ABSCHLUSSDOKUMENTATION

SICHERHEITSANALYSE DER WINDOWS-DIENSTE ÜBER PORTS 5357/5358 UND DEREN ABSICHERUNG

NAME: RONALD LINDE
AUSBILDUNG: IT-FACHKRAFT FÜR SYSTEMINTEGRATION UND IT-SICHERHEIT
ABGABETERMIN: APRIL 2025

Inhaltsverzeichnis

1. Einleitung

1.1 Projektbeschreibung und Zielsetzung

Ziel dieses Projekts ist die Durchführung einer tiefgreifenden Sicherheitsanalyse der Windows-Dienste, die über die Ports 5357/TCP und 5358/UDP kommunizieren. Diese Ports werden für das Device Profile for Web Services (DPWS) genutzt, er ist Bestandteil des Dienstes der Netzwerkerkennung von Web-Service-Funktionen auf ressourcenbeschränkten Geräten, sie gehören zur Microsoft Web Services Architektur (WSA) an, die zur automatischen Geräteerkennung dient. Diese Geräte kommunizieren mittels XML/SOAP-Nachrichten die durch den WSD (Web Service Discovery) erkannt und beantwortet werden.

Der WSD, wenn aktiv, sendet eine „Hallo-Nachricht“ als Multicast ins Netzwerk. Ist nun ein Gerät im Netzwerk würde es diese Nachricht empfangen. Daraufhin sendet der WSD eine Probe Nachricht, ähnlich wie: „Wer ist da draußen, der bestimmte Kriterien erfüllt?“ Nun antwortet jedes passende Gerät mit einer ProbeMatch-Nachricht indem es seine Metadaten wie Gerätetyp, Adresse (XAddrs) und Scopes sendet. Dies dient der ersten Gerätesuche ohne konkrete Adresse.

Dann gibt es noch die Probe/Resolve-Nachricht wo der Client bereits Informationen besitzt wie Beispielsweise EPR (Endpoint Reference / UUID) und nun weitere Details zu genau diesem Gerät beziehen will um eine gezielte Verbindung her zu stellen. Ein Beispiel einer Probe/Resolve-Nachricht im XML/SOAP-Format wäre: „Gib mir die XAddrs zu (uuid:printer-hp4200-xy).“ Das Gerät, in diesem Fall der Drucker, antwortet mit einer Resolve-Nachricht: „Hier bin ich, das ist mein Web-Service-Endpunkt.“ Wurde diese Nachricht gesendet, ist eine Verbindung hergestellt und weitere Routinen können ausgeführt werden, wie z.B. Setup oder Treiberinstallation. Das Ziel ist eine Härtung der Systemkonfiguration mit verstärkten Sicherheitsmaßnahmen, um diese potenziellen Schwachstellen in produktiven Systemen zu minimieren und so das Angriffsrisiko zu verringern und die Netzwerkstabilität zu erhöhen.

1.2 Relevanz der Sicherheitsanalyse für WSD-Dienste

Da diese Dienste anfällig für Angriffe wie Man-in-the-Middle (MITM), Spoofing und Denial-of-Services (DoS) sind, wird eine umfassende Sicherheitsanalyse durchgeführt.

Untersucht wird ob durch manipulierte XML-Nachrichten, das Verhalten der WSD-Kommunikation im Netzwerk, welche Daten offengelegt werden und wie sich Dienste manipulieren lassen.

2. Projektumfeld

2.1 Beschreibung der Testumgebung

Die Tests werden in Microsoft Hyper-V in einer isolierten Laborumgebung ausgeführt. Die Infrastruktur besteht aus 3 virtuellen Maschinen einem internen switch mit statisch zugewiesener IP von den Betriebssystemen Microsoft Windows Server und Microsoft Windows 10 als Zielsysteme für WSD-Dienste. Die Kali Linux Distribution dient als Angreifer- und Analysesystem. Das Netzwerk ist statisch konfiguriert um die Erreichbarkeit zu vereinfachen. Alle sind im selben virtuellen Netzwerk verbunden, sodass die Kali VM die beiden Windows VM erreichen kann. Anschließend werden verschiedene Angriffsarten auf Port 5357 (WSD) der Windows VM demonstriert und anschließend geeignete Sicherheitsmaßnahmen umgesetzt.

2.2 Genutzte Tools

Zur Umsetzung der Dokumentation wurden folgende Tools benutzt:

Nmap: Port- und Dienstscanner zum Auffinden offener Ports (insbesondere 5357/5358) und zum Identifizieren von Diensten.

Wireshark: Netzwerkprotokoll-Analyzer zum Mitschneiden und Analysieren des Netzwerkverkehrs der WSD-Dienste.

Metasploit: Penetrationstest-Framework zur Durchführung gezielter Exploits und Tests auf Schwachstellen in WSD-Diensten.

Impacket: Vielseitiges Tool mit doppeltem Verwendungszweck, das Python basierte Skripte verwendet, um legitime Windows Dienste und Protokolle aus zu nutzen.

SMB (Server Message Block): Ist ein Protokoll für die Freigabe von Dateien, Druckern und anderen Ressourcen

Powershell: mächtiges Werkzeug zum setzten der Systemeinstellungen und zum Ausführen von Skripten

3. IST-Analyse

3.1 Aktuelle System- und Netzwerkarchitektur

Der Switch (Netzwerkinterface/Netzwerkkarte) steht auf Intern damit eine sichere Isolierung gewährleistet ist. Die Kali VM hat die IP 192.168.2.200, die Windows Client VM hat die IP 192.168.2.20 und die Windows Server VM hat die IP 192.168.2.2. Die Windows VM haben anfänglich Netzwerkerkennung aktiviert, wodurch WSD auf dem privaten Netzwerkprofil lauscht. Die WSD-Dienste dienen der Geräte und Anwendungserkennung in einem Netzwerk.

3.2 Identifikation der genutzten WSD-Dienste

Port 5357 wird vom Web Services for Devices (WSD)-Dienst genutzt (HTTP auf TCP/5357, HTTPS auf TCP/5358, Discovery über UDP/3702). Dieser Dienst, wie schon erwähnt, dient dem auffinden aller Ressourcenbeschränkten Geräten. Dazu gehören alle Geräte die in den Bereich IoT (Internet of Thinks) fallen. Was unter anderem Drucker, Handys, Kameras, Kaffeemaschinen, Herdplatten, Backöfen, Geschirrspüler, Leuchtmittel usw. sein kann.

Nmap meldet, das Port 5357 offen ist. Da WSD aktiv ist, erscheint Port 5357 als open und wird als HTTP-Dienst erkannt. Ein Nmap Scan für die Windows VM zeigt dies in Abbildung 1

Port	State	Service	Version
5357/tcp	open	http	Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)

Dies deutet darauf hin, dass der Windows-WSD-Dienst auf Port 5357 horcht. Zusätzlich könnte Nmap andere offene Ports (wie 139/445 für SMB, 3389 für RDP etc.) anzeigen, wie in Abbildung 2 gezeigt, für diese Testzwecke ist jedoch Port 5357 besonders relevant.

4. Schwachstellenanalyse

4.1 Durchführung von Netzwerk- und Dienst-Scans (Nmap)

Durch den Befehl: sudo nmap -sS -p 5357,5358,3702 -sV -Pn <IP-Windows-VM>

Erläuterung: -sS führt einen TCP-SYN-Scan durch, -p 5357,5358,3702 beschränkt auf die relevanten WSD-Ports (5357/TCP, 5358/TCP, 3702/UDP), -sV versucht die Service Version zu ermitteln, und -Pn überspringt den Ping Vorcheck (nützlich, falls ICMP geblockt ist). Alternativ kann ein vollständiger Scan (-p-) durchgeführt werden, um alle offenen Ports zu finden. Ein Beispiel dafür ist Abbildung 3

4.2 Analyse des Netzwerkverkehrs (Wireshark)

Um den Portscan weiter zu untersuchen wird Wireshark hinzugezogen. Mittels Anzeige Filter (z.B. ip.addr==192.168.2.20 oder tcp.port==5357) wird der Datenverkehr übersichtlicher. Der Netzwerkverkehr wurde über die Netzwerkschnittstelle der Kali-VM mitgeschnitten, während gezielte Scans mit Nmap und erste Verbindungsversuche vom Windows-Client ausgingen. Zur gezielten Analyse wurden Port-Filter für TCP 5357, TCP 5358 und UDP 3702 gesetzt. So konnten alle WSD-relevanten Nachrichten erfasst werden. Im Beobachteten Datenverkehr zeigte sich ein typisches Kommunikationsmuster von WS-Discovery-Protokollen. Nach dem Einschalten des Windows-Clients sendete dieser regelmäßig Hallo-Nachrichten an die Multicast-Adresse 239.255.255.250 (UDP/3702), woraufhin die Antwortpakete mit Probe-Match-Informationen vom Server eintrafen. Bei aktiver Netzwerkerkennung erfolgten außerdem HTTP-Anfragen an Port 5357 mit SOAP-basierten XML-Nachrichten. Zu den Auffälligkeiten während der Tests ist zu sagen das beobachtet werden konnte, dass viele der SOAP-Nachrichten unverschlüsselt im Klartext übertragen wurden. In diesen Nachrichten waren teilweise spezifische Geräteinformationen wie Gerätename, Hersteller, Modellnummer, sowie interne IP-Adressen enthalten. Dies stellt ein Risiko für Informationsabfluss dar. Bei der Analyse der Protokollinhalte sah man das die entschlüsselten SOAP-Nachrichten strukturierte XML-Blöcke enthielten, darunter u.a. <d:Types>, <d:Scopes>, <d:XAddrs> und <d:MetadataVersion> siehe Abbildung 4. Diese Felder bieten einem Angreifer wertvolle Informationen über Netzwerkgeräte und deren

Dienste. Die Sicherheitsrelevanz ist besonders kritisch da die unverschlüsselte Übertragung von Metadaten über HTTP (Port 5357), leicht mitgeschnitten und analysiert werden können. Zudem zeigte sich, dass die Dienste ohne Authentifizierung antworteten – was eine Angriffsmöglichkeit für Spoofing-Angriffe oder DoS durch gefälschte Hello-Nachrichten bietet.

4.3 Identifikation potenzieller Sicherheitslücken

Für den WSD-Dienst sind bekannte Schwachstellen dokumentiert, unter anderem die CVE-2009-2512, die eine Remotecodeausführung durch manipulierte Netzwerkpakete ermöglicht. Microsoft hat dazu das Sicherheitsupdate MS09-063 veröffentlicht. Auch moderne Systeme sind potenziell verwundbar, wenn Sicherheitsupdates fehlen oder Standardkonfigurationen verwendet werden. Die Analyse verweist auf CVE-2009-2512, die WSD-Anfragen und somit ein Puffer Überlauf auslösen könnten. Weitere Schwachstellen sind die unverschlüsselte Kommunikation über HTTP (Port 5357) und die Verwendung von Multicast-Discovery ohne Authentifizierung. Die Testumgebung zeigte, dass die WSD-Dienste standardmäßig ohne Zugangsschutz aktiviert sind. Geräte identifizieren sich automatisch, ohne dass der Nutzer eine Freigabe erteilt. Die SOAP-Kommunikation erfolgt unverschlüsselt was bei sensitiven Geräten (z.B. Kameras oder Multifunktionsdrucker) zu Datenabfluss führen kann. Offene Ports 5357 / 5358 und 3702 bieten eine potenzielle Angriffsfläche und erlaubt Angreifern, sich selbst als Geräte auszugeben. Die SOAP-Nachrichten enthalten verwendbare Daten (z.B. bei XAddrs sah man interne/externe Verzeichnispfade), da diese in Klarschrift aufzufinden waren. Die Risiken schätze ich als mittel bis hoch ein, je nachdem in welchem Netzwerk man sie sich betrachtet. In internen Netzwerken mit vielen IoT-Geräten könnten unautorisierte Geräte sich einschleichen ohne dass man dies sofort mitbekommt.

5. Penetrationstests

5.1 Angriffsstrategie und Methoden

Bei dem Ziel der Tests lag der Fokus auf der gezielten Ausnutzung des WSD-Dienstes auf dem Windows-Server und dem Windows-Client. Die Vorgehensweise war zunächst Standard-Schwachstellen über das Metasploit-Framework zu beurteilen. Zusätzlich kamen selbst geschriebene PowerShell- und Python-Skripte zum Einsatz, um typische Spoofing- und

Replay-Angriffe nachzustellen. Bei der Priorisierung ist zu sagen das zuerst die passive Analyse durch Wireshark durchgeführt wurde, gefolgt von aktiven Scans mittels Nmap zum Schluss dann Tests mit unautorisierten Geräteidentifikation über SOAP-Nachrichten sowie Versuche zur Ausführung bösartiger INF-Dateien zur Simulation eines Fake-Druckers. Es ist an dieser stelle zu erwähnen das die Sicherheitsvorkehrungen der Tests so konzipiert waren als das sie isoliert durchgeführt wurden. Die SMB-Freigabe wurde ausschließlich für lokale Installationen genutzt.

5.2 Durchführung der Tests (Metasploit)

Bei der Konfiguration für die INF-Datei wurde ein Payload in EXE-Form über eine SMB-Freigabe ausgeliefert. Mittels WSD-Simulation antwortete ein Fake-Drucker auf Hallo-Nachrichten. Der Windows-Client erkannte das Gerät, installierte den Treiber aber nur nach manueller Auswahl. Die automatische Namensanzeige scheiterte an der Microsoft Signaturprüfung, siehe Abbildung 5. Die INF-Datei, lokal installiert löste erfolgreich eine Reverse Shell Sitzung aus. Das Ergebnis verlief nicht nach den Vorstellungen jedoch war die Simulation eines Backdoors über INF erfolgreich. SOAP-Nachrichten konnten erfolgreich manipuliert werden.

5.3 Ergebnisse und Bewertung der Sicherheitsrisiken

Als Zusammenfassung kann man sagen das die Kommunikation unverschlüsselt und ohne Authentifizierung bei Geräteidentifikation verlief. Die manipulierte INF-Datei ließ sich, wenn auch nur lokal, über SMB auf das Zielsystem verschieben und über die Systemsteuerung -> Geräte und Drucker installieren. Über Multicast Discovery ist es leicht manipulierbare Nachrichten zu senden. Die Impact-Analyse ergab das eine komplette Kompromittierung eines Windows-Clients über eine Fake-Druckerinstallation möglich ist. Weiterhin sind DoS durch Geräte-Duplikation und Informationsabfluss via SOAP/XML Nachrichten realistisch. Das Risikolevel bewerte ich als Hoch, insbesondere bei produktiven Systemen mit IoT-Komponenten. Die Risiken steigen bei fehlenden Updates und offener Multicast-Kommunikation. Die Erkenntnis daraus ist das WSD ein offenes Protokoll mit zahlreichen nicht abgesicherten Prozessen ist. Ohne Authentifizierungsmechanismen oder Verschlüsselung ist es nicht für sicherheitskritische Netzwerke, wie es bei Firmen mit produktiven Systemen der Fall ist, bedenklich bis nicht geeignet.

6. Absicherungsmaßnahmen

6.1 Implementierung von Sicherheitsmaßnahmen

Die Firewall-Regeln wurden so konfiguriert das sie die Ports 5357, 5358 und 3702 in privaten sowie öffentlichen Netzwerkprofilen blockieren. Durch die Deaktivierung der WSD-Dienste FDResPub und fdPHost, siehe Abbildung 6, auf Windows Systemen und die Deaktivierung der Netzwerkkennung über die Gruppenrichtlinie bei Domaincontrollern. Zu den punkten der Systemhärtung gehören, dass man nicht genutzte Geräte entfernt. Weiterhin sollte der Windows Defender und das Antivirenprogramm aktiviert sein. Ebenso wie das installieren aktueller Sicherheitspatches und Windows Updates. Empfehlenswert ist auch die Netzwerksegmentierung, WSD fähige Geräte sollten in ein separates VLAN verschoben werden. Weitere Beschränkungen der Kommunikation, mit anderen Netzen, werden über Netzwerk-ACLs geregelt.

6.2 Auswirkungen der Maßnahmen auf die Sicherheit

Die zuvor getesteten Angriffe konnten nach Umsetzung der Maßnahmen nicht mehr erfolgreich durchgeführt werden. Ein erneuter Nmap-Scan zeigte die Ports als geschlossen und auf SOAP-Nachrichten wurde nicht mehr geantwortet. Selbst Wireshark zeigte keine WSD-Kommunikation mehr an. Jedoch bleibt der Nebeneffekt das die Geräteerkennung im Netzwerk eingeschränkt ist und alles was sich vorher selbst gefunden und installiert hatte nun nur noch manuell eingebunden werden kann. Was zusätzlichen Aufwand und tieferes Fachwissen voraussetzt.

7. Ergebnisse und Fazit

7.1 Zusammenfassung der Testergebnisse

Die Analyse bestätigte, dass WSD-Dienste ohne Sicherheitsvorkehrungen ein hohes Risiko darstellen. Fake Geräte wurden automatisch erkannt so dass eine manipulierte INF-Datei als

Angriff dienen kann. Eine Verschlüsselung oder Authentifizierung fehlt komplett, so dass keinerlei Absicherung vorhanden ist.

7.2 Empfehlungen für zukünftige Sicherheitsmaßnahmen

Die Ports 5357 / 5358 sollten bestenfalls in isolierten oder segmentierten Netzwerken verwenden werden. Allerdings lässt sich das kaum vermeiden da die Ports schon Standartmäßig geöffnet sind. Die Unwissenheit das diese Ports geöffnet sind, aber auch mangelndes Fachwissen wie man mit dieser „Lücke“ umgeht sollte in Fachbereichs Schulungen für z.B. von System- und Netzwerkadministratoren als „need-to-know“ eingeführt werden. Hierbei ist wichtig darauf hin zu weisen was zu beachten ist, wenn man die Ports 5357 / 5358 geöffnet lässt. Weiterhin ist der Einsatz von Monitoring-Systemen zur Überwachung der Webdienste für SOAP-, GET/POST-, TCP- und ICPM-Protokolle zu empfehlen.

7.3 Fazit zur Sicherheitslage der WSD-Dienste

Die Ziele der Analyse wurden erreicht. WSD-Dienste erwiesen sich als gefährlich, wenn sie nicht gezielt abgesichert werden. Die Dokumentation zeigt, wie durch gezielte Tests reale Angriffe simuliert und mit technischen Maßnahmen verhindert werden können. Die Thematik sollte in zukünftigen Netzwerkdesigns beachtet und WSD bei Nichtgebrauch konsequent deaktiviert werden.

I Abkürzungsverzeichnis

ACL	Access Control List
API	Application Programming Interface
DPWS	Device Profile for Web Services
fdPHost	Function Discovery Provider Host
FDResPub	Funktionssuche-Ressourcenveröffentlichung
HTTP	Hypertext Transfer Protocol
ICMP	Internet Control Message Protocol
IoT	Internet of Things (bezeichnet ein Netzwerk vernetzter Geräte und Objekte)
IP	Internet Protocol (verbindungsloses Protokoll zur Adressierung von Geräten)
INF	Setupinformationsdatei
MITM	Man-in-the-Middle
SOAP	Simple Object Access Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UUID	Universally Unique Identifier
VLAN	Virtual Local Area Network
VM	Virtuelle Maschine
WSA	Web Service Architecture
WSD	Web Service for Devices
XML	Extensible Markup Language

II Quellenverzeichnis

Bei der gesamten Dokumentation, insbesondere bei dem Laboraufbau und dem testen der Ports mittels SOAP/XML Nachrichten wurde ChatGPT herangezogen. Die Abkürzungen wurden mit Hilfe von Google in seine ausgeschriebene Form gebracht.

III Bildanhang

Abbildung 1:

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sS -p 5357,5358,3702 -sV -Pn 192.168.2.20
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-11 13:24 CEST
Nmap scan report for 192.168.2.20
Host is up (0.00035s latency).

PORT     STATE    SERVICE      VERSION
3702/tcp filtered ws-discovery
5357/tcp open     http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
5358/tcp filtered wsdapi-s
MAC Address: 00:15:5D:98:D7:0B (Microsoft)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

Abbildung 2:

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sS -p 139,445,3389 -sV -Pn 192.168.2.20
[sudo] Passwort für kali:
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-13 12:46 CEST
Nmap scan report for 192.168.2.20
Host is up (0.00023s latency).

PORT     STATE    SERVICE       VERSION
139/tcp  open     netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open     microsoft-ds?
3389/tcp filtered ms-wbt-server
MAC Address: 00:15:5D:98:D7:0B (Microsoft)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

Abbildung 3:

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sS -p- -sV -Pn 192.168.2.20
[sudo] Passwort für kali:
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-11 13:14 CEST
Nmap scan report for 192.168.2.20
Host is up (0.00057s latency).
Not shown: 65529 filtered tcp ports (no-response)
PORT      STATE SERVICE       VERSION
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds?
5040/tcp  open  unknown
5357/tcp  open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
49668/tcp open  msrpc         Microsoft Windows RPC
MAC Address: 00:15:5D:98:D7:0B (Microsoft)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

Abbildung 4:

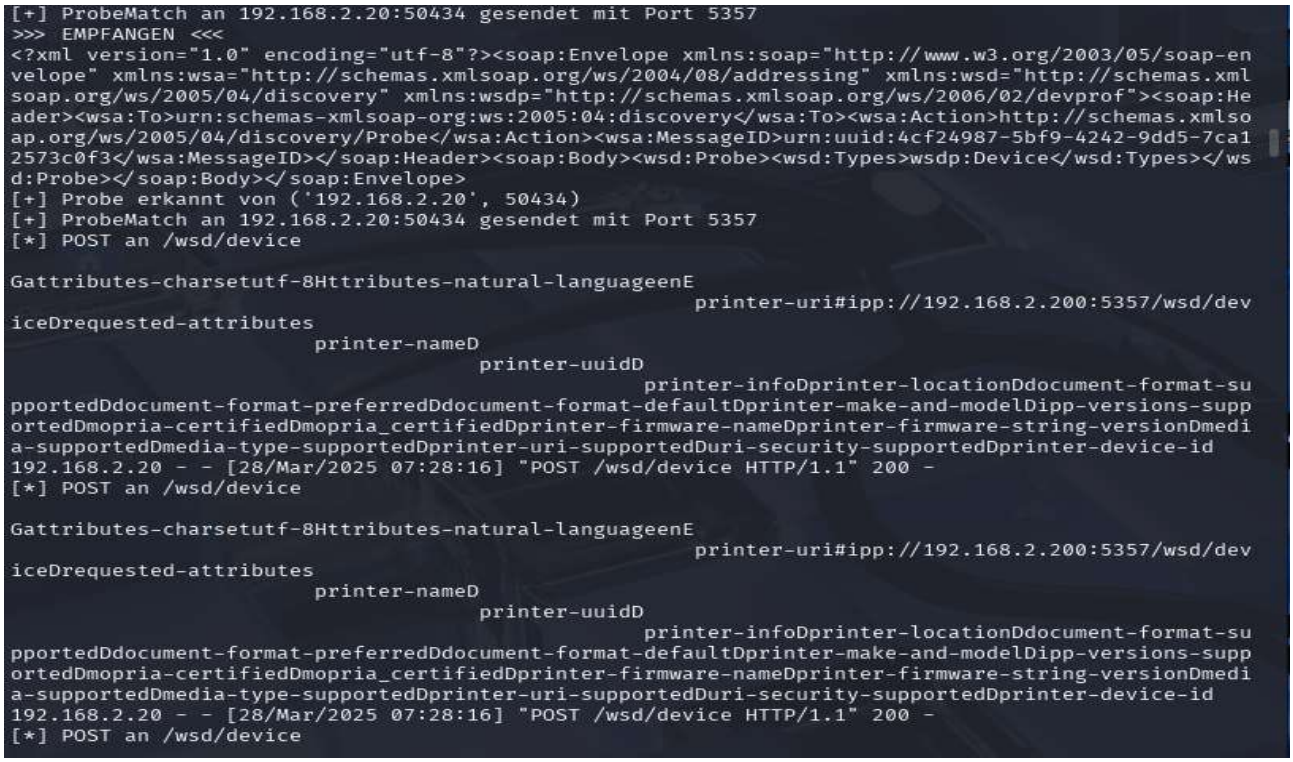

Abbildung 5:

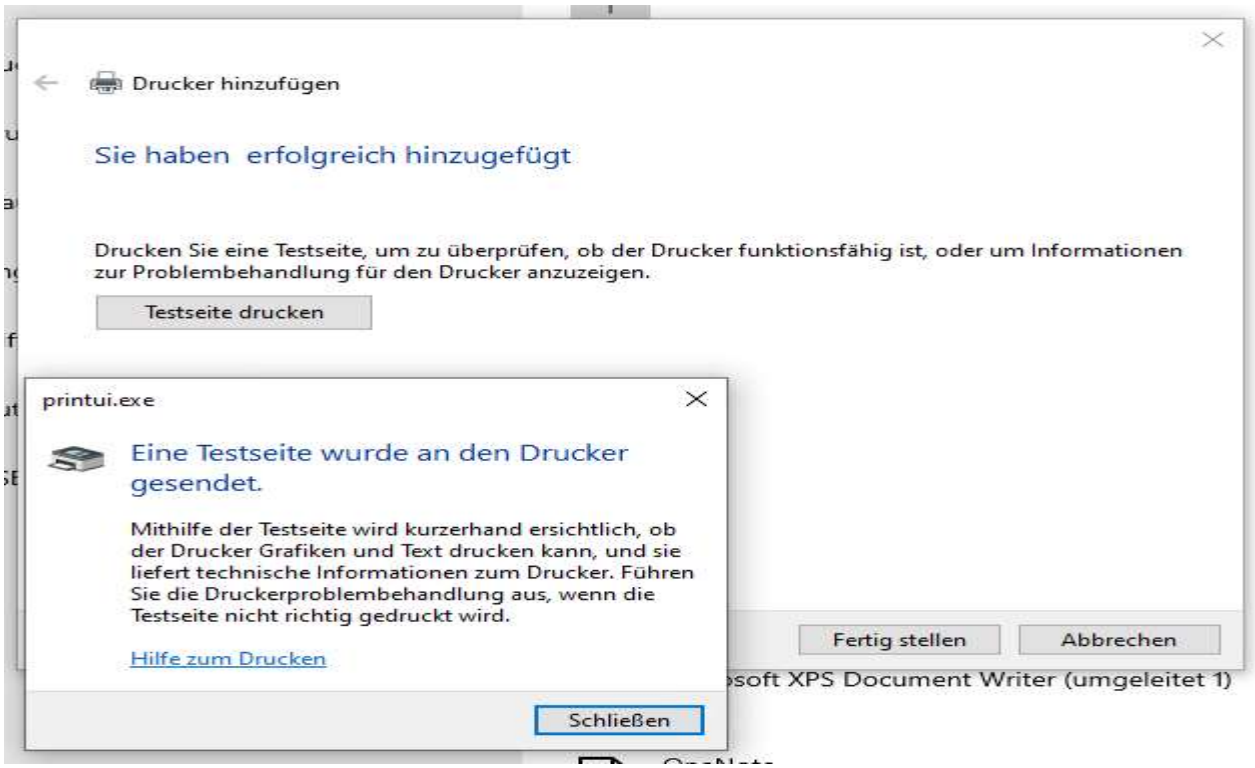

Abbildung 6:

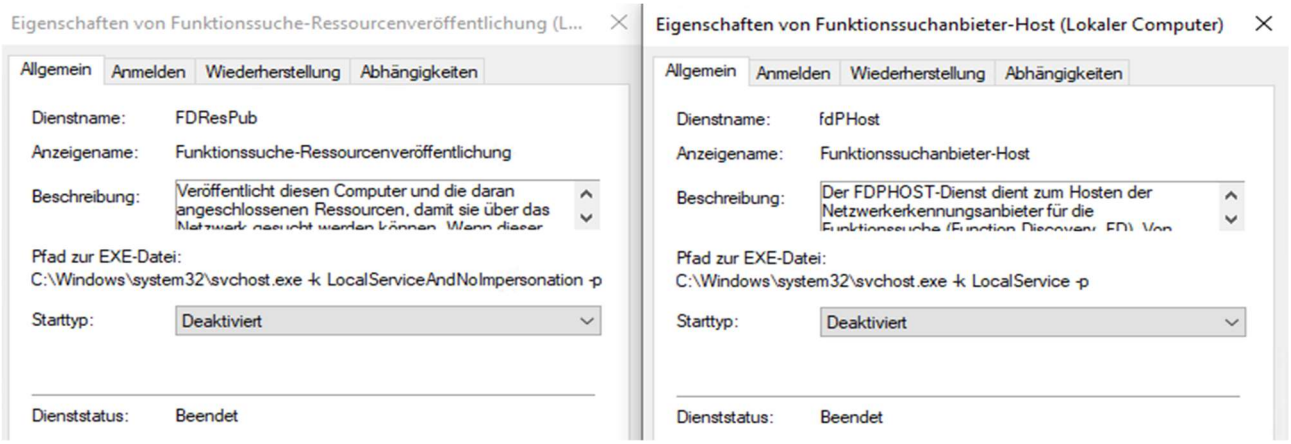

Abbildung 7 zeigt ein Python-Skript für einen SMB-Server