

Aufgabe 9.1: Erste Analyseszenarien

Zeichnen Sie den durch die unten stehenden Szenarien verursachten Netzwerkverkehr mittels ethereal auf.

Ein Konsolenfenster starten, su und das root-Passwort eingeben und ethereal starten. In Ethereal geht man ins Menü Capture - Options und aktiviert die Protokollaufzeichnung für das Netzwerk.

Um eine Telnet-Verbindung aufzubauen, muss der entsprechende Daemon gestartet werden. Das erreicht man mit `rcxinetd start`.

Beantworten Sie unter Verwendung der Analysemöglichkeiten von ethereal für jedes Szenario und den hierdurch verursachten Netzwerkverkehr folgende Fragen:

- Welche Protokolle werden jeweils auf MAC-, Vermittlungs- und Transportschicht verwendet?

MAC-Schicht: Ethernet (ARP, DNS, HTTP, ICMP, TCMP)

Vermittlungsschicht: IP (DNS, HTTP, ICMP, TCMP)

Transportschicht: UDP (DNS), TCP (HTTP)

- Welche Rechner werden für die Kommunikation der beiden Endgeräte jeweils auf MAC- und auf IP-Ebene adressiert?

Auf MAC-Ebene sind das Sender und Router, auf der IP-Ebene sind das der Sender und der Empfangsrechner.

- Wie lautet im Falle der Kommunikation über TCP oder UDP jeweils das verwendete socket-Paar?

Es kommt darauf an, auf welches Socket-Paar (Source und Destination, möglicherweise mit well-known Ports) sie sich einigen und auf welchem Port die Antwort zurückgegeben werden soll.

- Welche Auffälligkeiten können Sie sonst noch beobachten?

Zwischen ssh und telnet gibt es Unterschiede: ssh ist verschlüsselt und telnet "offen".

Hier nun die Experimente:

- Ein ping auf einen anderen PC im Security Labor sowie auf einen Rechner im FB I-Netz, z.B. linas.

ICMP

- Eine DNS-Anfrage mittels nslookup für einen beliebigen DNS-Namen.

UDP

- Eine telnet-Verbindung mittels telnet auf einen anderen PC im Security Labor.

TCP

- Eine ssh-Verbindung auf einen anderen PC im Security Labor.

TCP

- Eine Web-Anfrage auf einen externen Web-Server.

TCP

Hinweis:

ethereal bietet einige Möglichkeiten zur Unterstützung der Analyse, z.B. Filterung des anzuzeigenden Netzwerkverkehrs und einige Tools. Interessant ist u.a. die Möglichkeit, die Anzeige auf die Pakete der aktuellen TCP-Verbindung zu beschränken und die ausgetauschten Daten dieser TCP-Verbindung anzeigen zu lassen (Tools->Follow TCPStream)

Aufgabe 9.2: Vergleich telnet und ttcp

- Zeichnen Sie den durch die folgenden Experimente erzeugten Netzwerkverkehr jeweils mit ethereal auf (und speichern Sie die Ergebnisse ab ;-):
 - eine telnet-Verbindung mittels telnet auf einen anderen Rechner; führen Sie ein oder zwei kurze Kommandos aus und beenden Sie die telnet-Sitzung anschließend wieder;
 - eine Performance-Messung zwischen zwei Rechnern mittels ttcp; übertragen Sie dabei eine Datei von einigen hundert KByte
- Finden Sie die durch die obigen Experimente erzeugten TCP-Verbindungen heraus. Halten Sie jeweils fest, welcher Rechner das
 - "active open",
 - "passive open",
 - "active close",
 - "passive close"macht.

Gucken: Wer macht den ersten Schritt beim Verbindungsaufbau-/abbau. Nicht der Client, sondern der Server schließt die Verbindung (=active close) bei einem "exit".

- Vergleichen Sie die TCP-Sitzungen hinsichtlich der Entwicklung folgender Daten:
 - TCP-Segmentgrößen
 - RoundTripTime (RTT)
 - Window-Size
 - Datendurchsatz (throughput)

Hinweis: ethereal bietet einige Hilfsmittel zur Ermittlung der o.g. Daten. In Ethereal: TCP - Analyse TCP-Graph ...

Beide sind TCP-Sitzungen. Der Puffer-Platz auf der Empfangsseite wird extrem angepasst, 10-20 TCP-Segmente werden gesendet, bis auf ein ACK gewartet werden muss. Mit Ethereal kann man erkennen, daß der Anfang wie bei der telnet-Sitzung aussieht (Paket-Verkehr in beide Richtungen), mit der Zeit aber mehr Pakete in eine Richtung gehen und erst dann ACKs zurückgehen und die Window-Size angepasst wird.

Gründe:

- Overhead vermeiden (mittels Slow-start-Algorithmus)

Die TCP-Segmentgröße verändert sich nicht über Maximum (Größe wird bestimmt dadurch, daß sie in Pakete verpackt werden müssen (Rahmengröße des Ethernet-Rahmens) und Fragmentierungen vermieden werden). Die Daten können 1.460 Byte groß sein (Headerdaten werden abgezogen).

Aufgabe 9.3: Stati von TCP-Verbindungen

Kontrollieren Sie während und nach einer telnet-Sitzung den Status der TCP-Verbindung auf

Client und Server-Seite mittels netstat -t.

Analysieren Sie auch den mit netstat ersichtlichen Unterschied zwischen folgenden Situationen:

- Beendigung der telnet-Verbindung durch Eingabe des Kommandos exit im kommandozeilenbasierten telnet-Client.

Wenn man sich mit netstat den Status der tcp-Verbindung anguckt, erkennt man, daß ein time-wait-Status aufgerufen wird, in dem gewartet wird, weil keine Sicherheit besteht, ob das letzte Paket bestätigt wurde. Erst wenn ein time-out folgt, wird die Verbindung geschlossen.

- Beendigung der telnet-Verbindung durch Beenden des telnet-Prozesses auf Client-Seite (z.B. durch kill <telnet-pid>).

Analysieren Sie auch mittels ethereal für die o.g. Situationen, welche Seite jeweils das active- und welche Seite das passiv-close macht.

Hier schließt nicht der Server die Verbindung, sondern der Client. Man sieht auch den time-wait-Zustand auf Client- statt auf Serverseite.

Aufgabe 9.4: Detailanalyse einer ftp-Sitzung

- Vorbereitungen:

In dieser Aufgabe sollen Sie eine vergleichende Analyse zweier ftp-Sitzungen durchführen. Um eine ftp-Sitzung auf einen Nachbarrechner durchführen zu können, muss auf diesem Rechner ein ftp-Server gestartet werden.

ftp-Server werden unter Linux typischerweise auf Anforderung durch den inetd- bzw. xinetd- Prozess gestartet. Verändern Sie die Konfiguration Ihres PCs so, dass der ftp-Server gestartet wird, z.B. mittels yast.

Nun zu den eigentlichen Aufgaben:

- Zeichnen Sie den Netzwerkverkehr
 - einer ftp-Sitzung im active-mode und
 - einer ftp-Sitzung im passive-mode auf.

Zwischen active- und passive-mode können Sie im kommandozeilen-basierten ftp-Client mit Hilfe des Kommandos passive umschalten.

Jede der beiden ftp-Sitzungen sollte folgendermaßen ablaufen:

- Kommando: ftp <Nachbarrechner>
- Melden Sie sich als Benutzer seclab an.
- Schalten Sie in Ihrer ftp-Sitzung mit dem Kommando passive in active- bzw. passive-mode.
- Transferieren Sie eine kleine Datei vom Nachbarrechner in ein temporäres Verzeichnis auf Ihrem Rechner.
- Beenden Sie die ftp-Sitzung.

Analysieren Sie vergleichend die aufgezeichneten Sitzungen. Insbesondere sollen folgende Analysepunkte berücksichtigt bzw. Fragen beantwortet werden:

- Identifizieren Sie pro ftp-Sitzung alle TCP-Verbindungen und notieren Sie zu jeder TCP-Verbindung:

- IP-Adressen und Ports der kommunizierenden Rechner,
- die Verbindungsrichtung (also wer macht active- / passive - open / -close)?

(Hier liegen die wesentlichen Unterschiede zwischen der active- und passive-mode-Variante!!)

- Woher wissen die Rechner die zu verwendenden Port-Nummern?

Wichtig:

Wichtig

- Kurzbeschreibung des Zwecks bzw. der Verwendung der jeweiligen TCP-Verbindung
- Zeitliche Abfolge der Verbindungsauf- und abbauten.
- Welche Kommandos kann man im ftp-Protokoll auf Applikationsebene beobachten?

Einige Protokolle wie FTP geben sich nicht mit einer Applikationssitzung (wie einer TCP-Verbindung) zufrieden, sondern initiieren viele verschiedene Applikationssitzungen.

FTP-Client und FTP-Server tauschen beispielsweise dauernd Steuerungskommandos aus ("ich sende Benutzername", "schicke Passwort") aus, der Client fordert schließlich mittels "ls" das Directory-Listing an. Dazu wird eine zweite Verbindung mittels passive mode für die Datenverbindung geöffnet (eine eigene TCP-Verbindung wird auf einem nicht well-known Port, also ≥ 1024 , geöffnet, auf die sich Client und Server über die Steuerleitung einigen). Nach Übertragung der Daten wird die Verbindung wieder geschlossen. Folgt später nochmal ein Befehl wie z.B. get, wird erneut eine neue Datenverbindung eine neue TCP-Verbindung geöffnet.

Im active mode wird eine TCP-Verbindung vom Server zum Client hergestellt, die ebenfalls zur Datenübertragung genutzt wird. Auch hier wird auf einen Port ≥ 1024 zurückgegriffen, der über die Steuerleitung vereinbart wird. Da auf dem Client eingehende Pakete oft von Routern über Paketfilter oder Firewalls blockiert werden, kann es dabei zu Störungen kommen.



