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.**

Revision #1
Created 29 September 2026 10:10:16 by Admin
Updated 29 September 2026 10:11:29 by Admin