Netzwerkforensik Teil 2

Stand ihrer Übungsaufgaben

- Wie ist der Stand mit Ihren Übungsaufgaben?
- Gibt es Probleme?
- Denken Sie an die Möglichkeit diese hochzuladen
- Nächsten Freitag werden die Aufgaben und mögliche Lösungswege besprochen

Ihr aktueller Stand zur Netzwerk-Forensik

- Sie haben bisher die Grundlagen der Netzwerk-Forensik verstanden
- In der Praxis lassen sich Netzwerk-Forensik, Datenträger-Forensik und Live-Forensik nicht immer klar trennen…wir werden das u. a. am Beispiel von TLS sehen.
- Sie kennen Wireshark als Standard-Tool zur Aufzeichnung und Analyse von Netzwerkverkehr → wir schauen uns heute an wie sie dies, zumindest teilweise, mit betriebssystemeigenen Tools können
  - tcpdump für Linux
  - Netsh für Windows

- Die Übungsaufgaben werden nach der Vorlesung um die heutigen Inhalte ergänzt

Grundlagen TLS

- TLS (Transport Layer Security) ist der Nachfolger von SSL (Socket Secure Layer)
- Wird u. a. für die Verschlüsselung von Webseiten genutzt, aus historischen Gründen wird das oft noch als SSL vermarktet
- Heute Standard bei fast alles Webseiten → ein Katalysator waren die Veröffentlichungen von Snowden 2013
- TLS schützt vor Abhören, Manipulation und Fälschung von Daten im Netzwerk
- TLS verwendet asymmetrische Verschlüsselung (Public-Key-Verfahren) zur Authentifizierung und Schlüsselaushandlung
- TLS verwendet symmetrische Verschlüsselung (Session-Key-Verfahren) zur Verschlüsselung der Datenübertragung
- TLS nutzt digitale Zertifikate zur Identifikation von Servern und Clients
- TLS bietet verschiedene Versionen (TLS 1.0, TLS 1.1, TLS 1.2, TLS 1.3)
- TLS wird von vielen Internetprotokollen (z.B. HTTPS, SMTPS, FTPS) genutzt
- TLS erfordert eine sichere Implementierung und Konfiguration → dies ist für Forensiker interessant

Mögliche Angriffe auf TLS

Einige mögliche Angriffe auf TLS sind:

- Man-in-the-Middle-Angriffe (durch Umleitung des Datenverkehrs auf einen falschen Server) → diesen werden Sie in der Übung testen
- Brute-Force-Angriffe (z.B. durch Erraten von schwachen Passwörtern)
- Denial-of-Service-Angriffe (z.B. durch Überlastung des Servers oder Netzwerks)
- Heartbleed-Angriffe (durch Ausnutzung einer Sicherheitslücke in der OpenSSL-Bibliothek)

Wiederholung mitm-Angriffe

- Ziel: Abfangen, Manipulation und/oder Diebstahl von vertraulichen Daten und Informationen, wie beispielsweise Passwörtern, Kreditkartennummern oder sensiblen Geschäftsdaten
- Der Angreifer platziert sich zwischen den Kommunikationspartnern, um den Datenverkehr abzufangen
- Der Angreifer kann dies auf verschiedenen Wegen tun, z.B. durch Zugriff auf das lokale Netzwerk, durch bösartige Software auf dem Endgerät oder durch Einrichtung einer gefälschten WLAN-Verbindung
- Ein mögliches Tool ist ettercap (enthalten in Kali, aber in gängigen Linux-Distributionen leicht nach zu installieren.

Diskussion: Was sind Voraussetzungen für einen mitm-Angriff auf Basis von ARP-Spoofing (unabhängig von Verschlüsselung). Wo kann das in der Praxis klappen, wo eher nicht?

Analyse von TLS-Verkehr mit Wireshark

- Voraussetzung: Sie haben Zugriff auf den Schlüssel → z. B. durch Zugriff auf das Endgerät

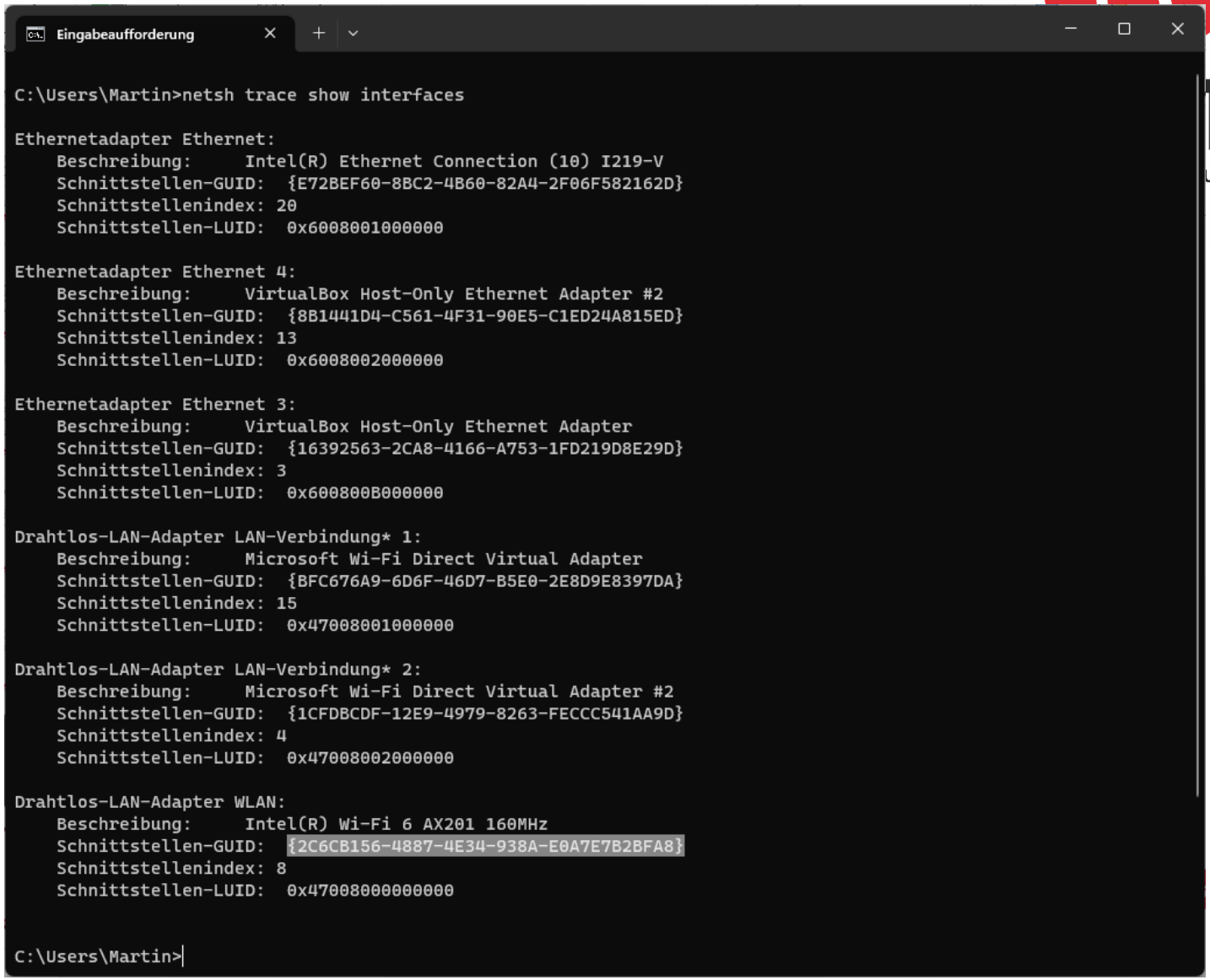
→ Praktisch Möglich wenn Netzwerk-Verkehr in Unternehmen / Behörden mit Wissen der Vorgesetzten überwacht werden soll…theoretisch auch durch Schadsoftware bei Opfer / Verdächtigen

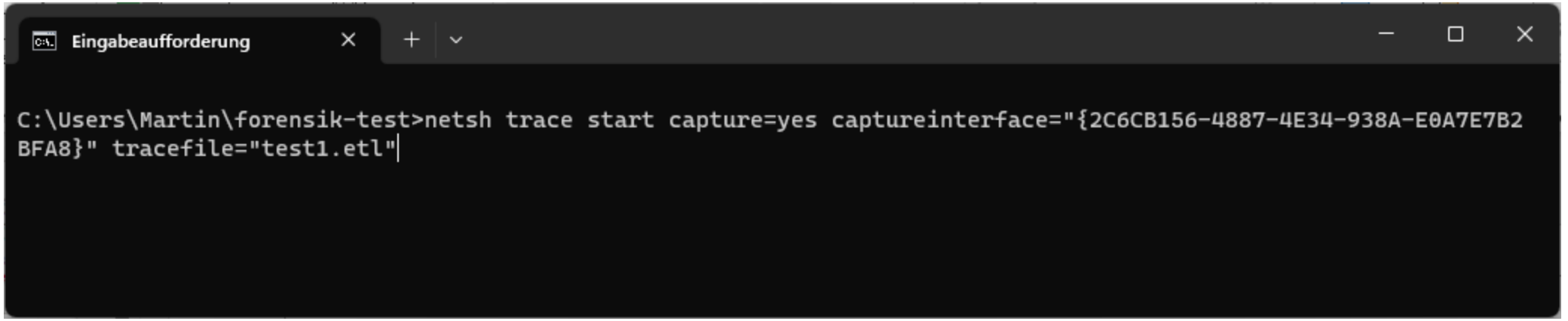
- Zum Export der Schlüssel aus gängigen Browsern und Betriebssystemen, sowie Integration in Wireshark existieren diverse Anleitungen im Internet
- Hierzu wird es eine Übungsaufgabe geben

tcpdump

- TCPDump ist ein Commandline-Tool für Linux
- ermöglicht das Aufzeichnen von Paketen, die auf einer oder mehreren Netzwerk-Interfaces empfangen oder gesendet werden
- TCPDump kann verwendet werden, um den Datenverkehr auf verschiedenen Netzwerkprotokollen, wie TCP, UDP und ICMP zu analysieren
- Das Tool bietet die Möglichkeit, Filter anzuwenden, um spezifische Datenverkehrsflüsse zu erfassen oder um bestimmte Arten von Paketen herauszufiltern, die nicht benötigt werden

Ablaufverfolgung in Windows mit netsh




Eingabeaufforderung
C:\Users\Martin\forensik-test>netsh trace start capture=yes captureinterface="{2C6CB156-4887-4E34-938A-E0A7E7B2
BFA8}" tracefile="test1.etl"

- g

- g

- g

- g

- g

- g

- g

- g

- g

- g

- g

- g

- g

- g

- g

- g

- g

- g

- g

- g

- g

- g

- g

- g

- g

- g

- g

- g

- g

- g

- g

- g

- g

- g

- g

- g