**Projektdokumentation: Netzwerkdatenerfassung mit Raspberry Pi 4B**

**Inhaltsverzeichnis**

1. [Projektübersicht](#projekt%25C3%25BCbersicht)
2. [Funktionsweise](#funktionsweise)
3. [Technische Details](#technische-details)
4. [Anwendungsfälle](#anwendungsf%25C3%25A4lle)
5. [Schritt-für-Schritt-Anleitung](#schritt-f%25C3%25BCr-schritt-anleitung)
6. [Datenanalyse](#datenanalyse)
7. [Fehlerbehebung](#fehlerbehebung)

**Projektübersicht**

Dieses Projekt ermöglicht die Einrichtung eines Raspberry Pi 4B als Netzwerk-Monitoring-System für Forschungszwecke im Bereich IT-Sicherheit. Es wurde speziell für ein akademisches Wochenend-Forschungsprojekt entwickelt, bei dem das Verhalten von Nutzerendgeräten in einer simulierten öffentlichen WLAN-Umgebung (wie einem Café oder einer Halle) untersucht werden soll.

**Hauptziele**

* Erstellung eines stabilen und zuverlässigen WLAN-Access-Points
* Umfassende Erfassung von Netzwerkdaten und Nutzerverhalten
* Ressourceneffiziente Verarbeitung auf dem Raspberry Pi 4B
* Sichere Übertragung der Daten an leistungsstärkere Systeme zur Analyse

**Systemanforderungen**

* Raspberry Pi 4B (mindestens 2GB RAM empfohlen)
* Zwei WLAN-Schnittstellen (eine für den Access Point, eine für die Internetverbindung)
* Micro-SD-Karte (mindestens 32GB empfohlen)
* Optional: Externe SSD über USB 3.0 für verbesserte Speicherleistung

**Funktionsweise**

Das System funktioniert nach folgendem Grundprinzip:

1. **Access Point Bereitstellung**: Der Raspberry Pi wird als WLAN-Access Point konfiguriert, der Internetkonnektivität für Studienteilnehmer bereitstellt.
2. **Datenerfassung**: Während der Nutzung werden verschiedene Datentypen erfasst:
   * Netzwerkverkehr in PCAP-Format (Rohdaten)
   * Verbindungsinformationen von WLAN-Clients
   * Systemstatistiken und Leistungsdaten
3. **Lokale Verarbeitung**: Grundlegende Datenvorverarbeitung direkt auf dem Pi:
   * Extraktion von Metadaten (Geräteidentifikation, Verbindungsmuster)
   * Grundlegende statistische Analyse
   * Komprimierung und Bereinigung von Daten
4. **Datenübertragung**: Regelmäßige Synchronisierung der gesammelten Daten mit einem leistungsstärkeren externen System zur tiefergehenden Analyse.
5. **Selbstüberwachung**: Umfassende Watchdog-Mechanismen stellen sicher, dass die Datenerfassung kontinuierlich und zuverlässig erfolgt.

**Technische Details**

**Systemarchitektur**

+------------------------+ +----------------------+  
| WLAN-Clients | | Internet |  
| (Forschungsteilnehmer) | | |  
+------------+-----------+ +----------+-----------+  
 | |  
 v |  
+------------+--------------------------+ |  
| Raspberry Pi 4B | |  
| +----------------------------------+ | |  
| | Access Point (wlan1) | | |  
| +----------------------------------+ | |  
| | Datenerfassung | | |  
| | - tcpdump/tshark | | |  
| | - Metadatenextraktion | | |  
| | - SQLite Datenbank | | |  
| +----------------------------------+ | |  
| | Datenvorverarbeitung | | |  
| | - Python-Scripts | | |  
| | - Kompression | | |  
| +----------------------------------+ | |  
| | Watchdog & Fehlerbehandlung | | |  
| +----------------------------------+ | |  
| | Datenübertragung | | |  
| +----------------------------------+ | |  
+------------+-------------------+------+ |  
 | | |  
 v v |  
+------------+-------+ +------+---------+  
| Lokale Speicherung | | Internet-Interface (wlan0)  
| - PCAP-Dateien | +-----------------------+  
| - JSON-Statistiken |  
| - SQLite-Datenbank |  
+--------------------+

**Hauptkomponenten**

1. **Access Point**: Konfiguriert mit hostapd und dnsmasq
   * DHCP-Server zur IP-Adressvergabe
   * DNS-Server mit Logging-Funktionalität
   * NAT und IP-Forwarding für Internetkonnektivität
2. **Datenerfassung**: Kombiniert verschiedene Erfassungsmethoden
   * tcpdump für PCAP-Erstellung mit automatischer Rotation
   * Direktes Monitoring des WLAN-Interfaces (iw, iwconfig)
   * Systemstatistiken (dstat, vnstat, Python psutil)
3. **Datenbank**: SQLite für strukturierte Datenspeicherung
   * Tabellen für Geräte, Verbindungen und Systemstatus
   * Indexierung für optimierte Abfrageleistung
   * Backup-Mechanismen für Datensicherheit
4. **Datenübertragung**: Sichere Synchronisierung mit externen Systemen
   * SSH/SCP-basierte Übertragung
   * Inkrementelle Synchronisierung (rsync)
   * Überwachung und Logging des Übertragungsprozesses
5. **Zuverlässigkeitsmechanismen**: Mehrschichtige Überwachung
   * Hardware-Watchdog (falls verfügbar)
   * Software-Watchdog mit Selbstheilungsfunktionen
   * Ressourcenüberwachung und -management

**Anwendungsfälle**

**IT-Sicherheitsforschung im Bildungskontext**

Dieses System eignet sich hervorragend für folgende Forschungsbereiche:

1. **Verhaltensmusteranalyse**: Untersuchung typischer Nutzungsmuster in öffentlichen WLANs
   * Erkennung von Anwendungsprotokollen und -präferenzen
   * Analyse von Verbindungshäufigkeiten und -dauern
   * Identifikation typischer Verkehrsmuster verschiedener Gerätetypen
2. **Sicherheitsbewusstsein und -praktiken**: Erfassung von Sicherheitsverhalten
   * Erkennung unverschlüsselter Verbindungen
   * Analyse von DNS-Anfragen und Webseitenbesuchen
   * Identifikation potenziell unsicherer Anwendungen
3. **Geräteerkennung und -profilerstellung**: Verständnis der genutzten Gerätelandschaft
   * Erkennung von Gerätetypen und Betriebssystemen
   * Analyse von Herstellerverteilungen (über MAC-Adressen)
   * Identifikation von IoT- und anderen nicht-standardmäßigen Geräten
4. **Netzwerkleistungs- und Nutzungsanalyse**: Bewertung der Ressourcennutzung
   * Bandbreitenverbrauch nach Gerät und Anwendung
   * Spitzenlastzeiten und -muster
   * Optimierungspotenziale für Netzwerkdesign

**Ethische Überlegungen**

Das Projekt wurde mit folgenden ethischen Gesichtspunkten konzipiert:

* Datenschutz durch Pseudonymisierungsfunktionen
* Beschränkung auf Metadaten ohne Erfassung von Kommunikationsinhalten
* Klare Informationen für Teilnehmer über die Datenerfassung
* Möglichkeit zur Anonymisierung vor der detaillierten Analyse

**Schritt-für-Schritt-Anleitung**

**1. Vorbereitung des Raspberry Pi**

# 1. Aktuelles Raspberry Pi OS installieren (64-bit empfohlen)  
# 2. Update des Systems durchführen  
sudo apt update && sudo apt upgrade -y  
  
# 3. Notwendige Pakete vorab installieren  
sudo apt install -y git jq python3-pip  
  
# 4. Repository klonen oder Skript herunterladen  
git clone https://github.com/benutzer/netzwerkerfassung.git  
cd netzwerkerfassung  
  
# 5. Skript ausführbar machen  
chmod +x setup.sh

**2. Installation ausführen**

# Als Root ausführen (sudo)  
sudo ./setup.sh  
  
# Alternativ direkt aus diesem Dokument:  
sudo bash -c "$(curl -fsSL https://raw.githubusercontent.com/benutzer/netzwerkerfassung/main/setup.sh)"

**3. Konfiguration anpassen (optional)**

Die Standard-Konfiguration wird während der Installation erstellt. Für benutzerdefinierte Einstellungen:

# Konfiguration bearbeiten  
sudo nano /etc/netcapture/config.json  
  
# Wichtige Parameter:  
# - network.ap\_interface: WLAN-Schnittstelle für Access Point  
# - network.internet\_interface: WLAN-Schnittstelle für Internetverbindung  
# - access\_point.ssid: Name des WLANs  
# - access\_point.passphrase: WLAN-Passwort  
# - data\_collection.capture\_interval: Erfassungsintervall  
# - remote\_sync: Einstellungen für Datenübertragung

**4. System starten**

# System neu starten, um alle Änderungen zu aktivieren  
sudo reboot  
  
# Nach dem Neustart: Dienststatus überprüfen  
sudo systemctl status netcapture  
sudo systemctl status hostapd  
sudo systemctl status dnsmasq

**5. Datenerfassung prüfen**

# Prüfen, ob die Datenerfassung läuft  
sudo ls -la /opt/netcapture/pcaps/  
  
# Log-Dateien prüfen  
sudo tail -f /opt/netcapture/logs/system.log  
  
# Verbundene Clients anzeigen  
sudo iw dev wlan1 station dump

**6. Datenübertragung einrichten (falls gewünscht)**

# 1. Öffentlichen SSH-Schlüssel vom Raspberry Pi anzeigen  
sudo cat /root/.ssh/transfer\_key.pub  
  
# 2. Diesen Schlüssel auf dem Zielserver in authorized\_keys eintragen  
# 3. Datenübertragung testen  
sudo /opt/netcapture/sync\_data.sh

**Datenanalyse**

**Erfasste Datentypen**

Das System sammelt folgende Daten:

1. **PCAP-Dateien**: Rohes Netzwerkverkehrsformat
   * Enthält vollständige Paketdaten
   * Analysierbar mit Wireshark, tshark, Zeek, Scapy
   * Rotiert automatisch zur Vermeidung übermäßiger Dateigröße
2. **Client-Metadaten**: In JSON und SQLite
   * MAC-Adressen und Gerätehersteller
   * Signal-Stärke und Verbindungsqualität
   * Datenübertragungsmengen und -raten
3. **Systemstatistiken**: Leistungsdaten
   * CPU-Auslastung und Temperatur
   * Speicherverwendung und -verfügbarkeit
   * Netzwerkdurchsatz und aktive Verbindungen

**Analyse-Workflow**

1. **Datenübertragung**: Zuerst werden die gesammelten Daten auf ein leistungsstärkeres System übertragen

# Auf dem Analyse-System  
rsync -avz -e "ssh -p 2222" researcher@raspberry-pi:/opt/netcapture/ ./analysis\_data/

1. **Basisanalyse**: Erste Untersuchung der Daten

# PCAP-Analyse mit tshark  
tshark -r capture\_20250314\_120000.pcap -q -z io,stat,1 -z endpoints -z conv,ip  
  
# SQLite-Datenbank untersuchen  
sqlite3 capture.db "SELECT mac\_address, device\_type, first\_seen, last\_seen, connection\_count FROM devices ORDER BY connection\_count DESC LIMIT 10;"

1. **Visualisierung**: Für fortgeschrittene Analyse
   * Import in Grafana oder ELK Stack
   * Analyse mit Python (pandas, matplotlib)
   * Netzwerkvisualisierung mit NetworkX
2. **Berichtserstellung**: Zusammenfassung der Ergebnisse
   * Gerätediversität und -verteilung
   * Verbindungsmuster und -präferenzen
   * Sicherheitsrelevante Erkenntnisse

**Fehlerbehebung**

**Typische Probleme und Lösungen**

1. **Access Point startet nicht**

# Log prüfen  
sudo journalctl -u hostapd -n 50  
  
# Konfiguration prüfen  
sudo cat /etc/hostapd/hostapd.conf  
  
# Schnittstelle prüfen  
sudo iw dev

1. **Keine Internetverbindung für Clients**

# NAT-Regeln prüfen  
sudo iptables -t nat -L -v  
  
# IP-Forwarding prüfen  
cat /proc/sys/net/ipv4/ip\_forward  
  
# Internet-Schnittstelle prüfen  
ping -c 3 8.8.8.8

1. **Datenerfassung stoppt unerwartet**

# Logs prüfen  
sudo tail -f /opt/netcapture/logs/system.log  
sudo tail -f /opt/netcapture/logs/watchdog.log  
  
# Speicherplatz prüfen  
df -h  
  
# Manuell neu starten  
sudo /opt/netcapture/stop\_capture.sh  
sudo /opt/netcapture/start\_capture.sh

1. **Datenübertragung funktioniert nicht**

# Logs prüfen  
sudo cat /opt/netcapture/logs/transfer.log  
  
# Verbindung testen  
sudo ssh -i /root/.ssh/transfer\_key researcher@192.168.1.100  
  
# Netzwerkverbindung prüfen  
ping -c 3 192.168.1.100

**Kurzanleitung: Netzwerkdatenerfassung mit Raspberry Pi 4B**

Diese Kurzanleitung führt Sie durch die wesentlichen Schritte zur Einrichtung und Nutzung des Netzwerkdatenerfassungssystems auf einem Raspberry Pi 4B.

**Voraussetzungen**

* Raspberry Pi 4B (mindestens 2GB RAM)
* Zwei WLAN-Schnittstellen (z.B. integriertes WLAN + USB-WLAN-Adapter)
* Micro-SD-Karte (mindestens 32GB)
* Raspberry Pi OS (64-bit empfohlen)

**1. System vorbereiten**

# Raspberry Pi OS aktualisieren  
sudo apt update && sudo apt upgrade -y  
  
# Repository klonen (falls verfügbar) oder Skript herunterladen  
wget https://raw.githubusercontent.com/benutzer/netzwerkerfassung/main/setup.sh  
chmod +x setup.sh

**2. Installation ausführen**

# Skript als Root ausführen  
sudo ./setup.sh  
  
# Während der Installation:  
# - WLAN-Schnittstellen für AP und Internet angeben  
# - Weitere Anweisungen befolgen

**3. System neu starten**

sudo reboot

**4. Funktionsprüfung**

# Dienststatus überprüfen  
sudo systemctl status netcapture  
sudo systemctl status hostapd  
sudo systemctl status dnsmasq  
  
# Prüfen, ob Access Point sichtbar ist  
# (Mit Ihrem Smartphone oder Laptop nach dem WLAN suchen)  
  
# Access Point Zugangsdaten anzeigen  
sudo cat /opt/netcapture/access\_credentials.txt

**5. Datenübertragung einrichten (optional)**

# SSH-Schlüssel anzeigen  
sudo cat /root/.ssh/transfer\_key.pub  
  
# Diesen Schlüssel auf dem Zielserver in ~/.ssh/authorized\_keys eintragen  
# In der Konfiguration den Host, Benutzer und das Verzeichnis anpassen:  
sudo nano /etc/netcapture/config.json

**6. Datenerfassung steuern**

# Datenerfassung manuell starten (falls nicht automatisch)  
sudo /opt/netcapture/start\_capture.sh  
  
# Datenerfassung stoppen  
sudo /opt/netcapture/stop\_capture.sh  
  
# Erfasste Daten anzeigen  
sudo ls -la /opt/netcapture/pcaps/  
sudo ls -la /opt/netcapture/stats/

**7. Logs und Überwachung**

# Hauptlog  
sudo tail -f /opt/netcapture/logs/system.log  
  
# Verbundene Clients  
sudo /opt/netcapture/collect\_clients.sh  
  
# Systemstatistiken  
sudo /opt/netcapture/collect\_stats.sh

**8. Daten manuell synchronisieren**

# Manuelle Synchronisierung starten  
sudo /opt/netcapture/sync\_data.sh

**9. System bereinigen oder zurücksetzen**

# Datenerfassung stoppen  
sudo systemctl stop netcapture  
sudo systemctl disable netcapture  
  
# Für komplette Deinstallation:  
sudo systemctl stop hostapd dnsmasq  
sudo systemctl disable hostapd dnsmasq  
sudo rm -rf /opt/netcapture /etc/netcapture

Bei Fragen oder Problemen konsultieren Sie die vollständige Dokumentation oder die Log-Dateien in /opt/netcapture/logs/.