Grundpraktikum Netz- und Datensicherheit

Thema:

Netzwerk-Analyse mit *nmap & Wireshark*

Lehrstuhl für Netz- und Datensicherheit
Ruhr-Universität Bochum

Versuchdurchführung: Raum ID 2/168

Betreuung: Marcus Niemietz
Zusammengestellt von: Sven Schäge, Michael Psarros, Endres Puschner, Dominik Noß
Stand: 2. Dezember 2018
Version: 3.0

Ziel dieses Praktikums ist u.A. das Verständnis von Abhörmaßnahmen. Seien Sie vorsichtig bei der Eingabe von persönlichen Zugangsdaten, wie etwa Email und Login-ID. Gehen Sie auch in den folgenden Versuchen von einer nicht vertrauenswürdigen Umgebung aus. Es wird keine Haftung übernommen. Benutzen Sie USB-Sticks für den Datentransfer.

1 Portscanner [6]

Ein Portscanner ist eine Software, mit der überprüft werden kann, welche Dienste ein mit TCP oder UDP arbeitendes System über das Internetprotokoll anbietet. Der Portscanner nimmt dem Anwender dabei die Arbeit ab, das Antwortverhalten eines Systems selbst mit einem Sniffer zu untersuchen und zu interpretieren. Oft bieten Portscanner auch Zusatzfunktionen wie Betriebssystem- und Diensterkennung an, obwohl sie nichts mit dem eigentlichen Portscannen zu tun haben.

Bei einem Portscan wird auf den gewünschten Ports versucht, eine Verbindung zu dem Rechner aufzubauen und das Antwort-Verhalten des Zielsystems je nach verwendetem Protokoll interpretiert. Im Falle von TCP z.B. werden i.d.R. SYN-Pakete von dem Portscanner verschickt. Ein Port, der mit einem SYN/ACK-Paket antwortet, wird als offen bezeichnet, ein Port, der mit einem RST/ACK-Paket antwortet, als geschlossen und ein Port, der nicht antwortet, als gefiltert.

Um auszuschließen, dass Antwort-Pakete übersehen werden, überprüfen Portscanner im letzteren Fall einen Port i.d.R. mehrfach, und passen ihren Timeout dynamisch an. Das Verfahren wird für alle zu untersuchenden Ports wiederholt, wobei, um Zeit zu sparen, mehrere Ports parallel geprüft werden.

SYN-Scans, bei denen der Portscanner ein erhaltenes SYN/ACK Paket mit einem RST-Paket terminiert, wurden in der Vergangenheit auch als „Stealth-Scans“ bezeichnet, da die Anwendung, die den entsprechenden Port bedient, in diesem Fall keinen Verbindungsversuch wahrnimmt und dies nicht protokollieren kann. Praktisch jede halbwegs aktuelle Firewall-Software kann jedoch die Pakete erfassen, und über deren Protokoll-Dateien kann der SYN-Scan erkannt werden.

Da es sich dabei um ein völlig protokollkonformes Verhalten handelt, sind Portscans teilweise legal (Anmerkung: dies ist rechtlich umstritten). Ausnahmen bestehen, wenn durch Portscans ein Denial of Service Angriff ausgelöst wird, wenn z.B. die dem Ziel zur Verfügung stehende Bandbreite verbraucht wird.

Die o.g. Zusatzfunktionen wie Betriebssystems- OS-Fingerprinting und Dienst-Erkennung, für die z.B. der Portscanner nmap bekannt ist, sind streng genommen keine Portscans mehr und ihr Einsatz kann aufgrund eines nicht ganz auszuschließenden Absturzrisikos beim Ziel problematisch sein.

2 Sniffer [7]

2.1 Engere Bedeutung

„Sniffer“ ist ein eingetragenes Warenzeichen des Herstellers „Network General“; es bezeichnet ein Produkt der sog. LAN-Analyse. Da dieses Produkt als eines der ersten auf dem Markt war, und da sein Name so eingängig ist, hat sich der Name „Sniffer“ allgemein durchgesetzt zur Bezeichnung vielfältigster Produkte der LAN-Analyse.

2.2 Allgemeine Bedeutung

Ein Sniffer (engl. „to sniff“ für riechen, schnüffeln) ist eine Software, die den Datenverkehr eines Netzwerks empfangen und darstellen kann. Ein Sniffer kennt den so genannten non-promiscuous mode und den Promiscuous Mode.

Im non-promiscuous mode wird der ankommende und abgehende Datenverkehr des eigenen Computers gesnifft.

Im Promiscuous Mode sammelt der Sniffer den gesamten Datenverkehr an die in diesen Modus geschaltete Netzwerkschnittstelle. Es werden also nicht nur die an ihn adressierten Frames empfangen, sondern auch die nicht an ihn adressierten. Der Adressat eines Frames wird in Ethernet-Netzwerken anhand der MAC-Adresse festgelegt.

Weiterhin ist es von der Netzwerkstruktur abhängig, welche Daten ein Sniffer sehen kann. Werden die Computer mit Hubs verbunden, kann sämtlicher Traffic von den anderen Hosts mitgeschnitten werden. Wird ein Switch verwendet, ist nur wenig oder gar kein Datenverkehr zu sehen, der nicht für das sniffende System selbst bestimmt ist. Allerdings gibt es in diesem Fall mehrere Möglichkeiten wie z. B. ARP-Spoofing, um trotzdem die Frames empfangen zu können. Ein Switch darf also nicht als Sicherheitsfeature gesehen werden.

Es gibt mehrere Gründe, einen Sniffer zu benutzen:

- Diagnose von Netzwerkproblemen
- Eindringungsversuche entdecken
- Netzwerktraffic-Analyse und Filterung nach verdächtigem Inhalt
- Datenspionage

Der Promiscuous Mode bezeichnet einen bestimmten Empfangsmodus für netzwerktechnische Geräte. In diesem Modus liest das Gerät den gesamten ankommenden Datenverkehr, an die in diesen Modus geschaltete Netzwerkschnittstelle mit und gibt die Daten zur Verarbeitung an das Betriebssystem weiter. Normalerweise würde das Gerät nur die an sich selbst gerichteten Pakete verarbeiten, was zum Beispiel in Ethernetnetzen über das Auswerten der MAC-Adresse geschieht.

Geräte, die diesen Modus benutzen, können Kombinationen aus Switch und Router, Netzwerktester oder auch normale Computer mit Anschluss an ein Netzwerk sein.

3 NMAP

NMAP[4] ist ein sehr umfassender und vermutlich der bekannteste Portscanner, insbesondere auf Unix-Systemen. Mit seinen vielfältigen Scantechniken ist er auch als Scan-Engine für andere Programme, wie den Sicherheitsscanner Nessus, sehr weit verbreitet. Es steht das grafische Frontend Zenmap[1] zur Verfügung, mit dem sich alle Einstellungen sehr komfortabel vornehmen lassen. Darüber hinaus gibt einem das Programm immer den äquivalenten Shell-Befehl an. Um NMAP in vollem Umfang nutzen zu können werden root-Rechte gefordert.

Für den sinnvollen Umgang mit NMAP ist zumindest grundlegende Kenntnis der verschiedenen Scantechniken nötig, die im Folgenden vermittelt werden soll.

3.1 Scannen allgemein

Für das Scannen eines Hosts oder IP-Bereichs gibt es genauso unterschiedliche Motivationen wie Techniken. Die ursprüngliche Motivation liegt darin begründet, die Verfügbarkeit von Ressourcen im Netzwerk zu überprüfen. Zu diesem Zweck steht zunächst das Inernet Control Message Protocol (ICMP) zur Verfügung, welches unter anderem den echo request bzw. reply, bekannter unter der Bezeichnung ping, definiert.

Darüber hinaus bietet das Transmission Control Protocol (TCP) eine gute Möglichkeit, detailierteren Aufschluss über die auf dem Host arbeitenden Dienste zu erlangen. Dazu wird ganz einfach ein Verbindungsaufbau auf einem bestimmten Port vorgenommen und die Reaktion abgewartet. Diese kann üblicherweise aus einer Verbindungsannahme oder Abweisung bestehen. Es ist auch denkbar, dass keinerlei Reaktion wahrzunehmen ist. In diesem Fall wurde das Paket vermutlich von einem Paketfilter (z.B. einer Firewall) fallen gelassen. Theoretisch denkbar ist auch, dass das Paket sein Ziel nicht erreicht hat, wobei dann zumeist wieder das ICMP-Protokoll zum Tragen kommt und eine Fehlermeldung an den Absender richtet. Abschließend dazu ist zu sagen, dass der Port, auf den eine Verbindung vorgenommen wird, nur in so weit aussagekräftige Informationen liefern kann, als dass den gängigen Diensten – also z.B. http, ftp, ssh, pop,... – durch die Internet Assigned Numbers Authority (IANA, www.iana.org) jeweils ein so genannter Well Known Port zugewiesen wird, unter dem sie sich üblicherweise auch ansprechen lassen, wenn sie überhaupt arbeiten.

Im Zeitalter globaler Vernetzung gerät eine weitere Motivation immer mehr in den Mittelpunkt: das Ausspähen fremder Netzwerkinfrastrukturen und von fremden Hosts angebotener Dienste, sowie der daraus eventuell resultierenden Schwachstellen im System. Diese Information ist die grundlegende Basis für einen Angriff. So ist das Scanning auf der einen Seite interessant für den potentiellen Angreifer, auf der anderen Seite, zur Gewährleistung eines hohen Sicherheitsniveaus, wichtig für den System-Administrator.

Im Laufe der vergangenen Jahre sind neben den oben genannten „regulären“ Scan-Techniken die so genannten „Stealth“-Techniken immer populärer geworden. Ziel ist es dabei, möglichst unbemerkt einen Scan durchzuführen, damit das potentielle Opfer keine Gefahr wittern und im Vorfeld Gegenmaßnahmen einleiten kann. An dieser Stelle sei auf die heutzutage weit verbreitete Nutzung so genannter Intrusion Detection Systems (IDS) hingewiesen, welche unter anderem aggresive Portscanns detektieren.

3.2 Stealth-Techniken

Die folgenden Scantechniken beziehen sich immer auf TCP-Verbindungen.

Der halboffene Scan ist die bekannteste und zugleich simpelste Methode eines Stealth-Scans. Hierbei

wird lediglich ein Verbindungsaufbau initiiert, nach dem Erhalt der Bestätigung aber nicht weiter vollzogen. Das hat den gravierenden Vorteil, dass dieser „missglückte“ Verbindungsaufbau bei der Gegenstelle im Allgemeinen nicht protokolliert wird.

Die Tatsache, dass die TCP-Richtlinien (RFC 793) die Beantwortung eines fehlerhaft empfangenen Paketes vorschreiben, machen sich die Scantechniken X-Mas-, Fin- und Null-Scan zu nutze. Beim X-Mas-Scan wird ein Paket mit gesetztem URG-, PSH- und FIN-Flag gesendet. Im Falle des FIN-Scans wird nur das FIN-Flag gesetzt und beim Null-Scan wird gar kein Flag gesetzt.

Die unterschiedlichen Scantechniken lassen sich hervorragend mit Hilfe von Wireshark nachvollziehen.

3.3 UDP-Scan

Das Scannen eines Hosts auf Basis des User Datagram Protocols (UDP) ist im Allgemeinen weniger interessant als ein TCP-Scan. Das hängt in erster Linie damit zusammen, dass UDP ein verbindungsloses Protokoll ist. Das Nichtvorhandensein eines Dienstes wird in diesem Fall durch die ICMP-Meldung „Destination unreachable“ signalisiert. Nachdem die Internet Engineering Task Force aber in Ihrer Richtlinie für IPv4-Routing (RFC 1812) die Verminderung von ICMP-Meldungen fordert, ist der UDP-Scan ein zumeist sehr langwieriger Prozess. Es gibt jedoch einige Ausnahmen, die den UDP-Scan wieder sehr interessant machen können. Dazu zählen trojanische Pferde, wie das inzwischen in die Tage gekommene Back Orifice, welche oftmals auf UDP-Ports lauschen.

3.4 NMAP-Parameter

Tabelle 1 fasst zu allen in diesem Teil aufgeführten Scantechniken die entsprechenden Parameter beim Aufruf von NMAP zusammen. Darüber hinaus gibt es noch viele weitere Möglichkeiten NMAP zu nutzen, die Sie beispielsweise durch das Aufrufen der man-Pages (man nmap) erfahren können.

Parameter	Bedeutung
-sP	Scan durch ICMP-Echo-Request
-sT	Einfacher TCP-Connect-Scan
-sS	Halboffener Scan bzw. SYN-Scan
-sX	X-Mas-Scan
-sF	FIN-Scan
-sN	Null-Scan
-sU	UDP-Scan

Tabelle 1: Scantechniken von NMAP

Um tiefgründigere Informationen über die Protokolle TCP, UDP und ICMP zu erhalten sei auf den jeweiligen Request for Comment (RFC) verwiesen. RFCs geben die de facto Standards eines jeden Internet-Protokolls wieder. Sie werden von der Internet Engineering Task Force (IETF) verwaltet.

Transmission Control Protocols	RFC 793, zzgl. RFC 1122, RFC 1323
User Datagram Protocol	RFC 768
Internet Control Message Protocol	RFC 792

Tabelle 2: RFCs zu Protokollen der Internetschicht

Alle RFCs lassen sich von den Webseiten der IETF abrufen: `http://www.ietf.org/rfc.html`

4 Wireshark [3]

Ethereal war jahrelang die bekannteste Software zum komfortablen protokollieren von Netzwerkverbindungen. 2006 hat der Hauptentwickler die Firma Ethereal (welche die Namensrechte besitzt) verlassen, deswegen wurde die Software zu Wireshark umbenannt.

Wireshark bietet eine sehr gute Möglichkeit, die einzelnen Protokolle des TCP/IP-Modells anschaulich kennen zu lernen, Fehleranalysen durchzuführen, aber auch Verbindungen zu belauschen und dabei zum Beispiel Kennwörter unverschlüsselter Verbindungen zu erspähen. Auch bei der Analyse von Software ist es ein nützliches Werkzeug, um Verbindungen ins Internet zu entdecken. Das Programm ist für jede gängige Hardware-Plattform und ebenso für jedes gängige Betriebssystem erhältlich.

4.1 Mitschnitt starten

Nach dem Öffnen von Wireshark haben Sie mehrere Möglichkeiten, das Abhören zu starten. Die schnellste Variante besteht aus einem Klick auf "Start" (vgl. Abb. 1, grünes Symbol). Dabei benutzt Wireshark die Standard-Einstellungen.

Alternativ gelangen Sie durch einen Klick auf "Capture Options" in einen Dialog, in welchem Sie zahlreiche Einstellungen vornehmen können (vgl. Abb. 2).

Der *Promiscious Mode* erlaubt es, alle ankommenden Pakete mitzuschneiden - inklusive derjenigen, die an andere Teilnehmer adressiert sind. Wenn Sie das Häckchen setzen, dann werden alle über das Netzwerkkabel empfangenen Daten verarbeitet.

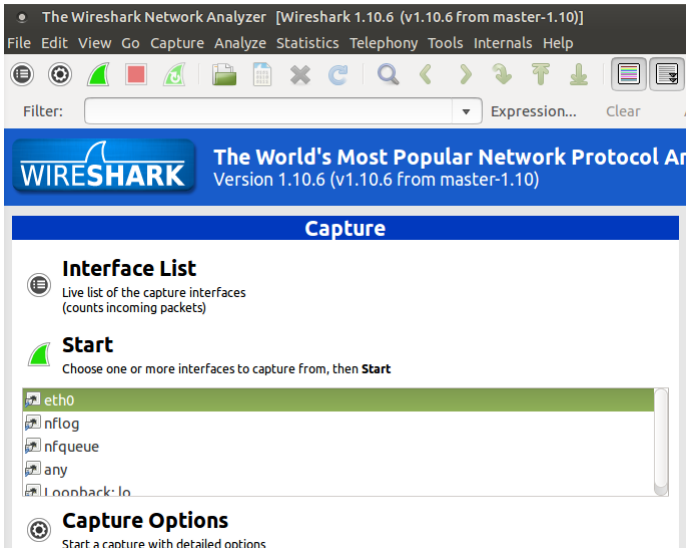


Abbildung 1: Startbildschirm

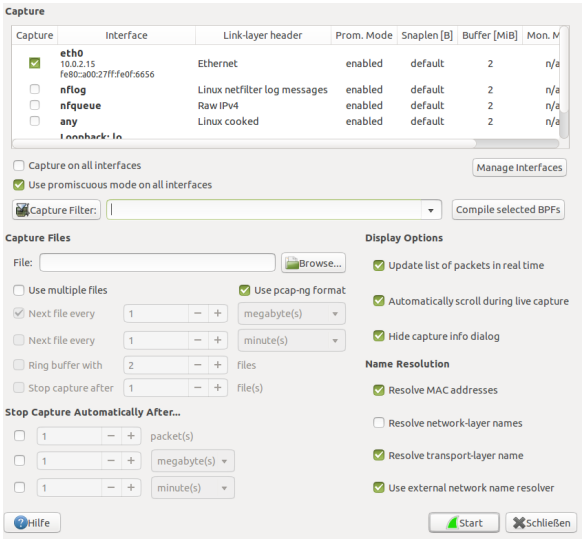


Abbildung 2: Konfiguration

Während des Mitschnitts können die Pakete bereits beobachtet werden. Um den Vorgang abzuschließen klicken Sie "Stop" (rotes Viereck).

Hinweis: Wireshark als root auszuführen, stellt ein Sicherheitsrisiko dar! Das Programm ist bereits korrekt konfiguriert, um unter dem Benutzer student zu laufen.

4.2 Mitschnitt auswerten

Nach klicken des Stop-Buttons wird der mitgeschnittene Datenverkehr angezeigt (vgl. Abb. 3). Sie können ihn auch schon während der Aufnahme durchsuchen. Im oberen Bereich finden Sie die mitge-

schnittenen Pakete in chronologischer Reihenfolge. Im mittleren Bereich lassen sich die detaillierten Paketinhalte disassembliert, das heißt im „Klartext“ betrachten, während im untersten Fenster die HEX-Darstellung zu finden ist.

Weil in vielen Fällen mehrere Verbindungen parallel ablaufen, ist es zunächst schwer, einen guten Überblick über die Daten zu gewinnen. **Wireshark bietet Ihnen einige Werkzeuge, die Ihnen das Leben leichter machen.** Es wird nun eine Auswahl davon zusammengefasst:

- Wireshark stellt eine Filter-Sprache zur Verfügung. Oberhalb der chronologischen Auflistung befindet sich ein Text-Feld, mit welchen die angezeigten Pakete gefiltert werden können. So führt die Eingabe des Filters "`tcp`" dazu, dass ausschließlich TCP-Pakete angezeigt werden. "`tcp contains 'example'` " zeigt nur TCP-Pakete, in welchen die Zeichenfolge "`example`" vorhanden ist. "`dns.flags == 0x0100`" gibt an, dass nur DNS-Queries gezeigt werden sollen.

  Nun können Sie beispielsweise bei einer mitgeschnittenen FTP-Verbindung nach dem Schlüsselwort USER oder PASS suchen.

  Da diese Art der Filterung sehr mächtig ist, sollten Sie sich mit ihrer **offiziellen Dokumentation**[10] befassen.

- Die Pakete, die zu einer TCP-, UDP- oder SSL-Verbindung gehören, können von Wireshark automatisch zusammengefasst werden. Klicken Sie dazu mit der rechten Maustaste auf ein Paket der gewünschten Verbindung und wählen Sie *Follow TCP Stream*.
  Es öffnet sich ein Fenster, in welchem die Payloads aller Pakete in einem Textfeld dargestellt werden. Das ist sehr praktisch, um etwa HTTP- oder POP3-Verbindungen für den Menschen lesbar darzustellen. Weiterhin wird der Display-Filter so gesetzt, dass ausschließlich Pakete angezeigt werden, die zur Verbindung gehören. So kann die untersuchte Verbindung Paketweise nachvollzogen werden.

- In den Paket-Details können Sie viele der dargestellten Elemente als Filter benutzen. Klicken Sie dazu rechts auf ein Detail, z.B. das IPv4-Adressfeld oder das Domain-Name-Feld in DNS, um es als Filter zu setzen oder mit dem derzeitigen logisch zu verknüpfen (vgl. Abb. 4).

  Damit ist es wesentlich komfortabler, interessante Funde zu verfolgen ohne diese Abtippen zu müssen.

- Über das Menu *Statistics* können Sie die Erstellung von Statistiken anregen. So können Sie beispielsweise zusammenfassen lassen, welche Protokolle in den Daten vorhanden sind (*Protocol Hierarchy*) und welche Teilnehmer im Netzwerk vorhanden sind oder miteinander kommunizieren (*Endpoints* und *Conversations*).

  Um die Datenübertragungsrate anzuzeigen, können Sie den *IO-Graph* öffnen. Hier können Unregelmäßigkeiten visuell erkannt werden.

  Wireshark kann Ihnen sogar Statistiken auf höheren Protokollen erstellen, wie etwa eine Zusammenfassung aller HTTP-Requests (*Statistics - HTTP - Requests*)

- Im Menu *File/Export Objects* können Sie mitgeschnittene Dateien exportieren, welche unverschlüsselt über SMB und HTTP übertragen wurden, wie z.B. HTML-Dokumente, Bilder, Executables.

Mit Wireshark können prinzipiell alle unverschlüsselt übertragenen Daten durchsucht werden. Bei verschlüsselnden Verbindungsprotokollen ist das Ausspähen m Allgemeinen nicht mehr möglich. Als Ersatz für z.B. FTP wird heutzutage das zur SSH-Suite gehörende SFTP verwendet. Die Standard-Konfiguration vieler Mail-Clients heutzutage sieht eine Übertragung mittels SSL vor.

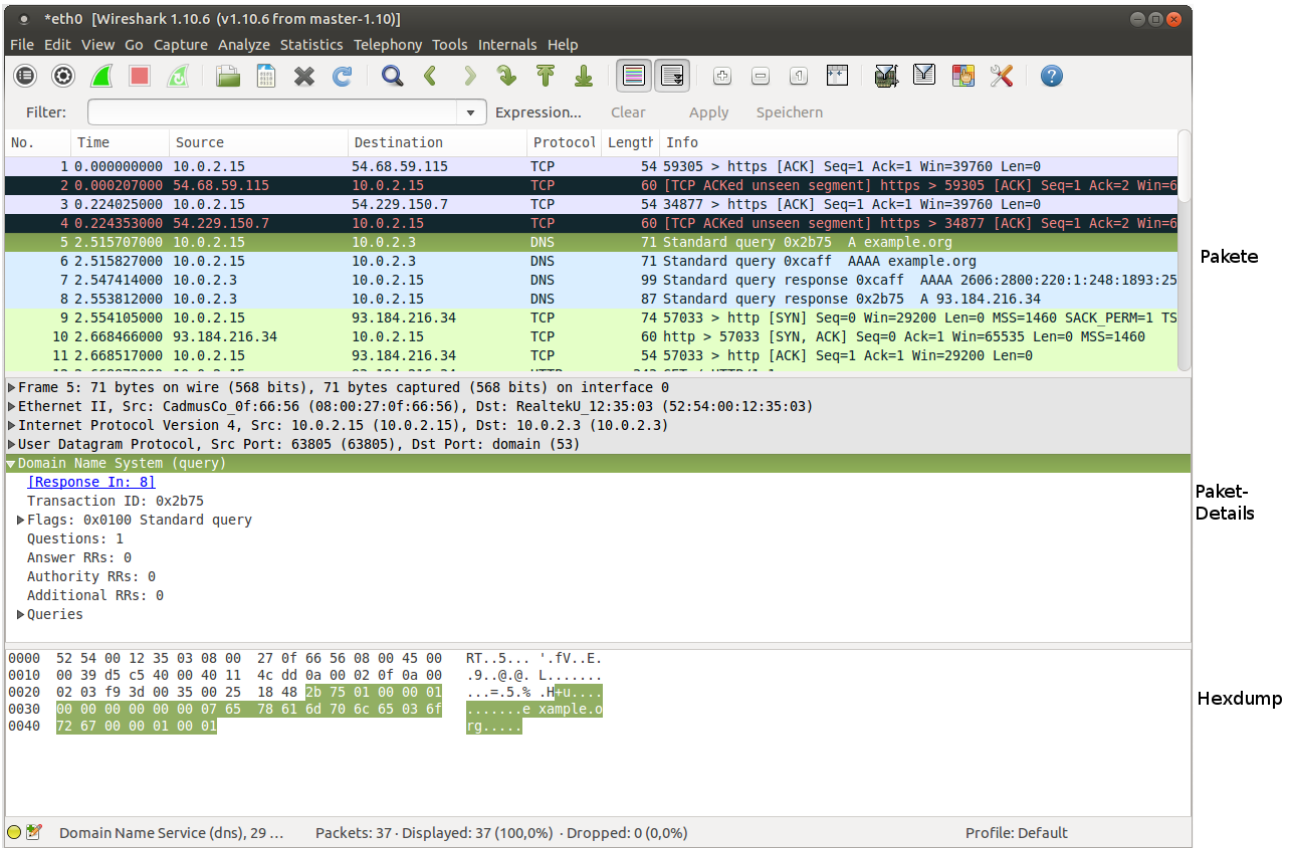


Abbildung 3: Die drei Bereiche von Wireshark: oben die chronologische Auflistung, mittig die Detail-Ansicht und unten der Hexdump.

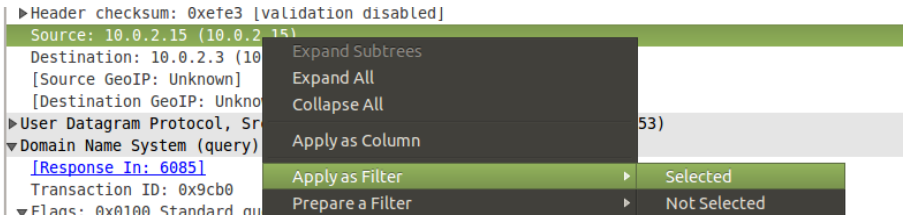


Abbildung 4: Durch Rechtklick auf ein Paket-Detail kann dieses als Filter benutzt werden.

5 Versuch

5.1 Versuchsaufbau

Auf den Rechnern ist Linux (Ubuntu Mate) installiert. Sie bearbeiten den Versuch mit Kommandozeilen-Befehlen und GUIs. Auf den Rechner sind alle vorher erwähnte Tools installiert.

Für diesen Versuch sind Sie unter dem Account „student“ angemeldet. Das Passwort für root-Rechte wird für den Versuch jeweils auf 12345 gesetzt.

Falls Sie etwas nicht verstehen, fragen Sie bitte zu Beginn den Mitarbeiter, der das Praktikum betreut. Ziel ist es, dass Sie die Vorgabe jedes Schrittes erfüllen und das Ergebnis für sich dokumentieren. **Schreiben Sie bei jedem Schritt das Zwischenergebnis auf**, damit der Bericht vollständig den Versuch abdeckt. Unvollständige Berichte werden vom Versuchsleiter nicht akzeptiert, was dazu führen kann, dass Sie die Schritte dann noch mal durchführen müssen.

Vorsicht: Während den Schritten 1-3 wird das Netzlabor möglicherweise keine Verbindung zum Internet haben (als Schutz gegenüber falsch gewählten Parametern bei nmap)

5.2 Schritt 1

Machen Sie sich mit der Netzwerktopologie vertraut. Ein Netzplan befindet sich im Anhang. Stellen Sie fest, welche IP-Adresse(n) Ihr PC hat und welche anderen Rechner er via TCP/IP erreichen kann. Stellen Sie Abweichungen zum Plan fest. Dokumentieren Sie das Ergebnis sorgfältig, wie auch alle folgenden Ergebnisse.

5.3 Schritt 2

Finden Sie mittels eines Stealth-Scans heraus, welche Dienste der Server in Ihrem Subnetz bereitstellt.

5.4 Schritt 3

Finden Sie mittels OS-Fingerprinting heraus, welches Betriebssystem die Rechner im Netzlabor benutzen. Gibt es nur Linux-Maschinen?

5.5 Schritt 4

Aktivieren Sie anschließend den Sniffer. Sniffen Sie für 5 Minuten die Netzwerksegmente, die für Sie (in Bezug auf Sniffen) erreichbar sind. Rufen Sie eine einzelne Website Ihrer Wahl auf, um interessanten Traffic für Sie und die anderen Gruppen zu erzeugen. *Beschränken Sie sich auf Seiten, die keine großen Dateien übertragen (Video-Portale etc.), da dies unter Umständen die Speicher-/Rechenkapazität der Rechner erschöpfen könnte.*
Sie dürfen gerne unterhaltsame Websites wählen.
Deaktivieren Sie den Sniffer nach Ablauf der 5 Minuten.

Erstellen Sie währenddessen eine stichpunktartige Auswertung des Ergebnisses. Geben Sie auch an, welche Filter Sie benutzt haben. Die Auswertung muss mindestens folgende Punkte umfassen:

1. Fügen Sie den Abschnitt "Statistics" in der *Comments Summary* in Ihren Bericht ein.
2. Welche Domains wurden mittels DNS-Anfragen angefragt?

3. Welche Ressourcen (z.B. "/index.html", "/img/doge.gif") wurden mittels HTTP abgerufen? Geben Sie auch die zugehörigen Domains an. Benutzen Sie die HTTP *Requests*-Statistik.
   Rufen Sie zwei gefundene Ressourcen ab (z.B. mit Firefox) und dokumentieren Sie das Ergebnis per Screenshot oder durch speichern der Ressource im Falle von Bildern. Bonuspunkte gibt es für besonders unterhaltsame Funde[1].
4. Fügen Sie ein abgefangenes Bild in ihren Bericht ein. Benutzen Sie dafür die Export-Funktion unter *File/Export Objects*.
5. Welche Website wurde von Ihnen aufgerufen? Wie können Sie den Aufruf im Mitschnitt finden?
6. Welche IPs sind im Mitschnitt enthalten? Benutzen Sie dafür die *Endpoints* Statistik. Welche der IPs sind vom Netzlabor, welche befinden sich im Internet?
7. Welche Protokolle wurden verwendet? Beschreiben Sie dies anhand der *Protocol Hierarchy*
8. Zeigen Sie, sofern vorhanden, einen Port-Scan. Eine Möglichkeit, einen solchen zu entdecken, bietet der Reiter *TCP* in der *Conversations* Statistik. Woran haben Sie den Scan erkannt?
9. Zeigen Sie, sofern vorhanden, das Zertifikat einer SSL-Verbindung. Sie finden dieses häufig in den ersten paar Paketen einer jeden SSL Verbindung. Auf welchen *Common Name* ist es ausgestellt?

Hinweis: Falls die Ihre Ergebnisse flächenmäßig zu groß sind für den Bericht, dürfen Sie sie sinnvoll reduzieren, z.B. mit GIMP.

5.6 Schritt 5

Aktivieren Sie den Sniffer. Führen Sie dann mindestens zwei Protokolle aus, die Passwörter übertragen. Beispielsweise Der Server *server.netzlabor* bietet Ihnen *HTTP Basic Authentication* sowie ein Login-Feld einer Web-App-Attrappe, welches über HTTP POST übertragen wird. Sie können auch mit einer Suchmaschine nach öffentlichen FTP- oder IRC-Servern suchen. Hinweis: nutzen Sie unverschlüsselte Protokolle, da die Login-Daten sonst nicht sichtbar sind.

Deaktivieren Sie dann den Sniffer. Suchen Sie die Passwörter im Daten-Mitschnitt und geben Sie die Position an. Geben Sie auch an, auf welche Weise man solche Passwörter protokollabhängig schnell finden kann (wenn man sie vorher nicht kennt). Deaktivieren Sie den Sniffer.

Ende des Versuchs

[1] Disclaimer: es gibt keine Bonuspunkte

6 Hinweise

6.1 Voraussetzungen für die Teilnahme

- Grundkenntnisse zum Arbeiten unter Linux
- Sie müssen mit Kommandozeilen umgehen können!
- Dieses Dokument muss vorher gelesen werden; siehe Kontrollfragen zum Verständnis des Dokuments
- Grundkenntnisse in TCP/IP sind erforderlich
- Machen Sie sich mit der Dokumentation zu nmap und Wireshark vertraut

Falls sie eine Voraussetzung nicht erfüllen, sprechen Sie unbedingt mindestens eine Woche vor dem Versuch den Versuchsleiter an, um das Wissen rechtzeitig aufholen zu können.

6.2 Schriftliche Versuchsauswertung

Jedes Team fertigt eine schriftliche Auswertung an. Diese sollte insbesondere die bei jedem Schritt verwendeten Befehle enthalten (also unbedingt dokumentieren, was Sie bei der Versuchsdurchführung getan haben) und die Ausgabe der Befehle erläutern.

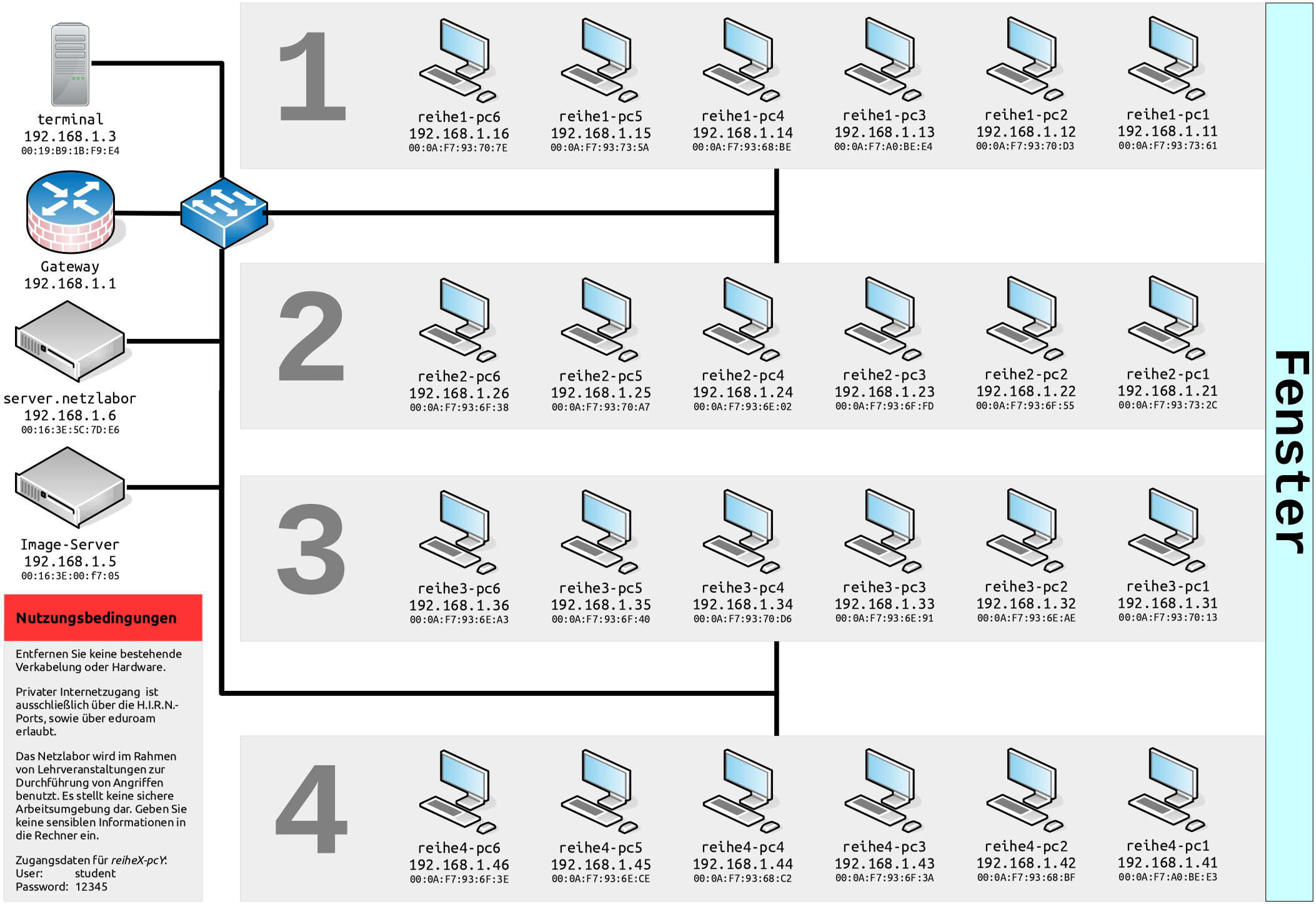
Geben Sie bitte Ihre Einschätzung als Versuchsteam wieder, was Sie von diesen Tools halten. Bitte geben Sie auch Feedback, ob Sie den Praktikumsversuch als interessant empfunden haben und ob dieses Dokument für Sie bei der Versuchsdurchführung hilfreich war. Verbesserungsvorschläge sind willkommen!

Die Versuchsauswertung ist spätestens vor dem nächsten Versuch abzugeben!

7 Kontrollfragen

Sie sollen folgende Fragen beantworten können (bevor der Versuch beginnt):

1. Was ist ein Port, was ist ein Portscanner?
2. Welche Art von Sicherheitslücken kann ein Portscanner finden?
3. Welche Sicherheitslücken kann ein Portscanner nicht finden?
4. Welche Scan-Techniken gibt es?
5. Was bedeuten SYN, ACK und TCP-Sequenznummer?
6. Was bedeutet Stealth-Scan? Erläutern Sie den (möglichen) Ablauf.
7. Was ist ein Sniffer?
8. Was ist der Unterschied zwischen Sniffen und (Port-)Scannen?
9. Welche Daten kann ein Sniffer auswerten?
10. Welche Daten kann ein Sniffer nicht auswerten?
11. Was ist der Unterschied zwischen einer MAC- und einer IP-Adresse?
12. Was hat Sniffen mit IT-Sicherheit zu tun?
13. Kann man nur im Ethernet sniffen?
14. Wozu dienen die Protokolle ARP, IP, TCP, ICMP, UDP, DNS, POP3, HTTP, FTP?
15. Welche bekannten Protokolle übertragen Passwörter im Klartext?



Netzplan Raum ID 2/168, Stand 01.05.2017

Literatur

[1] Nmap. `https://nmap.org/zenmap/`.

[2] Nmap. `http://nmap.org/`.

[3] Philipp Südmeyer. MiniHowTo zu ETHEREAL, 2008.

[4] Philipp Südmeyer. MiniHowTo zu NMAP, 2008.

[5] Wikipedia. NMAP. `http://de.wikipedia.org/wiki/NMAP`.

[6] Wikipedia. Portscanner. `http://de.wikipedia.org/wiki/Portscanner`.

[7] Wikipedia. Sniffer. `http://de.wikipedia.org/wiki/Sniffer`.

[8] Wikipedia. Wireshark. `http://de.wikipedia.org/wiki/Wireshark`.

[9] Wireshark. `http://www.wireshark.org/`.

[10] Wireshark. Wireshark display-filter. `https://wiki.wireshark.org/DisplayFilters`.