Kapitel 2

Metasploit-Framework: Hintergrund, Historie

2.1 Die Geschichte des Metasploit-Frameworks

Bevor das Buch sich in die technischen Tiefgründe von Metasploit stürzt, möchte ich an dieser Stelle erst einmal ein wenig die Hintergründe und die Historie von Metasploit aufzeigen.

Das Metasploit-Projekt wurde im Sommer 2003 von H. D. Moore gegründet. Die ersten Versionen des Frameworks wurden in Perl geschrieben, mit dem Ziel, die Entwicklung und Anwendung von Exploits zu vereinheitlichen und zu vereinfachen.

Gegen Ende 2003 trat der zweite Kern-Entwickler »spoonm« dem Projekt bei und schaffte die Grundstruktur, nach der auch heute noch das Metasploit-Framework aufgebaut ist und bedient wird.

Kurz danach kam auch Matt Miller »skape« an Bord und komplettierte als drittes Mitglied das Metasploit-Kernteam.

Im Jahr 2006 kam mit der Version 2.7 die letzte in Perl geschriebene Version des Frameworks heraus. Bereits 2005 begann das Team damit, Metasploit in Ruby neu zu schreiben, und brachte letztendlich 2007 mit Version 3.0 eine komplett in Ruby geschriebene Version heraus.

In einem weiterhin auf Github verfügbaren Artikel wird als einer der Hauptgründe hierfür angeführt, dass Perl gewisse Nachteile in fehlender Objektorientierung aufweist, sowie der einfache Grund, dass das Entwickler-Team mehr Spaß an Ruby als an Perl hatte.

Im Oktober 2009 wurde dann durch das Metasploit-Projekt-Team verkündet, dass das Projekt von der Firma Rapid7 gekauft und übernommen wurde. Rapid7 ist eine Firma mit dem Fokus auf IT-Security-Software, die neben Metasploit auch für ihren Schwachstellenscanner »Nexpose« bekannt ist.

Seit 2020 ist das Framework in Version 6.0 und derzeit in Version 6.1.41 verfügbar.

2.2 Die Editionen von Metasploit

Nach der Übernahme durch Rapid7 war Metasploit zwischenzeitlich in vier Versionen verfügbar:

- Metasploit Framework Edition
- Metasploit Community Edition (eingestellt)
- Metasploit Express (eingestellt)
- Metasploit Pro

Im Jahr 2022 bietet Rapid7 allerdings nur noch eine kommerzielle Version an: Metasploit Pro.

Metasploit-Framework-Edition

Die Framework-Edition ist weiterhin Open Source und steht unter einer »BSD Style«-Lizenz.

Das Framework ist unter der URL `https://github.com/rapid7/metasploit-framework` quelloffen verfügbar und kann weiterhin von jedem mitentwickelt und unter Berücksichtigung der geltenden Lizenz weiterverwendet werden.

Dieses Buch wird sich ausschließlich mit dieser Version des Metasploit-Frameworks befassen.

Metasploit Pro

Für diese Edition werden keine Preise öffentlich auf der Rapid7-Webseite genannt. Sie dürften allerdings im fünfstelligen Bereich zuzüglich jährlicher Support-Kosten liegen.

Also ganz klar für den professionellen Einsatz durch große Pentesting-Firmen oder Firmen mit großen hausinternen Penetrationstest-Abteilungen gedacht.

Diese Edition weist neben dem Webinterface auch wieder ein erweitertes Konsolen-Interface ähnlich dem der Framework-Edition auf.

Eine ausführliche Auflistung aller Features findet sich auf der Rapid7-Homepage. Ein paar der interessantesten Features sind im Folgenden aufgelistet:

- Interaktion mit dem kommerziellen Schwachstellenscanner Nexpose
- Social-Engineering-Module (z.B. Phishing)
- AV-Evasion (Umgehen von Virenscannern)
- IDS/IPS-Evasion (Umgehen von Netzwerk-Traffic-Scannern)
- VPN-Pivoting (siehe auch Abschnitt 9.3.1 »Pivoting«)
- 24/7-Support durch Rapid7

Um welche Metasploit-Edition dreht sich dieses Buch?

Dieses Buch befasst sich abseits der kurzen Einleitung in das Metasploit-Framework ausschließlich mit der quelloffenen Framework-Edition.

Zwar bieten die kommerziellen Ableger der Firma Rapid7 sicherlich nützliche Zusatzfeatures, allerdings sind sie dafür auch kostenpflichtig und für das Erlernen des Frameworks nicht unbedingt notwendig.

Wer das Bedienen von Metasploit mit der freien Open-Source-Framework-Edition erlernt, wird ohne Weiteres in der Lage sein, bei Bedarf die kommerziellen Editionen zu benutzen.

2.3 Die Wahl der Open-Source-Framework-Edition

Warum habe ich mich auf die Open-Source-Framework-Edition fokussiert und basiere dieses Buch auf dieser Version?

Zum einen aus Überzeugung. IT-Security-Tools müssen nicht kostenpflichtig, kommerziell und Closed Source sein. Metasploit ist eines der Vorzeigeprojekte hierfür.

Auch die Firma Rapid7 gehört an dieser Stelle dafür gelobt, dass sie dem »Open Core«-Modell folgt und den Kern – also die Framework-Edition – weiterhin als kostenlose Open-Source-Version am Leben hält.

Darüber hinaus bin ich der Meinung, dass das Erlernen der Metasploit-Konsole gegenüber den vereinfachten Webinterfaces den großen Vorteil bringt, dass man sich genau überlegen muss, was man vorhat und erreichen will.

Den professionellen Penetrations-Tester unterscheidet vom »Script Kiddy«, dass er genau weiß, was er tut und seine Tools zielgerichtet und so schadfrei wie möglich einsetzt.

Genau dieses zielgerichtete Wissen und Anwenden möchte ich den Lesern dieses Buches zugutekommen lassen.

Als letzten Grund möchte ich dann noch die »Offenheit« der Version anführen, die dazu führt, dass kurz nach Bekanntwerden von großen Schwachstellen sehr häufig zeitnah ein Metasploit-Modul zum Auffinden der Schwachstelle (Auxilliary-Modul) sowie ein Exploit für die Schwachstelle für Metasploit bereitsteht.

Ein Beispiel stellt die Sicherheitslücke MS17-010 in Microsofts SMB-Dienst von Windows dar, die am Wochenende 13.05./14.05.2017 weltweite Bekanntheit dadurch erlangte, dass sie von der Ransomeware (Verschlüsselungstrojaner)

»WannaCry« wurmartig ausgenutzt wurde und die schadhafte Verschlüsselung vieler Computer auf der ganzen Welt ermöglichte.

Das passende Scanner-Modul (`smb_ms17_010`) tauchte erstmals am 29.03.2017 in Metasploit auf.

Der passende Exploit (`ms17_010_eternalblue`) war am 14.05.2017 kurz nach den Schlagzeilen verfügbar und wäre wahrscheinlich durch seine Bekanntheit wegen seiner Herkunft bei der NSA sicherlich auch schon früher in Metasploit integriert gewesen, wenn er nicht große »SMB-Protokoll-Binary-Blobs« enthalten hätte, die mühselig von den Exploit-Modul-Entwicklern revers-engineert, also ohne Dokumentation manuell nachvollzogen und erprobt werden mussten.

Ein weiteres aktuelleres Beispiel stellt zur Zeit des Verfassens dieses Buches die Sicherheitslücke CVE-2021-44228 Log4Shell dar. Log4Shell wurde am 09.12.2021 veröffentlicht und löste so bei vielen Administratoren ein arbeitsintensives Wochenende und einen arbeitsintensiven Dezember im Jahr 2021 aus.

Das passende Scanner-Modul (`log4shell_scanner`) tauchte erstmals am 15.12.2021 – also 6 Tage nach Veröffentlichung der Schwachstelle – im Metasploit-Github-Repository auf.

Passende Exploits-Module wie beispielsweise `log4shell_header_injection` und `vmware_vcenter_log4shell` wurden erstmalig am 07.01. respektive 13.01.2022 im Github-Repository eingecheckt.

Metasploit ermöglicht es also, seine Systeme gezielt und zeitnah nach der Bekanntmachung solcher Schwachstellen zu auditieren.

Ich ermutige trotzdem jeden Leser dieses Buches dazu, auch die Pro-Edition von Metasploit zu erproben und zu testen. Gegebenenfalls weisen sie ja trotzdem Funktionen auf, die das Verwenden oder gar Erwerben dieser Editionen rechtfertigt.

IT-Security-Expertise basiert nicht zuletzt darauf, möglichst viele Systeme zu verstehen und so einen möglichst großen Überblick und Wissensschatz aufzubauen.