Betriebsysteme und Netzwerke Übung

Aufgabe: Wie funktionieren Firewalls?

Datum: 10.06.2020

Autoren: Istvan Galfi, Oliver Dragschitz

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Einleitung

Heutzutage sind Firewalls ein fester Bestandteil aller Betriebsysteme und sind meist im Betriebsystem Kernel intergriert (Linux IP-Tables) oder werden in einer vorinstallierten Variante angeboten(Windows Defender Firewall).
Rein Grundsätzlich überwacht die Firewall den Datenverkehr eines Systems und entscheidet anhand festgelegter Regeln ob Netzwerkpakete durchgelassen werden oder nicht.[7]

Firewalls werden dabei in zwei Arten unterschieden: In Personal Firewall sowie in externe Firewall. Ersteres entspricht beispielsweiese der Desktop Firewall eines PC. Ein Router, aus relativer Betrachtweise, findet hingegen zweiterem Zuordnung[7]

Zielgerichtet ausgedrückt, ist die Firewall ein Teilgebiet, welches der IT-Sicherheit zuzuordnen ist.

1.1 Firewall Grundregel

Elementar ausgedrückt liegt die Grundregel einer Firewall darin: „Alle Verbindungen unterdrücken und nur jene zu erlauben die gewollt sind.“Diese Art der Ansichtsweise steht jener gegenüner die besagt: Explizit Schadquellen zu blockieren. Letzteres bringt den Nachteil mit sich, in einem kontinuirlichen Vorgang immer neu endeckte Schadquellen ausdrücklich zu verbieten.

2 Wie Funktionieren Firewalls?[1][2]

Die Filterung von Netzwerkpaketen kann aus Basis einzelner oder komposite Filterungen von IP-Adressen, Ports oder Netzwerkdomainen(Privat, öffentlich) geschehen.

2.1 Arten in mehr Detail

1. Packet filtering firewall
    - Diese Methode wurde von den ersten Firewalls verwendet. Hier werden die Parameter der Packete(Ursprung Adresse, Port) mit erlaubten und verbotenen Parameter verglichen. Dies kann einfach durch IP-Spoofing umgegangen werden - Veränderung des Ursprunges der Addresse des Packetes.
2. Circuit-level gateway
    - Hier werden die Session Etablierungsnachrichten auf ihre Gültigkeit überprüft. TCP Handshake zum Beispiel.
3. Stateful inspection firewall
    - Diese Methode gehört zur zweiten Generation von Firewalls und ist grundsätzliche eine Kombination von Packet Filtering und Circuit-Level Gateway. Diese Methode ist sehr sicher, aber besitzt einen erheblichen Einfluss an die Leistung.
4. Application-level gateway
    - Diese Methode benutzt einen so genannten Proxy(der kann auch allein als firewall fungieren), zwischen Einheit, kommuniziert wird hier durch diesen, dadurch bleibt die Addresse des Benutzers unbekannt. Außerdem verwendet diese Methode Elemente des Packetfilterings sowie der Circuit-Level gateway. Erweitert aber diese durch die Filterung von Applikation spezifischen Packete. Außerdem wird die Leistung nicht erheblich beeinträchtigt.
5. Next-generation firewall (NGFW)
    - Beinhaltet alles bereits erwähnte, nur in einem leistungschonendem Format. Weiters ergänzt es mit einer Deep Packet Inspection, in welcher die Packete auch validiert werden. Z.B. das Zusammenfügen einer HTTP Anfrage und Prüfen des Formats.

2.2 Firewalls unter Windows

Windows kommt seit Windows XP und Windows Server 2003 mit einem eingebauten Firewall. Vor Windows XP Service Pack 2, hieß dieser Internet Connection Firewall, danach nur Windows Firewall und seit Windows 10 Version 1709 heißt es Windows Defender Firewall. Dieses Dokument konzentriert sich jedoch auf die Windows 10 Version post Version 1709.

2.3 Firewalls unter Linux [4]

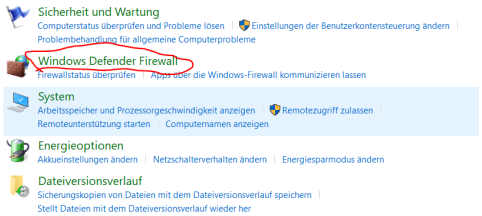
Moderne Linux Firewall Lösungen bedienen sich heutzutage über einen eingebauten Netfilter welcher Bestandteil des Kernels ist. Jeder eingehende Netzwerk-Traffic wird daher von dem Nefilter System überprüft und bei Notwendigkeit manipuliert.
Die Gesamtheit der Netfilter stellen daher die Firewall dar.
Die Konfiguration des Firewall (der Netfilter) wird dabei durch eine Linux-Komponente gewährleistet, bekannt als „iptables“. Auf iptables wird dazu in Kapitel 4.1 noch näher eingegangen.

3 Einstellung einer Windows Firewall

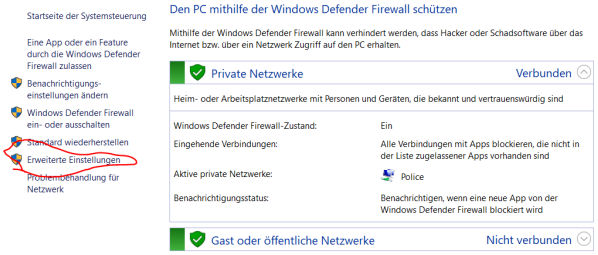
Unter Windows ist es möglich, sowohl mithilfe der Konsole oder durch die Verwendung der graphischen Benutzeroberfläche die Firewall einzustellen.

3.1 Durch die Verwendung der graphischen Benutzeroberfläche

(a) Systemsteuerung



(b) System und Sicherheit



(c) Windows Defender Firewall

Abbildung 1: Wegbeschreibung

Unter Windows 10 ist es möglich durch die Key-Kombination Strg+R und das Eingeben von 'control' die Systemsteuerung zu öffnen, hier können die Firewall-Einstellungen durch das Klicken des Menüpunkts 'System und Sicherheit' (1a) und 'Windows Defender Firewall' (1b) erreicht werden, hier ist es möglich durch das Menüpunkt 'Erweiterte Einstellungen' (1c) den Window 'Windows Defender Firewall erweiterte Sicherheit' zu öffnen. Hier ist es möglich neue Regeln für ein- und ausgehende Pakete zu erstellen, bestehende zu ändern, sowie Verbindungssicherheitsregeln zu erstellen und zu ändern.
Die Regeln für ein- und ausgehende Pakete sind zuständig für die Regulierung des Eingang und Ausgang von TCP- und UDP-Pakete. Durch sie ist möglich sehr spezifische Regeln zu erstellen, die nur unter bestimmte umstände den Eingang bzw. Ausgang eines Pakets eraluben.
Beispiel:

- Das Programm Firefox sollte eingehende Pakte nur auf Ports: 443 und 80 Empfangen können und nur dann wenn der Computer auf einem Privaten Netzwerk gebunden ist.

3.2 Mit der Command Line

In der Command Line ist es möglich dieselbe Dinge durch die Verwendung des Networkshells einzustellen, was durch das Benutzen des 'netsh' Befehls möglich ist. Die Einstellungen des Firewalls können mit Hilfe des "netsh advfirewall" Befehls, früher "netsh firewall", geändert werden, wobei außerdem geänderten Befehlen Syntax, "netsh firewallërlaubt die spezifizieren von Profielen, wie Private, Domain und Public nicht.
Hier wird auf die neuere Variante konzentriert, also hier sind ein paar nützliche Befehle:

- netsh advfirewall reset ⇒ Stellt die standard Richtlinien zurück
- netsh advfirewall firewall add rule name="Firefox" dir=in action=allow program="Pfad\firefox.exe" enable=yes ⇒ Dieses Befehl erlaubt das Programm "Firefox" durch den Firewall zu kommunizieren.
- netsh advfirewall firewall add rule name="Open Port 80" dir=in action=allow protocol=TCP localport=80 ⇒ Öffnet den Port 80 für TCP Packete.

Der Commandline bietet die Funktion an, die Liste von möglichen Befehlen mit dem "?" symbol am Ende ab zu rufen ⇒ netsh advfirewall add rule ? ⇒ ruft alle Mögliche Befehle mit deren Hilfe einen Gesetz zum Firewall zugefügt werden kann.
Anmerkung: Administrator Rechte sind benötigt!
Für die Quelle sehe Kapitel 5.2.

4 Linux Firewall

Als Herzeige - Distributionen für folgende gezeigten Firewall Beispiele wurde die Router Software RouterOS (MikroTik) wie auch die linux Distribution ubuntu gewählt.

4.1 iptables

Wie bereits in 2.3 erwähnt wird die Software iptables dazu verwendet um die Firewall zu modifizieren.

"Die Grundidee von iptables: Listen von Regeln, wovon jede angibt, was in einem Paket überprüft wird und was dann mit diesem Paket geschehen soll.„[5] Die Konfiguration wird dabei als Tabelle dargestellt, in der Regeln und Ketten hinzugefügt, verändert und gelöscht werden können.

		#	Action	▼ Chain	Src. Address	Dst. Address	Prot...	Src. Port	Dst. Port	Any. Port	In. Interf...	Out. Interf...
- D		17	✔ accept	output								

Abbildung 2: iptables Regel

In Abbildung 2 ist exemplarisch eine solche Firewall-Regel zu sehen. Das zusehende Bild stammt dabei aus dem GUI eines Routers welcher ebenfalls iptables implementiert.
Die zu sehende Regel bestimmt dabei, dass alle (da keine Source Adresse angegeben) ausgehenden Verbindungen (chain=output sprich aus dem System) erlaubt sind.
Bei iptables werden die Regeln nach der Reihenfolge abgearbeitet, weshalb diese zu beachten ist.
Würde man beispielsweise eine solche Regel an oberster Stelle der Liste hinufügen und action=drop und chain=input setzen würde dass bedeuten das alle eingehenden pakete sofort fallen gelassen werden. Dies würde bedeuten dass keine Netzwerkkommunikation mit dem System mehr möglich ist. Das wäre beispielsweise bei einem Router äußerst Fatal da das ganze Netzwerk still gelegt werden würde und nicht einmal der Zugriff auf den Router selbst mehr möglich wäre, um die Regel wieder zu verändern. In diesem Fall hilft nur mehr ein "Hard-Reset" weshalb der Umgang mit der Firewall bei auf Netzwerk angewiesenen Systemen sehr vorsichtig geübt werden sollte.

4.1.1 Tabellen[6]

Firewall-Regeln werden nach ihrem Anwendungsfall in Tabellen zusammengefasst. Es gibt dabei vier verschiede Tabellen, die je Ihre eigene Aufgabe besitzen.

- filter - enthält Filter-Regeln
- nat - Addressumsetzung und Weiterleitung. Diese Regeln beinhaltet z.B. die wichtige "masquerade" Action, die die Funktion besitzt IP Adressen durch eine andere zu ersetzen. Dies ist besonders bei Routern sehr wichtig, um Pakete unter der public IP in die exterene Umwelt (Internet) zu routen.
- mangle - Für Paketmanipulationen
- raw - Ausnahmen für connection tracking.

4.1.2 Arten von Ketten[6]

Tabellen enthalten sog. chains (Ketten). Diese regeln fest wann ein paket geprüft wird. Dabei gibt es fünf Regeln wie folgt:

- Prerouting - Auf Pakete angewendet bevor diese geroutet werden
- Input- Auf Pakete die für einen lokalen Prozess bestimmt sind
- Forward - Auf alle Pakete die geroutet werden
- Output - Auf Pakete die von einem lokalen Prozess stammen
- Postrouting - Auf Pakete angewendet nachdem diese geroutet werden

4.2 Uncomplicated Firewall - ufw[4]

Das Haupt-Tool von ubuntu zur Konfiguration von der Firewall ist ufw, gennant Uncomplicated Firewall. Das Kürzel ufw stellt sinngemäß auch gleich den Bash Command dar.
Hier sei gesagt, dass ufw durch seine Einfachheit und nicht durch seine Komplexität/Vollständigkeit überzeugt (kongruent zu den Command „iptables"), weshalb ufw meist auch nur im Bereich der Host-Based Firewall Anwendung findet, wie auch die man page bestätigt (Command: „man ufw").

4.3 Konfiguration einer Regel

Folgendes Beispiel zeigt exemplarisch die Konfiguration von einer Filter Firewall Regel welche der Filter Tabelle zuzuordnen ist.
Standartmäßig ist die Firewall bei Linux deaktiviert. Die Aktivierung erfolgt mittels dem Befehl: “ufw enable„. Ports können mit dem Befehl “ufw allow port 22” (für port 22 ssh) geöffnet werden. Mit dem Befehl “ufw deny port 22” wird die vorhin erstellte Regel wieder deaktiviert. Mit dem Keyword “delete” z.B. “ufw delete deny port 22” wird die Regel hingegen komplett gelöscht.

```
root@ubuntuserver1:~# ufw status
Status: inactive
root@ubuntuserver1:~# ufw allow proto tcp from 192.168.88.254 to any port 22
Rules updated
root@ubuntuserver1:~# ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
root@ubuntuserver1:~# reboot
```

```
root@ubuntuserver1:~# ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To                         Action      From
--                         ------      ----
22/tcp                     ALLOW IN    192.168.88.254
```

Abbildung 3: Firewall Aktivierung

Mit dem Befehl (siehe 3) “ufw allow proto tcp from 192.168.88.254 to any port 22” wird der ssh port geöffnet und nur Verbindungen von der angegebenen IPv4 Addresse mittels tcp erlaubt.
Dies stellt eine etwas sichere und elegantere SSH Portfreigabe dar.

5 Quellenverzeichnis

5.1 Funktionsweise von Firewalls

[1] https://searchsecurity.techtarget.com/feature/The-five-different-types-of-firewalls
Zugriffszeit: 05.06.2020, Vgl.

[2] https://cs.stanford.edu/people/ashgup/Reports/network.pdf
Zugriffszeit: 05.06.2020, Vgl.

[4] https://ubuntu.com/server/docs/security-firewall
Zugriffszeit: 05.06.2020, Vgl.

[7] https://de.wikipedia.org/wiki/Firewall
Zugriffszeit: 06.06.2020, Vgl.

5.2 Windows-Commandline-Firewall

[3] https://support.microsoft.com/en-us/help/947709/how-to-use-the-netsh-advfirewall-firewall-context-instead-of-the-netsh
Zugriffszeit: 21.05.2020, Vgl.

5.3 Linux

[5] https://de.wikipedia.org/wiki/Iptables
Zugriffszeit: 05.06.2020, Direkter Zitat

[6] https://wiki.ubuntuusers.de/iptables2/
Zugriffszeit: 06.06.2020, Vgl.