine neue Kette mit dem Namen PORTSCAN (siehe Zeile 1). Anschließend definieren wir eine Regel damit alle Pakete dieser Kette protokolliert werden (siehe Zeile 2).

```
1 sudo iptables -N PORTSCAN
2 sudo iptables -A PORTSCAN -j LOG --log-prefix "PORTSCAN erkannt --"
```

Listing 3: Portscan protokollieren

Danach erstellen wir eine einfache Regel fest, die die eingehende Kommunikation für localhost zulässt. Als Aktion nehmen wir diesmal aber *PORTSCAN* (siehe Zeile 1).

```
1 sudo iptables -A INPUT -i lo -j PORTSCAN
```

Listing 4: Regel fÃ¼r PORTSCAN

Wenn jetzt mit nmap ein Portscan durchgeführt wird, kann man in der *syslog*-Datei sehen, dass unsere Ausgabe dort erscheint (Abbildung 4).

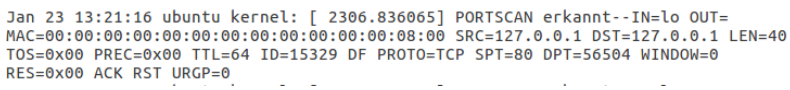

Abbildung 4: Ausgabe in syslog

Ergebnis

Wir können unsere Portscans mit der LOG-Funktion protokollieren.

1.6 Teilaufgabe g)

Frage- bzw. Aufgabenstellung

Hping3 ist ein Tool, um die eigene Firewall zu testen. Es kann benutzerdefinierte TCP-Pakete generieren und das auch mit langsamerer Geschwindigkeit als nmap. Schicken Sie einer VM mehrere TCP-Pakete mit gesetzter SYN-Flag beginnend mit Port 0.

Welchen Befehl verwenden Sie und woran erkennt man, welche Ports offen sind?

Lösung

Wir konnten diese Aufgabe nicht bearbeiten, da wir das Paket hping3 nicht installieren konnten.

2 Quellen

Literatur

[1] damadmai, iptables2
`https://wiki.ubuntuusers.de/iptables2/`, 23.01.2018

[2] Markus Mansshardt, iptables Firewall-Script für Debian
`http://blog.mansshardt.net/iptables-firewall-script-fuer-debian/`, 23.01.2018

[3] Heinrich Schwietering, netcat
`https://wiki.ubuntuusers.de/netcat/`, 23.01.2018

[4] heise.de, SSH vor Brute-Force-Angriffen schützen
`https://www.heise.de/security/artikel/recent-271032.html`, 23.01.2018

[5] FK, Portscans mit iptables erkennen und protokollieren.
`https://www.kworx.de/portscans-iptables-protokollieren/`, 23.01.2018