Ausarbeitung zum Versuch IIS 7

Netzwerksicherheit am Beispiel von Firewalls

Gruppe 6:

Kai Schmitz-Hofbauer

Thomas Espeter

Protokollführer: Kai Schmitz-Hofbauer

1 Einleitung

1.1 Ziel des Versuches

Dieser Versuch sollte einen Einstieg in das umfangreiche Gebiet der Netzwerksicherheit am Beispiel von Firewalls geben. In diesem Praktikum wurden zwei Komponenten eines Firewall-Systems thematisiert, nämlich Paketfilter und Proxies.

1.2 Sicherheit in Netzwerken

Jeder, der seinen Rechner mit dem Internet verbindet, muss sich darüber im Klaren sein, dass er damit seinen Rechner potentiell mit einigen Millionen anderer Rechner in Verbindung bringt. So wie man selbst alle möglichen fremden Rechner erreichen kann, ist man auch für jedermann erreichbar. Daraus resultieren zahlreiche Gefahren. Für einige Privatanwender mag der diesbezügliche Sicherheitsaspekt nur eine untergeordnete Rolle spielen. Aber Unternehmen, die ihr Betriebsnetzwerk an das Internet anbinden, müssen sich vor Angriffen aus dem Internet schützen. Ein gutes Sicherheitskonzept ist unverzichtbar. Angriffsarten lassen sich überwiegend in Kategorien *Spionage*, *Manipulation*, *Verhinderung von Diensten (Denial of Service)* und *Unberechtigte IT-Nutzung* einteilen.

1.3 Firewalls

Eine Firewall (*Brandschutzmauer*) ist ein System, das die Kommunikation zwischen zwei Netzwerken, meist einem betriebsinternen LAN und dem Internet, kontrolliert. Ein Firewall-System besteht in der Regel aus mehreren zusammenarbeitenden Komponenten. Die einzelnen Komponenten arbeiten

auf unterschiedlichen Ebenen des ISO/OSI - Referenzmodells.

1.3.1 Paketfilter

Paketfilter sind ein zentraler Bestandteil eines Firewall- Systems. Sie arbeiten in der Regel auf Ebene 3 (Vemittlungsschicht) und Ebene 4 (Transportschicht) des ISO/OSI - Referenzmodelles. Paketfilter überprüfen Pakete an Hand zuvor definierter Regeln und entscheiden, ob die Pakete verworfen werden oder ob sie passieren dürfen. Filterkriterien können z.B. Quell- und Zieladresse, Quell- und Zielport sowie sämtliche Flags eines Headers (ACK-Bit,SYN-Bite etc.) sein.

1.3.2 Proxies

Proxies (*Stellvertreter*), auch Applikationsfilter genannt, arbeiten auf der Anwendungsebene des ISO/OSI - Referenzmodelles. Proxies nehmen stellvertretend für ein Netzwerk Verbindungen aus einem anderen Netzwerk entgegen, bauen eine Verbindung zu dem zu vertretenden Netzwerk auf und vermitteln Daten nach evtl. Überprüfung zwischen den Verbindungen. Proxies können gemäß vorgegebener Regeln Filterfunktionen durchführen. Für jeden Dienst (Telnet, FTP, HTTP etc.), der erlaubt werden soll, führt man einen Proxy ein. Dieser Proxy steuert den Zugriff auf diesen Dienst. So kann z. B. ein FTP-Proxy bestimmte FTP-Befehle, wie den `delete`-Befehl, verbieten. Gegebenenfalls werden weitere Bedingungen an die Nutzung geknüpft. So besteht beispielsweise die Möglichkeit einer ausführlichen Protokollierung für die unterschiedlichen Dienste.

2 Versuchsdurchführung

2.1 Paketfilter

Im ersten Versuchsteil sollten Paketfilter untersucht werden. Folgendes Szenario war vorgegeben : Eine Firma möchte ihren Mitarbeitern den Zugriff auf Webseiten, ihren Kunden Zugriff auf den eigenen Webserver gestatten und ihren Kunden die Möglichkeit geben, via FTP auf den FTP-Server der Firma zuzugreifen. Diese Gegebenheiten wurden mit den Praktikumsrechnern simuliert. Dazu wurde die Praktikumsgruppe in zwei Teile unterteilt, eine Gruppe repräsentierte das *Internet* und die andere Gruppe das (firmeninterne)*Intranet*. Auf den einzelnen Rechnern waren verschiedene Dienste aktiv. Aufgabe war es jetzt herauszufinden, welche Dienste laufen und ob das für die oben erwähnten Anforderungen aus sicherheitstechnischen Aspekten sinnvoll ist. Zunächst wurden mit einem Portscanner zwei Rechner des *Internets* untersucht. Ein Portscanner ist ein Programm, mit dessen Hilfe die Ports eines Zielrechners nach offenen Diensten abgetastet werden können. Von unserer Gruppe wurden die Rechner `192.168.2.230` und `192.168.2.210` gescannt. Dabei ergab sich folgendes Ergebnis :

Rechner	Dienste
`192.168.2.230`	ftp (Port 21), ssh (Port 22), telnet (Port 23), time (Port 37), finger (Port 79), pop-3 (Port 110), login (Port 513), shell(Port 514), samba-swat(Port 901), X11 (Port 6000)
`192.168.2.210`	ftp (Port 21), ssh (Port 22), telnet (Port 23), domain (Port 53),finger (Port 79), http (Port 80), login (Port 513), X11 (Port 6000)

Hier sind wesentlich mehr Dienste aktiv als benötigt. Das ist aus sicherheitstechnischen Aspekten natürlich nicht vertretbar, da sich ein Angreifer die nicht benötigten Dienste zu Nutze machen könnte. Neben Diensten, die das Betriebssystem benötigt, sollten hier nur FTP und HTTP zugelassen werden. Im Folgenden wurde durch den Praktikumsleiter ein Paketfilter aktiviert, der die Ports, denen nicht benötigte Dienste zugeordnet waren, sperrt. Im Anschluss wurde der Rechner `192.168.2.210` erneut gescannt.

Rechner	Dienste
`192.168.2.210`	ftp (Port 21), domain (Port 53), http (Port 80)

Diese Variante ist wesentlich sicherer als die vorherige. Im Anschluss an diesen Versuchsteil wurde das Erstellen von Regeln für Paketfilter geübt. Zunächst wurden Regeln für eine HTTP-Verbindung aufgestellt. Pakete, die auf die Regeln zutreffen, sollten vom Paketfilter akzeptiert werden, alle weiteren zurückgewiesen werden.

Quell-Adr.	Ziel-Adr.	Protokoll	Quell-Port	Ziel-Port	ACK-Bit
extern	intern	TCP	≥ 1024	80	*
intern	extern	TCP	80	≥ 1024	$\surd$
intern	extern	TCP	≥ 1024	80	*
extern	intern	TCP	80	≥ 1024	$\surd$

Der Verbindungsaufbau im TCP funktioniert mittels drei-Wege-Handshake. Der Client sendet an den Server ein Segment, welches mitteilt, dass der Client mit dem Server einen Verbindungsaufbau wünscht. In diesem Segment ist ein das SYN-Bit gesetzt. Der Server bestätigt, indem er ein Segment sendet, in dem das ACK-Bit (Acknowledgement) und das SYN-Bit gesetzt ist. Daraufhin bestätigt der Client den Empfang dieses Segments (ACK) und beginnt mit der Übertragung der Daten. In den weiteren Segmenten ist immer

das ACK-Bit gesetzt. Aus diesem Vorgang erklären sich die obigen Filterregeln. Die ersten beiden Zeilen beinhalten Filterregln, die das Zugreifen von Außen auf den eigenen HTTP-Server ermöglichen. Dazu wird zunächst vom (externen) Client eine Verbindungsanfrage gestellt. Das ACK-Bit ist nicht gesetzt. Der Server bestätigt mit ACK. Bei allen weiteren Paketen ist das ACK-Bit gesetzt. Bei Paketen, die vom Client kommen, kann das ACK-Bit gesetzt sein, muss aber nicht gesetzt sein. Das ACK-Bit ist also beliebig (* in der Filterregel-Tabelle). Bei allen Paketen, die den Server verlassen, muss hingegen das ACK-Bit gesetzt sein ($\surd$ in der Filterregel-Tabelle). Die letzten beiden Zeilen der Tabelle beinhalten Filterregeln, die das Zugreifen auf einen externen HTTP-Server ermöglichen. Die Regeln lassen sich analog erklären. Im Rahmen dieser Ausarbeitung sollten die Filterregln für einen FTP-Zugriff von einem externen Client auf einen (firmen)internen Server erstellt werden. FTP benötig zwei separate TCP-Verbindungen: eine um Kommandos und Antworten zwischen Client und Server zu übertragen (Kommandokanal), und eine weitere um Daten auszutauschen (Datenkanal). Es wird zwischen aktiven und passiven Modus unterschieden. Im aktiven Modus baut der Server einen Datenkanal auf Port 20 auf. Im passiven Modus initiiert der Client eine weitere Verbindung für den Datenkanal. Daraus ergeben sich die folgenden Regeln :

Quell-Adr.	Ziel-Adr.	Protokoll	Quell-Port	Ziel-Port	ACK-Bit
extern	intern	TCP	≥ 1024	21	*
intern	extern	TCP	21	≥ 1024	$\surd$
extern	intern	TCP	≥ 1024	≥ 1024	*
intern	extern	TCP	≥ 1024	≥ 1024	$\surd$
intern	extern	TCP	20	≥ 1024	*
extern	intern	TCP	≥ 1024	20	$\surd$

2.2 Proxies

2.2.1 HTTP-Proxy

Im zweiten Teil des Praktikums wurden Proxies näher betrachtet. Hierzu wurde mit Hilfe des Browsers Netscape die Seite `http://www.inter.net/hydra.html` aufgerufen. Zu diesem Zeitpunkt wurde noch kein Proxy verwendet. Die Seite `hydra.html` enthielt aktive Inhalte in Form von Javascript. Folgende Zeilen des Quelltextes der Seite enthielten Javascript-Elemente :

```
  .
  .
  .
  <script language="javascript">
  <!--
    var herakles = false;

    function hydra() {
      if ( ! herakles ) {
        window.open("http://www.inter.net/hydra.html");
        window.open("http://www.inter.net/hydra.html");
      }
    }
  //-->
  </script>
   .
   .
   .
   .
<map name="hydra">
  <area shape=rect coords="160,100,179,119" href="javascript:this.close()"
                   onClick="herakles=true">
</map>
  .
  .
  .
```

Die Javascript Anweisungen hatten zur Folge, dass jedesmal, wenn man versuchte, das Fenster zu schließen, zwei neue erzeugt wurden. Dieses Verhalten ist natürlich vollkommen unerwünscht. Im Folgenden wurde für das HTTP-Protokoll ein HTTP-Proxy verwendet. Nachdem der HTTP-Proxy bei den Proxy-Einstellungen des Browsers eingetragen wurde, konnte die Seite erneut aufgerufen werden. Diesmal wurden Javascript - Inhalte durch den Proxy her-

ausgefiltert. Die Seite konnte jetzt ohne Komplikationen geschlossen werden. Anstelle von Javascript war im Quelltext Folgendes zu erkennen:

```
    .
    .
    .
  <fwtk removed script at firewall></fwtk>
    .
    .
    .
<map name="hydra">
  <area shape=rect coords="160,100,179,119" href="filtered://-removed-"
                   noClick="herakles=true">
</map>
    .
    .
    .
```

2.2.2 FTP-Proxy

Im weiteren Verlauf wurde ein Versuch mit einem FTP-Proxy durchgeführt. Zunächst war der FTP-Proxy nicht aktiv. Mittels FTP-Programm konnte auf einen FTP-Server zugegriffen werden. Es war u. a. mit dem `delete`-Befehl möglich, Dateien auf dem FTP-Server zu entfernen. Im Folgenden wurde der FTP-Poxy durch den Praktikumsleiter aktiviert. Nun war es nicht mehr möglich, einen `delete`-Befehl auf dem FTP-Server abzusetzen. Der `delete`-Befehl wurde durch den FTP-Proxy abgefangen.

3 Diskussion der Ergebnisse

Der Versuch hat die Notwendigkeit von Sicherheitsvorkehrungen in einem Netzwerk aufgezeigt. Ohne Sicherheitsvorkehrungen können Angreifer ohne größere Schwierigkeiten auf einen Rechner bzw. ein Netzwerk mit Internet-Anbindung zugreifen. Ein kombinierter Einsatz von Paketfiltern und Proxies verringert das Risiko, Opfer einer Attacke zu werden, deutlich. Mit Hilfe von Paketfiltern lässt sich auf den Ebenen drei und vier des ISO/OSI - Refe-

renzmodells der Zugriff auf das eigene Netzwerk deutlich einschränken (z. B. durch das Sperren nicht benötigter Ports). Mit Hilfe von Proxies lassen sich die Protokolle der Anwendungsschicht relativ gut kontrollieren. So können z. B. durch einen HTTP-Proxy kritische Inhalte aus Webseiten herausgefiltert werden. Zu einer leistungsfähigen Firewall gehören natürlich nicht nur Paketfilter und Proxies, sondern auch weitere Komponenten, wie z. B. ein Intrusion Detection System. Auf eine Sache sei an dieser Stelle aber noch hingewiesen :

Einen hundertprozentigen Schutz vor Angriffen kann auch der Einsatz einer Firewall nicht gewährleisten. Eine Firewall kann nur gegen bereits bekannte Angriffs-Methoden schützen. Absolute Sicherheit gegenüber Fremdübergriffen können nur Insellösungen bieten. Einzelplatz-Arbeitsstationen und LANs, die nicht weiter vernetzt sind, können auch nicht von Außerhalb des eigenen Systems angegriffen werden.