Windows unter Kontrolle – IPFire Teil4

1. Kein Windows ohne Firewall IPFire Windows

Windows (10) ist ein Datenschutzsuper-GAU – das werden die meisten wissen. Aufgrund mangelnder Alternativen werden viele dennoch weiterhin **notgedrungen** auf Microsoft setzen und versuchen, dem Kontrollverlust irgendwie entgegenzuwirken.

Das Problem kurz zusammengefasst: Selbst wenn man im Privat- oder Unternehmensumfeld auf Linux umgestiegen ist, so gibt es immer Anwendungen, die ausschließlich für Microsoft (Windows) konzipiert sind. Dazu gehören bspw. Anwendungen zur Steuererklärung oder Buchhaltungssoftware – aufgrund mangelnder (guter) Alternativen für Linux benutzen selbst eingeschworene Linux-Fans und Open-Source Anhänger noch immer Systeme mit Windows. Nicht weil sie es wollen, sondern weil sie es müssen.

Im vierten und letzten Teil der IPFire-Artikelserie möchte ich aufzeigen, wie ihr Level 3 (siehe Ziffer 6) erreichen könnt und den **Datenverkehr** von Windows auf der IPFire kontrolliert und einschränkt. Am Beispiel von einer Steuer- und Buchhaltungssoftware werde ich euch demonstrieren, wie ihr nur jene Verbindungen freigebt, die für das Funktionieren eurer Software erforderlich sind.

DIESER BEITRAG IST TEIL EINER ARTIKELSERIE:

- Hardware und Netzwerkaufbau – IPFire Teil1
- DNS-Adblocker Skript für IPFire – IPFire Teil2
- ASN-Skript: Datensammler haben ausgeschnüffelt – IPFire Teil3
- Windows unter Kontrolle – IPFire Teil4

2. Ausgangslage

Für echte Kontrollfreaks gibt es nur eine Lösung, wenn sie auf Anwendungen angewiesen sind, die unter Windows laufen und gleichzeitig nicht wollen, dass sie von Microsoft ausspioniert werden: Die Filterung des Netzwerkverkehrs mittels einer externen Firewall, die **ausgehende** Verbindungen von einem Windows-Rechner reglementiert.

Das im vorliegenden Beitrag dargestellte Szenario eignet sich für all jene, die Windows nicht als Hauptsystem nutzen, sondern lediglich als Werkzeug, um mit Anwendungen zu arbeiten, die ausschließlich unter Windows lauffähig sind. Das System soll später so stark durch die IPFire reglementiert sein, dass nur jene Verbindungen »nach draußen« erlaubt sind, die für das Funktionieren eurer Software erforderlich sind. Weitere Einsatzszenarien, wie Surfen, E-Mails schreiben / lesen oder Videostreaming sind **ausgeschlossen** bzw. aufgrund der strikten Beschränkung des Netzwerkverkehrs durch die IPFire nicht vorgesehen.

2.1 Windows Updates nur manuell

Das **regelmäßige** Einspielen von (System-) Updates stellt eine sehr wichtige Maßnahme gegen bekannte bzw. bekanntgewordene Sicherheitslücken dar. Im Grunde bedeutet das: Sowohl das

Betriebssystem, als auch **alle** auf dem System installierten Anwendungen, Plug-Ins, Browser-Erweiterungen etc. sollten stets auf dem aktuellen Stand sein und gehalten werden. Es gilt daher die einfache Regel:

Wenn du etwas installiert hast, dann update es auch.

Für unser Projekt machen wir keine Ausnahme, werden Software- und Windows-Updates allerdings **manuell** einspielen. Unser Windows-System wird zu keinem Zeitpunkt Verbindung zu Microsoft aufnehmen können / dürfen.

Über sogenannte »Update Packs« kann man Windows aktuell halten. Das Einspielen von Updates via Update Pack ist zwar weniger zeitnah, als die in Windows integrierte Update-Funktion, für unser Vorhaben allerdings vertretbar, da unser Nutzungsprofil eine relativ geringe Angriffsfläche bietet.

Die Update Packs für Windows 7 und 8 werden unter anderem von DrWindows (erster angepinnter Thread), WinFuture oder Chip angeboten und monatlich aktualisiert. Für Windows 10 bekommt man die Update Packs bei DrWindows (erster angepinnter Thread) und Chip. Nach Angaben der Betreiber sind alle (relevanten) Updates enthalten und dennoch bleibt ein mulmiges Gefühl, von einem Drittanbieter (Betriebssystem-)Updates zu beziehen. Letztendlich müsst ihr den Anbietern **vertrauen** oder eben eure Windows-Maschinen in der IPFire soweit öffnen, dass zumindest Updates via Microsoft möglich sind.

Alternativ stellt ihr euch einen Windows Server Update Service (WSUS) in euer Netzwerk oder nutzt das WSUS Offline Update Tool der c't bzw. Torsten Wittrock. Beide Varianten sind den Update Packs vorzuziehen, erfordern allerdings etwas mehr Aufwand. Das Positive: Auch hierbei kommuniziert euer System nicht direkt mit Microsoft.

Hinweis

Windows lässt sich übrigens via Telefon aktivieren. Es ist nicht zwingend eine Internetverbindung erforderlich:

- Aktivieren von Windows 7 oder Windows 8.1
- Aktivieren von Windows 10 per Telefon

2.2 Software-Updates

Nicht nur Windows, sondern auch unsere Steuer- und Buchhaltungssoftware wollen wir natürlich aktuell halten. Da heutige Software meist einen Update-Check inklusive Download und Installationsroutine integriert hat, können wir die dafür notwendigen IP-Adressen der Gegenstelle beim Anbieter einfach in der IPFire freischalten. Alternativ könnt ihr hin und wieder auch selbst direkt beim Hersteller nach Updates suchen und diese gegebenenfalls einspielen.

3. IPFire Vorbereitung

Euer Windows-System wird vermutlich imblauen (WLAN) oder grünen (LAN) Netzbereich der IPFire

untergebracht sein. Zur Erinnerung nochmal mein eigener Aufbau:

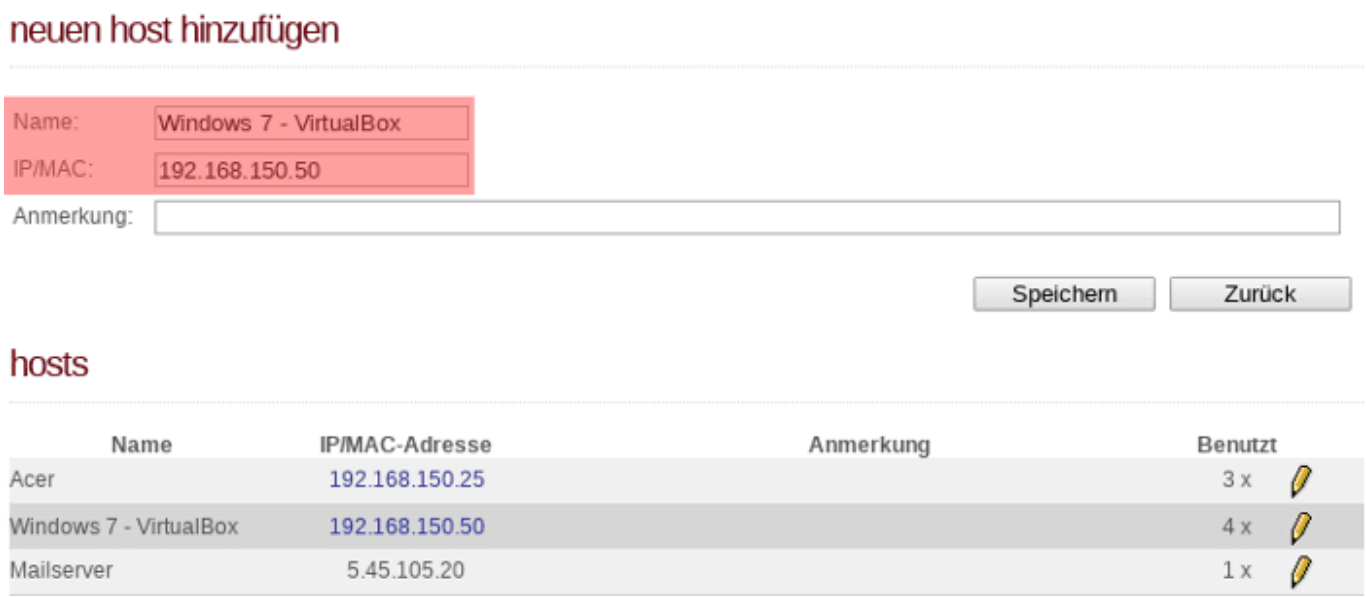
- **Rot**: Externes Netzwerk, dass typischerweise direkt mit dem Internet bzw. ISP verbunden ist. Bei mir ist das rote Netzwerk der interne IP-Adressbereich der FRITZ!Box (192.168.50.0/24).
- **Grün**: Das interne private Netzwerk (192.168.200.0/24), bei dem Geräte mit einem Netzwerkkabel verbunden sind.
- **Blau**: Ein separates Netzwerk (192.168.150.0/24) für Wireless Geräte wie bspw. Smartphones.
- **Orange**: Nicht vorhanden. Typischerweise die DMZ, für Geräte, die direkt aus dem Internet erreichbar sein sollen.

Weitere Details zum Aufbau könnt ihr dem ersten Beitrag der Artikelserie entnehmen.

3.1 Windows-System mit eigener IP-Adresse

Persönlich betreibe ich mein Windows 7 in einer VirtualBox über eine Netzwerkbrücke (Bridged Networking) und fester IP-Adresse. Den Netzwerkmodus »Netzwerkbrücke« habe ich natürlich bewusst gewählt, damit ich dem Windows-System eine eigene IP-Adresse in der IPFire zuweisen kann – das würde beim NAT-Modus nicht funktionieren, da VirtualBox die IP-Adresse vom Hostsystem verwendet.

Ihr solltet eurem Windows-System -egal ob das System virtualisiert ist oder nicht- also eine feste bzw. **eigene IP-Adresse** zuweisen. Legt dazu am besten auf der IPFire unter »Firewall → Firewallgruppen → Hosts« einen neuen Eintrag an:



4. IP-Adressen bzw. Gegenstelle identifizieren

Wie bereits erwähnt erhält Windows selbst keinerlei Zugriff »nach draußen« bzw. auf das Internet, sondern ausschließlich **ausgewählte** Software, die dies auch tatsächlich benötigt. Hierfür ist es notwendig, dass ihr die Server bzw. Gegenstellen identifiziert, die eure Software zur Funktionserbringung benötigt. In meinem Fall ist das bspw. das Finanzamt Karlsruhe, an das ich meine monatliche Umsatzsteuervoranmeldung (Buchhaltung) und die jährliche

Einkommensteuererklärung (Steuern) übermittle.

Bevor wir die IP-Adressen identifizieren, möchte ich euch noch kurz das Whitelist-Prinzip vorstellen – einem **empfehlenswerten** Ansatz für die Firewall-Konfiguration bzw. der Regelsätze:

Verboten ist alles, was nicht ausdrücklich erlaubt ist.

Es werden also nur jene Verbindungen nach außen zugelassen, die explizit **erlaubt** sind. Ein solches Regelwerk zu definieren ist je nach Anwendung und Anzahl der installierten Programme, die mit der Außenwelt kommunizieren dürfen, ein langwieriger Prozess. Für unser Vorhaben ist das allerdings durchaus machbar, denn wir reduzieren uns auf jene Software, die wir notgedrungen unter Windows einsetzen müssen – im vorliegenden Beispiel also auf eine Steuer- und Buchhaltungssoftware.

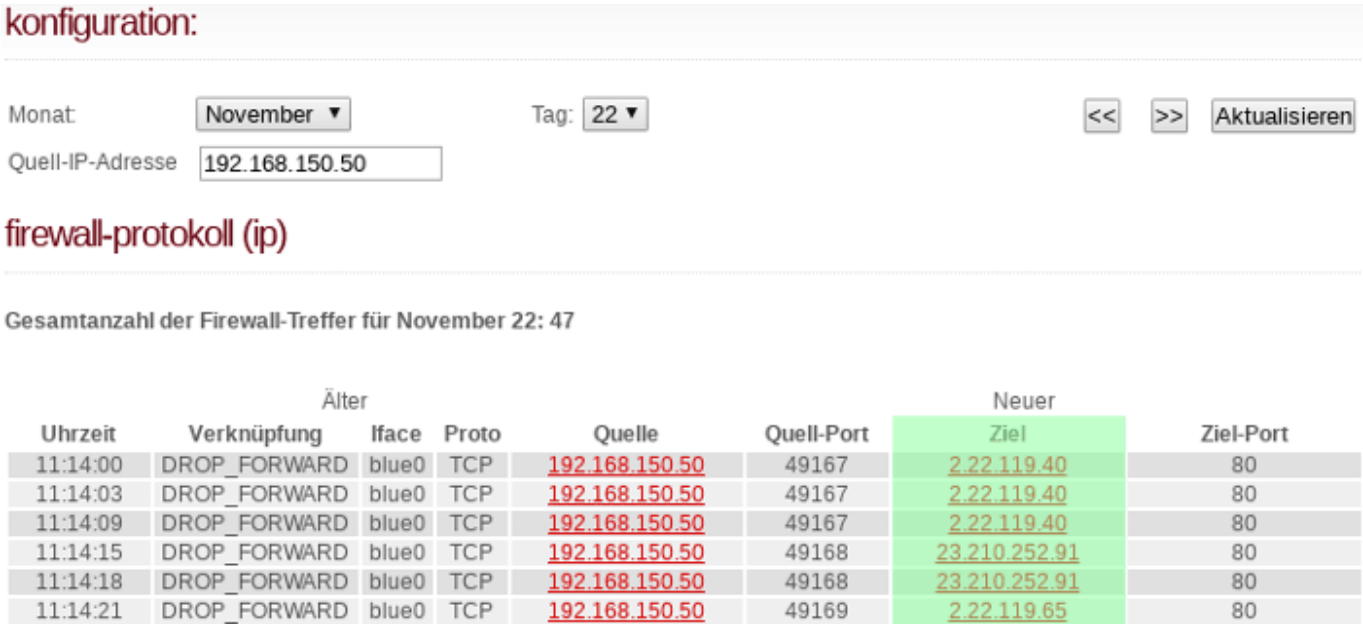
4.1 Logging in der IPFire

Die IPFire bietet eine Logging-Funktionalität, mit der ihr grundsätzlich in der Lage seid, die Gegenstellen bzw. Server für eure Software zu identifizieren. Grundsätzlich bedeutet an dieser Stelle, dass dies nach meiner Auffassung nicht unbedingt der ideale Weg ist. Ich werde euch dennoch kurz demonstrieren, wie ihr für ein System die aus- und eingehenden Verbindungen direkt auf der IPFire einsehen könnt.

Klickt auf »Protokolle → Fw-Protokolldiagramme (IP)«. Dort bekommt ihr neben einem Diagramm auch folgende Auflistung angezeigt:

	IP-Adresse	Land	Anzahl	Prozent
Mehr	192.168.150.50	?	53	33
Mehr	192.168.150.15	?	45	28
Mehr	192.168.150.30	?	35	22
Mehr	192.168.150.10	?	25	15
Mehr	192.168.50.1	?	4	2
	Andere IP		0	0

Die IP-Adresse des Windows-Systems (192.168.150.50) taucht dort ebenfalls auf. Klickt auf »Mehr« und ihr erhaltet folgende Ansicht:


konfiguration:

Monat: November ▼ Tag: 22 ▼ << >> Aktualisieren

Quell-IP-Adresse 192.168.150.50

firewall-protokoll (ip)

Gesamtanzahl der Firewall-Treffer für November 22: 47

	Älter					Neuer	
Uhrzeit	Verknüpfung	Iface	Proto	Quelle	Quell-Port	Ziel	Ziel-Port
11:14:00	DROP_FORWARD	blue0	TCP	192.168.150.50	49167	2.22.119.40	80
11:14:03	DROP_FORWARD	blue0	TCP	192.168.150.50	49167	2.22.119.40	80
11:14:09	DROP_FORWARD	blue0	TCP	192.168.150.50	49167	2.22.119.40	80
11:14:15	DROP_FORWARD	blue0	TCP	192.168.150.50	49168	23.210.252.91	80
11:14:18	DROP_FORWARD	blue0	TCP	192.168.150.50	49168	23.210.252.91	80
11:14:21	DROP_FORWARD	blue0	TCP	192.168.150.50	49169	2.22.119.65	80

Dort werden euch einige Informationen angezeigt. Unter anderem auch das Ziel, wohin das Windows-System eine Verbindung versucht aufzubauen. Da wir mit dem Whitelist-Prinzip arbeiten, ist der Verbindungsaufbau allerdings nicht erfolgreich, sondern die Anfrage wird von der IPFire (DROP_FORWARD) verworfen.

Das Problem ist allerdings, wir können die IP-Adressen bzw. Gegenstelle nicht ohne größeren Aufwand einer Anwendung zuordnen. Daher ist dieser Ansatz für unsere Zwecke eher nicht von Vorteil.

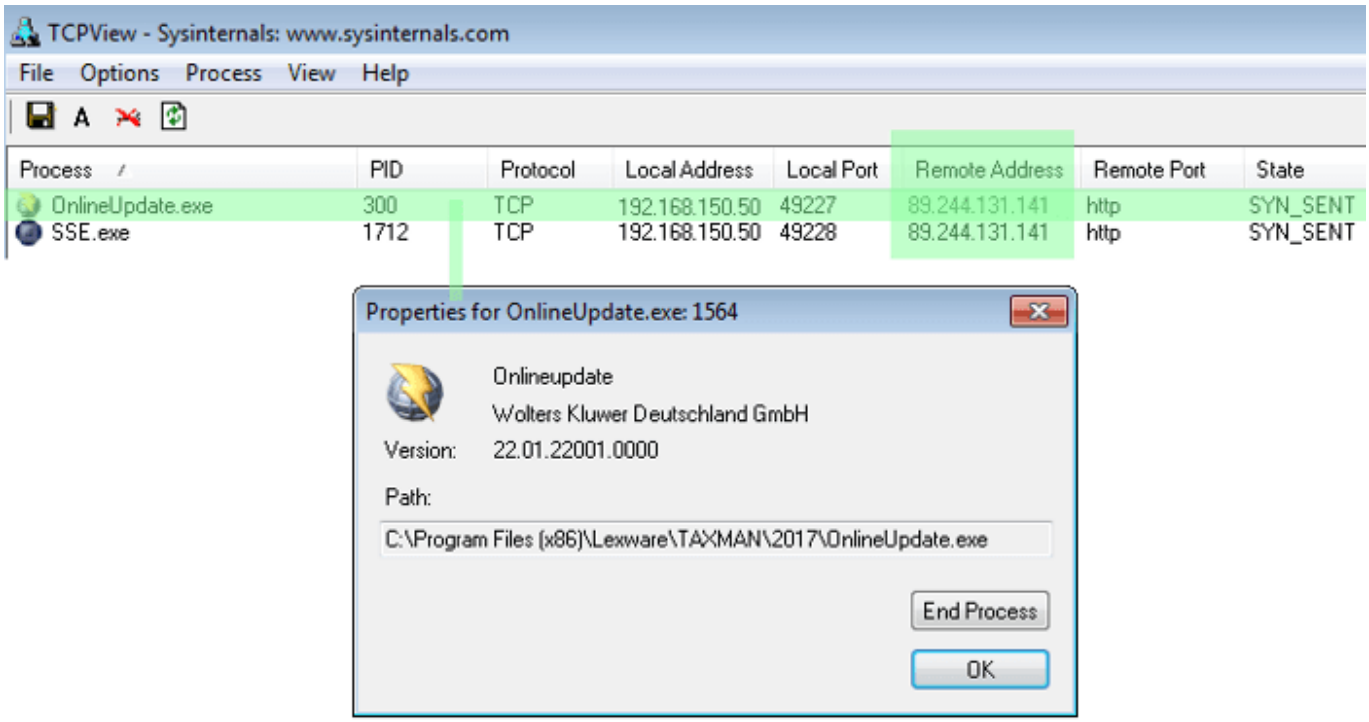
4.2 TCPView

Für Windows stellt Microsoft ein kleines Tools (TCPView) bereit, mit dem man eine detaillierte Auflistung aller TCP- und UDP-Endpunkte auf dem System erhält. Wir nutzen TCPView, um herauszufinden, mit welchen **Endpunkten** bzw. Servern unsere Software kommunizieren möchte. Die IP-Adressen können wir anschließend für unser Windows-System auf der IPFire freigeben und dadurch die Kommunikation erlauben.

Zunächst starten wir TCPView und entfernen unter »Options« die Häkchen bei:

- Show Unconnected Endpoints
- Resolve Addresses

Anschließend wird die Steuersoftware gestartet. TCPView wird kurz nach dem Start die ersten **Verbindungsversuche** der Anwendung registrieren und darstellen:



Mit einem Rechtsklick auf »OnlineUpdate.exe« und einem weiteren Klick auf »Process Properties« können wir **sicherstellen**, dass der Verbindungsversuch tatsächlich von der eben gestarteten Anwendung stammt.

Mit diesem Vorgehen können wir die IP-Adressen der Gegenstellen ermitteln und die Adressen anschließend als Host-Objekte in der Firewall anlegen oder diese in einer Gruppe zusammenfassen.

Anschließend müssen wir den Vorgang wiederholen, denn die Freigabe einer IP-Adresse bedeutet nicht, dass der Update-Prozess anschließend reibungslos abläuft. Im Falle der Steuersoftware erfolgt die Prüfung auf eine neue Version über die IP-Adresse 89.244.131.141 – der Download von Updates allerdings über eine andere IP-Adresse, nämlich der 212.211.139.206.

Hinweis

Der Unterschied zum Logging auf der IPFire besteht insbesondere darin, dass wir mit TCPView auf Anwendungsebene sehen, zu welchem Ziel eine Verbindung aufgebaut wird. Das erleichtert eine Zuweisung zu einer IP-Adresse ungemein und ist daher dem Logging-Verfahren auf der IPFire vorzuziehen – jedenfalls bei unserem Anwendungsfall.

Hilf mit die Spendenziele zu erreichen!
Mitmachen ➡

4.3 IPFire Konfiguration

Öffnet erneut die Hosts-Ansicht über »Firewall → Firewallgruppen → Hosts«. Dort legt ihr dann Hosts-Objekte für alle identifizierten IP-Adressen an, zu denen eure Software eine Verbindung initiieren darf:

hosts

Name	IP/MAC-Adresse	Anmerkung	Benutzt
Acer	192.168.150.25		3 x
LibreELEC	192.168.200.105		2 x
Mailserver	5.45.105.20		1 x
Windows 7 - VirtualBox	192.168.150.50		4 x
Windows7 - Finanzamt Karlsruhe01	62.157.211.59		1 x
Windows7 - Finanzamt Karlsruhe02	80.157.84.22		1 x
Windows7 - MonkeyOffice - Update Server	178.250.10.46		1 x
Windows7 - TAXMAN - Update Server 01	89.244.131.141		1 x
Windows7 - TAXMAN - Update Server 02	212.211.139.206		1 x

Insgesamt werden **fünf IP-Adressen** bzw. Hosts benötigt, damit die Steuer- und Buchhaltungssoftware ihre Funktion erbringen kann. Drei der IP-Adressen sind notwendig, um den automatischen Update-Prozess zu erlauben. Die anderen beiden IP-Adressen wiederum sind vom Finanzamt Karlsruhe, an das im Anschluss die Daten übermittelt werden.

5. Fazit

Windows kann so viel »schreien« und »rufen« wie es will – die IPFire lässt das neugierige Betriebssystem verstummen. Mit diesem Ansatz könnt ihr **ausgehende** Verbindungen kontrollieren und reglementieren. Letztendlich solltet ihr nur jene Verbindungen zulassen, die eure Anwendungen für die Funktionserbringung benötigen – alles andere ist in diesem Setup nicht erlaubt.

An dieser Stelle sei noch erwähnt, dass ihr für die Umsetzung nicht zwangsläufig eine IPFire benötigt. Auch andere externe Firewalls, die ihr nach Vorbild des Whitelist-Prinzips konfiguriert, erzielen das selbe Ergebnis: Windows ist unter Kontrolle. Oder anders formuliert: Es hält endlich seine Klappe.

Bildquellen:

Windows: Smashicons from www.flaticon.com is licensed by CC 3.0 BY
Jail: Freepik from www.flaticon.com is licensed by CC 3.0 BY