

Studienarbeit

Nicolas Flöter

Leistungsfähigkeit von kostenlosen Firewalls

Nicolas Flöter

Leistungsfähigkeit von kostenlosen Firewalls

Studienarbeit im Rahmen der Prüfungsordnung 98
im Studiengang Informatik
Studienrichtung Softwaretechnik
am Fachbereich Elektrotechnik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuer: Prof. Dr. Ing. Martin Hübner

Abgegeben am 05.10.2004

Nicolas Flöter

Thema der Studienarbeit

Leistungsfähigkeit von kostenlosen Firewalls

Stichworte

Firewall, IPCop, Fli4l, Nutzwertanalyse, Firewallklassen, Hacking, Scannen, NMAP, SuperScan, Spoofing, DoS Attacken

Kurzzusammenfassung

Diese Arbeit befasst sich mit einer Leistungsanalyse von kostenlosen Firewalls aus Open Source Projekten. Nach einer Einführung und der Übersicht über die verschiedenen Angriffsmethoden auf Computernetzwerke und die Arten von Firewallsystemen folgt der Entwurf eines Zielsystems, welches anhand eines theoretischen Anwendungsszenarios erstellt wird. An diesem Zielsystem werden die beiden Probanden IPCop und Fli4l gemessen. Am Ende wird eine Zusammenfassung das Gesamtergebnis dieser Leistungsanalyse präsentieren und aufzeigen wo die Stärken und Schwächen der Testteilnehmer liegen. Der darauf folgende Ausblick enthält eine kritische Betrachtung dieser Analyse und zeigt Möglichkeiten, auf welche weiteren Merkmale zu interessanten Leistungsanalysen führen könnten.

Danksagung

An dieser Stelle möchte ich allen danken, die dazu beigetragen haben, dass ich diese Arbeit fertig stellen konnte.

Insbesondere danke ich:

- Der Firma BERATA die, mir ermöglicht mein Studium zu finanzieren. Insbesondere Hermann Winter, Thilo Horenburg und Arne Brehmer, die mir ermöglichen, genügend Zeit für die Vollendung meines Studiums aufzubringen.
- Prof. Dr. Martin Hübner, der mir mit gutem Rat zur Seite stand, um die Schwierigkeiten zu überwinden, die auftreten, wenn man zum ersten mal eine Arbeit in einem derartigen Umfang verfasst.
- Meiner Lebenspartnerin Michaela, die mir mit viel Verständnis durch die schweren Tage geholfen hat, an denen man kein Ende kommen sieht.
- Meinem Freund Hauke Schacht, der mir trotz unserer gemeinsamen Firma die Zeit gegeben hat, um mich auf mein Studium konzentrieren zu können.
- Meinen Eltern und meiner Schwester, die mir Mut zugesprochen und mich unterstützt haben, um dieses Studium zu beginnen.
- Meinen Kollegen Tobias Kilian und Frank Pohler, die mir geholfen haben diese Arbeit auf Rechtschreibfehler und Ausdrucksschwächen zu korrigieren.

Inhaltsverzeichnis

1	Einleitung	9
2	Inhalt / Aufgabenstellung	10
3	Angriffstechniken	11
3.1	Computernetze Scannen	11
3.1.1	ICMP-Scan	11
3.1.2	TCP-Connect-Scan	11
3.1.3	TCP-Ping-Scan	11
3.1.4	TCP-Syn-Scan	12
3.1.5	TCP-ACK-Scan	12
3.1.6	Slow-Scan	12
3.1.7	Null-Scan/Xmas-Tree-Scan	12
3.1.8	UDP-Scan	12
3.2	Legitime Verbindungen ausspionieren	13
3.2.1	Paket-Sniffing	13
3.3	Mittels Datenmaterial angreifen	13
3.3.1	Buffer Overflows	13
3.4	Legitime Verbindungen übernehmen	14
3.4.1	IP-Spoofing	14
3.4.2	ARP-Spoofing	14
3.4.3	TCP-Session-Hijacking	14
3.4.4	Man-In-The-Middle-Attacke	15
3.5	Netzwerkressourcen ausschalten	15
3.5.1	SYN-Flooding	15
3.5.2	Smurf	15
3.5.3	Fraggle	16
3.6	Schwachstellen	16
4	Einbruchswerkzeuge	17
4.1	Ping-Scanner	17
4.2	Port-Scanner	17
4.3	Schwachstellen-Scanner	17
4.4	In dieser Arbeit verwendete Scanner	18
4.4.1	SuperScan 4.0	18
4.4.2	NMAP 1.3.1 für Windows	18
5	Firewalls	19
5.1	Was sind Firewalls	19
5.2	Firewallklassen	19
5.2.1	Paketfilter	19
5.2.2	Zustandsbasierte Paketfilter	20
5.2.3	Proxybasiert	20
6	Zielsystementwurf	22
6.1	Testszenario	22
6.2	Testkriterien	22
6.2.1	Komfort	22
6.2.2	Kosten	23
6.2.3	Sicherheit	24
6.2.4	Netzwerkleistung	25

7	Testen der Leistungsfähigkeit	27
7.1	Testaufbau	27
7.1.1	Scantests	27
7.1.2	Testen der Verarbeitungsverzögerung	28
7.2	Testdurchführung	28
7.2.1	Installationsdauer.....	28
7.2.2	Konfigurationsdauer.....	28
7.2.3	Scantests	29
7.2.4	Erkennen des Scanversuchs	29
7.2.5	Feedback bei Angriffsversuch	29
7.2.6	Verarbeitungsverzögerung	29
7.3	Testergebnisse	31
7.3.1	IPCOP v1.3.0 Sicherheitsupdate 9.....	31
7.3.2	Fli4l 2.0.8	47
8	Auswertung/Zusammenfassung	58
8.1	Kosten	58
8.2	Sicherheit.....	59
8.3	Netzwerkleistung.....	60
8.4	Komfort	60
8.5	Gesamtergebnis	61
9	Ausblick	62
10	Literatur.....	63
11	Anhang	64
11.1	Zielsystementwurf.....	64
11.2	NMAP.....	64
11.3	SuperScan	64
11.4	Netcps	64
11.5	Fli4l	64
11.6	IPCop.....	64
11.7	CDs	65

Abbildungsverzeichnis

Abbildung 7.1-1 Versuchsaufbau Scantests	27
Abbildung 7.1-2 Versuchsaufbau Verarbeitungsverzögerung ermitteln	28
Abbildung 7.2-1 Messung der Übertragungsrate ohne Firewall, 1000MB, Serverseite	30
Abbildung 7.2-2 Messung der Übertragungsrate ohne Firewall, 1000MB, Clientseite	30
Abbildung 7.3-1 IPCop GUI	31
Abbildung 7.3-2 IPCop Betriebssystem	32
Abbildung 7.3-3 IPCop Konfigurationsdateien für den Paketfilter	33
Abbildung 7.3-4 IPCop sperren der Ports 137-139	33
Abbildung 7.3-5 IPCop Verhindern von Xmas- und Null-Scan	33
Abbildung 7.3-6 IPCop Fernwartungszugang	34
Abbildung 7.3-7 IPCop SSH	34
Abbildung 7.3-8 IPCop Updateassistent	35
Abbildung 7.3-9 IPCop DHCP	36
Abbildung 7.3-10 IPCop Webproxy	37
Abbildung 7.3-11 IPCop Webproxy Analyse	38
Abbildung 7.3-12 IPCop Webproxy Protokollierung	38
Abbildung 7.3-13 IPCop VPN Steuerung	39
Abbildung 7.3-14 IPCop VPN Verbindungen	39
Abbildung 7.3-15 IPCop Ergebnis Übertragungsrate	40
Abbildung 7.3-16 IPCop Verbindungsprotokollierung	40
Abbildung 7.3-17 IPCop Firewall Protokollierung	41
Abbildung 7.3-18 IPCop IDS Protokoll	42
Abbildung 7.3-19 IPCop Snort	43
Abbildung 7.3-20 IPCop NMAP TCP-Syn-Scan	44
Abbildung 7.3-21 IPCop NMAP Null-Scan	45
Abbildung 7.3-22 IPCop Angriffserkennung	46
Abbildung 7.3-23 Fli4I Zusatzpakete zur Installation	47
Abbildung 7.3-24 Fli4I Imonc	48
Abbildung 7.3-25 Fli4I Imonc Überblick	48
Abbildung 7.3-26 Fli4I Webinterface	49
Abbildung 7.3-27 Fli4I FliwizNG	50
Abbildung 7.3-28 Fli4I Paketweiterleitung	51
Abbildung 7.3-29 Fli4I Paketfilterkonfiguration	51
Abbildung 7.3-30 Fli4I Fernadministration	52

Abbildung 7.3-31 Fli4l RoutingEinstellungen	53
Abbildung 7.3-32 Fli4l DNS Konfiguration	54
Abbildung 7.3-33 Fli4l DHCP Konfiguration	54
Abbildung 7.3-34 Fli4l Internetproxy Konfiguration	55
Abbildung 7.3-35 Fli4l Ergebnis Übertragungsrate	55
Abbildung 7.3-36 Fli4l DMZ Möglichkeit	56
Abbildung 7.3-37 Fli4l SSH Aktivierung	56
Abbildung 7.3-38 Fli4l Null-Scan	57

1 Einleitung

Die Vernetzung von Datenverarbeitungsanlagen gewinnt immer mehr an Bedeutung. Virtual Private Network (VPN) bietet die Möglichkeit, über die Infrastruktur des Internets Systeme standortübergreifend zu verbinden. So können sich externe Mitarbeiter von außerhalb mit ihren mobilen Geräten in das interne Netz ihrer Firma anmelden. Öffentliche Dienste wie zum Beispiel Webservices werden über Server, die teilweise auch in einem lokalen Computernetz angeschlossen sind, angeboten.

Diese Möglichkeiten der Vernetzung bergen allerdings die Gefahr, dass sich Unbefugte Zutritt verschaffen. Systeme, die direkt an das Internet angeschlossen sind, werden ständig der Gefahr ausgesetzt, von Unbekannten angegriffen zu werden.

Um die Gefahren der Kompromittierung der eigenen Systeme durch Dritte zu minimieren, werden verschiedene Sicherheitsmechanismen verwendet.

Ein wichtiger Netzwerkmechanismus ist das Einteilen des Netzes in verschiedene Zonen mit unterschiedlichen Sicherheitsgraden. Es werden unterschiedliche Ansätze notwendig, je nachdem wie die Datenverarbeitungslandschaft, die es zu schützen gilt, angelegt ist. Eine pauschal beste Architektur gibt es nicht. Die Anforderung an die Funktionalitäten innerhalb eines Computernetzes sind unterschiedlich und fordern eine passend zugeschnittene Architektur.

Um die Sicherheit des eigenen Computernetzes zu erhöhen können Komponenten wie DNS-, SMTP- oder WWW-Server, die einen geringeren Sicherheitsgrad besitzen, in ein Subnetz mit niedrigerer Sicherheitsstufe ausgegliedert werden. Diese Subnetze werden *de-militarized zone* (DMZ) genannt. **[3]**

Die Einteilung eines Computernetzes in Zonen hat den Vorteil, dass die unsicheren Komponenten den Sicherheitsgrad des inneren Netzes nicht mehr beeinträchtigen.

Um die Zonen sicherheitstechnisch voneinander trennen zu können, müssen Netzwerknoten benutzt werden, die den Datenverkehr für ihren jeweils geltenden Abschnitt überwachen. Diese sogenannten Netzwerkfirewalls haben die Verantwortung über die Sicherheit eines Netzwerks zwar nicht allein, tragen aber erheblich zur Netzwerksicherheit bei. Sie haben meist direkten Kontakt zum Internet und müssen bei einer akzeptablen Übertragungsleistung direkten Angriffen standhalten.

2 Inhalt / Aufgabenstellung

Ziel dieser Studienarbeit ist die Leistungsfähigkeit kostenloser Firewalls zu ermitteln. Der Focus liegt auf Softwarefirewalls aus Open Source Projekten, die in kleinen bis mittleren Netzwerken eingesetzt werden können.

Firewalls können Opfer verschiedener Angriffstechniken werden, wie sie im Kapitel *Angriffstechniken* beschrieben werden. Sicherheit allein macht eine gute Firewall aber noch nicht aus. Für die Güte der Testobjekte werden deshalb weitere Testkriterien herangezogen, die sich auf die Betriebskosten, der Netzwerkleistung und den Komfort beziehen. Dazu wird ein Zielsystem entwickelt, an dem die gewünschte Leistungsfähigkeit gemessen werden kann.

Um den Testaufwand in einem übersichtlichen Rahmen zu halten, werden nicht alle aufgeführten Angriffstechniken in die Tests einfließen.

Am Ende der Tests wird eine Nutzwertanalyse über alle Testkriterien gemacht, die als Ergebnis einen Testsieger hervorbringt.

Dieser Test ist als Entscheidungshilfe zu sehen, da die Einsatzgebiete für Firewalls zu vielfältig sind. Er zeigt die Güte der Testobjekte in Zusammenhang mit einem bestimmten Szenario, welches wiederum Entscheidungsgrundlage für die Gewichtung der einzelnen Testkriterien ist.

3 Angriffstechniken

Um zu verstehen, was eine Firewall alles leisten müsste, um ein lokales Netz von einem unsicheren externen Netz zu schützen, muss man verstehen, wie ein Netz über einen fremden Host angegriffen werden kann. Dieses Kapitel erklärt die unterschiedlichen Angriffsmöglichkeiten auf Computernetze und erklärt deren Grundfunktion.

3.1 Computernetze Scannen

Ein Scan ist der Versuch herauszufinden, welche Hosts aktiv sind und welche Dienste auf ihnen bereitgestellt werden. Es gibt verschiedene Methoden, dies zu tun.

3.1.1 ICMP-Scan

Mit dieser Technik kann der Angreifer herausfinden, ob ein Host aktiv ist bzw. es werden ICMP-Nachrichten verschiedener Typen an mehrere IP-Adressen eines Segmentes gesendet, um alle aktiven Hosts zu entdecken. Ist einer dieser Nachrichten-Typen nicht geblockt erkennt der Angreifer den jeweiligen Host als aktiv. Am häufigsten verwenden Angreifer den ICMP-Nachrichten Typ 8 = ECHO (oder auch Ping genannt). [1]

3.1.2 TCP-Connect-Scan

Beim klassischen Portscan, auch TCP-Connect-Scan genannt, wird versucht, vollständige TCP/UDP-Verbindungen (3-Wege-Handshake SYN, SYN/ACK, ACK) auf einzelnen oder mehreren Ports aufzubauen. Antwortet der Server mit SYN/ACK, ist der Port aktiv. Damit lässt sich ermitteln welche Dienste auf einem System laufen, die angegriffen werden können. Die Verbindung wird dann erfolgreich durchgeführt. Diese Art des Scannens ist sehr leicht zu bemerken, da in den Log-Dateien des Zielrechners eine große Anzahl von Verbindungen und Fehlermeldungen der akzeptierten und sofort wieder geschlossenen Verbindungen zu sehen sind. Deshalb wurden intelligentere Scan-Methoden entwickelt. [1]

3.1.3 TCP-Ping-Scan

Der TCP-Ping-Scan erreicht dasselbe Ziel wie der ICMP-Scan, indem er TCP-Syn Pakete über einen gewählten Port, meist Port 80 für http, an den Zielhost schickt und auf ein TCP-RST wartet. Meldet sich der Host mit einem TCP-RST, dann ist er aktiv. Der Verbindungsaufbau wird nicht vollständig vollzogen, aber auch nicht beendet. Dieser Scan wird eingesetzt, um aktive Systeme zu erkennen, bei denen die ICMP-Nachrichten geblockt werden. [1]

3.1.4 TCP-Syn-Scan

Der TCP-Syn-Scan wird als „halb-offenes“ Scannen bezeichnet, da die TCP-Verbindungen nur zur Hälfte aufgebaut werden. Der Angreifer sendet ein Syn-Paket zum Verbindungsaufbau. Das Opfer sendet ein SYN/ACK und zeigt, dass der Port aktiv ist, oder es sendet ein RST/ACK und zeigt, dass der Port nicht erreichbar ist. Als Antwort auf einen SYN/Ack sendet der Angreifer sofort wieder ein RST und beendet die Verbindung. Der Vorteil in dieser Scan-Methode liegt darin, dass weniger Stellen diesen Scan loggen. [1]

3.1.5 TCP-ACK-Scan

Mit Hilfe dieses Scans kann bestimmt werden, ob es sich bei dem jeweiligen Host um ein einfachen Paketfilter oder einen zustandsbasierten Paketfilter (SPF) handelt. Dieser Scan schickt ein ACK-Paket mit einer zufälligen Sequenznummer an einen beliebigen Port. Wenn ein RST zurückkommt, wird der Port als *unfiltered* eingestuft und deutet somit auf einen einfachen Paketfilter hin. Wenn nichts (oder ein ICMP unreachable error) zurückkommt, wird der Port als *filtered* eingestuft, was auf einen zustandsbasierten Paketfilter schließen lässt. [1]

3.1.6 Slow-Scan

Viele Intrusion Detection Systeme (IDS) erkennen einen Portscan erst dann als solchen, wenn eine gewisse Anzahl an Verbindungsversuchen in einem festen Zeitintervall überschritten wird. Der Slow-Scan, der nur eine bestimmte Anzahl von Ports innerhalb einer bestimmten Zeit scant, fällt weniger auf. [1]

3.1.7 Null-Scan/Xmas-Tree-Scan

Diese beiden Angriffe beruhen auf der gleichen Idee. Es wird ein Paket an den Zielhost gesendet, das zu keiner gültigen Verbindung gehört. Ist der Port geschlossen, so wird auf dieses Paket mit RST geantwortet. Wenn der Port allerdings geöffnet ist, so wird das Paket verworfen (siehe RFC 739). Der Unterschied zwischen der Xmas-Tree- und Null-Scan Attacke liegt in den gesendeten Paketen. Bei der Xmas-Tree Attacke sind alle Flags gesetzt (FIN, URG und PUSH) und bei der Null-Scan Attacke sind keine Flags gesetzt. Ein weiterer Vorteil dieser Scan-Methoden ist der, dass sich nicht alle Betriebssysteme an die Standards nach RFC 739 halten. Eine Windowsmaschine reagiert auf solche Scans mit ungewöhnlich vielen offenen Ports. Der Angreifer kann somit erkennen, ob es sich bei dem Zielhost um einen Rechner mit Windowsbetriebssystem handelt oder nicht. [1]

3.1.8 UDP-Scan

Es wird ein UDP-Paket zum Zielport gesendet, wenn der Host mit *ICMP Port unreachable* antwortet, ist der Port inaktiv.

Diese Methode ist jedoch unzuverlässig, da bei Erfolg keine Rückmeldung durch das Zielsystem erfolgt und ungültige Pakete von den Zielsystemen verworfen werden können. [1]

3.2 Legitime Verbindungen ausspionieren

3.2.1 Paket-Sniffing

Über das Paket-Sniffing versuchen Angreifer Informationen, die in einem Netz übertragenen Pakete herauszulesen.

Pakete, die an einen Rechner adressiert sind, werden normalerweise von allen anderen ignoriert. In Broadcast-Netzen (z.B. Ethernet oder Token Ring) ist es möglich, dass jeder angeschlossene Rechner jedes Datenpaket „hören“ kann. Dies funktioniert jedoch nur soweit sich der *sniffende* Rechner im selben Netzwerksegment befindet. Im Falle von Ethernet wäre dies derselbe Hub.

Sniffer gehören oft zum Standardumfang eines Betriebssystems (z.B. *tcpdump* bei Linux). Sie erlauben die Inhaltsanalyse sämtlicher Datenpakete (TCP-/IP-Header, Nutzdaten), die an der Netzwerkkarte lesbar sind. Gefährlich wird dies bei Diensten, die ihre Authentisierung ohne Verschlüsselung durchführen wie z.B. FTP, Telnet oder POP3. Angreifer können sich dieser Methode bedienen, um insbesondere Passwörter auszuspähen. [4]

3.3 Mittels Datenmaterial angreifen

Diese Arten von Angriffen basieren auf der Formatierung des Datenstroms bzw. auf der Veränderung dessen zeitlicher Abfolge.

3.3.1 Buffer Overflows

Buffer Overflows werden verwendet, um auf einem vorher ausgewählten Opferhost möglichst unbeschränkten Zugriff zu erhalten. Der Angreifer versucht zu diesem Zweck, einen ausführenden Prozess zu übernehmen. Dieser Prozess sollte als Ausführungsrecht nach Möglichkeit *root-* bzw. *Administratorenrechte* besitzen. Die Übernahme erfolgt über den Stack des Rechners. Eingabedaten mit Überlänge überschreiben Teile des Stacks und tauschen die echte Rücksprungadresse durch eine Adresse auf das vom Angreifer eingeschleuste Programm aus. Auf manchen Systemen lässt sich auf dem Stack sogar ausführbarer Code platzieren und verwenden. Möglich ist dies durch nachlässige Programmierung und unsichere Programmiersprachen wie z.B. C/C++. Eine unzureichende Längenprüfung bzw. Absicherung der Eingabedaten (Pre- und Postconditions) von Prozeduren ermöglicht so einen Angriff über den Stack. [2], [4]

3.4 Legitime Verbindungen übernehmen

3.4.1 IP-Spoofing

IP-Spoofing ist eine Angriffsmethode, bei der falsche IP-Adressen verwendet werden, um dem anzugreifenden System eine falsche Identität vorzuspielen.

Bei vielen Protokollen der TCP/IP-Familie erfolgt die Authentisierung der kommunizierenden IT-Systeme nur über die IP-Adresse. Diese kann aber leicht gefälscht werden. Nutzt man darüber hinaus noch aus, dass die von den Rechnern zur Synchronisation einer TCP/IP-Verbindung benutzten Sequenznummern leicht zu erraten sind, ist es möglich, Pakete mit jeder beliebigen Absenderadresse zu verschicken. Der Angreifer kann somit eine Adresse annehmen, die in dem Segment des anzugreifenden Hosts vorkommt und dem Opfer vorgaukeln, er sei im selben Netzsegment und damit vertrauenswürdig. [4]

3.4.2 ARP-Spoofing

In LANs in denen das *address resolution protocol* (ARP) eingesetzt wird, sind sehr viel wirkungsvollere Spoofing-Angriffe möglich. ARP dient dazu IP-Adressen auf Mac-Adressen umzusetzen. Hierfür wird intern eine Tabelle aufgebaut, die zu jeder IP-Adresse eine Mac-Adresse vorsieht. Gibt es keinen Eintrag zu einer IP-Adresse, wird ein ARP-Broadcast-Paket mit der unbekanntenen IP-Adresse ausgesandt. Der Rechner mit dieser IP-Adresse sendet dann ein ARP-Antwort-Paket mit seiner Mac-Adresse zurück. Da die ARP-Antwort-Pakete nicht manipulationssicher sind, reicht es dann meist schon, die Kontrolle über einen der Rechner im LAN zu erlangen und die gewünschten IP-Adressen mit der Mac-Adresse des Angreifers zu versehen, um das gesamte Netz zu kompromittieren. [4]

3.4.3 TCP-Session-Hijacking

Um TCP-Session-Hijacking zu praktizieren, muss der Angreifer vorher die IP-Adresse des Opfers kennen. Er muss in Erfahrung bringen, welche Anwendung hinter der Verbindung steht, die er zu übernehmen wünscht, und welche Sequenznummer gerade aktuell ist. Die Art der Anwendung, die hinter einem TCP Paket steht, erhält der Angreifer über den TCP Port. Er steht im Kopf des TCP Rahmens. Die Schwierigkeit dieses Angriffes besteht darin, die richtige Sequenznummer herauszufinden, die das nächste gültige Paket identifiziert.

Diese Methode zielt auf eine im Aufbau oder im Betrieb befindliche TCP Verbindung zwischen dem Opfer und seinem Partner. Bevor der Angreifer die Verbindung übernimmt, setzt er den ursprünglichen Partner mittels Syn-Flooding außer Gefecht. Danach gibt sich der Angreifer dem Opfer gegenüber als Partner aus, indem er Pakete mit der Zieladresse des Opfers, der Quelladresse des legitimen Partners, dem TCP Port und der korrekten Sequenznummer in die Verbindung einspeist. Auf diese Weise gefälschte Pakete enthalten z.B. Kommandos mit der die Anwendung, die hinter der Verbindung steht, auf dem Host gesteuert werden kann und somit dem Angreifer die Möglichkeit gibt, das Opfer zu übernehmen. [2], [4]

3.4.4 Man-In-The-Middle-Attacke

Diese Angriffsmethode wird verwendet, um geheime Informationen aus einer legitimen, verschlüsselten Verbindung zu ermitteln.

Der Angreifer setzt sich als Zwischenstation zwischen die beiden Verbindungspartner, liest deren ausgetauschte Pakete und leitet sie an den jeweiligen Adressaten weiter. Voraussetzung für das Lesen der verschlüsselten Pakete ist ein gültiger Zugangscode.

[4]

Mögliche Schwachpunkte, die diese Angriffsmethode begünstigen, sind Netzwerke mit schwacher Authentifikation oder interne Netze in denen die Zugangsdaten „ausgeschnüffelt“ werden. (Siehe Paket-Sniffing).

3.5 Netzwerkressourcen ausschalten

Viele IT-Systeme sind direkt am Internet angeschlossen und bieten öffentliche Dienste an, ohne beeinflussen zu können, wer wie viele Ressourcen in Anspruch nimmt.

Schlecht implementierte Systeme können bei einer massenhaften Dienstanforderung mit ungültigen Parametern sogar zum Absturz gebracht werden.

So genannte *denial of service* Attacken (DoS Attacken) verhindern die Nutzung von Diensten für legitime Nutzer, indem sie die Verfügbarkeit von Netzwerkressourcen massiv einschränken. **[2]**

3.5.1 SYN-Flooding

Beim Syn-Flooding wird eine große Anzahl „halboffener“ Verbindungen erzeugt, indem TCP-Syn-Pakete (Verbindungsanfrage) mit einer ungültigen Absenderadresse verschickt werden.

Das Ergebnis ist, dass das Opfer für jede Verbindungsanfrage Ressourcen reserviert, in dem Fall TCP-Pufferspeicher, und jeweils Syn/Ack-Antworten zurück sendet. Um die Verbindung zu vervollständigen, wartet das Opfersystem auf ein ACK des Angreifers, welcher aber nie kommt. So wartet das System jeweils auf ein Timeout für die Verbindung und belegt Pufferplätze mit diesen „halboffenen“ Verbindungen. Bei einem extremen Ansturm von Verbindungsanfragen kann der TCP/IP-Stack zusammenbrechen und das System wird lahmgelegt. **[2], [1]**

3.5.2 Smurf

Der Smurf-Angriff ist ein *distributed DoS* Angriff und dient dem Überfluten des Opfers mit ICMP Echo-Message.

Dazu wählt der Angreifer ein verstärkendes Netzwerk aus, welches im Idealfall eine größere Bandbreite besitzt als das Opfer und eine große Anzahl an Hosts besitzt. Er sendet massenhaft ICMP Echo-Anfragen an die Broadcastadresse des Verstärkernetzes mit der IP-Adresse des Opfers als Quelladresse. Jeder Host des Verstärkernetzes wird auf diese ICMP-Anfrage antworten und ein Paket an das Opfersystem schicken.

Bei einer ausreichenden Anzahl an Verstärkernetzen wird das Opfersystem überlastet und bricht unter Umständen sogar zusammen. **[2], [1]**

3.5.3 Fraggle

Der Fraggle-Angriff ist im Wesentlichen ein Smurf-Angriff, nur dass statt ICMP-Nachrichten UDP-Pakete an die Broadcastadresse des verstärkenden Computernetzes geschickt werden. Typischerweise wird das der Port 7 für *echo* sein. Jeder Host, bei dem *echo* aktiviert ist, wird darauf antworten. **[2], [1]**

3.6 Schwachstellen

Schwachstellen in Form von Fehlern auf der Applikationsebene entziehen sich der allgemeinen Kontrolle der Firewalls, da sie, bis auf die dedizierten Proxies, maximal in der Transportebene des ISO/OSI Schichtenmodells operieren. Angriffe auf Schwachstellen der Applikationen werden in dieser Arbeit nicht betrachtet. **[3], [2]**

4 Einbruchswerkzeuge

Es gibt viele Werkzeuge, derer sich Angreifer bedienen können. Ein Programm allein reicht nicht aus, um einen erfolgreichen Angriff zu ermöglichen. Bevor der Angreifer einen Host angreifen kann, muss er ihn über seine IP-Adresse ausfindig machen. Ein Angriff erfolgt meist in Stufen, in denen der Angreifer sich eines geeigneten Werkzeugs bedient.

Die 1. Stufe ist das Sammeln von Informationen über die gesamte Datenverarbeitungsstruktur des Ziels wie z.B. Netzwerktopologie, Domänen, etc.. Die 2. Stufe ist das Durchsuchen des Zielnetzes nach einem geeigneten Opferhost und das Herausfinden dessen Betriebssystems, das Suchen nach offenen Ports und laufenden Applikationen. Die 3. Stufe ist das Auswerten der gesammelten Informationen und das Ausnutzen von Schwachstellen der Betriebssysteme und Applikationen, um die Kontrolle über den Host zu übernehmen.

Absolviert der Angreifer erfolgreich die Ziele der 2. Stufe, dann hat eine vorhandene Firewall bereits ihren Dienst versagt. Sensible Daten können von dem Angreifer über Einleitung der 3. Stufe unbefugt erreicht werden. [1]

4.1 Ping-Scanner

Mit Ping-Scannern kann per ICMP-ECHO-Request (ICMP Typ 8) festgestellt werden, welche Systeme aktiv sind. [1]

4.2 Port-Scanner

Über Portscanner kann man herausfinden, welche Hosts aktiv sind, welche Dienste ein Host zur Verfügung stellt bzw. welche Ports bei dem Zielsystem offen sind, um die dahinter liegende Applikation anzusprechen. [1]

4.3 Schwachstellen-Scanner

Schwachstellenscanner haben eine Datenbank über Schwachstellen in bekannten Applikationen. Angreifer decken mit Ihnen Unsicherheiten und Lücken beim Opfer auf, um sich mit deren Hilfe Zugang zum Zielsystem zu verschaffen. [1]

4.4 In dieser Arbeit verwendete Scanner

4.4.1 SuperScan 4.0

Diese Software von Foundstone ist ein TCP/UDP Portscanner. Er bietet die Möglichkeit, ganze IP-Adressbereiche zu scannen und Portlisten in Form von Textdateien zu verarbeiten. Dabei sind Feineinstellungen möglich, wie die Scangeschwindigkeit, Anzahl der Durchläufe pro Host und der Timeout für die abgeschickten Pakete.

2 Arten des TCP-Port-Scans und 2 des UDP-Port-Scans unterstützt SuperScan. Den TCP-Connect-Scan, den TCP-Syn-Scan, den UDP-Data-Scan und den UDP-Data-Scan mit ICMP Unterstützung. Im Bereich des ICMP-Protokolls unterstützt SuperScan die Abfrage von Echo-, Timestamp-, Address Mask- und Information Request. Der Scanner ist kostenlos und kann unter Windows 2000 und Windows XP verwendet werden. [1]

4.4.2 NMAP 1.3.1 für Windows

NMAP bietet die Möglichkeit unter Windows sowie Linux verwendet zu werden. Für die Betriebssysteme von Microsoft gibt es sogar eine Benutzeroberfläche, mit der man diesen Scanner komfortabel bedienen kann. Unter der Oberfläche von NMAP Win 1.3.1 arbeitet der Kommandozeilenscanner NMAP 3.0 für Windows.

NMAP beherrscht Ablenkungsmanöver mit gefälschten Scans parallel zum echten Scan, sowie das Fragmentieren von Paketen zum Unterlaufen von einfachen Paketfiltern. Weitere Scanoptionen sind die Einstellung der Scanfrequenz und die Angabe des Portnummernraums, der untersucht werden soll.

Zur Erkennung von aktiven Systemen bietet NMAP folgende Scantechniken:

- ICMP-Suchläufe
- TCP-Ping-Scan
- TCP-Ack-Scan

Zur Erkennung der horchenden/bereitgestellten Dienste bietet NMAP noch weitere Scantechniken an:

- TCP-Connect-Scan
- TCP-Syn-Scan
- TCP-Null-Scan
- TCP-Xmas-Tree-Scan
- UDP-Scan
- Ident-Scan (ansprechen von Port 113) Viele Versionen des Dienstes *ident* geben den Eigentümer eines bestimmten Ports als Antwort zurück.
→ primär für UNIX Zielsysteme

Mit der Bestimmung der Quelladresse und des Quellports bietet NMAP die Möglichkeit eine Spoofing-Attacke durchzuführen. [1]

5 Firewalls

5.1 Was sind Firewalls

Eine Firewall ist eine Kombination aus Hard- und Software, die bestimmte Pakete einlässt und andere aussperrt, so dass ein zu schützendes Netzwerk einer Organisation von einem unsicheren Netz abgeschirmt werden kann. [5]

Eigenschaften, die eine Firewall beinhalten kann, sind das einfache Filtern von Paketen, das Merken der Zustände von Verbindungen und deren ausgetauschte Pakete, Virens Scanner, Überwachung auf Anwendungsebene und ein Intrusion Detection System (IDS).

Je nachdem wie eine Firewall ein- und ausgehende Pakete überprüft, kann sie einer Klasse zugeordnet werden.

Als Grundfunktion können Firewalls mit definierten Filterregeln entscheiden, welche ankommenden und herausgehenden Pakete akzeptiert werden oder nicht. Die Entscheidungsgrundlagen sind hier die IP-Adressen und Portnummern. Bessere Firewalls können den Datenverkehr filtern, indem sie den Inhalt der ankommenden Pakete analysieren.

Firewalls sind meist eigenständige, geschlossene Systeme, können aber auch zusätzlich als Software auf einem Computer laufen oder in einer Kombination mit einem anderen Gerät implementiert sein. Es gibt z.B. Router mit Paketfilterfunktionen (*screening router*), die als erste Verteidigungslinie eingesetzt werden. [3], [2]

5.2 Firewallklassen

5.2.1 Paketfilter

Einfache Paketfiltersysteme sind auf der Netzwerk- und der Transportschicht des ISO/OSI Modells angesiedelt. Ihre Funktionalität ist beschränkt.

Sie überwachen die ein- und ausgehenden Pakete wobei jedes einzelne Paket getrennt betrachtet wird ohne dabei Informationen, die aus der Betrachtung vorangegangener Pakete stammen könnten, zu berücksichtigen.

Die Filterung geschieht aufgrund eines definierten Regelwerkes und den Filterkriterien, die dem Paketfilter zur Verfügung stehen.

Folgende Filterkriterien stehen einem einfachen Paketfilter zur Verfügung: Quell IP-Adresse, Ziel IP-Adresse, Protokolltyp, Quell- und Zielport, ICMP- oder IGMP Nachrichtentyp, SYN- oder ACK-bits, Empfangende Physikalische Schnittstelle, Physikalische Zielschnittstelle.

Aufgrund dieser Informationen entscheidet der Paketfilter ob ein Paket weitergereicht, modifiziert, ignoriert oder zurückgewiesen werden soll.

Die Aufgaben, die z.B. mit diesem Paketfilter wahrgenommen werden können, sind das Sperren von bestimmten Ports für Anwendungen innerhalb und außerhalb des zu schützenden Netzes sowie die Sperrung des externen Datenverkehrs.

Die Beschränktheit der einfachen Paketfilter hat einen großen Vorteil: Sie sind sehr schnell und benötigen nur wenig Ressourcen. [2], [3]

5.2.2 Zustandsbasierte Paketfilter

Zustandsbasierte Paketfilter (SPF) sind wie die einfachen Paketfilter auf der Netzwerk- und der Transportschicht angesiedelt.

Wie der Name bereits aussagt, verwalten diese Paketfilter Zustände. Über diese Zustände ist es einem Paketfilter erst möglich, Verbindungen zwischen zwei verschiedenen Netzwerken zu erkennen und zu überwachen.

Zustandsbasierte Paketfilter protokollieren mit, wann welche Verbindung aufgebaut wird und überprüfen mit Hilfe der so gewonnenen Daten alle nachfolgenden Pakete. So kann z.B. zu einer bestehenden FTP-Datenverbindung die Kontrollverbindung zugeordnet werden. [2], [3]

5.2.3 Proxybasiert

Ziel der Verwendung von proxybasierten Firewalls ist weniger die Performanz als die maximal mögliche Absicherung des Netzwerkes.

Es gibt zwei Arten von proxybasierten Firewalls. Zum einen gibt es die generischen Proxies, die auch Verbindungsgateways genannt werden, und zum anderen gibt es dedizierte Proxies die als Applikationsfilter bekannt sind.

Beide Arten müssen in Verbindung mit einem Screening Router betrieben werden bzw. können selbst als *dual homed host* laufen.

Proxybasierte Firewalls benötigen die Implementierung auf einem regulären Betriebssystem dessen TCP/IP-Stack sie verwenden können.

Die Verbindung des Screening Routers mit der Firewall hat den Vorteil, dass die TCP/IP Implementierung der Firewall nicht auf die Probe gestellt wird. Der Router ist in der Lage, illegale Pakete schon vorher zu verwerfen bevor sie die Applikationsebene erreichen. Außerdem wird die Verbindung von Client und Zielhost in zwei separate Verbindungen aufgebrochen und die direkte Etablierung zwischen Client und Server durch die Firewall hindurch so verhindert. Client und Server befinden sich dementsprechend in verschiedenen Netzen, wodurch die Firewall implizit gezwungen wird, eine Adressumsetzung durchzuführen. Der Vorteil ergibt sich daraus, dass die Netzstrukturen hinter der Firewall verborgen bleiben.

5.2.3.1 Generische Proxies

Der generische Proxy arbeitet auf der Transportebene des ISO/OSI Schichtenmodells und ist daher anwendungs- und herstellerunabhängig. Unterschiedliche Anwendungen können daher auf einem einheitlichen Dienst aufgesetzt werden. Das bedeutet allerdings auch, dass aufgrund der fehlenden Kenntnisse über die Anwendung keine gezielte Beschränkung der Funktionalität der einzelnen Dienste möglich ist. Sie können nur als Ganzes zugelassen oder gesperrt werden.

5.2.3.2 **Dedizierte Proxies**

Dedizierte Proxies bzw. Applikationsfilter arbeiten auf der Anwendungsebene des ISO/OSI Schichtenmodells. Sie haben die Möglichkeit, anwendungsspezifische Protokolle zu berücksichtigen. Es ist ihnen möglich, eine genaue Analyse der Daten durchzuführen, die durch sie hindurchfließen. Sie ermöglichen eine dienstspezifische Filterung mit der sie Dienste in ihrem Funktionsumfang benutzerabhängig einschränken können. Sicherheitsrelevante Entscheidungen treffen sie aufgrund von semantisch abhängigen Informationen über die Applikation.

Um diese Funktionalität zu gewährleisten, muss für jeden Client in dem dahinter liegenden Netzwerk eine speziell angepasste Clientsoftware installiert werden und für jede Anwendung muss ein anwendungsspezifischer Proxy zur Verfügung stehen.

Bisher gibt es standardmäßig Proxies für *HTTP*, *FTP*, *SMTP* und *POP 3*. Für andere Applikationen besteht nur die Möglichkeit, die Ports durchzuschleifen, welches den Applikationsfilter auf einen Paketfilter reduziert. Nachteile können sich aus einer schlechten Benutzerverwaltung ergeben. Es können Effekte auftreten, bei denen Nutzer in ihren Zugriffsrechten so sehr eingeschränkt werden, dass es ihren Arbeitsablauf behindert. **[2], [3]**

6 Zielsystementwurf

6.1 TestszENARIO

Für ein Unternehmen ist eine Firewall interessant, welche zu minimalen Kosten einen möglichst hohen Sicherheitsstandard und gute Leistungen bringt.

Das Zielsystem wird für ein klein- bis mittelständisches Unternehmen entwickelt, welches als Anbindung an das Internet nur eine Standleitung mit einem Volumentarif besitzt.

Das Unternehmen möchte einen effektiven und komfortablen Schutzmechanismus vor Gefahren aus dem Internet zu möglichst geringen Kosten erhalten. Die Entscheidung fiel daher auf eine kostenlos erhältliche Softwarefirewall, für die man ältere Hardware nutzen kann. Ein tiefgehendes Knowhow im Bereich Netzwerksicherheit und Konfiguration von Firewalls fehlt und muss angeeignet werden. Die Firewall soll möglichst schnell einsatzbereit sein und wird vorerst nur in der Standardkonfiguration benutzt. Mit wachsendem Knowhow soll das System immer weiter angepasst werden.

Das System wird in einem speziellen Raum installiert, wo sich die gesamte Hardware für das Firmennetzwerk befindet.

Der Kauf von zusätzlichen Geräten, die DHCP, DNS, Routing, Proxieing, VPN und DMZ zur Verfügung stellen, soll vermieden werden.

Zusätzlich will die Firma mehrere Dienste wie einen Mailserver, dessen Postfächer von außerhalb benutzt werden sollen, einen FTP-Server und ein Intranet betreiben.

Externe Mitarbeiter sollen per VPN Zugriff auf das interne Netz zugreifen können.

6.2 Testkriterien

Die zu testenden Systeme werden auf 4 Hauptkriterien geprüft: Kosten, Sicherheit, Netzwerkleistung und Komfort. Aus Ihnen entwickeln sich Subkriterien, die das Zielsystem beschreiben, so dass das TestszENARIO bestmöglich unterstützt wird.

6.2.1 Komfort

6.2.1.1 GUI

Zur besseren Administration soll das Zielsystem eine Benutzeroberfläche zur Verfügung stellen.

Gewichtung: $g = 0,25$

Bewertungsfunktion: $f(x) = x * g$, $x = 1$ wenn GUI vorhanden / sonst 0

6.2.1.2 Installationsroutine

Das Zielsystem soll eine Installationsroutine besitzen, um den Installationsvorgang zu vereinfachen.

Gewichtung: $g = 0,25$

Bewertungsfunktion: $f(x) = x * g$, $x = 1$ wenn Installationsroutine vorhanden / sonst 0

6.2.1.3 Updateassistent

Ein vorhandener Updateassistent soll selbstständig nach neuen Updates für das Zielsystem suchen und den Administrator über neue Updates informieren.

Gewichtung: $g = 0,2$

Bewertungsfunktion: $f(x) = x * g$, $x = 1$ wenn Updateassistent vorhanden / sonst 0

6.2.1.4 Fernadministration

Das Testszenario sieht vor, dass das Zielsystem in einem separaten Raum steht. Möglicherweise werden die Administrationsaufgaben auch an eine externe Firma vergeben. Daher soll das Zielsystem eine Möglichkeit bieten aus der Ferne administriert zu werden.

Gewichtung: $g = 0,3$

Bewertungsfunktion: $f(x) = x * g$, $x = 1$ wenn Fernadministrationsfunktionalität vorhanden / sonst 0

6.2.2 Kosten

Eines der wichtigsten Hauptkriterien, die an das Zielsystem gestellt werden, sind die Kosten. Sie sollen möglichst gering gehalten werden.

6.2.2.1 Installationsdauer

Die Dauer der Installation bindet unter Umständen die Arbeitszeit des Administrators und verursacht Kosten. Eine kurze Installationszeit ist daher erwünscht, um die Personalkosten gering zu halten.

Gewichtung: $g = 0,2$

Bewertungsfunktion: $f(x) = (1 - (x - y) / y) * g$, $y =$ bester im Test, $x =$ Benötigte Zeit des Probanden.

6.2.2.2 Konfigurationsdauer

Die Dauer der Konfiguration bis zur Einsatzfähigkeit der Firewall bindet Arbeitszeit des Administrators und verursacht Kosten. Daher ist eine geringe Konfigurationszeit erwünscht, um die Personalkosten gering zu halten und eine schnelle Einsatzfähigkeit zu erreichen.

Gewichtung: $g = 0,3$

Bewertungsfunktion: $f(x) = (1 - (x - y) / y) * g$, $y =$ bester im Test, $x =$ Benötigte Zeit des Probanden.

6.2.2.3 Betriebssystem

Das Zielsystem soll als Betriebssystem Linux unterstützen, um entstehende Lizenzkosten für den Betrieb zu vermeiden.

Gewichtung: $g = 0,5$

Bewertungsfunktion: $f(x) = x * g$, $x = 1$ wenn Linux unterstützt / sonst 0

6.2.3 Sicherheit

Sicherheit ist ein Kriterium an dem die Effektivität des Zielsystems gemessen wird. Es ist das wichtigste neben allen anderen Kriterien.

6.2.3.1 Firewallklasse

In diesem Test wird zwischen 3 Arten von Firewallklassen unterschieden. Paketfilter, zustandsbasierte Paketfilter und dedizierte Proxies. Je nachdem welcher Klasse das Testsystem angehört, steigt die Sicherheit, die es bietet. Der dedizierte Proxy bietet den größten Schutz für das Testszenario. Bei der Bewertung wird eine einfache Abstufung verwendet, wobei der einfache Paketfilter die niedrigste und der dedizierte Proxy die höchste Stufe bildet.

Um als dedizierter Proxy eingestuft zu werden, muss der Proband die üblichen Applikationen wie HTTP, FTP, SMTP und POP 3 unterstützen und deren eingehende und ausgehende Verbindungen überwachen.

Gewichtung: $g = 0,1$

Bewertungsfunktion: $f(x) = x * g$, $x = 1$ wenn AppProxy; $x = 0,5$ wenn SPF; $x = 0$ wenn Paketfilter

6.2.3.2 DMZ

Das Testszenario sieht die Möglichkeit vor, ein Intranet aufzubauen und andere verschiedene Dienste nach außen anzubieten. Der Aufbau einer DMZ unterstützt die Möglichkeit einen Bereich zu schaffen, der eine mögliche Unsicherheit, die aus einer fehlerhaften Implementierung der Dienste entstehen könnte, zu kapseln. Das interne Netzwerk würde dadurch geschützt werden, indem diese Hosts in ein separates Netzwerk zusammengefasst werden. [3]

Gewichtung: $g = 0,15$

Bewertungsfunktion: $f(x) = x * g$, $x = 1$ wenn DMZ möglich / sonst 0

6.2.3.3 IDS

Intrusion Detection Systeme tragen zur Sicherheit bei, indem sie einen möglichen Angriff erkennen und Abwehren können. Sie erhöhen den Grad der Sicherheit, den das Zielsystem leisten kann.

Gewichtung: $g = 0,1$

Bewertungsfunktion: $f(x) = x * g$, $x = 1$ wenn IDS vorhanden / sonst 0

6.2.3.4 Konfigurierbares Paketfilterregelsystem

Um die eingehenden und ausgehenden Verbindungen zu kontrollieren, ist ein Regelsystem notwendig, nach dem die Pakete für die Weiterleitung bewertet werden.

Gewichtung: $g = 0,1$

Bewertungsfunktion: $f(x) = x * g$, $x = 1$ wenn Konfigurierbares Paketfilterregelsystem vorhanden / sonst 0

6.2.3.5 Verbindungsprotokollierung

Die Verbindungsprotokollierung bietet dem Administrator die Möglichkeit, Verbindungsversuche aus beiden Richtungen zu analysieren. Es unterstützt die Konfiguration des Zielsystems. Eine fehlerhafte Konfiguration, die sicherheitsrelevant ist, wie z.B. dass nur bestimmte Ports weitergeleitet werden, kann dadurch erfasst werden.

Gewichtung: $g = 0,1$

Bewertungsfunktion: $f(x) = x * g$, $x = 1$ wenn Verbindungsprotokollierung vorhanden / sonst 0

6.2.3.6 Portscan

Das Zielsystem soll keine offenen Ports zu erkennen geben, damit ein direkter Angriff auf das interne Netzwerk schwieriger wird.

Das beste Ergebnis wird erzielt, wenn alle Ports standardmäßig geschlossen sind.

Gewichtung: $g = 0,15$

Bewertungsfunktion: $f(x) = x * g$, $x = 1$ wenn Alle Ports geschlossen / sonst 0

6.2.3.7 Betriebssystemidentifikation

Das Betriebssystem des Zielsystems soll durch das Werkzeug NMAP nicht erkannt werden. Wird es erkannt, so kann der Angreifer bekannte Schwachstellen dafür suchen und entsprechend ausnutzen.

Gewichtung: $g = 0,15$

Bewertungsfunktion: $f(x) = x * g$, $x = 1$ wenn Betriebssystem nicht identifiziert werden kann/ sonst 0

6.2.3.8 Scan-Angriffserkennung

Scanversuche soll das Zielsystem erkennen, um dem Administrator die Möglichkeit zu geben darauf zu reagieren.

Gewichtung: $g = 0,05$

Bewertungsfunktion: $f(x) = x * g$, $x = 1$ wenn Scan erkannt / sonst 0

6.2.3.9 Feedback bei Angriffsversuch

Ein wichtiges Sicherheitsmerkmal ist die Art und Weise wie das Zielsystem auf sich aufmerksam macht, wenn es gerade aktiv angegriffen wird. Da sich das Zielsystem in einem separaten Raum befindet, der möglicherweise schallgedämpft ist, muss der Administrator zur Kontrolle diesen Raum betreten. Daher wird zwischen einem akustischen Signal und einem visuellen Signal nicht unterschieden.

Gewichtung: $g = 0,1$

Bewertungsfunktion: $f(x) = x * g$, $x = 1$ wenn Feedback vorhanden / sonst 0

6.2.4 Netzwerkleistung

Effektivität wird auch an der Netzwerkleistung gemessen. Mit Netzwerkleistungen sind die Leistungen gemeint, die der Proband mit sich bringt, um IT-Infrastruktur zu bereichern.

6.2.4.1 Routingfunktion

Das Zielsystem soll eine Routingfunktion beinhalten, um als Verbindungsgateway ins externe Netz zu dienen.

Gewichtung: $g = 0,15$

Bewertungsfunktion: $f(x) = x * g$, $x = 1$ wenn Routingfunktion vorhanden / sonst 0

6.2.4.2 DNS

Das Zielsystem soll eine Domännennamensauflösung zur Verfügung stellen.

Gewichtung: $g = 0,05$

Bewertungsfunktion: $f(x) = x * g$, $x = 1$ wenn DNS vorhanden / sonst 0

6.2.4.3 DHCP

Das Zielsystem soll einen *DHCP* Dienst leisten, damit die IP-Adressen im Netzwerk automatisch vergeben werden.

Gewichtung: $g = 0,05$

Bewertungsfunktion: $f(x) = x * g$, $x = 1$ wenn DHCP vorhanden / sonst 0

6.2.4.4 Internetproxy

Um die Internetverbindung zu entlasten, soll das Zielsystem einen Internetproxy zur Verfügung stellen, damit die Aufrufe aus einem Zwischenspeicher erfolgen können. Das bringt eine erhöhte Performance bei der Internetnutzung und spart Onlinekosten.

Gewichtung: $g = 0,15$

Bewertungsfunktion: $f(x) = x * g$, $x = 1$ wenn Internetproxy vorhanden / sonst 0

6.2.4.5 VPN

VPN ermöglicht das Anmelden eines Computers aus einem externen Netz in das zu schützende Netz, so dass der Computer einen autorisierten Verbindungstunnel erhält, der durch Verschlüsselung geschützt wird. Mit diesem Service können sich externe Mitarbeiter mit ihrem Computer am Firmennetzwerk so beteiligen, als wären sie physisch direkt angeschlossen.

Gewichtung: $g = 0,1$

Bewertungsfunktion: $f(x) = x * g$, $x = 1$ wenn VPN vorhanden / sonst 0

6.2.4.6 Userverwaltung mit Zugriffsregeln

Um den Zugriff auf das externe Netz kontrollieren zu können, damit bestimmte Internetseiten für bestimmte User gesperrt werden können, braucht das Zielsystem eine Userverwaltung mit einem Regelsystem.

Gewichtung: $g = 0,2$

Bewertungsfunktion: $f(x) = x * g$, $x = 1$ wenn Userverwaltung mit Zugriffsregelsystem vorhanden / sonst 0

6.2.4.7 Verarbeitungsverzögerung

Die Verarbeitungsverzögerung beschreibt die wichtigste Netzwerkleistung, die das Zielsystem mitbringen muss. Eine Firewall sollte die Übertragungsgeschwindigkeit in externe Netzwerke so wenig wie möglich beeinträchtigen .

Gewichtung: $g = 0,3$

Bewertungsfunktion: $f(x) = x / y * g$, $x =$ gemessene Übertragungsrate,
 $y =$ Übertragungsratenreferenz

7 Testen der Leistungsfähigkeit

7.1 Testaufbau

Die Firewalls werden auf einem *IBM* kompatiblen PC installiert, der folgende Leistungsdaten besitzt:

Pentium 133 Mhz

96MB EDO RAM Arbeitsspeicher

2 GB Festplatte

7.1.1 Scantests

Für die Scantests an der Firewall wird ein Angriffsrechner, ein Opferrechner und ein Firewallrechner, der den Opferrechner schützt, aufgebaut und vernetzt. Das unsichere Netz wird in diesem Test durch ein weiteres LAN simuliert (Siehe Abbildung 7.1-1).

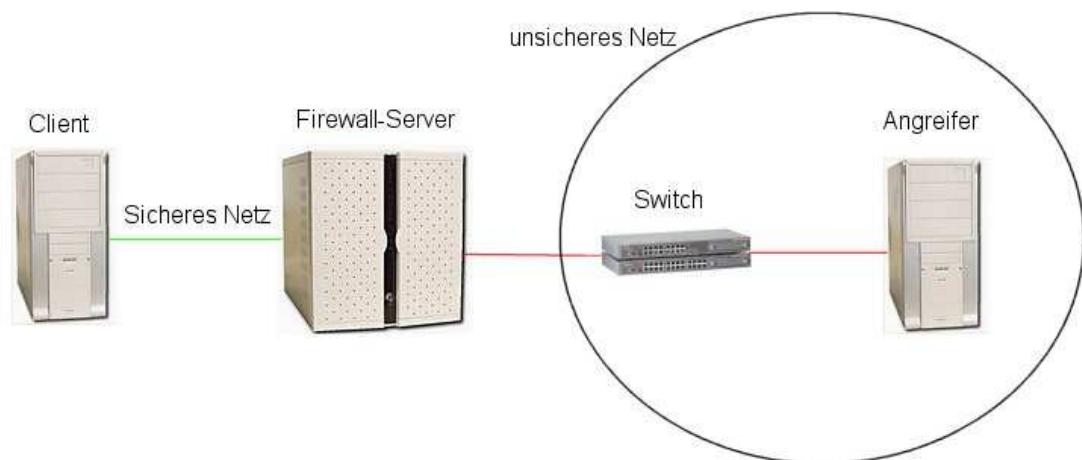


Abbildung 7.1-1 Versuchsaufbau Scantests

7.1.2 Testen der Verarbeitungsverzögerung

Für diesen Test wird der Client und der Server direkt per Kreuzkabel an die Firewall angeschlossen, um Verzögerungen durch eine andere Netzwerkperipherie auszuschließen (Siehe Abbildung 7.1-2).

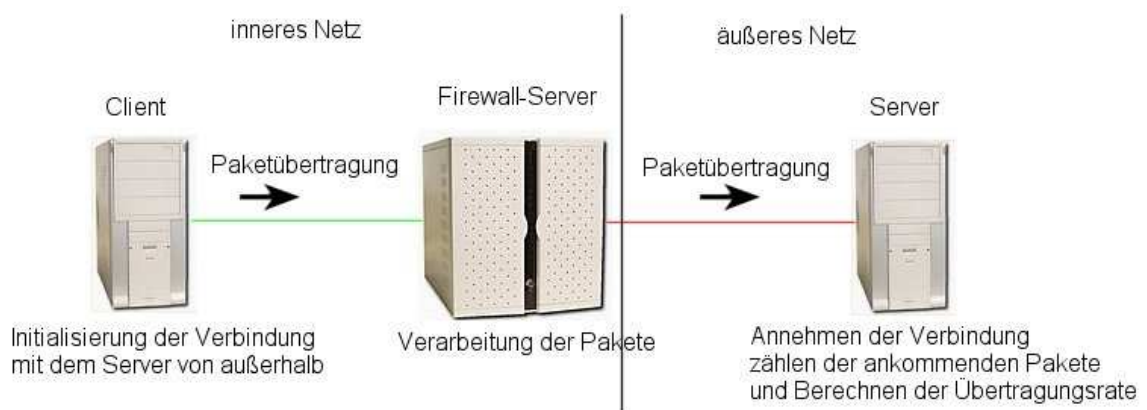


Abbildung 7.1-2 Versuchsaufbau Verarbeitungsverzögerung ermitteln

7.2 Testdurchführung

Die folgenden Punkte erklären die Art und Weise unter welchen Parametern die Tests im Einzelnen ablaufen werden. Die meisten Testkriterien werden aus den Beschreibungen der Anbieter/Ersteller oder durch Sichtung im laufenden Betrieb geprüft und in diesem Kapitel nicht extra aufgeführt.

7.2.1 Installationsdauer

Zur Messung der Installationsdauer wird der Testrechner in einen Zustand versetzt indem die Software direkt aufgespielt werden kann. Die Festplatte ist formatiert. Die Software ist in Form von Compact Disks vorhanden. Es wird die Zeit vom Einlegen des Datenträgers bis zur Fertigstellung der Installation gemessen. Nach der Installation folgt die Konfiguration.

7.2.2 Konfigurationsdauer

Hier wird die Zeit gemessen, die benötigt wird, um das Testgerät in einer Standardkonfiguration einsatzbereit zu machen. Die Konfigurationsdauer wird direkt nach der Installation bis zur Inbetriebnahme gemessen. Zur Standardkonfiguration gehören die Zuweisung der IP-Adressen zu den Netzwerkadapter, die DNS Einstellungen, die Gateway Einstellungen und die DHCP Einstellungen.

7.2.3 Scantests

Es wird davon ausgegangen, dass der Angreifer bereits die IP-Adresse des anzugreifenden Hosts kennt und versucht, diesen auszuspionieren.

Es ist das Ziel, dass der Angreifer in Erfahrung bringen kann, welche Ports offen sind, welche Dienste auf diesen Ports horchen und auf welchem Betriebssystem die Firewall basiert.

Folgende Scantypen werden dafür verwendet:

TCP-Connect-Scan, TCP-Syn-Scan, TCP-Ack-Scan, Null-Scan, Xmas-Tree-Scan, Fin-Stealth-Scan und der UDP-Scan.

Zum Identifizieren des Betriebssystems wird ein Null-Scan von NMAP mit der Option *os detection* verwendet.

7.2.4 Erkennen des Scanversuchs

Um zu prüfen, ob das zu testende System einen Angriff in Form eines Scanversuchs erkennt, wird ein Null-Scan mit NMAP vollzogen. Die Zeiteinstellung der Abtastrate ist auf *insane* (sehr schnell) eingestellt.

Anschließend werden die Protokolldateien untersucht, ob der Angriff identifiziert wurde. Hat das Testsystem keine Protokolldateien, ist der Test negativ ausgefallen.

7.2.5 Feedback bei Angriffsversuch

Das Testsystem wird daraufhin untersucht, ob ein akustisches oder visuelles Signal von ihm ausgeht, wenn es aktiv angegriffen wird. Der Testangriff erfolgt mit SuperScan. Es wird ein TCP-Syn-Scan bei einer Abtastrate von 10ms erzeugt und auf das Testsystem angewendet.

7.2.6 Verarbeitungsverzögerung

Zum Test der Verarbeitungsverzögerung wird das Programm *netcps* benutzt. Es stellt einen einfachen Server und Client zur Verfügung, die Datenpakete über eine TCP/IP-Verbindung austauschen.

Folgende Einstellungen sind bei dem Programm möglich:

- Port auf dem gesendet/gehorcht wird
- Übertragungsmenge in Megabyte
- Adresse des Servers zu dem gesendet werden soll

Das Programm sendet über eine TCP/IP-Verbindung auf dem Port 4455 ein Datenvolumen von 1000 Megabyte vom Client zum Server. Zum Ende der Übertragung errechnet das Programm die durchschnittliche und die maximal gemessene Übertragungsrate, die erreicht wurde (Siehe Abbildung 7.2-1 und 7.2-2).

Um für die Tests einen Vergleichswert festzulegen, wird die Übertragungsrate der Hosts gemessen, ohne dass eine Firewall zum Einsatz kommt. Die Hosts sind dann über eine 100 Megabit Leitung per Kreuzkabel direkt miteinander verbunden.

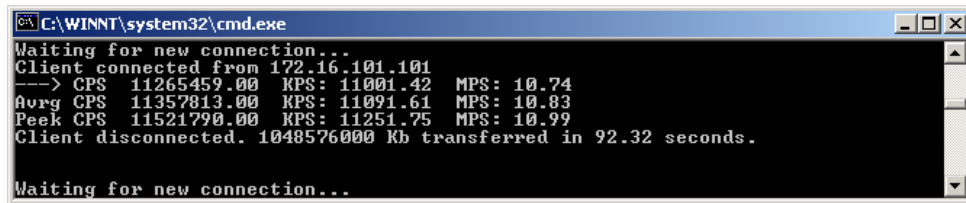
Das Ergebnis wird als 100% Marke für die Berechnung der Verarbeitungsverzögerung herangezogen.

Alle Messungen werden sechsmal vollzogen und das arithmetische Mittel aus deren Ergebnissen entnommen.

Die Messungen ergaben folgende Werte in Megabyte pro Sekunde (MB/s):

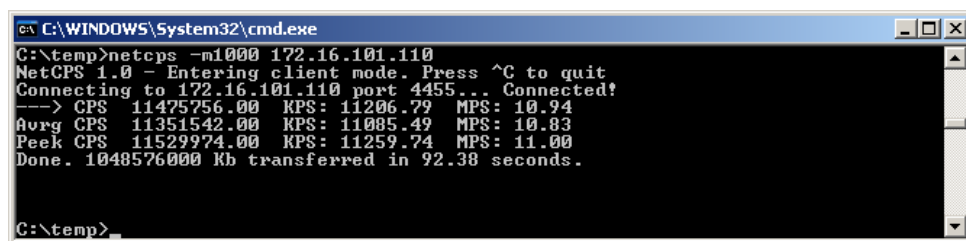
10,76/10,77/10,82/10,83/10,84/10,86

Daraus entsteht ein Mittelwert von: 10,81 MB/s



```
C:\WINNT\system32\cmd.exe
Waiting for new connection...
Client connected from 172.16.101.101
--> CPS 11265459.00 KPS: 11001.42 MPS: 10.74
Avg CPS 11357813.00 KPS: 11091.61 MPS: 10.83
Peak CPS 11521790.00 KPS: 11251.75 MPS: 10.99
Client disconnected. 1048576000 Kb transferred in 92.32 seconds.
Waiting for new connection...
```

Abbildung 7.2-1 Messung der Übertragungsrate ohne Firewall, 1000MB, Serverseite



```
C:\WINDOWS\System32\cmd.exe
C:\temp>netcps -m1000 172.16.101.110
NetCPS 1.0 - Entering client mode. Press ^C to quit
Connecting to 172.16.101.110 port 4455... Connected!
--> CPS 11475756.00 KPS: 11206.79 MPS: 10.94
Avg CPS 11351542.00 KPS: 11085.49 MPS: 10.83
Peak CPS 11529974.00 KPS: 11259.74 MPS: 11.00
Done. 1048576000 Kb transferred in 92.38 seconds.
C:\temp>
```

Abbildung 7.2-2 Messung der Übertragungsrate ohne Firewall, 1000MB, Clientseite

Die Übertragungsrate von 10,81 MB/s ist die Referenz zur Berechnung der Verarbeitungsverzögerung.

Die tatsächlich erreichte Übertragungsrate der Probanden wird mit der Referenz verglichen. Das Ergebnis gibt Aufschluss darüber, wie viel Leistung durch die Verarbeitung der Firewall verloren gegangen ist.

7.3 Testergebnisse

7.3.1 IPCOP v1.3.0 Sicherheitsupdate 9

IPCop ist eine auf Linux basierende Netzwerkfirewall. Die Paketfilterfunktion dieser Firewall ist mit Hilfe von IP-Tables implementiert. Zusätzlich besteht die Möglichkeit, IPCop im IDS-Modus zu betreiben, der über das integrierte Snort realisiert wird. Die Tests werden durchgeführt während die Dienste *DHCP*, *IDS*, *VPN*, Cron-Server, Webproxy, DNS-Proxy, Kernel-Protokollierungs-Server, Web-Server und Secure Shell Server gestartet sind. Snort als *IDS* wurde aktiviert.

7.3.1.1 GUI

Eine GUI ist in Form einer dynamischen HTML-Oberfläche, die über einen Browser abgerufen werden kann, vorhanden. Aufrufen kann man die Oberfläche über den Port 81. Mit dieser Implementierung ist die Firewall von jedem browserfähigen Rechner administrierbar. (Siehe Abbildung 7.3-1)

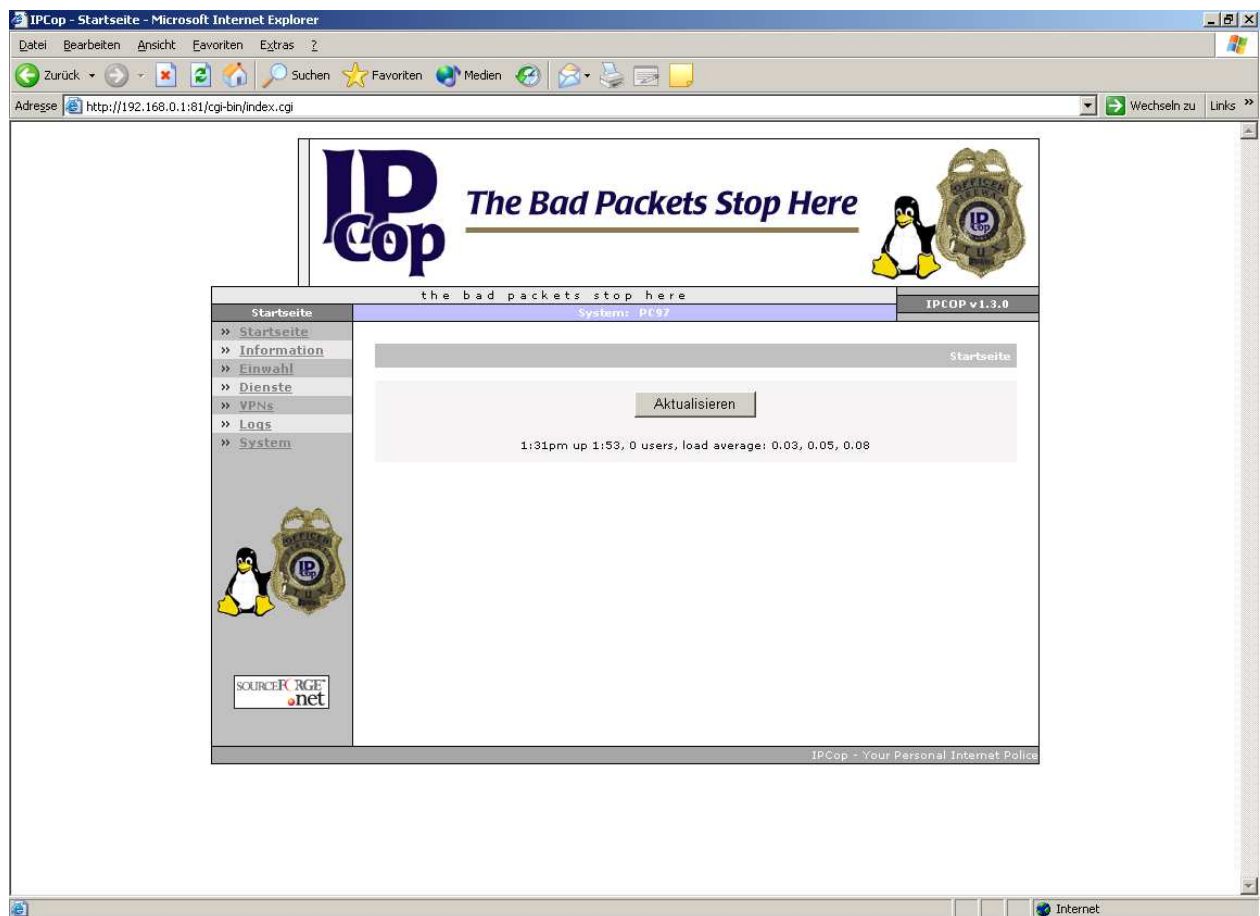


Abbildung 7.3-1 IPCop GUI

7.3.1.2 Betriebssystem

Die Software basiert auf einem mitgelieferten Linuxbetriebssystem mit der Kernelversion 2.4.24. (Siehe Abbildung 7.3-2)

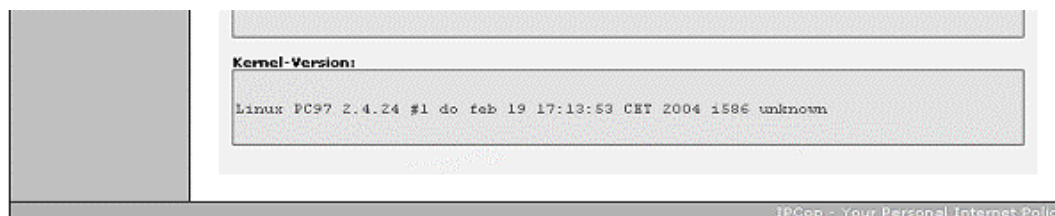


Abbildung 7.3-2 IPCop Betriebssystem

7.3.1.3 Installationsroutine

Für den IPCop gibt es als Download ein ISO-Image, welches aus dem Internet heruntergeladen werden kann [8]. Mit diesem Image kann eine Bootfähige CD-Rom erstellt werden. Bei einem Neustart mit dieser eingelegten CD startet die Installationsroutine automatisch. Sie löscht alle vorhandenen Daten auf der Festplatte und legt die für die Firewall erforderlichen Partitionen an und installiert die erforderlichen Programmteile.

7.3.1.4 Installationsdauer

Die Installation per CD dauerte 3 Minuten (2 min 55 sec.)

7.3.1.5 Konfigurationsdauer

Beim erstmaligen Einrichten werden 5 min. (5 min. 7 sec.) benötigt. Es gibt die Möglichkeit, die Konfigurationseinstellungen auf eine Diskette zu speichern. Diese kann bei der nächsten Installation verwendet werden, falls die zuvor gespeicherte Konfiguration von nutzen ist.

7.3.1.6 Userverwaltung mit Zugriffsregeln

Eine Userverwaltung mit Zugriffsregeln, die den Netzverkehr für User beschränken, bietet der IPCop nicht an.

7.3.1.7 Konfigurierbares Paketfiltersystem

Das Paketfiltersystem von IPCop ist konfigurierbar.

Im Verzeichnis ../etc/rc.d (Siehe Abbildung 7.3-3) befindet sich die Datei rc.firewall mit der ein Script ausgeführt wird, welches die Paketfilteroptionen für die IP-Tables konfiguriert. In der Abbildung 7.3-4 wird gezeigt wie die Ports 137 bis 139 in beide Richtungen gesperrt werden. Die Abbildung 7.3-5 zeigt wie die Scan-Angriffe Xmas-Scan und Null-Scan verhindert werden, indem Pakete, bei denen alle Flags oder keine Flags gesetzt sind, gesondert überprüft werden.


```

192.168.26.13 - PuTTY
init.d rc.alcatelusb rc.firewall rc.isdn rc.netaddress.down rc.network rc.red
root@ipcop2:/etc/rc.d # ls -all
total 59
drwxr-xr-x  4 root  root    1024 Sep  9 14:43 .
drwxr-xr-x 28 root  root    2048 Sep  9 14:43 ..
drwxr-xr-x  2 root  root    1024 Sep  9 14:41 helper
drwxr-xr-x  2 root  root    1024 Apr 16  2003 init.d
-rwxr-xr-x  1 root  root    3312 Apr 16  2003 ipsec
-rwxr-xr-x  1 root  root    601 Apr 16  2003 rc.alcatelusb
-rwxr-xr-x  1 root  root    2878 Jan  8  2004 rc.eciadsl
-rwxr-xr-x  1 root  root    6901 Sep 14  09:07 rc.firewall
-rwxr-xr-x  1 root  root    590 Apr 16  2003 rc.halt
-rwxr-xr-x  1 root  root    738 Apr 16  2003 rc.isdn
-rwxr-xr-x  1 root  root     11 Apr 16  2003 rc.local
-rwxr-xr-x  1 root  root    298 Apr 16  2003 rc.netaddress.down
-rwxr-xr-x  1 root  root    1086 Sep  9 14:15 rc.netaddress.up
-rwxr-xr-x  1 root  root    1628 Apr 16  2003 rc.network
-rwxr-xr-x  1 root  root    1048 Jul 31  2003 rc.pulsardsl
-rwsr-x--  1 root  nobody 18657 Jul 31  2003 rc.red
-rwxr-xr-x  1 root  root    4835 Apr 16  2003 rc.sysinit
-rwxr-xr-x  1 root  root    3635 Jan  8  2004 rc.updatered
root@ipcop2:/etc/rc.d #

```

Abbildung 7.3-3 IPCop Konfigurationsdateien für den Paketfilter

```

192.168.26.13 - PuTTY
echo 1024 > /proc/sys/net/ipv4/tcp_max_syn_backlog

# Flush all rules and delete all custom chains
/sbin/iptables -F
/sbin/iptables -t nat -F
/sbin/iptables -X
/sbin/iptables -t nat -X

# Set up policies
/sbin/iptables -P INPUT DROP
/sbin/iptables -P FORWARD DROP
/sbin/iptables -P OUTPUT ACCEPT

# Hier sperren wir die Ports 137-139
/sbin/iptables -A INPUT -p tcp --destination-port 137:139 -j DROP
/sbin/iptables -A INPUT -p udp --destination-port 137:139 -j DROP

/sbin/iptables -A OUTPUT -p tcp --destination-port 137:139 -j DROP
/sbin/iptables -A OUTPUT -p udp --destination-port 137:139 -j DROP

# This chain will log, then DROPS "Xmas" and Null packets which might
-- INSERT --

```

Abbildung 7.3-4 IPCop sperren der Ports 137-139

```

192.168.26.13 - PuTTY
/sbin/iptables -A PSCAN -p udp -m limit --limit 10/minute -j LOG --log-prefix "UDP Scan
? "
/sbin/iptables -A PSCAN -p icmp -m limit --limit 10/minute -j LOG --log-prefix "ICMP Sca
n? "
/sbin/iptables -A PSCAN -f -m limit --limit 10/minute -j LOG --log-prefix "FRAG Sca
n? "
/sbin/iptables -A PSCAN -j DROP

# Disallow packets frequently used by port-scanners, XMas and Null
/sbin/iptables -A INPUT -p tcp --tcp-flags ALL ALL -j PSCAN
/sbin/iptables -A FORWARD -p tcp --tcp-flags ALL ALL -j PSCAN
/sbin/iptables -A INPUT -p tcp --tcp-flags ALL NONE -j PSCAN
/sbin/iptables -A FORWARD -p tcp --tcp-flags ALL NONE -j PSCAN
}

iptables_red() {
/sbin/iptables -F RED
/sbin/iptables -t nat -F RED

# PPPoE / PPTP Device
if [ "$IFACE" != "" ]; then
-- INSERT --

```

Abbildung 7.3-5 IPCop Verhindern von Xmas- und Null-Scan

7.3.1.8 Fernadministration

Der Fernwartungszugang kann individuell konfiguriert werden. Einstellungsmöglichkeiten gibt es beim Protokoll, der Quelladresse, der Ziel-Adresse, und des Zielports (Siehe Abbildung 7.3-6). Zusätzlich kann die Fernwartung über SSH erfolgen (Siehe Abbildung 7.3-7).

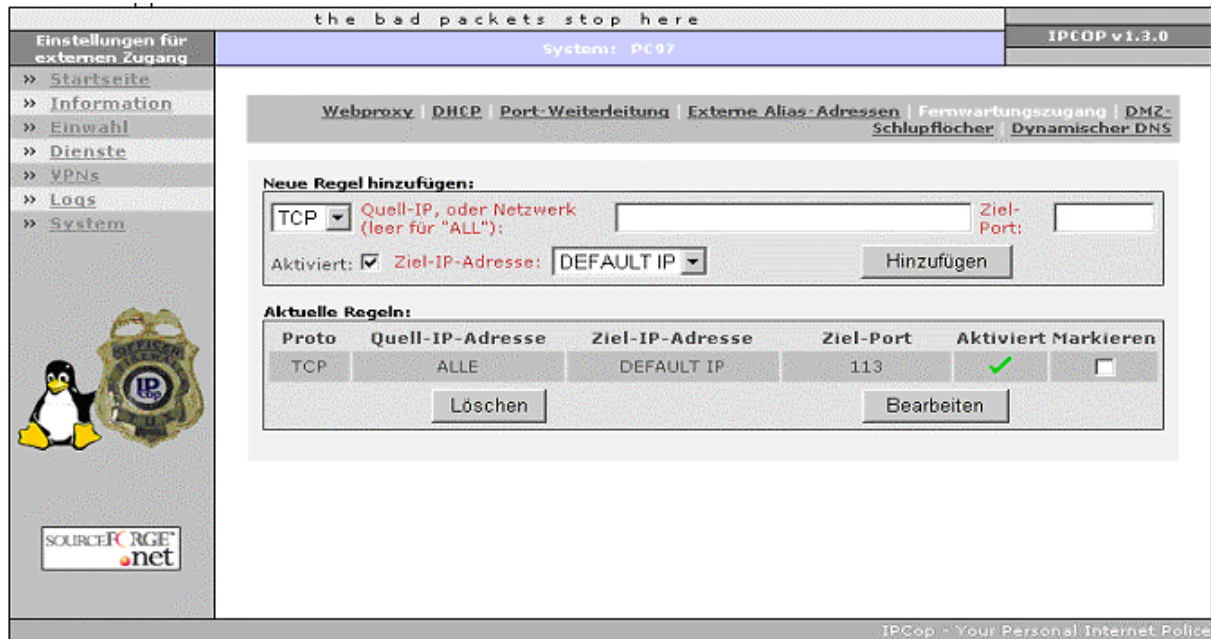


Abbildung 7.3-6 IPCop Fernwartungszugang

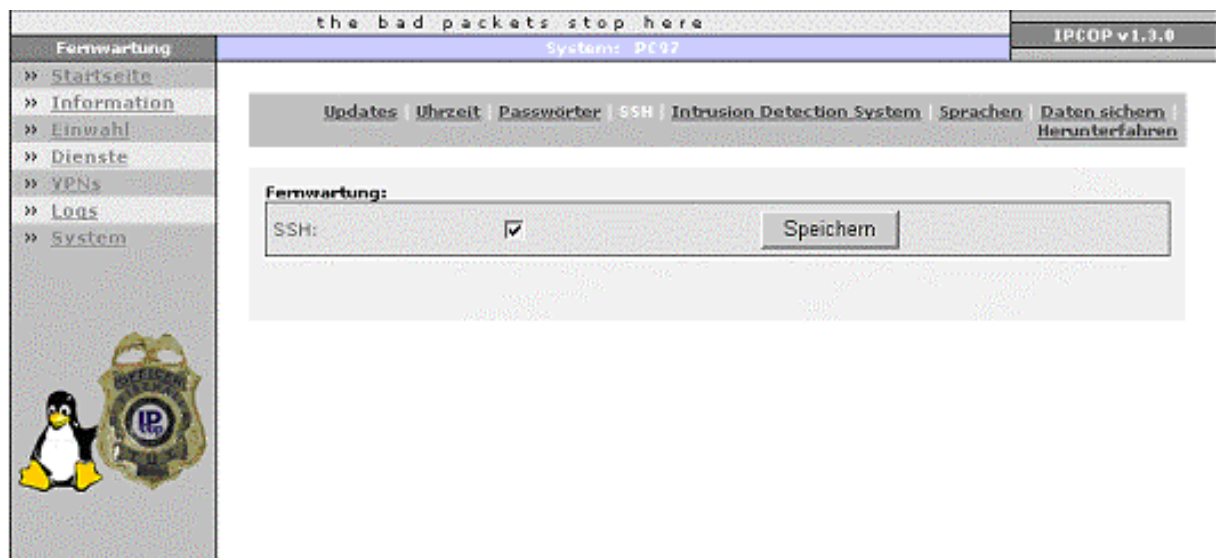


Abbildung 7.3-7 IPCop SSH

7.3.1.9 Updateassistent

Es gibt einen integrierten Updateassistenten, der jedes mal bei Aufruf der Oberfläche anzeigt, ob ein neues Update verfügbar ist. Diese Datei muss separat aus dem Netz heruntergeladen werden, kann dann aber über den Updateassistenten ausgewählt und installiert werden (Siehe Abbildung 7.3-8).

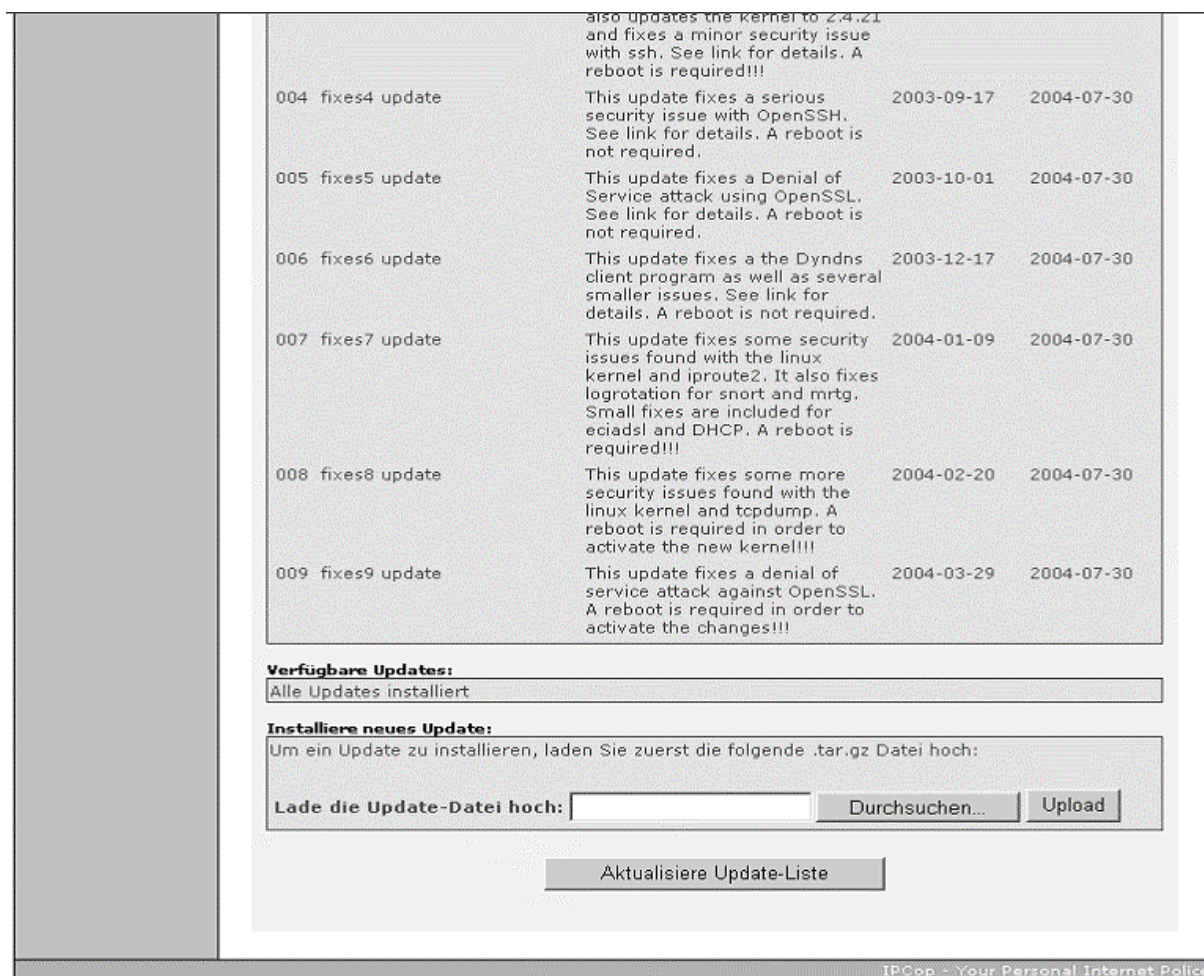


Abbildung 7.3-8 IPCop Updateassistent

7.3.1.10 Firewallklasse

Es handelt sich um einen Paketfilter, der durch IP-Tables Implementiert wird. IP-Tables unterstützt das zustandsbasierte Überwachen der Pakete.

7.3.1.11 Routingfunktion

Eine Routingfunktion ist vorhanden. Der IPCop verwaltet bis zu 3 Netzwerkadapter.

7.3.1.12 DNS

DNS wird wahlweise durch eine Weiterleitung der Anfragen oder durch einen DNS-Proxy unterstützt.

7.3.1.13 DHCP

DHCP wird unterstützt. Einstellungen in der Anfangs- und Endadresse des IP-Adressraums, der Adresse eines primären und sekundären DNS Servers, ein Domain Name Suffix und die Adresse eines WINS-Servers sind möglich (Siehe Abbildung 7.3-9).

the bad packets stop here

system: PC97

IPCop v1.3.0

DHCP-Konfiguration

» Startseite
» Information
» Einwahl
» Dienste
» YPNs
» Logs
» System

Webproxy | DHCP | Port-Weiterleitung | Externe Alias-Adressen | Fernwartungszugang | DMZ-Schlupflocher | Dynamischer DNS

DHCP-Server-Parameter:

Anfangsadresse: 192.168.0.2 Endadresse: 192.168.0.50
Primärer DNS: 192.168.0.1 Sekundärer DNS:
Haltezeit-Voreinstellung in min: 60 Max. Haltezeit in min: 120
Domain-Name-Suffix: WINS-Server-Adresse:
Aktiviert:

Dieses Feld kann leer bleiben.

Neue Zuordnung definieren

MAC-Adresse IP-Adresse
Aktiviert:

Aktuelle feste Zuordnungen

MAC-Adresse	IP-Adresse	Aktiviert	Markieren
<input type="button" value="Löschen"/>		<input type="button" value="Bearbeiten"/>	

Abbildung 7.3-9 IPCop DHCP

7.3.1.14 Internetproxy

Ein Internetproxy wird unterstützt. Einstellungen in der Cache-Größe, Objektgröße, Benutzername und Passwort sind möglich. Es gibt zusätzlich die Option, den Proxy in einen transparenten Modus zu versetzen, so dass spezielle Browsereinstellungen beim Client vermieden werden können (Siehe Abbildung 7.3-10).

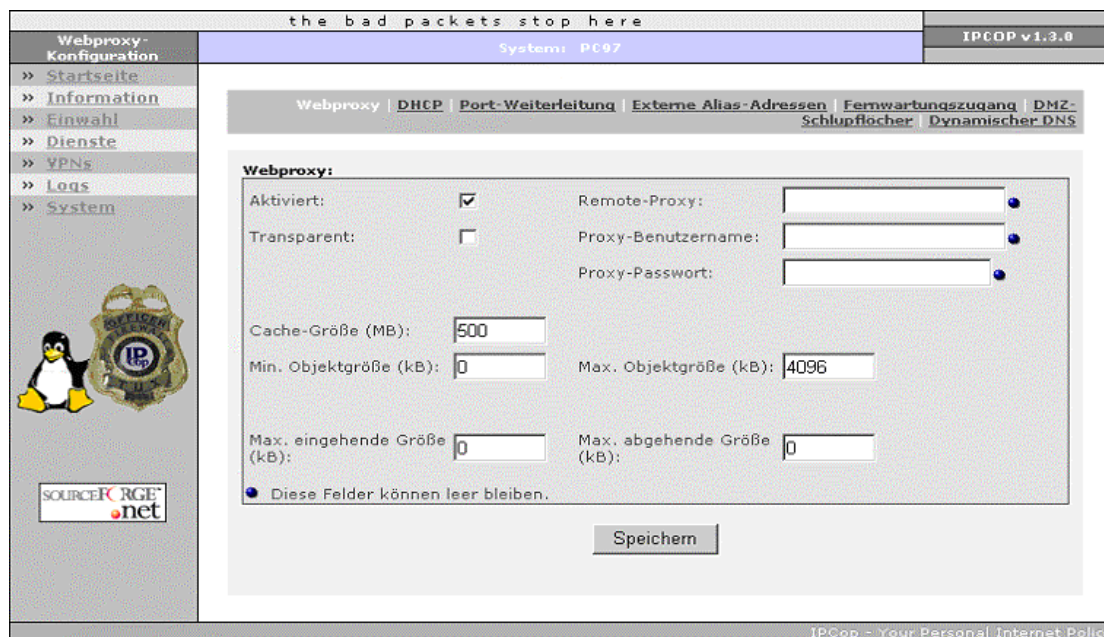


Abbildung 7.3-10 IPCop Webproxy

Für den Webproxy gibt es Auswertungsmöglichkeiten in Form von Graphen, die die Cachetreffer aufzeigen (Siehe Abbildung 7.3-11) und ein Protokoll der Aufrufe von Webseiten und den Rechneradressen, von denen der Aufruf kommt (Siehe Abbildung 7.3-12).

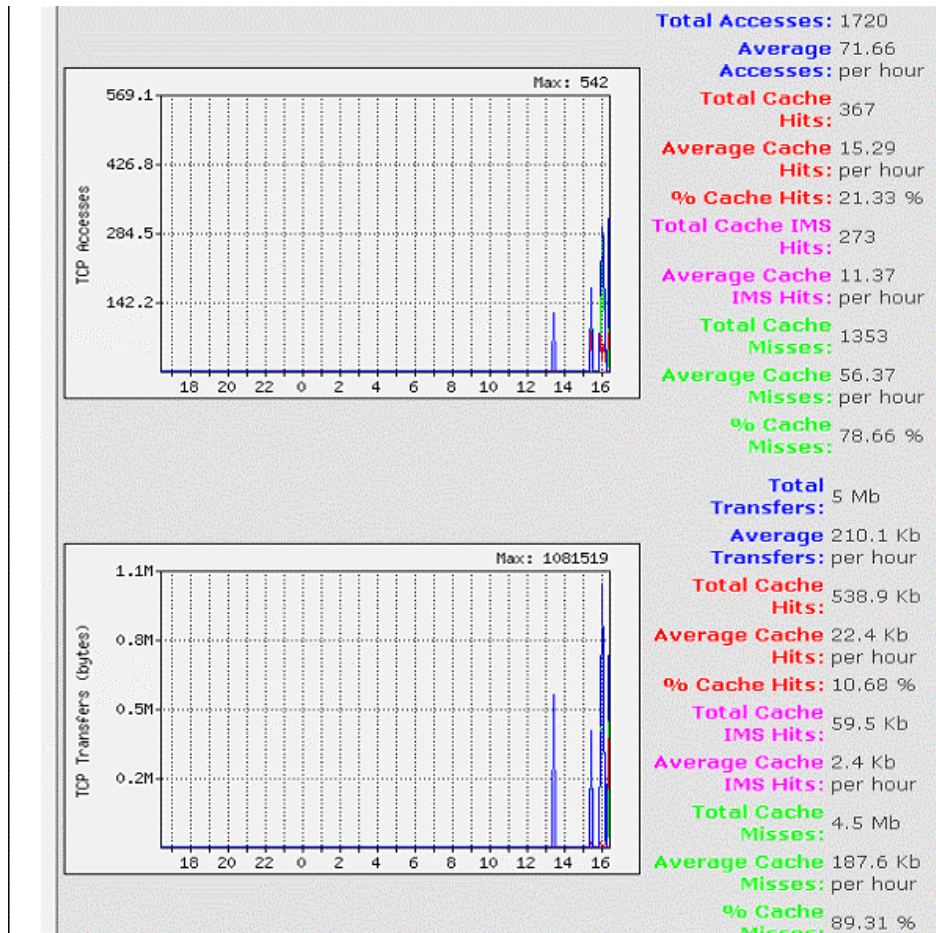


Abbildung 7.3-11 IPCop Webproxy Analyse

the bad packets stop here

System: PC97 IPCOP v1.3.0

Ansicht Proxy-Log

» Startseite
 » Information
 » Einwahl
 » Dienste
 » VPNs
 » Logs
 » System

Andere | Webproxy | Firewall | Intrusion Detection System

Konfiguration

Monat: August Tag: 3
 Quell-IP-Adresse: ALLE

"Ignorieren"-Filter: [.]([gif]p[eg]l[jpg]pnglcssjs)\$ "Ignorieren"-Filter ein:

Voreinstellungen wiederherstellen Aktualisieren Export

Protokoll:

Gesamtanzahl der Websites zum ausgewählten Kriterium August 03: 29

Uhrzeit	Quell-IP-Adresse	Website
13:26:10	192.168.0.2	http://sourceforge.net/sfloqo.php?
13:26:51	192.168.0.2	http://www.heise.de/
13:26:51	192.168.0.2	http://www.heise.de/support/js3.js?
13:26:51	192.168.0.2	http://www.heise.de/RealMedia/ads/adstream_lx.ads/www.heise....
13:26:52	192.168.0.2	http://www.heise.de/RealMedia/ads/adstream_lx.ads/www.heise....
13:26:52	192.168.0.2	http://www.heise.de/RealMedia/ads/adstream_lx.ads/www.heise....
13:26:52	192.168.0.2	http://heise.ivwbox.de/cgi-bin/ivw/CP/homepage/?
13:26:53	192.168.0.2	http://www.heise.de/RealMedia/ads/adstream_lx.ads/www.heise....
13:26:55	192.168.0.2	http://www.heise.de/bilder/49431/0/2
13:26:56	192.168.0.2	http://www.heise.de/ivw-bin/ivw/CP/
13:27:04	192.168.0.2	http://www.qamestar.de/
13:27:05	192.168.0.2	http://www.qamestar.de/cgi-bin/heft.mol

Abbildung 7.3-12 IPCop Webproxy Protokollierung

7.3.1.15 VPN

VPN wird unterstützt. Der IPCop bietet die Möglichkeit mehrere VPN-Verbindungen zuzulassen (Siehe Abbildung 7.3.13) Jede Verbindung kann einzeln verwaltet werden (Siehe Abbildung 7.3.14).

Abbildung 7.3-13 IPCop VPN Steuerung

Name	Status
Nicolas (172.16.101.0/24)	BEENDET

Abbildung 7.3-14 IPCop VPN Verbindungen

7.3.1.16 Verarbeitungsverzögerung

IPCop erreichte in der Übertragungsgeschwindigkeit folgende Messwerte in MB/s:

4,29/4,31/4,32/4,32/4,32/4,33

daraus ergibt sich für den IPCop eine durchschnittliche Übertragungsrate von 4,32 MB/s (Siehe Abbildung 7.3.15).

```

C:\WINDOWS\System32\cmd.exe
C:\temp>netcps -m1000 172.16.101.110
NetCPS 1.0 - Entering client mode. Press ^C to quit
Connecting to 172.16.101.110 port 4455... Connected!
--> CPS 4568615.50 KPS: 4461.54 MPS: 4.36
Avg CPS 4527667.00 KPS: 4421.55 MPS: 4.32
Peak CPS 4587029.00 KPS: 4479.52 MPS: 4.37
Done. 1048576000 Kb transferred in 231.59 seconds.

C:\temp>
    
```

Abbildung 7.3-15 IPCop Ergebnis Übertragungsrate

Damit erreichte der IPCop gegenüber der Referenz eine Übertragungsleistung von:

$$\frac{(4,32MB/s) * 100}{10,81MB/s} = \underline{\underline{39,96\%}}$$

Das ergibt einen Verlust von 60,04% durch die Verarbeitungsverzögerung.

7.3.1.17 Verbindungsprotokollierung

Die Verbindungen von Innen nach Außen und von Außen nach Innen werden protokolliert. Das Protokoll enthält die Informationen über das verwendete Übertragungsprotokoll, den Verbindungsstatus, das Ablaufdatum der Verbindung, die Quelladresse mit Port und die Zieladresse mit Port (Siehe Abbildung 7.3.16). Zusätzlich bietet IPCop noch andere Protokolle für die Firewall und das IDS.

Protokoll	Ablaufdatum (sek.)	Verbindung Status	Original Quell-IP:Port	Original Ziel-IP:Port	Erwartet Quell-IP:Port	Erwartet Ziel-IP:Port	Markiert	Einsatz
tcp (6)	431995	ESTABLISHED	192.168.0.2:1313	192.168.0.1:81	192.168.0.1:81	192.168.0.2:1313	[ASSURED]	1
tcp (6)	431992	ESTABLISHED	192.168.0.2:1252	172.16.101.211:445	172.16.101.211:445	172.16.101.101:1252	[ASSURED]	1
tcp (6)	97	TIME_WAIT	192.168.0.2:1308	192.168.0.1:81	192.168.0.1:81	192.168.0.2:1308	[ASSURED]	1
tcp (6)	115	TIME_WAIT	192.168.0.2:1311	66.35.250.203:80	66.35.250.203:80	172.16.101.101:1311	[ASSURED]	1
tcp (6)	115	TIME_WAIT	192.168.0.2:1310	192.168.0.1:81	192.168.0.1:81	192.168.0.2:1310	[ASSURED]	1
tcp (6)	98	TIME_WAIT	192.168.0.2:1309	66.35.250.203:80	66.35.250.203:80	172.16.101.101:1309	[ASSURED]	1


Abbildung 7.3-16 IPCop Verbindungsprotokollierung


Das Protokoll der Firewall (Siehe Abbildung 7.3-17) zeigt die erfolgreich gefilterten Pakete. Es enthält Informationen über den genauen Zeitpunkt, die Richtung (von innen oder außen), das betroffene Interface, das verwendete Protokoll und Angaben über die Quelladresse mit Port und Zieladresse mit Port.

the bad packets stop here
IPCop v1.3.0

Firewall-Protokoll

- » Startseite
- » Information
- » Einwahl
- » Dienste
- » VPNs
- » Logs
- » System





[Andere](#) | [Webproxy](#) | [Firewall](#) | [Intrusion Detection System](#)

Konfiguration

Monat: August Tag: 3
 << >>
 Aktualisieren
 Export

Firewall-Protokoll:

Gesamtanzahl der Firewall-Treffer für August 3: 400

Uhrzeit	Verknüpfung	Iface	Proto	Quelle	Quell-Port	Ziel	Ziel-Port
12:21:43	INPUT	eth1	UDP	<input type="checkbox"/> 172.16.101.211	137 (NETBIOS-NS)	<input type="checkbox"/> 172.16.101.255	137 (NETBIOS-NS)
12:21:48	INPUT	eth1	UDP	<input type="checkbox"/> 172.16.101.109	138 (NETBIOS-DGM)	<input type="checkbox"/> 172.16.101.255	138 (NETBIOS-DGM)
12:21:57	INPUT	eth1	UDP	<input type="checkbox"/> 172.16.101.109	138 (NETBIOS-DGM)	<input type="checkbox"/> 172.16.101.255	138 (NETBIOS-DGM)
12:22:01	INPUT	eth1	UDP	<input type="checkbox"/> 172.16.101.109	138 (NETBIOS-DGM)	<input type="checkbox"/> 172.16.101.255	138 (NETBIOS-DGM)
12:22:07	INPUT	eth1	UDP	<input type="checkbox"/> 172.16.101.211	137 (NETBIOS-NS)	<input type="checkbox"/> 172.16.101.255	137 (NETBIOS-NS)
12:26:06	INPUT	eth1	UDP	<input type="checkbox"/> 172.16.101.210	137 (NETBIOS-NS)	<input type="checkbox"/> 172.16.101.255	137 (NETBIOS-NS)

Abbildung 7.3-17 IPCop Firewall Protokollierung

Das Protokoll des IDS (Siehe Abbildung 7.3-18) erfasst alle erfolgreich angewendeten Regeln und versucht mögliche Angriffe zu klassifizieren.

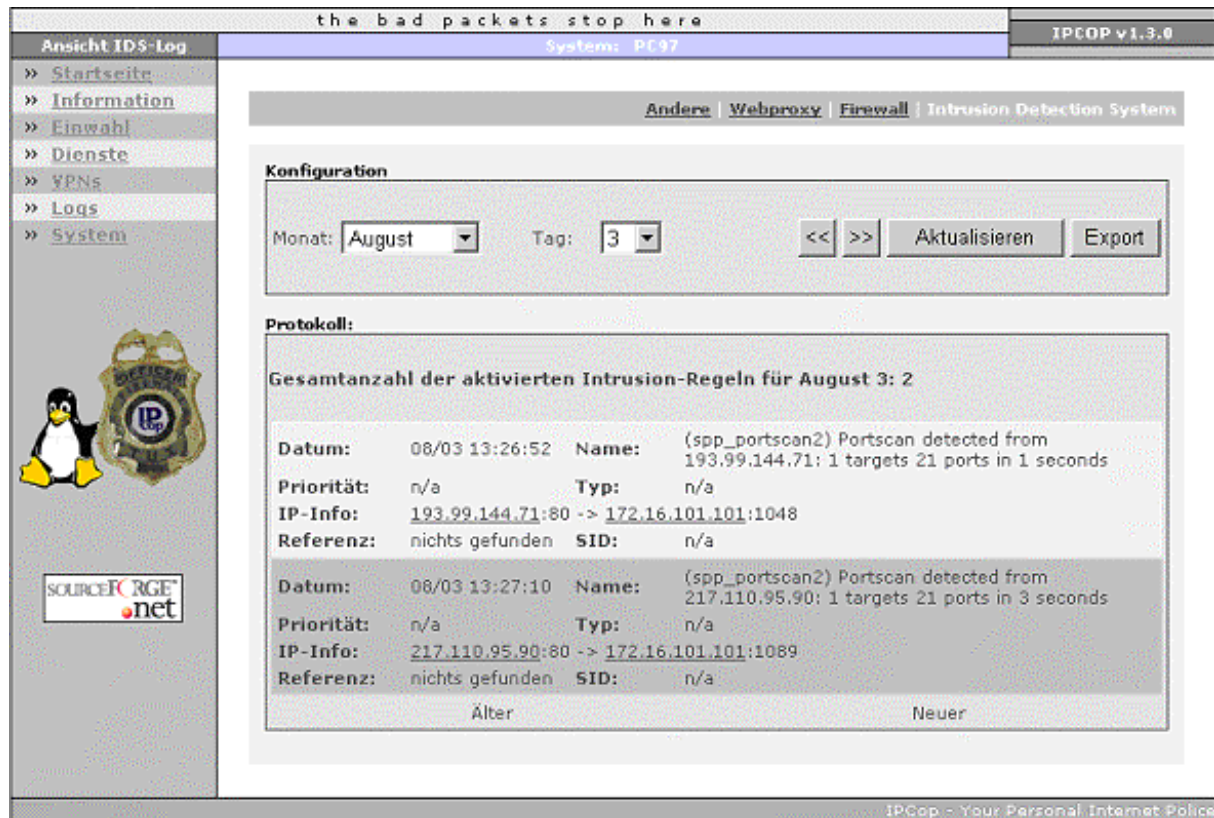


Abbildung 7.3-18 IPCop IDS Protokoll

Das Protokoll des Webproxies (Siehe Abbildung 7.3-12) gibt eine Übersicht über die ausgehenden Verbindungen mit dem *hypertext transfer protocol*. Es gibt Aufschluss darüber, wann welche Internetseiten von welcher internen IP Adresse aus aufgerufen wurden.

7.3.1.18 DMZ

Eine DMZ wird unterstützt. IPCop teilt die Sicherheitsbereiche nach Farben auf. Grün für das sichere Netz, orange für die DMZ und rot für das unsichere Netz.

7.3.1.19 IDS

Snort ist als IDS integriert und kann aktiviert werden (Siehe Abbildung 7.3-19).

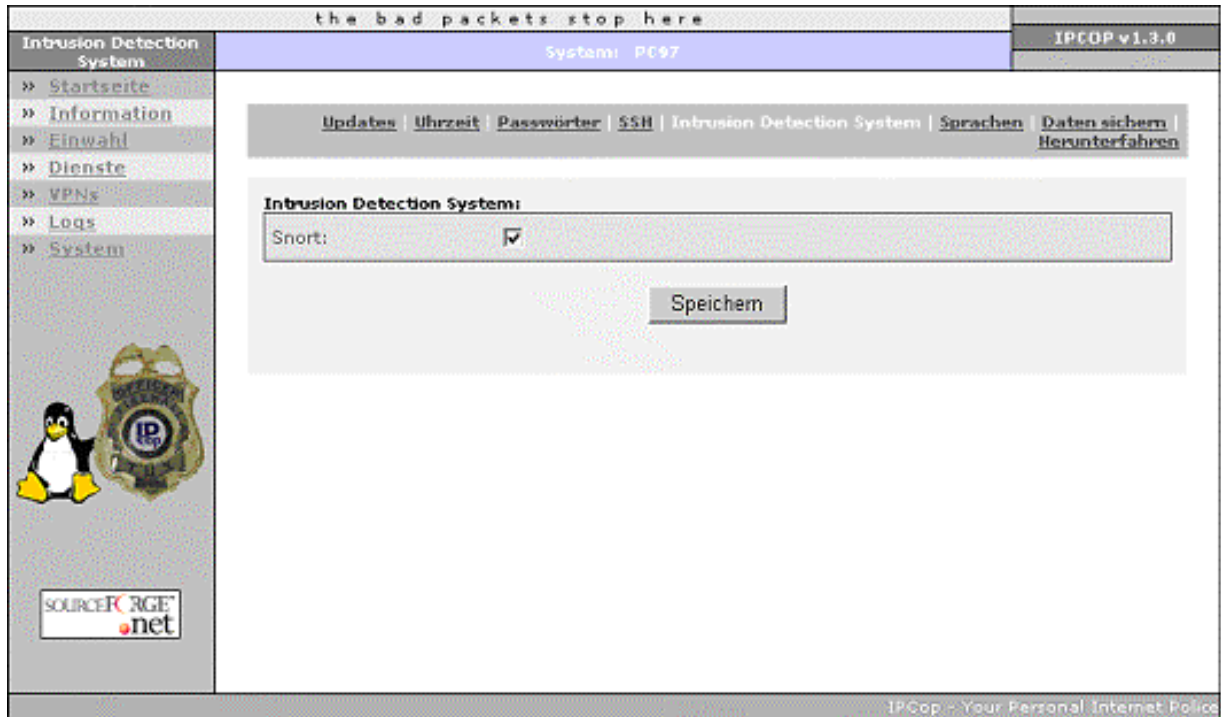


Abbildung 7.3-19 IPCop Snort

7.3.1.20 Portscanversuch

Die Scans TCP-Connect-Scan, TCP-Syn-Scan, TCP-Ack-Scan, Null-Scan, Xmas-Tree-Scan, Fin-Stealth-Scan und der UDP-Scan entdeckten keine offenen Ports im Nummernraum von 1-32768. Der Syn-Stealth-Scan klassifizierte lediglich einen inaktiven Dienst auf Port 113 (Siehe Abbildung 3.7-20).

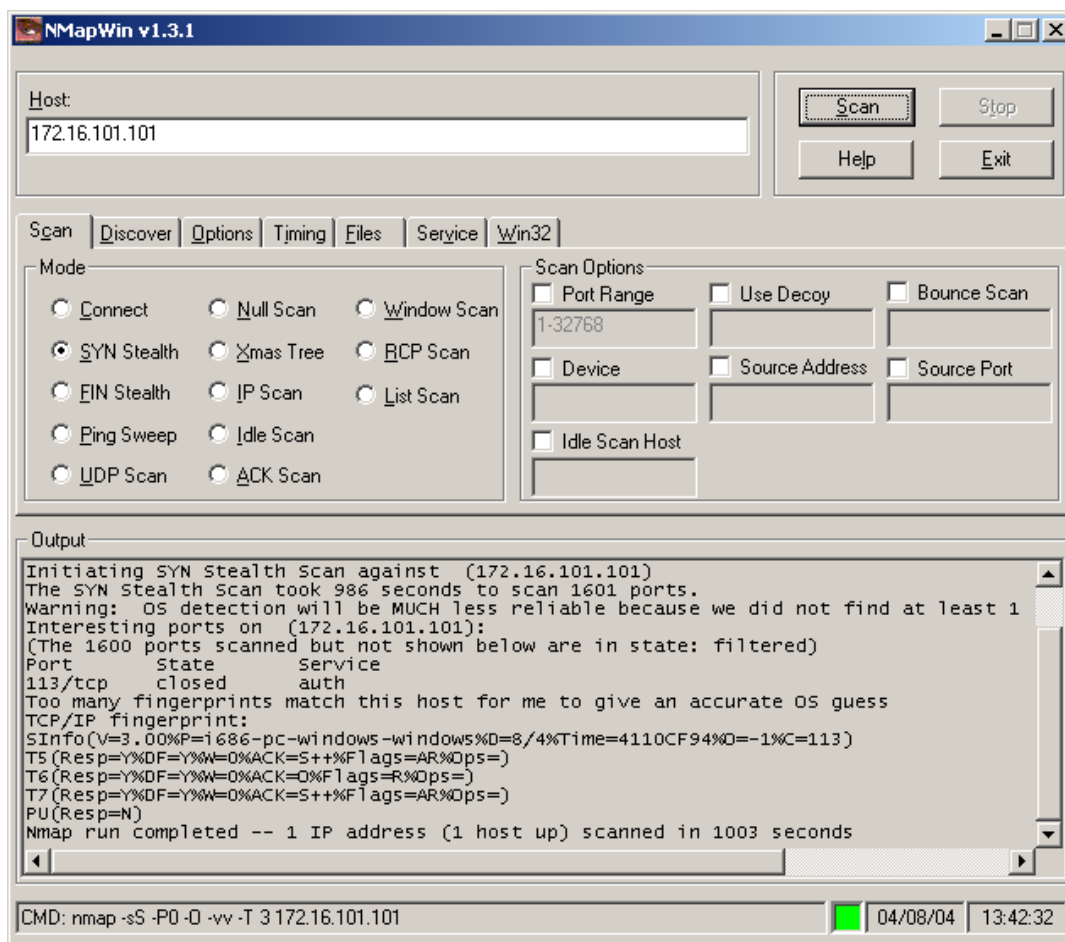


Abbildung 7.3-20 IPCop NMAP TCP-Syn-Scan

7.3.1.21 Betriebssystemidentifikation

NMAP konnte mit den durch den Null-Scan gelieferten Parametern kein positives Ergebnis erzielen. Das Betriebssystem konnte nicht identifiziert werden (Siehe Abbildung 7.3-21).

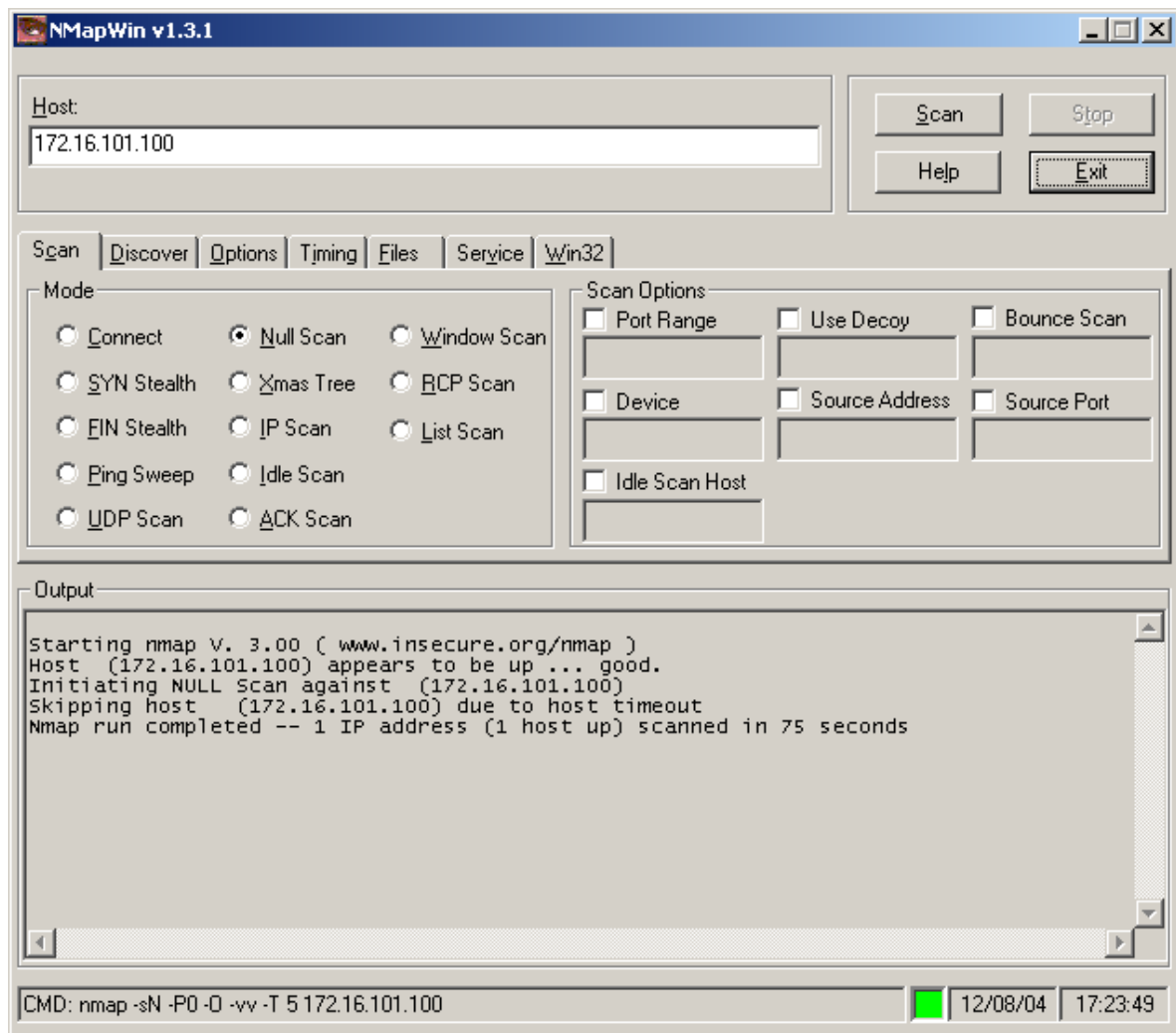


Abbildung 7.3-21 IPCop NMAP Null-Scan

7.3.1.22 Scan-Angriffserkennung

Der Null-Scan wurde erkannt und in der Protokolldatei des IDS festgehalten (Siehe Abbildung 7.3-22).

Datum:	08/12 17:35:53	Name:	(spp_stream4) STEALTH ACTIVITY (NULL scan) detection
Priorität:	n/a	Typ:	n/a
IP-Info:	172.16.101.110:35118 -> 172.16.101.100:1019		
Referenz:	nichts gefunden	SID:	n/a
Datum:	08/12 17:35:53	Name:	(spp_stream4) STEALTH ACTIVITY (NULL scan) detection
Priorität:	n/a	Typ:	n/a
IP-Info:	172.16.101.110:35118 -> 172.16.101.100:1401		
Referenz:	nichts gefunden	SID:	n/a
Datum:	08/12 17:35:53	Name:	(spp_stream4) STEALTH ACTIVITY (NULL scan) detection
Priorität:	n/a	Typ:	n/a
IP-Info:	172.16.101.110:35118 -> 172.16.101.100:528		
Referenz:	nichts gefunden	SID:	n/a
Datum:	08/12 17:35:53	Name:	(spp_stream4) STEALTH ACTIVITY (NULL scan) detection
Priorität:	n/a	Typ:	n/a
IP-Info:	172.16.101.110:35118 -> 172.16.101.100:3001		
Referenz:	nichts gefunden	SID:	n/a
Datum:	08/12 17:35:53	Name:	(spp_stream4) STEALTH ACTIVITY (NULL scan) detection
Priorität:	n/a	Typ:	n/a
IP-Info:	172.16.101.110:35118 -> 172.16.101.100:610		
Referenz:	nichts gefunden	SID:	n/a
Datum:	08/12 17:35:53	Name:	(spp_stream4) STEALTH ACTIVITY (NULL scan) detection
Priorität:	n/a	Typ:	n/a
IP-Info:	172.16.101.110:35118 -> 172.16.101.100:1492		
Referenz:	nichts gefunden	SID:	n/a
Datum:	08/12 17:35:53	Name:	(spp_stream4) STEALTH ACTIVITY (NULL scan) detection
Priorität:	n/a	Typ:	n/a
IP-Info:	172.16.101.110:35118 -> 172.16.101.100:17007		
Referenz:	nichts gefunden	SID:	n/a
		Älter	Neuer

IPCop - Your Personal Internet Police

Abbildung 7.3-22 IPCop Angriffserkennung

7.3.1.23 Feedback bei Angriffsversuch

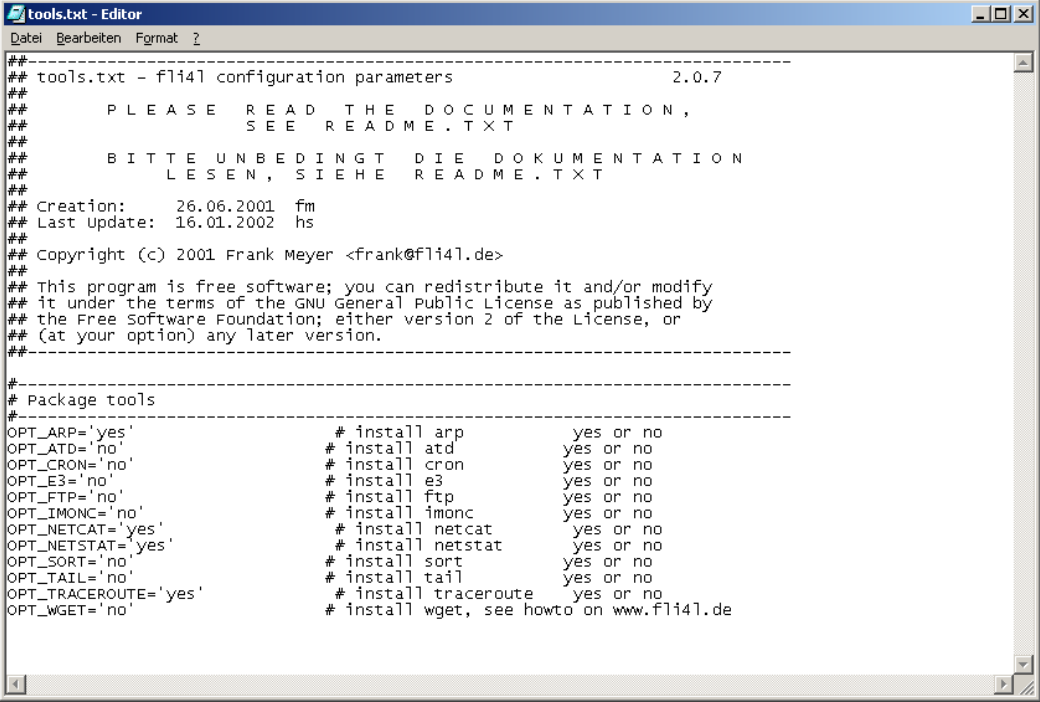
Der IPCop gibt kein akustisches oder visuelles Signal aus, wenn ein Portscan stattfindet und erkannt wird.

7.3.2 Fli4l 2.0.8

Der Fli4l enthält einen Paketfilter auf Linux/Unix Basis, der in einigen Bereichen konfiguriert werden kann. Die Basissoftware und Zusatzmodule werden als komprimierte Dateiarhive der Form tar.gz angeboten und können zum Basisverzeichnis modular hinzugefügt werden.

Folgende Pakete werden für diesen Test installiert:

- Basis: Basispaket für den Paketfilter Fli4l
- DHCP: Zusatzpaket zum Betreiben eines DHCP-Servers
- proxy: Zusatzpaket zum Betreiben eines HTTP Proxies
- httpd: Zusatzpaket zum Betreiben eines Mini-Webservers
- inet: Zusatzpaket zum Betreiben der Services FTP, SSH und Telnet
- tools: Zusatzpaket für Tools zur Administration des Fli4l (Siehe Abbildung 7.3-23)



```
##-----##
## tools.txt - fli4l configuration parameters                2.0.7
##
##      PLEASE READ THE DOCUMENTATION,
##      SEE README.TXT
##
##      BITTE UNBEDINGT DIE DOKUMENTATION
##      LESEN, SIEHE README.TXT
##
## Creation:      26.06.2001  fm
## Last Update:  16.01.2002  hs
##
## Copyright (c) 2001 Frank Meyer <frank@fli4l.de>
##
## This program is free software; you can redistribute it and/or modify
## it under the terms of the GNU General Public License as published by
## the Free Software Foundation; either version 2 of the License, or
## (at your option) any later version.
##-----##
#-----#
# Package tools
#-----#
OPT_ARP='yes'      # install arp          yes or no
OPT_ATD='no'       # install atd           yes or no
OPT_CRON='no'      # install cron          yes or no
OPT_E3='no'        # install e3            yes or no
OPT_FTP='no'       # install ftp           yes or no
OPT_IMONC='no'     # install imonc         yes or no
OPT_NETCAT='yes'   # install netcat        yes or no
OPT_NETSTAT='yes'  # install netstat       yes or no
OPT_SORT='no'      # install sort          yes or no
OPT_TAIL='no'      # install tail          yes or no
OPT_TRACEROUTE='yes' # install traceroute   yes or no
OPT_WGET='no'      # install wget, see howto on www.fli4l.de
```

Abbildung 7.3-23 Fli4l Zusatzpakete zur Installation

7.3.2.1 GUI

Der Fli4l hat 2 Möglichkeiten für eine GUI. Einmal in Form der von Frank Meyer und Nico Wallmeyer programmierten Oberfläche Imonc für Windows 2.0.7c (Siehe Abbildung 7.3-24 und 7.3.25) oder über das Webinterface, das über den zusätzlich installierten Miniwebserver bereitgestellt wird (Siehe Abbildung 7.3-26).

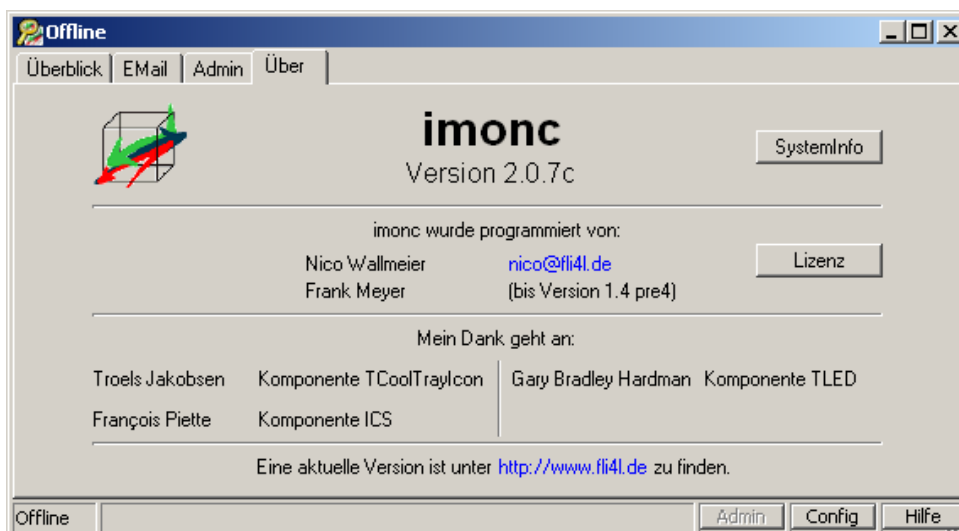


Abbildung 7.3-24 Fli4l Imonc

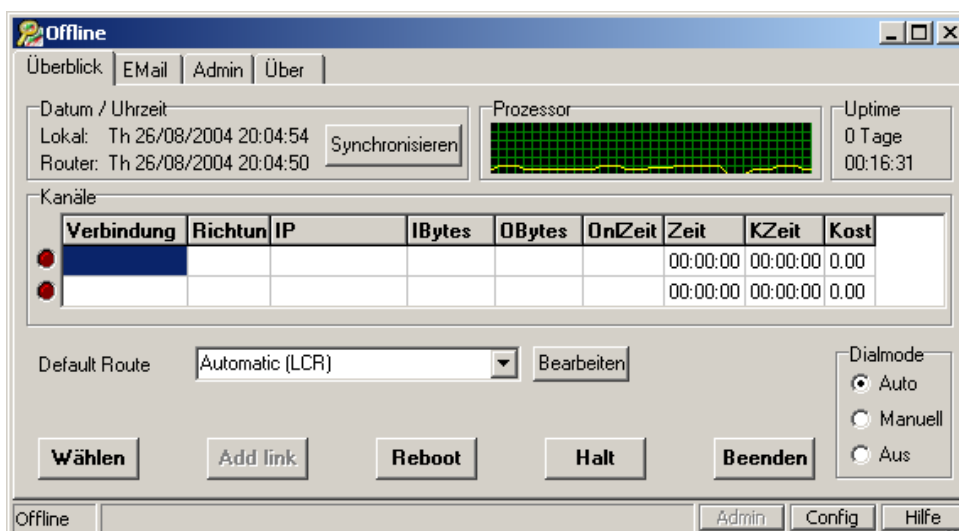


Abbildung 7.3-25 Fli4l Imonc Überblick

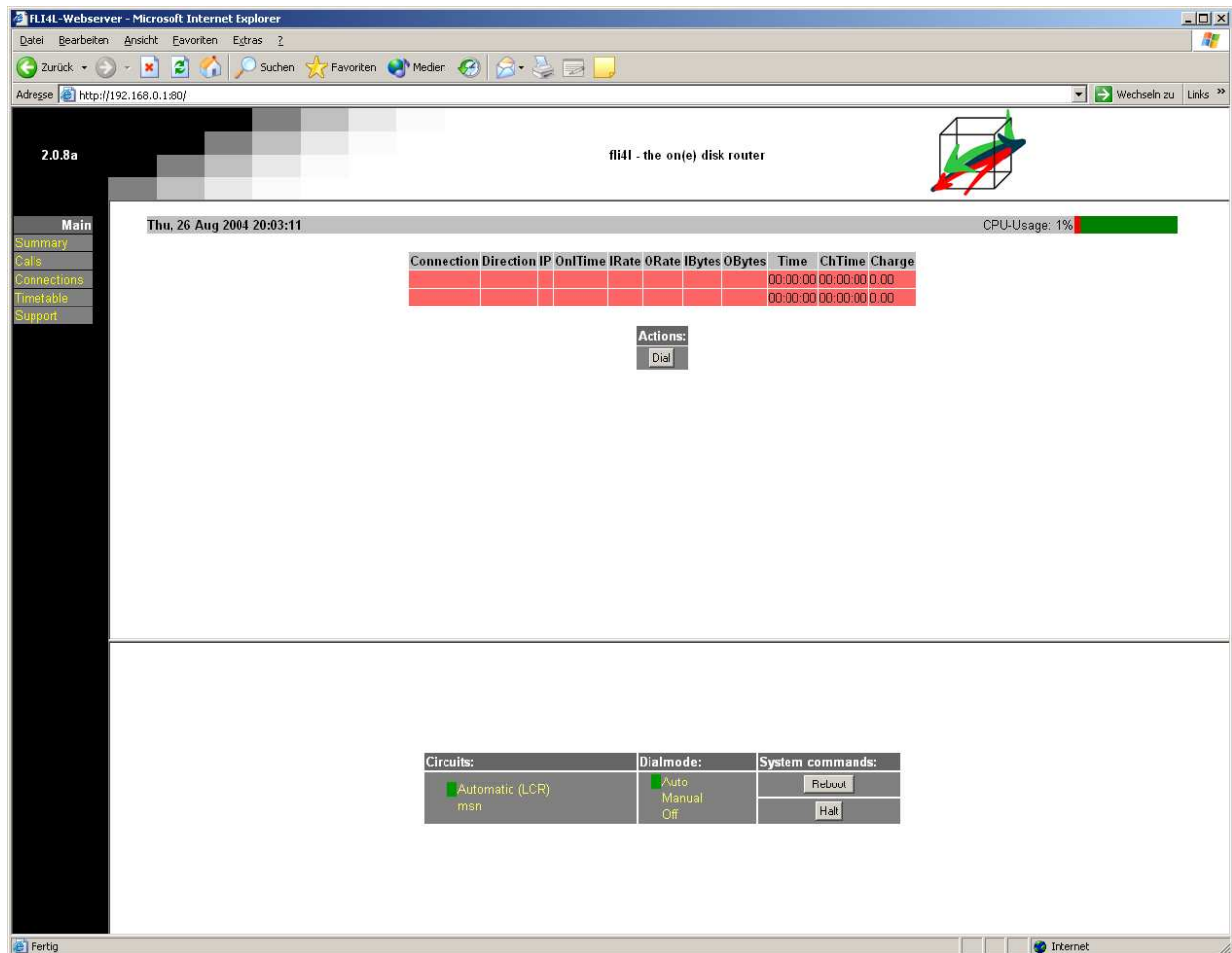


Abbildung 7.3-26 Fli4l Webinterface

7.3.2.2 Betriebssystem

Der Fli4l basiert auf einem mitgelieferten Linuxbetriebssystem mit einem reduzierten Kernel der Version 2.2.22. [7]

7.3.2.3 Installationsroutine

Der fli4l enthält keine übliche Installationsroutine. Die Software wird als komprimiertes Dateiarchiv im tar.gz-Format angeboten. Die Ordnerstruktur im Archiv gibt vor, wo Pakete für weitere Funktionen hinzugefügt werden müssen.

Eine Stapelverarbeitungsdatei für Windows und ein Skript für Linux erstellt aus der Zusammenstellung eine Bootdiskette, mit der der Paketfilter in Betrieb genommen werden kann. Diese Diskette enthält dann alle Funktionen, die man zuvor hinzugefügt hat. Eine Festplatteninstallation ist nur notwendig, wenn alle 17 Zusatzpakete installiert werden sollen.

7.3.2.4 Installationsdauer

Die Installationsgeschwindigkeit von fli4l hängt bei einigen Teilen von der Arbeitsgeschwindigkeit des installierenden Administrators ab, daher ist dieser Wert mit einer gewissen Toleranz zu sehen.

Es wird davon ausgegangen, dass der Installierende weiß, welche Handgriffe er zu erledigen hat und in welchen Dateien sie getätigt werden müssen.

Der Installationsvorgang teilt sich auf in das Entpacken der einzelnen Softwarepakete und dem Aufspielen der Software auf eine formatierte Diskette per Batchdatei. Die Konfiguration findet zwischen dem Entpacken und dem Aufspielen statt.

Der reine Installationsvorgang dauerte 2:42 min.

7.3.2.5 Konfigurationsdauer

Der Paketfilter wird über Textdateien konfiguriert. Jedes Modul hat eine eigene Textdatei, in der Anpassungen für die Netzwerkkonfiguration getätigt werden können.

Durch den Konfigurationsassistenten FliwizNG von Carsten Cerny und Jürgen Bauer wird ein Möglichkeit angeboten, die Einstellungen der Konfigurationsdateien interaktiv zu machen. Da dieser Assistent noch nicht alle möglichen Funktionen des Fli4l unterstützt, fließt er nicht in die Wertung mit ein.

Die Einrichtung dauerte 5:36 min.

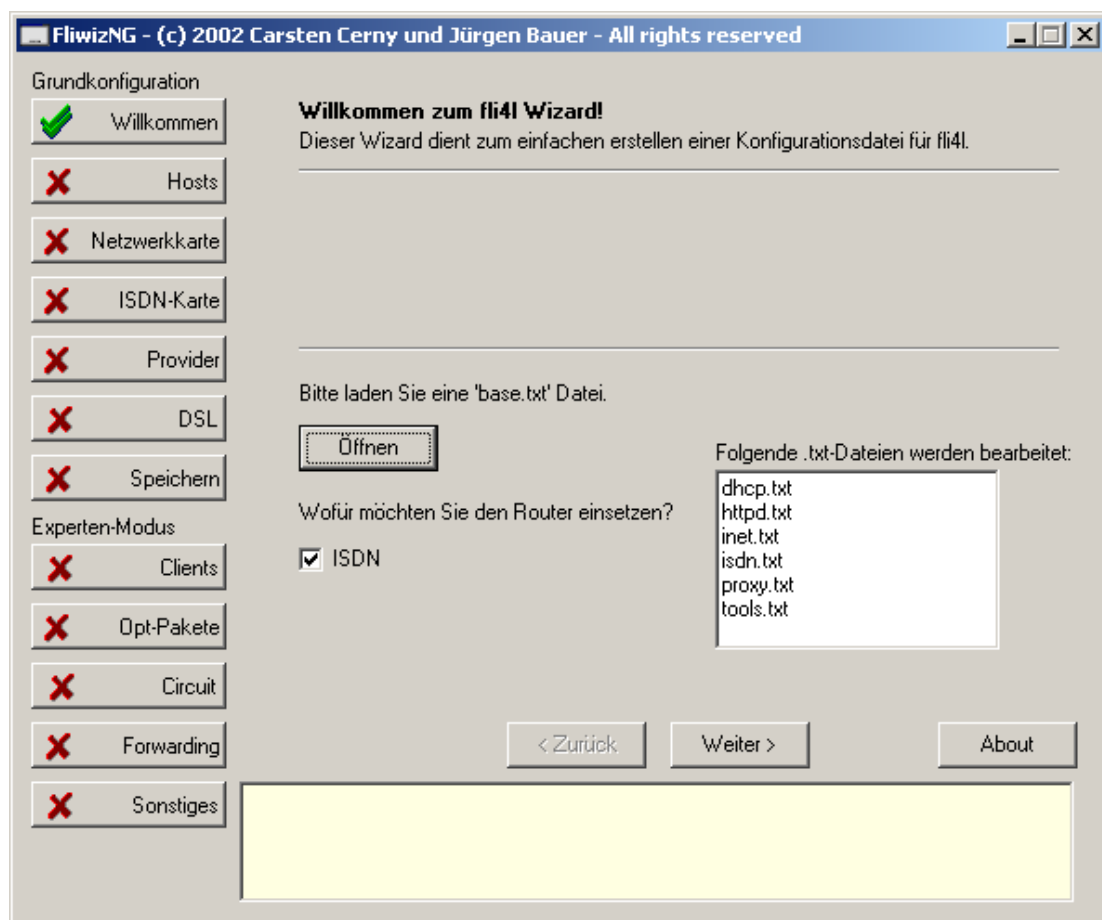


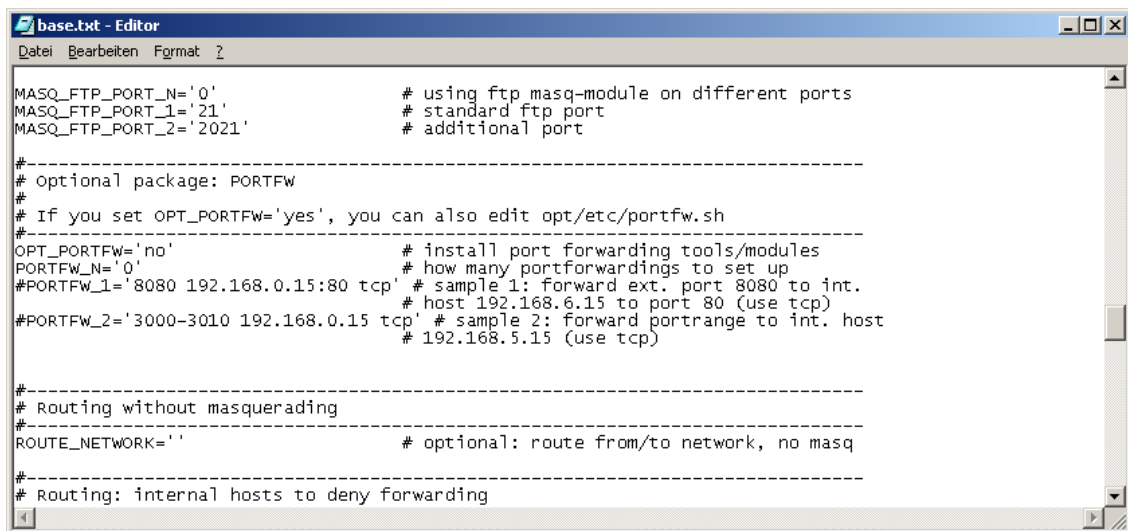
Abbildung 7.3-27 Fli4l FliwizNG

7.3.2.6 Userverwaltung mit Zugriffsregeln

Fli4l bietet keine Möglichkeit einer Userverwaltung, um Benutzerrechte für Dienste zu verwalten.

7.3.2.7 Konfigurierbares Paketfiltersystem

Das Paketfiltersystem von Fli4l kann konfiguriert werden. Es besteht die Möglichkeit, in der textbasierten Datei *base.txt* das Weiterleiten von Paketen auf bestimmte Ports an eine festgelegte IP Adresse zu konfigurieren (Siehe Abbildung 7.3-28) sowie die Ports anzugeben, auf denen Pakete abgelehnt oder angenommen werden (Siehe Abbildung 7.3-29).



```
base.txt - Editor
Datei Bearbeiten Format ?

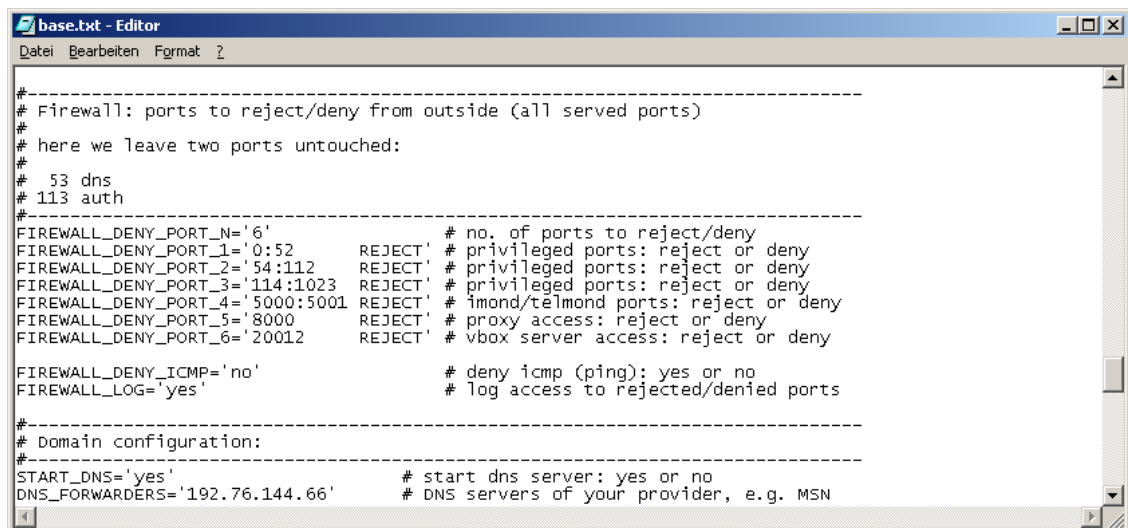
MASQ_FTP_PORT_N='0'           # using ftp masq-module on different ports
MASQ_FTP_PORT_1='21'          # standard ftp port
MASQ_FTP_PORT_2='2021'        # additional port

#-----
# Optional package: PORTFW
#
# If you set OPT_PORTFW='yes', you can also edit opt/etc/portfw.sh
#-----
OPT_PORTFW='no'               # install port forwarding tools/modules
PORTFW_N='0'                  # how many portforwardings to set up
#PORTFW_1='8080 192.168.0.15:80 tcp' # sample 1: forward ext. port 8080 to int.
# host 192.168.0.15 to port 80 (use tcp)
#PORTFW_2='3000-3010 192.168.0.15 tcp' # sample 2: forward portrange to int. host
# 192.168.5.15 (use tcp)

#-----
# Routing without masquerading
#
ROUTE_NETWORK=''             # optional: route from/to network, no masq

#-----
# Routing: internal hosts to deny forwarding
```

Abbildung 7.3-28 Fli4l Paketweiterleitung



```
base.txt - Editor
Datei Bearbeiten Format ?

#-----
# Firewall: ports to reject/deny from outside (all served ports)
#
# here we leave two ports untouched:
#
# 53 dns
# 113 auth
#-----
FIREWALL_DENY_PORT_N='6'      # no. of ports to reject/deny
FIREWALL_DENY_PORT_1='0:52 REJECT' # privileged ports: reject or deny
FIREWALL_DENY_PORT_2='54:112 REJECT' # privileged ports: reject or deny
FIREWALL_DENY_PORT_3='114:1023 REJECT' # privileged ports: reject or deny
FIREWALL_DENY_PORT_4='5000:5001 REJECT' # imond/telmond ports: reject or deny
FIREWALL_DENY_PORT_5='8000 REJECT' # proxy access: reject or deny
FIREWALL_DENY_PORT_6='20012 REJECT' # vbox server access: reject or deny

FIREWALL_DENY_ICMP='no'      # deny icmp (ping): yes or no
FIREWALL_LOG='yes'           # log access to rejected/denied ports

#-----
# Domain configuration:
#
START_DNS='yes'               # start dns server: yes or no
DNS_FORWARDERS='192.76.144.66' # DNS servers of your provider, e.g. MSN
```

Abbildung 7.3-29 Fli4l Paketfilterkonfiguration

7.3.2.8 Fernadministration

Fli4I bietet die Möglichkeit per Telnet oder SSH von außen oder innerhalb des lokalen Netzwerkes über die Oberfläche *Imonc* administriert zu werden.

Zu diesem Zweck muss das Paket *inet* installiert werden.

Imonc bietet die Möglichkeit, die Konfigurationsdateien des Fli4I, die auf einem beliebigen Windowsrechner im LAN liegen können, zu konfigurieren, neu zu kompilieren und anschließend an den laufenden Router zu übertragen (Siehe Abbildung 7.3-30).

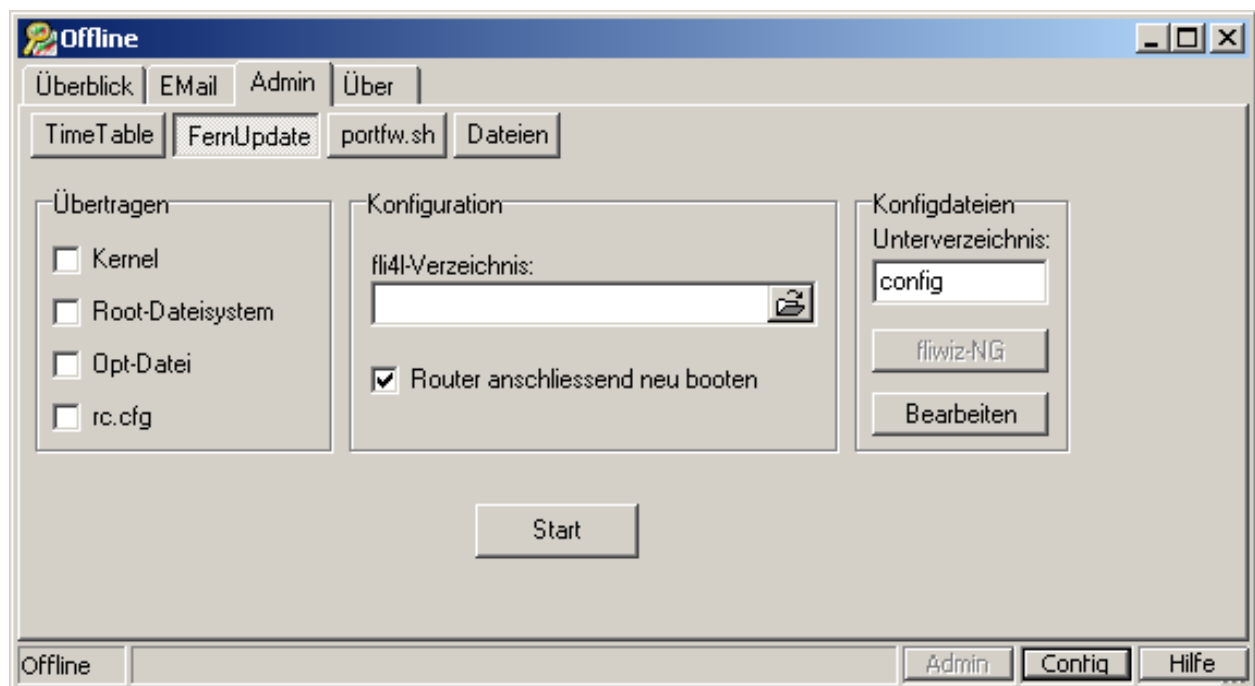


Abbildung 7.3-30 Fli4I Fernadministration

7.3.2.9 Updateassistent

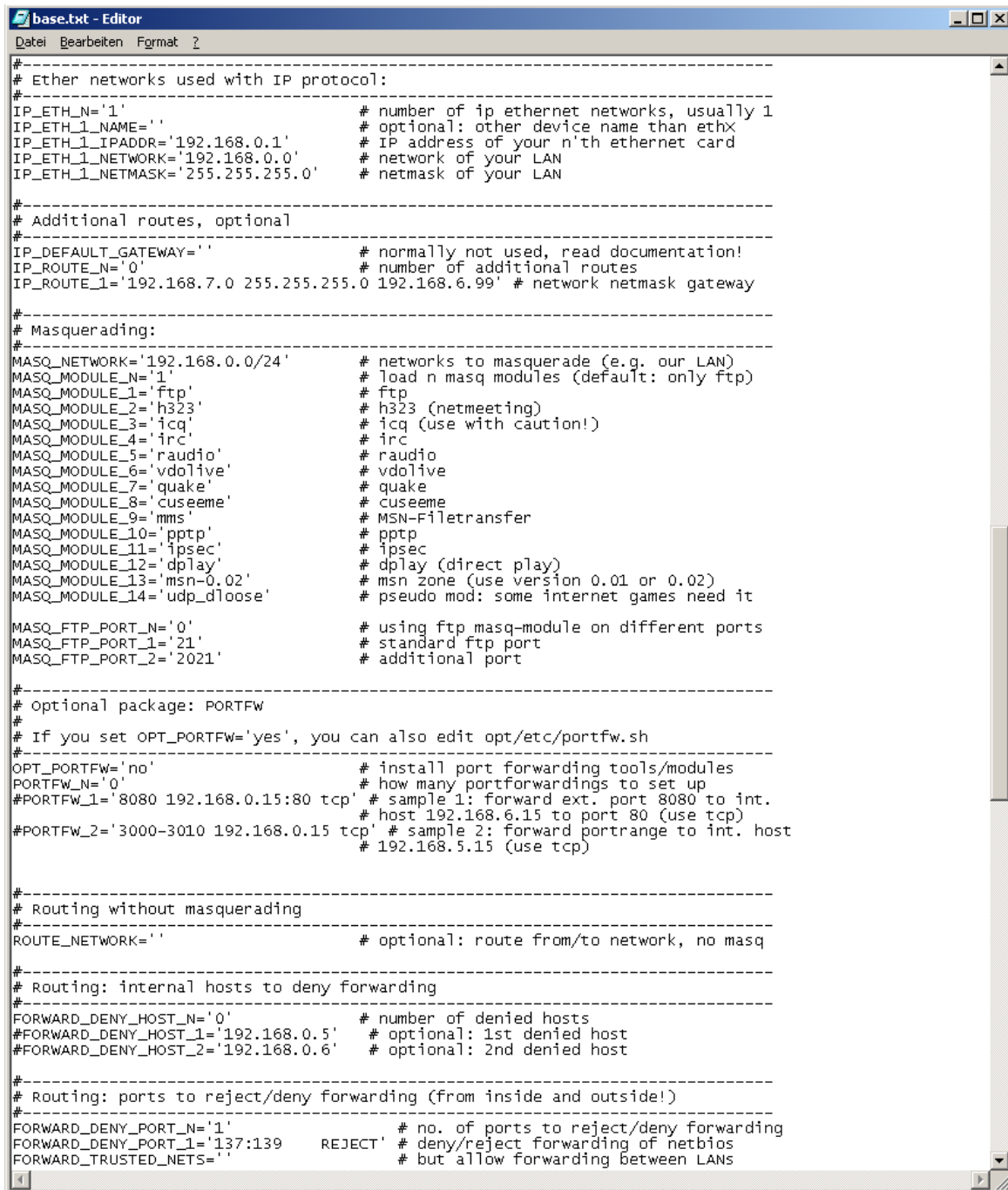
Imonc bietet über die Fernadministration die Möglichkeit den Router aus dem lokalen Netz zu aktualisieren. Wenn Updates für die Pakete im Internet verfügbar sind, können diese auf einem Rechner im LAN entpackt werden. Diese Pakete können von diesem Rechner auf den laufenden Router übertragen werden. Neue Updates werden über die Internetseite des Fli4I Projekts publiziert [7] (Siehe Abbildung 7.3-30)

7.3.2.10 Firewallklasse

Fli4I ist ein einfacher Paketfilter ohne jegliche Verbindungsüberwachung. Er richtet sich ausschließlich danach, welche Ports ihm vorgegeben sind, bei denen er Pakete verwirft oder ablehnt. [7]

7.3.2.11 Routingfunktion

Der Fli4l besitzt eine Routingfunktion und kann mehrere Netzwerkadapter auf einmal verwalten (Siehe Abbildung 7.3-31).



```
#-----#
# Ether networks used with IP protocol:
#-----#
IP_ETH_N='1' # number of ip ethernet networks, usually 1
IP_ETH_1_NAME='' # optional: other device name than ethx
IP_ETH_1_IPADDR='192.168.0.1' # IP address of your n'th ethernet card
IP_ETH_1_NETWORK='192.168.0.0' # network of your LAN
IP_ETH_1_NETMASK='255.255.255.0' # netmask of your LAN

#-----#
# Additional routes, optional
#-----#
IP_DEFAULT_GATEWAY='' # normally not used, read documentation!
IP_ROUTE_N='0' # number of additional routes
IP_ROUTE_1='192.168.7.0 255.255.255.0 192.168.6.99' # network netmask gateway

#-----#
# Masquerading:
#-----#
MASQ_NETWORK='192.168.0.0/24' # networks to masquerade (e.g. our LAN)
MASQ_MODULE_N='1' # load n masq modules (default: only ftp)
MASQ_MODULE_1='ftp' # ftp
MASQ_MODULE_2='h323' # h323 (netmeeting)
MASQ_MODULE_3='icq' # icq (use with caution!)
MASQ_MODULE_4='irc' # irc
MASQ_MODULE_5='raudio' # raudio
MASQ_MODULE_6='vdolive' # vdolive
MASQ_MODULE_7='quake' # quake
MASQ_MODULE_8='cuseeme' # cuseeme
MASQ_MODULE_9='mms' # MSN-Filetransfer
MASQ_MODULE_10='pptp' # pptp
MASQ_MODULE_11='ipsec' # ipsec
MASQ_MODULE_12='dplay' # dplay (direct play)
MASQ_MODULE_13='msn-0.02' # msn zone (use version 0.01 or 0.02)
MASQ_MODULE_14='udp_dloose' # pseudo mod: some internet games need it

MASQ_FTP_PORT_N='0' # using ftp masq-module on different ports
MASQ_FTP_PORT_1='21' # standard ftp port
MASQ_FTP_PORT_2='2021' # additional port

#-----#
# Optional package: PORTFW
#
# If you set OPT_PORTFW='yes', you can also edit opt/etc/portfw.sh
#
OPT_PORTFW='no' # install port forwarding tools/modules
PORTFW_N='0' # how many portforwardings to set up
#PORTFW_1='8080 192.168.0.15:80 tcp' # sample 1: forward ext. port 8080 to int.
# host 192.168.6.15 to port 80 (use tcp)
#PORTFW_2='3000-3010 192.168.0.15 tcp' # sample 2: forward portrange to int. host
# 192.168.5.15 (use tcp)

#-----#
# Routing without masquerading
#
ROUTE_NETWORK='' # optional: route from/to network, no masq

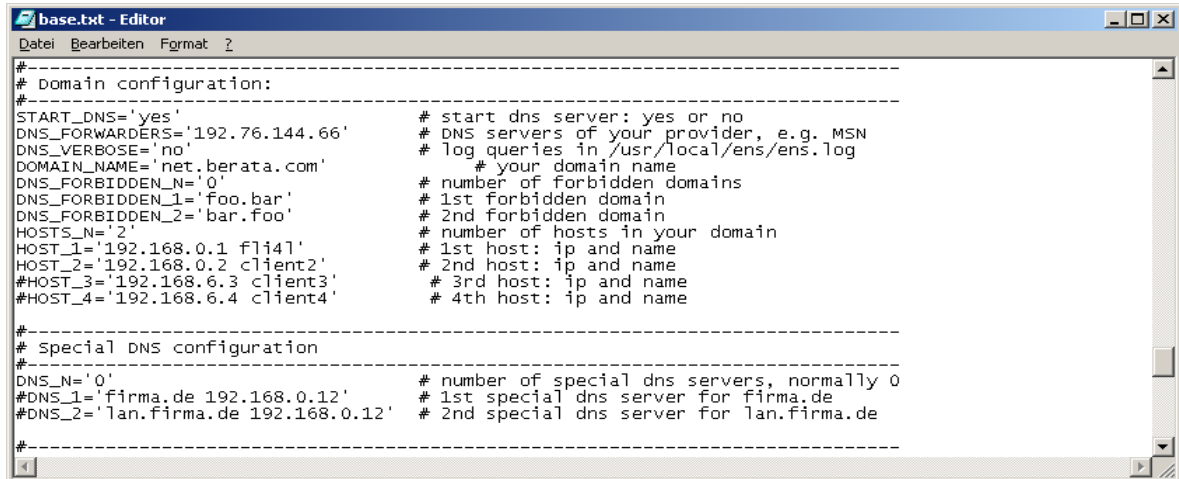
#-----#
# Routing: internal hosts to deny forwarding
#
FORWARD_DENY_HOST_N='0' # number of denied hosts
#FORWARD_DENY_HOST_1='192.168.0.5' # optional: 1st denied host
#FORWARD_DENY_HOST_2='192.168.0.6' # optional: 2nd denied host

#-----#
# Routing: ports to reject/deny forwarding (from inside and outside!)
#
FORWARD_DENY_PORT_N='1' # no. of ports to reject/deny forwarding
FORWARD_DENY_PORT_1='137:139 REJECT' # deny/reject forwarding of netbios
FORWARD_TRUSTED_NETS='' # but allow forwarding between LANs
```

Abbildung 7.3-31 Fli4l RoutingEinstellungen

7.3.2.12 DNS

Bei DNS-Anfragen bietet der Fli4l die Möglichkeit, diese selbst zu beantworten oder bei Nichterfüllung an den nächsten DNS-Server weiterzuleiten (Siehe Abbildung 7.3-32).

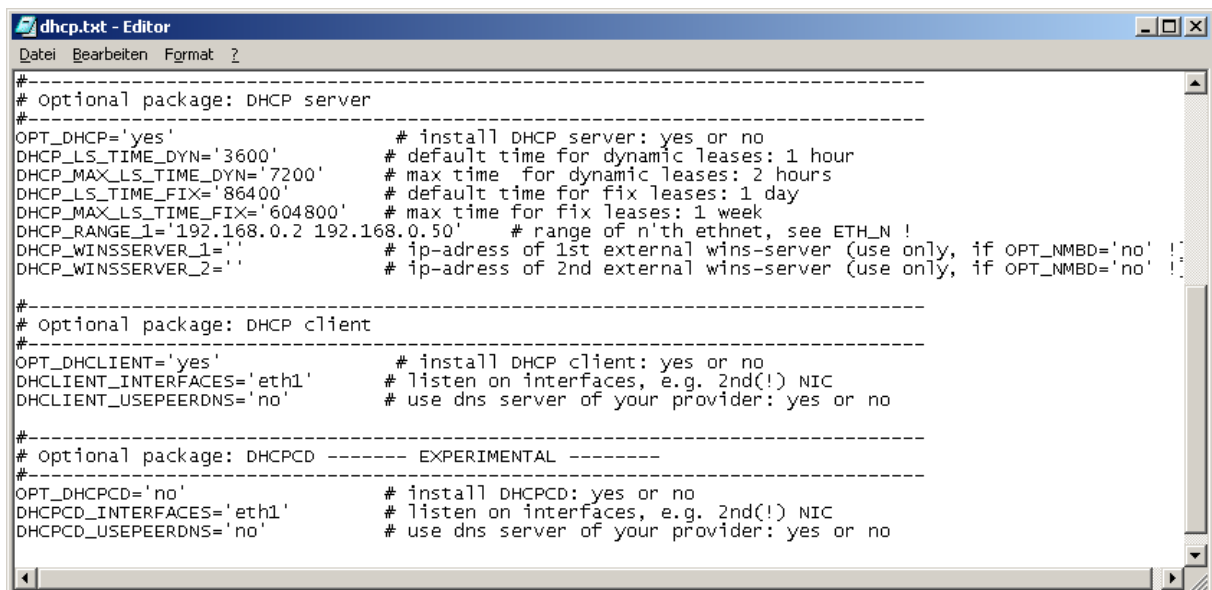


```
#-----  
# Domain configuration:  
#-----  
START_DNS='yes' # start dns server: yes or no  
DNS_FORWARDERS='192.76.144.66' # DNS servers of your provider, e.g. MSN  
DNS_VERBOSE='no' # log queries in /usr/local/ens/ens.log  
DOMAIN_NAME='net.berata.com' # your domain name  
DNS_FORBIDDEN_N='0' # number of forbidden domains  
DNS_FORBIDDEN_1='foo.bar' # 1st forbidden domain  
DNS_FORBIDDEN_2='bar.foo' # 2nd forbidden domain  
HOSTS_N='2' # number of hosts in your domain  
HOST_1='192.168.0.1 fli4l' # 1st host: ip and name  
HOST_2='192.168.0.2 client2' # 2nd host: ip and name  
#HOST_3='192.168.6.3 client3' # 3rd host: ip and name  
#HOST_4='192.168.6.4 client4' # 4th host: ip and name  
#-----  
# Special DNS configuration  
#-----  
DNS_N='0' # number of special dns servers, normally 0  
#DNS_1='firma.de 192.168.0.12' # 1st special dns server for firma.de  
#DNS_2='lan.firma.de 192.168.0.12' # 2nd special dns server for lan.firma.de  
#-----
```

Abbildung 7.3-32 Fli4l DNS Konfiguration

7.3.2.13 DHCP

Der Fli4l kann ein internes Netz mit IP-Adressen dynamisch versorgen. Das Paket *DHCP* muss dafür installiert werden (Siehe Abbildung 7.3-33).

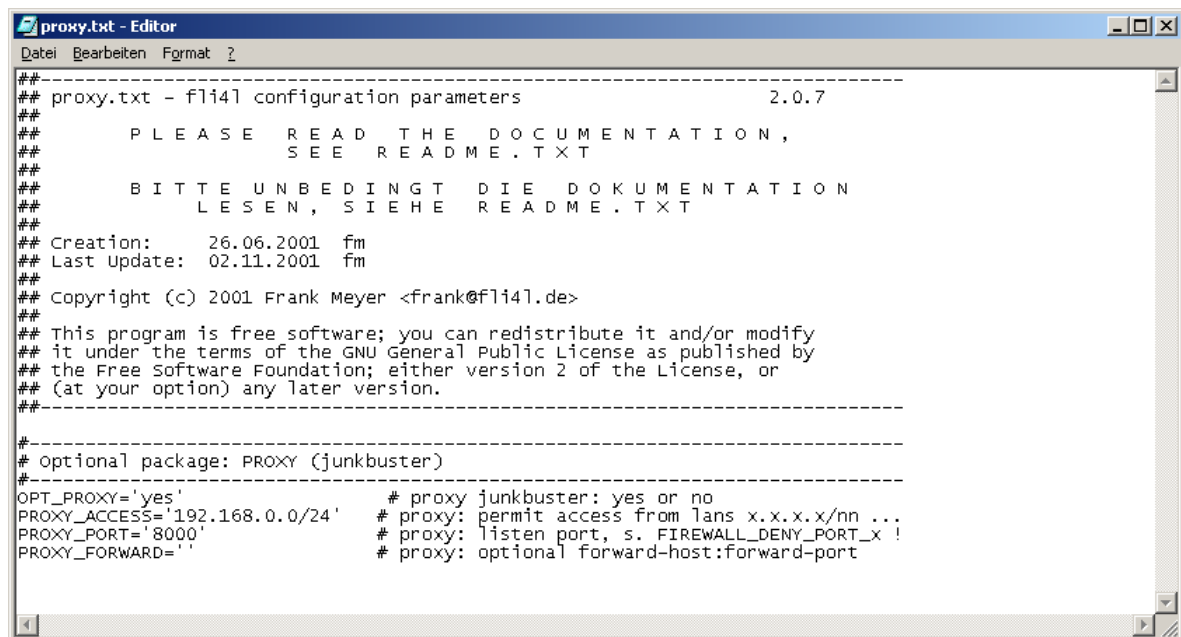


```
#-----  
# optional package: DHCP server  
#-----  
OPT_DHCP='yes' # install DHCP server: yes or no  
DHCP_LS_TIME_DYN='3600' # default time for dynamic leases: 1 hour  
DHCP_MAX_LS_TIME_DYN='7200' # max time for dynamic leases: 2 hours  
DHCP_LS_TIME_FIX='86400' # default time for fix leases: 1 day  
DHCP_MAX_LS_TIME_FIX='604800' # max time for fix leases: 1 week  
DHCP_RANGE_1='192.168.0.2 192.168.0.50' # range of n'th ethnet, see ETH_N !  
DHCP_WINSERVER_1='' # ip-adress of 1st external wins-server (use only, if OPT_NMBD='no' !)  
DHCP_WINSERVER_2='' # ip-adress of 2nd external wins-server (use only, if OPT_NMBD='no' !)  
#-----  
# optional package: DHCP client  
#-----  
OPT_DHCLIENT='yes' # install DHCP client: yes or no  
DHCLIENT_INTERFACES='eth1' # listen on interfaces, e.g. 2nd(!) NIC  
DHCLIENT_USEPEERDNS='no' # use dns server of your provider: yes or no  
#-----  
# optional package: DHCPD ----- EXPERIMENTAL -----  
#-----  
OPT_DHCPD='no' # install DHCPD: yes or no  
DHCPD_INTERFACES='eth1' # listen on interfaces, e.g. 2nd(!) NIC  
DHCPD_USEPEERDNS='no' # use dns server of your provider: yes or no  
#-----
```

Abbildung 7.3-33 Fli4l DHCP Konfiguration

7.3.2.14 Internetproxy

Ein Internetproxy mit dem Namen *Junkbuster* wird über das Paket *proxy* bereitgestellt. Über die Textdatei *proxy.txt* kann man den Port, auf dem der Proxy horcht, verändern (Siehe Abbildung 7.3-34).



```
##-----  
## proxy.txt - fli4l configuration parameters                2.0.7  
##  
##      PLEASE READ THE DOCUMENTATION ,  
##      SEE      README.TXT  
##  
##      BITTE UNBEDINGT DIE DOKUMENTATION  
##      LESEN , SIEHE  README.TXT  
##  
## Creation:      26.06.2001  fm  
## Last Update:   02.11.2001  fm  
##  
## Copyright (c) 2001 Frank Meyer <frank@fli4l.de>  
##  
## This program is free software; you can redistribute it and/or modify  
## it under the terms of the GNU General Public License as published by  
## the Free Software Foundation; either version 2 of the License, or  
## (at your option) any later version.  
##-----  
#-----  
# optional package: PROXY (junkbuster)  
#-----  
OPT_PROXY='yes'          # proxy junkbuster: yes or no  
PROXY_ACCESS='192.168.0.0/24' # proxy: permit access from lans x.x.x.x/nn ...  
PROXY_PORT='8000'        # proxy: listen port, s. FIREWALL_DENY_PORT_x !  
PROXY_FORWARD=''         # proxy: optional forward-host:forward-port
```

Abbildung 7.3-34 Fli4l Internetproxy Konfiguration

7.3.2.15 VPN

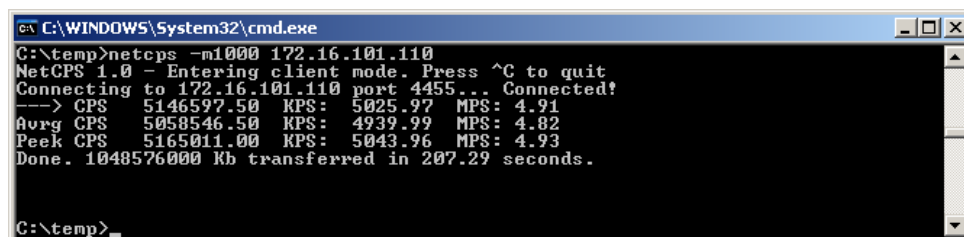
Ein VPN Server mit Userverwaltung stellt der Fli4l nicht zur Verfügung.

7.3.2.16 Verarbeitungsverzögerung

Fli4l erreichte bei sechs Messungen folgende Ergebnisse in MB/s:

4,82/4,83/4,84/4,87/4,88/4,90

daraus ergibt sich eine durchschnittliche Übertragungsrate von 4,86 MB/s.



```
C:\WINDOWS\System32\cmd.exe  
C:\temp>netcps -m1000 172.16.101.110  
NetCPS 1.0 - Entering client mode. Press ^C to quit  
Connecting to 172.16.101.110 port 4455... Connected!  
--> CPS 5146577.50 KPS: 5025.97 MPS: 4.91  
Avg CPS 5058546.50 KPS: 4939.99 MPS: 4.82  
Peek CPS 5165011.00 KPS: 5043.96 MPS: 4.93  
Done. 1048576000 Kb transferred in 207.29 seconds.  
C:\temp>
```

Abbildung 7.3-35 Fli4l Ergebnis Übertragungsrate

Gegenüber der Referenz erreichte der Fli4l eine Übertragungsleistung von:

$$\frac{(4,86\text{MB/s}) * 100}{10,81\text{MB/s}} = \underline{\underline{44.95\%}}$$

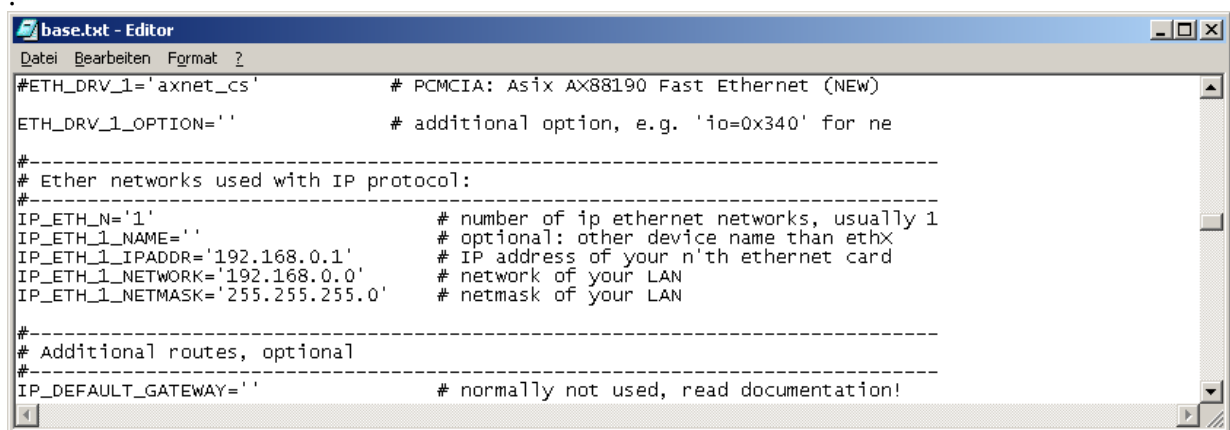
Das ergibt einen Verlust von 55,05% durch die Verarbeitungsverzögerung.

7.3.2.17 Verbindungsprotokollierung

Eine Verbindungsprotokollierung stellt Fli4l nicht zur Verfügung.

7.3.2.18 DMZ

Fli4l bietet keine direkte Möglichkeit, eine DMZ zu verwalten, kann aber mehrere Netzwerkadapter und somit mehrere Netzwerke verwalten (Siehe Abbildung 7.3-36). Dem Netzwerkadministrator steht entsprechend die Möglichkeit zur Verfügung, verschiedene Teilnetze mit unterschiedlichen Sicherheitsgraden zu erstellen.



```
base.txt - Editor
Datei Bearbeiten Format ?
#ETH_DRV_1='axnet_cs'          # PCMCIA: Asix AX88190 Fast Ethernet (NEW)
ETH_DRV_1_OPTION=''          # additional option, e.g. 'io=0x340' for ne
#-----
# Ether networks used with IP protocol:
#-----
IP_ETH_N='1'                  # number of ip ethernet networks, usually 1
IP_ETH_1_NAME=''              # optional: other device name than ethx
IP_ETH_1_IPADDR='192.168.0.1' # IP address of your n'th ethernet card
IP_ETH_1_NETWORK='192.168.0.0' # network of your LAN
IP_ETH_1_NETMASK='255.255.255.0' # netmask of your LAN
#-----
# Additional routes, optional
#-----
IP_DEFAULT_GATEWAY=''        # normally not used, read documentation!
```

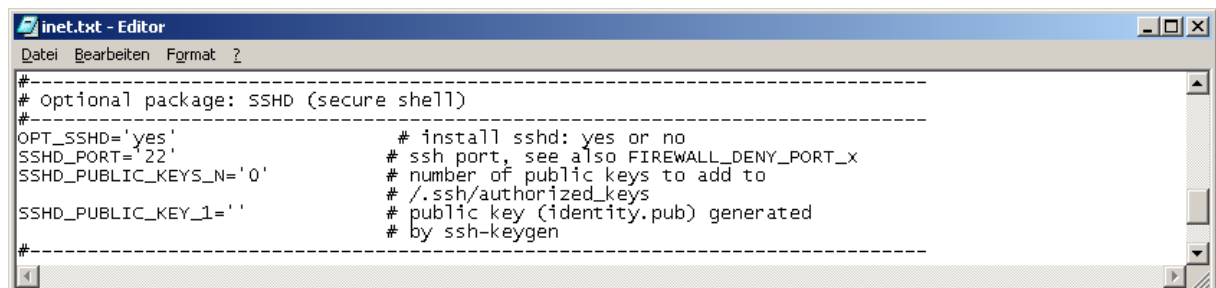
Abbildung 7.3-36 Fli4l DMZ Möglichkeit

7.3.2.19 IDS

Fli4l hat kein *intrusion detection system*.

7.3.2.20 Portscanversuch

Der Null-Scan und der Xmas-Tree-Scan förderten bei dem Fli4l einen offenen Port zu Tage, den Port 22. Dieser ist für den Service SSH vor dem Scan manuell geöffnet worden, um diesen Dienst bereit zu stellen (Siehe Abbildung 7.3-37). Sonst wurden keine offenen Ports entdeckt (Siehe Abbildung 7.3-38).



```
inet.txt - Editor
Datei Bearbeiten Format ?
#-----
# optional package: SSHD (secure shell)
#-----
OPT_SSHD='yes'                # install sshd: yes or no
SSHD_PORT='22'                # ssh port, see also FIREWALL_DENY_PORT_x
SSHD_PUBLIC_KEYS_N='0'        # number of public keys to add to
                               # /ssh/authorized_keys
SSHD_PUBLIC_KEY_1=''          # public key (identity.pub) generated
                               # by ssh-keygen
#-----
```

Abbildung 7.3-37 Fli4l SSH Aktivierung

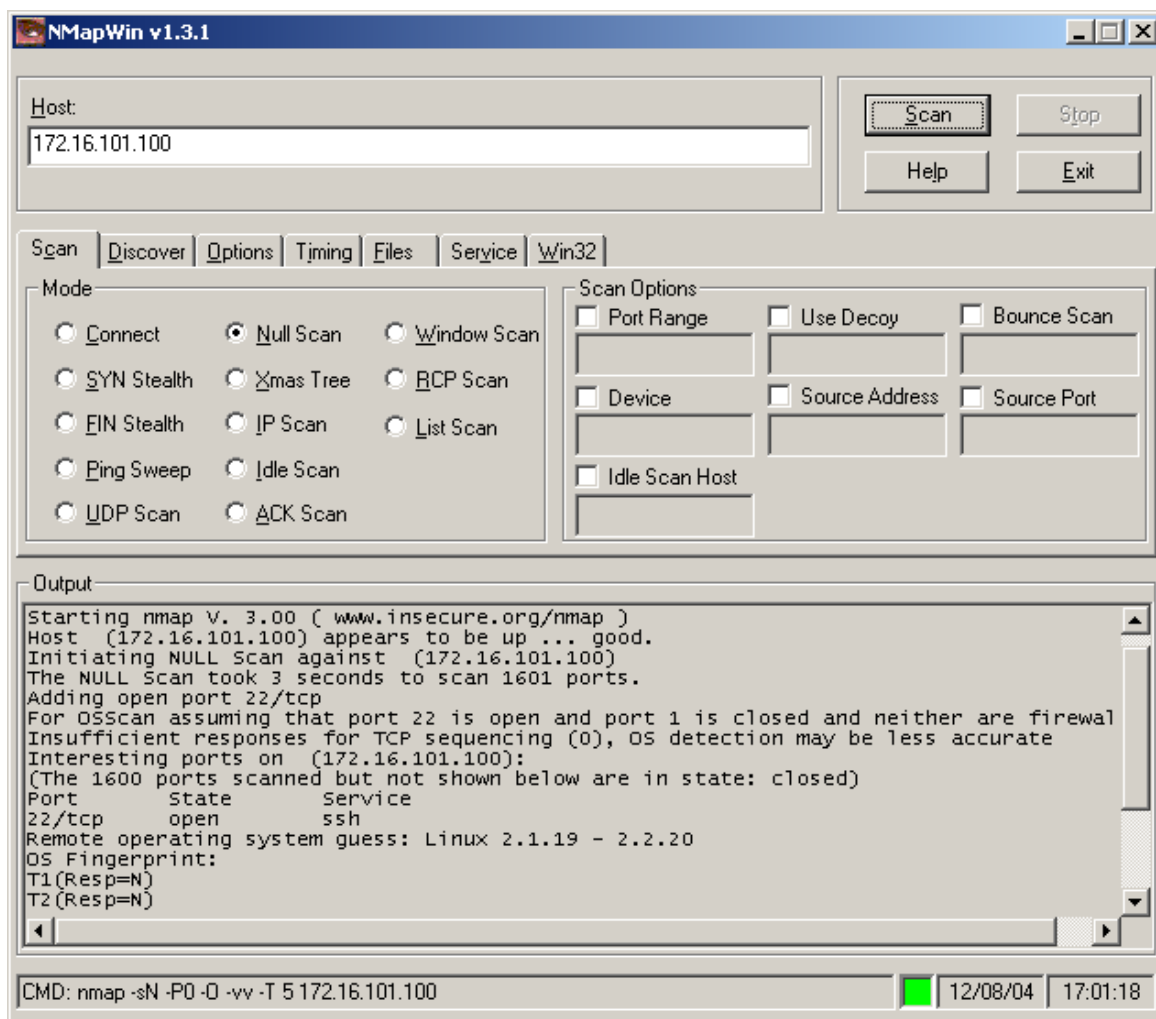


Abbildung 7.3-38 Fli4l Null-Scan

7.3.2.21 Betriebssystemidentifikation

Der Null-Scan von NMAP mit der Option *os detection* hat das Betriebssystem von Fli4l samt Kernelversion richtig erkannt (Siehe Abbildung 7.3-38).

7.3.2.22 Scan-Angriffserkennung

Eine Angriffserkennung besitzt der Fli4l nicht. Die Verbindungsversuche werden auf dem Bildschirm des Rechners zeilenweise im Textmodus angezeigt. Eine Auswertung, die zu dem Ergebnis eines Angriffs führen könnte, findet nicht statt.

7.3.2.23 Feedback bei Angriffsversuch

Da fli4l nicht die Möglichkeit hat Angriffe zu identifizieren, gibt es kein Feedback über Angriffsversuche.

8 Auswertung/Zusammenfassung

In diesem Kapitel werden die Testergebnisse zu einer aussagekräftigen Form aufbereitet und präsentiert. Zur besseren Übersicht werden die Ergebnisse in Prozent(%) angegeben.

Das hier verwendete Bewertungssystem wurde gewählt, um die Güte anhand des gewählten Testszenarios messbar zu machen. Interessant ist, wie sich die Abstände der erreichten Prozentpunkte in Bezug auf die tatsächliche Leistung auswirken.

Außerdem interessiert die Frage, wie einsetzbar der jeweilige Proband im Testszenario wirklich ist.

8.1 Kosten

Kosten	Gewichtung	IPCop 1.3		Fli4l 2.0.8	
		Ergebnis	Bewertung	Ergebnis	Bewertung
BS = Linux	50%	Ja	50%	Ja	50%
Installationsdauer (in Sekunden)	20%	175	18%	162	20%
Konfigurationsdauer (in Sekunden)	30%	307	30%	336	27%
Gesamt:	100%	98%		97%	

Beide Firewallsysteme basieren auf dem Betriebssystem Linux, wobei der Fli4l eine ältere Version des Linuxkerns benutzt. Die Installations- und Einrichtungszeiten sind von beiden Systemen nahezu identisch. Das Gesamtergebnis des Bereichs Kosten zeigt daher nur einem minimalen Unterschied. Beide Systeme sind sehr kostensparend.

8.2 Sicherheit

Sicherheit	Gewichtung	IPCop 1.3		Fli4l 2.0.8	
		Ergebnis	Bewertung	Ergebnis	Bewertung
FW-Klasse	10%	SPF*	5%	PF*	0%
Alle Ports geschlossen*	15%	Ja	15%	Ja	15%
BS kann nicht identifiziert werden	15%	Ja	15%	Nein	0%
Erkennen eines Scanversuchs	5%	Ja	5%	Nein	0%
IDS	10%	Ja	10%	Nein	0%
DMZ	15%	Ja	15%	Ja	15%
Feedback bei Angriff	10%	Nein	0%	Nein	0%
Verbindungsprotokollierung	10%	Ja	10%	Nein	0%
Konfigurierbares Paketfilterregelsystem	10%	Ja	10%	Ja	10%
Gesamt:	100%		85%		40%

*SPF steht für *stateful paket filter*; PF für *paket filter*

Im Bereich Sicherheit fallen die Unterschiede der Ergebnisse am größten aus. Es liegen 45% zwischen den beiden Systemen.

Während beide Systeme alle Ports sicher verschlossen halten, ein konfigurierbares Paketfiltersystem besitzen und die Möglichkeit einer DMZ zur Verfügung stellen, zeigt sich deutlich, dass der Sicherheitsstandard des Fli4l weit unter dem Niveau des Standards von dem IPCop ist.

5 Merkmale kann der Fli4l nicht erfüllen, während der IPCop gerade mal einen einzigen nicht erfüllen kann. Außerdem ist der Fli4l im Gegensatz zum zustandsbasierten Paketfilter des IPCops nur ein einfacher Paketfilter. Am auffälligsten ist, dass das Betriebssystem des Fli4l über den Null-Scan mit NMAP erkannt werden konnte. Das birgt die Gefahr, dass ein potenzieller Angreifer bekannte Schwachstellen im Betriebssystem ausnutzen könnte.

Da beide Probanden kein visuelles oder optisches Feedback auf einen Angriffsversuch geben, ist der Administrator praktisch blind, sollte ein Angriff geschehen. Bei dem IPCop ist zumindest das IDS *Snort* vorhanden, welches potenzielle Angriffe identifiziert und auflistet.

Bis auf das fehlende Feedback bei einem erfolgreich identifizierten Angriff liefert der IPCop gute Leistungen, die die Sicherheitsanforderungen des Testszenarios gut abdecken. Wünschenswert wäre es, wenn zusätzlich ein visuelles Signal auf dem Bildschirm erfolgen würde, während Snort einen möglichen Angriff identifiziert.

8.3 Netzwerkleistung

Netzwerkleistung	Gewichtung	IPCop 1.3		Fli4l 2.0.8	
		Ergebnis	Bewertung	Ergebnis	Bewertung
Verarbeitungsverzögerung (in MB/s)	30%	4,32	12%	4,86	14%
VPN	10%	Ja	10%	Nein	0%
Routingfunktion	15%	Ja	15%	Ja	15%
DNS	5%	Ja	5%	Ja	5%
DHCP	5%	Ja	5%	Ja	5%
Userverwaltung mit Zugriffsregeln	20%	Nein	0%	Nein	0%
Internetproxy	15%	Ja	15%	Ja	15%
Gesamt:	100%	62%		54%	

Eigentlich erfüllen die beiden Firewalls den Großteil der Merkmale für die Netzwerkleistung. Einen Internetproxy, DNS-Dienst, DHCP-Dienst und eine Routingfunktion besitzen beide. Den Unterschied von 10% am Gesamtergebnis zwischen den beiden macht der fehlende VPN-Dienst beim Fli4l aus. Trotzdem ist das Gesamtergebnis von beiden nicht so gut. Grund dafür ist die schlechte Übertragungsleistung und die fehlende Userverwaltung, mit welcher intern der Zugriff auf bestimmte Dienste kontrolliert werden kann. Beide Probanden haben Bandbreitenverluste von mehr als 50%, wobei der Fli4l leicht besser abschneidet. Da die Übertragungsleistung und die Userverwaltung als sehr wichtig eingestuft und die Gewichtungen entsprechend hoch gesetzt wurden (30% und 20%), ist das Ergebnis entsprechend schlecht ausgefallen.

8.4 Komfort

Komfort	Gewichtung	IPCop 1.3		Fli4l 2.0.8	
		Ergebnis	Bewertung	Ergebnis	Bewertung
Installationsroutine	25%	Ja	25%	Nein	0%
Updateassistent	20%	Ja	20%	Ja	20%
GUI	25%	Ja	25%	Ja	20%
Fernadministration	30%	Ja	30%	Ja	30%
Gesamt:	100%	100%		75%	

Der IPCop ist sehr komfortabel. Die Einrichtung und Konfiguration über dessen Installationsroutine ist sehr benutzerfreundlich. Nachträgliche Veränderungen an der Konfiguration und die gesamte Administration des IPCops kann über das Webinterface von jedem Host im internen Netz getätigt werden.

Der Fli4l schneidet zwar schlechter ab, da er keine Installationsroutine hat, das Hinzufügen der Installationspakete und die Erstellung der Bootdiskette ist aber leicht zu bewerkstelligen und führt darüber hinaus zu einer schlanken Installation, die auf eine Diskette passt. Der sich in der Entwicklung befindende Konfigurationsassistent wird nach Fertigstellung erheblich zum Komfort beitragen.

Abschließend ist zu sagen, dass der IPCop das Gefühl vermittelt ein zusammenhängendes ausgereiftes Produkt zu sein. Der Fli4l hat zwar keine Installationsroutine, verfügt aber über die Qualität eines modular aufgebauten Systems, welches einfach und platzsparend konfiguriert werden kann.

8.5 Gesamtergebnis

Oberziele	Gewichtung	IPCop 1.3		Fli4l 2.0.8	
		Ergebnis	Bewertung	Ergebnis	Bewertung
Kosten	30%	98%	29%	97%	29%
Sicherheit	40%	85%	34%	40%	16%
Netzwerkleistung	20%	62%	12%	54%	11%
Komfort	10%	100%	10%	75%	8%
Gesamtergebnis	100%	85%		64%	

Mit 85% ist der IPCop von beiden Probanden am besten für das gewählte Szenario geeignet. Der Fli4l schneidet mit 64% wesentlich schlechter ab. Interessant ist der Grund für den großen Unterschied. Den großen Vorsprung erhält der IPCop gegenüber dem Fli4l durch die guten Leistungen im Bereich der Sicherheit, welcher 40% zum Gesamtergebnis beiträgt. Sie bringen dem IPCop einen Vorsprung von 18% im Gesamtergebnis ein. Die übrigen Merkmale tragen mit einer Differenz von 3% nur wenig zum Unterschied des Gesamtergebnisses bei. Im Ganzen wirkt der IPCop ausgereifter als der Fli4l. Die Verbindung von einer guten Ausstattung im Bereich der Netzwerkleistung mit dem Komfort einer guten Administrierbarkeit und einem guten Sicherheitsstandard erzeugen ein sehr konkurrenzfähiges Produkt für Vollpreisprodukte zu unschlagbaren Kosten.

Ein weiterer interessanter Punkt ist, ob der Fli4l wirklich so wenig leistet, wie das Gesamtergebnis es erscheinen lässt. Für diese Fragestellung müssen die Merkmale und deren Gewichtungen im Bereich der Sicherheit betrachtet werden, denn wie oben erwähnt, ist dieses Hauptmerkmal für einen Unterschied von 18% im Gesamtergebnis zwischen IPCop und Fli4l verantwortlich. Was ist wenn einige Merkmale als weniger wichtig eingestuft werden? Wie schneidet der Fli4l dann ab? Ein IDS-System, die Scan-Erkennung, das Feedback auf einen erkannten Scan und die Verbindungsprotokollierung können durchaus als weniger wichtig eingestuft oder ganz weggelassen werden, so dass sie nur wenig zum Erfolg des Zwischenergebnisses und somit zum Gesamtergebnis beitragen. Das Ergebnis würde für den Fli4l wesentlich besser ausfallen. Der einzige wichtige Unterschied zum IPCop wäre dann der, dass bei dem Fli4l das Betriebssystem erkannt wurde. Diese Nichterfüllung des Merkmals könnte unter Umständen unwichtig sein, wenn für den schlanken Kernel des Fli4l-Betriebssystems keine Schwächen bekannt sind, so dass einem möglichen Angreifer diese Information nichts nützt. Unter diesen Gesichtspunkten wäre der Fli4l ein ernst zu nehmender Konkurrent für den IPCop. Ein weiterer Vorteil des Fli4l ist der, dass man seine Firewall praktisch in der „Hosentasche“ tragen kann, denn bei einer Installation mit den gebräuchlichsten Diensten, wie sie in dieser Arbeit konfiguriert wurde, passt diese auf eine bootfähige Diskette.

9 Ausblick

Diese Arbeit hat keinen direkten Vergleich zwischen Vollpreisprodukten und Firewallsystemen aus Open Source Projekten angestrebt. Sie legt lediglich dar, ob die hier genannten Firewalls als Alternative in Betracht gezogen werden können.

Es wäre sicher interessant, die Leistungsunterschiede zwischen Vollpreisprodukten und Open Source Firewalls zu testen. Die Ausarbeitung der Hauptmerkmale Kosten, Sicherheit, Netzwerkleistung und Komfort könnten zusätzlich in einigen Punkten noch aufschlussreicher gestaltet werden, um eine genauere Betrachtung zuzulassen. Der Punkt Komfort kann z.B. um den Bereich der Ergonomie erweitert werden, so dass eine intensivere Begutachtung der Bedienbarkeit einer Firewall in einem Ergonomielabor geleistet werden könnte. Auch eine differenziertere Untersuchung der Sicherheit wäre lohnenswert, denn in diesem Test kamen nur die Scan-Angriffe zum Einsatz, um Aufschluss darüber zu geben, ob die getestete Firewall als sicher gilt oder nicht.

Die Wahl und Bewertungen hängen von dem gewählten Testszenario ab. Durch die Wahl anderer Szenarien würden sich auch die Anforderungen und deren Gewichtungen entsprechend verändern. Ein Szenario, welches hochsensible Daten vorsieht, die einen maximalen Schutz gegen Angriffe von außen erfordern, hätte diese Studienarbeit in einen Test mit dem Schwerpunkt auf die Sicherheitsmerkmale verändert. Angriffe mittels Datenmaterial und die Übernahme von legitimen Verbindungen müssten dafür näher untersucht werden.

Für das hier verwendete Testszenario reichen die gewählten Anforderungen aus, so dass das eigentliche Ziel, zu ergründen, ob Firewalls aus Open Source Projekten, für eine Erweiterung im Netzwerk in Betracht gezogen werden können, erfüllt wurde. Gerade für Anwender im Heimbereich oder kleinere Unternehmen, die sich nur auf den Schutz ihres Betriebssystems und ihres Virenschanners verlassen, sind diese Firewalls attraktiv. Wäre jeder Internetrechner mit solch einem kostenlosen Schutzmechanismus ausgestattet, würde das Internet ein Stück sicherer vor unerlaubten Operationen Dritter sein, denn oft bauen diese darauf auf, dass sie die schwächste Stelle in einem Netzwerk ausnutzen.

10 Literatur

[1] Georg Kurtz, Stuart McClure, Joel Scambray: Das Anti-Hacker-Buch; MITP-Verlag 2002, 3. Auflage

[2] Christoph Busch, Stephen D. Wolthusen: Netzwerksicherheit; Spektrum Akademische Verlag 2002

[3] C. Eckert: IT-Sicherheit, Konzepte-Verfahren-Protokolle; Oldenbourg Verlag 2003

[4] Prof. Dr. Ing. Martin Hübner: Folien zur Vorlesung IT-Sicherheit. Vorlesung Sommersemester 2003

[5] James F. Kurose, Keith W. Ross: Computernetze, ein Top-Down-Ansatz mit Schwerpunkt Internet; Addison-Wesley/Pearson Studium 2002

[6] Anonymous: Der neue Hacker's Guide, Sicherheit im Internet und im lokalen Netz; Markt + Technik Verlag 2002

[7] Internetseite zum verwendeten Software-Paketfilter: www.fli4l.de

[8] Internetseite zur verwendeten Software-Firewall: www.ipcop.org

11 Anhang

Hier befindet sich die in dieser Arbeit verwendete Software.

11.1 Zielsystementwurf

Auf der ersten CD befindet sich eine Exceltabelle, die die Zielsystemmerkmale und Ergebnisse festhält.

11.2 NMAP

Auf der ersten CD befindet sich der in dieser Arbeit verwendete Scanner NMAPWin v1.3.1

11.3 SuperScan

Auf der ersten CD befindet sich der in dieser Arbeit verwendete Scanner SuperScan v4.0

11.4 Netcps

Auf der ersten CD befindet sich das in dieser Arbeit verwendete Analysewerkzeug für die Übertragungsrate Netcps v1.0

11.5 Fli4l

Die in dieser Arbeit getestete Firewallsoftware Fli4l 2.0.8 befindet sich auf der ersten CD.

11.6 IPCop

Die in dieser Arbeit getestete Firewallsoftware IPCop v 1.3 befindet sich auf der zweiten, bootfähigen CD.

11.7 CDs

CD1:

- Zielsystementwurf, Exceltabelle
- NMAPWin v1.3.1
- SuperScan v4.0
- Netcps v1.0
- Fli4l – Installationspakete

CD2:

- IPCop Installations-CD

Erklärung

Hiermit erkläre ich, Nicolas Flöter, geboren am 17.11.1973 in Bad Oldesloe, dass ich meine Studienarbeit mit dem Titel

„Leistungsfähigkeit von kostenlosen Firewalls“

im Sinne der Prüfungsordnung 98 nach § 23 selbständig angefertigt habe. Ich habe keine anderen als die in der Studienarbeit angegebenen Hilfsmittel benutzt.

Hamburg, Oktober 2004

