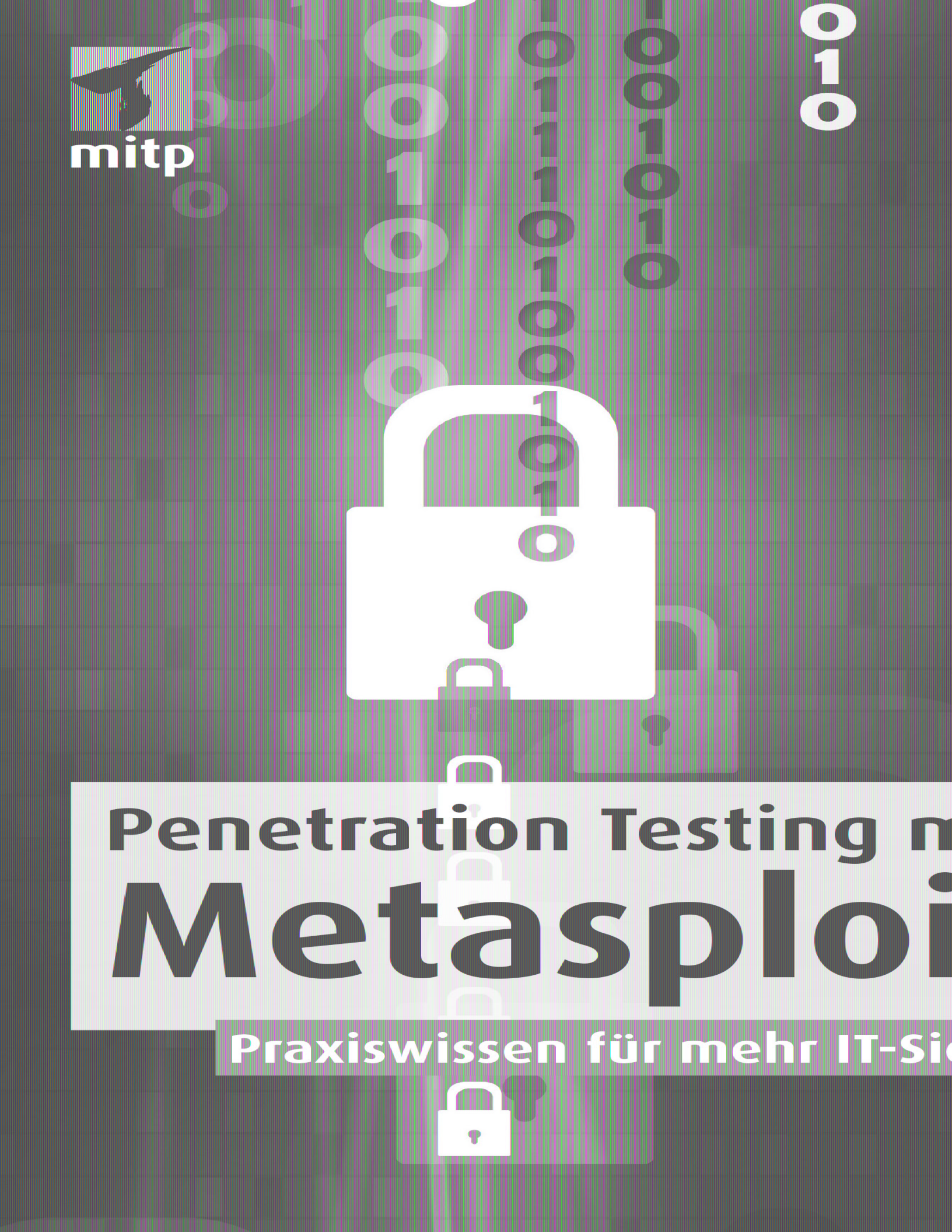
mitp
Penetration Testing m
Metasploi
Praxiswissen für mehr IT-Si

Hinweis des Verlages zum Urheberrecht und Digitalen Rechtemanagement (DRM)

Liebe Leserinnen und Leser,

dieses E-Book, einschließlich aller seiner Teile, ist urheberrechtlich geschützt. Mit dem Kauf räumen wir Ihnen das Recht ein, die Inhalte im Rahmen des geltenden Urheberrechts zu nutzen. Jede Verwertung außerhalb dieser Grenzen ist ohne unsere Zustimmung unzulässig und strafbar. Das gilt besonders für Vervielfältigungen, Übersetzungen sowie Einspeicherung und Verarbeitung in elektronischen Systemen.

Je nachdem wo Sie Ihr E-Book gekauft haben, kann dieser Shop das E-Book vor Missbrauch durch ein digitales Rechtemanagement schützen. Häufig erfolgt dies in Form eines nicht sichtbaren digitalen Wasserzeichens, das dann individuell pro Nutzer signiert ist. Angaben zu diesem DRM finden Sie auf den Seiten der jeweiligen Anbieter.

Beim Kauf des E-Books in unserem Verlagsshop ist Ihr E-Book DRM-frei.

Viele Grüße und viel Spaß beim Lesen,

Ihr mitp-Verlagsteam

Neuerscheinungen, Praxistipps, Gratiskapitel,
Einblicke in den Verlagsalltag –
gibt es alles bei uns auf Instagram und Facebook

instagram.com/mitp_verlag

facebook.com/mitp.verlag

Inhaltsverzeichnis

Penetration Testing mit Metasploit

Impressum

Kapitel 1: Einleitung

- 1.1 Ziel und Inhalt des Buches
- 1.2 Rechtliches
- 1.3 Die Einverständniserklärung
- 1.4 Begrifflichkeiten und Glossar

Kapitel 2: Metasploit-Framework: Hintergrund, Historie

- 2.1 Die Geschichte des Metasploit-Frameworks
- 2.2 Die Editionen von Metasploit
- 2.3 Die Wahl der Open-Source-Framework-Edition

Kapitel 3: Kali-Linux-Umgebung aufsetzen

- 3.1 Die richtige Plattform
- 3.2 Hintergründe zu Kali Linux
  - 3.2.1 Offensive Security
  - 3.2.2 Kali-Versionen – Rollierende Releases
- 3.3 Muss es unbedingt Kali Linux sein?
- 3.4 Fluch und Segen zugleich
- 3.5 Nativ oder als VM?
  - 3.5.1 Kali als native Installation – volle Performance
  - 3.5.2 Kali als VM – immer dabei

3.5.3 32 Bit oder 64 Bit?
3.6 Erste Schritte nach der Installation
3.6.1 Kali-Passwort setzen
3.6.2 Zurechtfinden
3.6.3 Aktualisieren des Systems
3.7 Ein erstes Zielsystem: Metasploitable2

Kapitel 4: Pentesting-Grundlagen

4.1 Begriffsdefinition
4.2 Der Scope
4.3 Schwarz, Weiß, Grau
4.4 Häufigkeit und Zeitpunkt
4.5 Verschiedene Arten von Pentests
4.5.1 Infrastruktur-Pentests
4.5.2 Windows-Domain-/Active-Directory-Pentest
4.5.3 Webapplication-Pentests
4.5.4 Application-Pentests
4.5.5 Wireless-, Physical-Pentests … and so much more
4.5.6 All-In!
4.6 Pentesting-Phasen
4.6.1 Phase 1: Reconnaissance/Information Gathering
4.6.2 Phase 2: Exploitation
4.6.3 Phase 3: Privilege Escalation & Post Exploitation
4.6.4 Phase 4: Go back to 1 / Pivot and Escalate
4.7 Unterschied zwischen Schwachstellenscans, Pentests und Schwachstellenmanagement

4.7.1 Penetrationstest
4.7.2 Schwachstellenscan
4.7.3 Schwachstellenmanagement
4.7.4 Der Mix macht’s!

Kapitel 5: Schwachstellen und Exploits

5.1 Softwareschwachstellen/Schwachstellen im Quelltext
5.2 Konfigurationsschwachstellen
5.3 Standard-, keine und unsichere Passwörter
5.3.1 Standard-Passwörter
5.3.2 Keine Passwörter
5.3.3 Unsichere Passwörter
5.3.4 Eine Lösung für das Passwort-Problem?
5.4 Exploits
5.4.1 Remote Buffer Overflow in SLmail 5.5
5.4.2 Der passende Exploit
5.4.3 Verschiedene Kategorien von Exploits

Kapitel 6: Nmap-Exkurs

6.1 Wie funktionieren Portscanner?
6.1.1 Ein wenig TCP/IP-Theorie
6.1.2 Wie erhebt ein Portscanner offene Ports?
6.2 Keine Angst vor der Nmap-Hilfe
6.3 Erste Gehversuche mit Nmap
6.3.1 Spezifizieren der zu scannenden Ports
6.3.2 Offene Ports vs. lauschende Dienste
6.3.3 Nmap-Script-Scanning
6.3.4 Schwachstellenscanning mit Nmap

6.4 Intensität und Datenmengen von Portscans verstehen

6.5 Die wichtigsten Nmap-Parameter

Kapitel 7: Metasploit-Basics

7.1 Der erste Start

7.1.1 Die Datenbankanbindung

7.1.2 Initialisierung der Datenbank

7.2 Erste Gehversuche im Framework

7.2.1 Tab Autocomplete und Befehlshilfen

7.2.2 Modul-Workflow – Arbeitsschritte

7.3 Metasploit-Module

7.3.1 Die Modulfamilien

7.4 Eine Ablaufkette am Beispiel von Metasploitable2

7.4.1 Portscan mit Nmap

7.4.2 Validierung mit Metasploit-Auxiliary-Modul

7.4.3 Eindringen mittels Metasploit-Exploit und Payload

7.4.4 Post-Exploitation

7.4.5 Übungsaufgabe

Kapitel 8: Metasploit in der Verteidigung

8.1 Von der Heise-Meldung über das Patchen bis zur Validierung

8.1.1 Transparenz und Awareness schaffen mit Metasploit

8.1.2 Auffinden und Ausnutzen von Heartbleed mit Metasploit

8.1.3 Schließen von Schwachstellen validieren

8.2 Andere Einsatzmöglichkeiten in der Verteidigung

Kapitel 9: Praxisbeispiele

- 9.1 Vom Word-Dokument zum Domänen-Administrator
  - 9.1.1 Die Laborumgebung
- 9.2 Initialvektor: Word-Makro
  - 9.2.1 Der soziale Aspekt – Social Engineering
  - 9.2.2 Research, Research, Research
  - 9.2.3 Erstellung der Word-Datei mit Metasploit
  - 9.2.4 Reverse Listener starten
  - 9.2.5 Übertragen und Ausführen der Word-Datei
  - 9.2.6 Local Privilege Escalation
  - 9.2.7 Situational Awareness
- 9.3 Eskalation in der Windows-Domäne
  - 9.3.1 Pivoting
  - 9.3.2 Pass-The-Hash
  - 9.3.3 Clientside-Exploitation
  - 9.3.4 Letzte Hürde: Domänen-Administrator
- 9.4 Erkenntnisse für die Verteidigung
  - 9.4.1 User-Awareness-Maßnahmen als erste Verteidigungslinie
  - 9.4.2 Antiviren- und Endpoint-Security-Software
  - 9.4.3 Office-Makros und Sicherheit allgemein
  - 9.4.4 Keinerlei Ports aus dem LAN ins Internet öffnen
  - 9.4.5 Berechtigungshygiene
  - 9.4.6 Privilege Escalation mittels UAC unterbinden
  - 9.4.7 Auffinden von Schwachstellen durch aktive Schwachstellenscanner
  - 9.4.8 Alarmierung mittels Microsoft Defender for Identity (MDI)

9.4.9 Office-Angriffsfläche mit Windows Defender Exploit Guard verringern

9.5 Das IT-Security-Wettrüsten

Kapitel 10: Anti-Virus-Evasion

10.1 Grundlagen der Anti-Virus-Evasion

10.1.1 Wann kann man von einem Virenscanner erkannt werden?

10.1.2 Pattern-Matching

10.1.3 Wenn Virenscanner selbst zur Schwachstelle werden

10.1.4 Network-, Filetype-, Clientside- und Post-Exploitation

10.2 Generierung von Stand-alone-Payloads mit Metasploit

10.2.1 msfvenom generiert Payloads

10.2.2 Standardtechniken: Packer und Encoder

10.3 AV-Evasion mit PowerShell

10.3.1 Gar nicht erst in Berührung mit dem Virenscanner kommen: PowerShell

10.3.2 Invoke-Shellcode.ps1

10.3.3 Base64-All-The-Things by Hand

10.3.4 Klickbare Payload

10.3.5 Verteidigung: PowerShell sperren

10.4 Katz-und-Maus-Spiel – Neue Techniken und Tricks nutzen

Kapitel 11: Nessus-Schwachstellenscanner

11.1 Schwachstellen scannen

11.2 Vergleich Schwachstellenscanner

11.2.1 Unterschied Schwachstellenscanner zu Virenscanner
11.2.2 Funktionsweise von Schwachstellenscannern
11.3 Nessus-Versionen
11.4 Nessus-Anwendung in der Praxis
11.4.1 Installation und Start
11.4.2 Erstellen einer passenden Policy
11.4.3 Scan der bekannten Labor-Umgebung
11.4.4 Schwachstellen-Bewertungssysteme
11.4.5 Auswertung der Nessus-Scan-Ergebnisse
11.4.6 Effiziente Filtermöglichkeiten
11.5 Schwachstellenmanagement
11.5.1 Tenable.sc-SecurityCenter

Kapitel 12: Schlusswort

Anhang A: Glossar

Sebastian Brabetz

Penetration Testing mit Metasploit

Praxiswissen für mehr IT-Sicherheit

Impressum

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de abrufbar.

ISBN 978-3-7475-0564-9
2. Auflage 2022

www.mitp.de
E-Mail: mitp-verlag@sigloch.de
Telefon: +49 7953 / 7189 - 079
Telefax: +49 7953 / 7189 - 082

© 2022 mitp Verlags GmbH & Co. KG

Dieses Werk, einschließlich aller seiner Teile, ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlages unzulässig und strafbar. Dies gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Lektorat: Janina Bahlmann, Sabine Janatschek
Sprachkorrektorat: Petra Heubach-Erdmann
Covergestaltung: Christian Kalkert, www.kalkert.de
Coverbild: © j-mel / stock.adode.com
electronic **pub**lication: III-satz, Flensburg, www.drei-satz.de

Dieses E-Book verwendet das ePub-Format und ist optimiert für die Nutzung mit dem iBooks-reader auf dem iPad von Apple. Bei der Verwendung anderer Reader kann es zu Darstellungsproblemen kommen.

Der Verlag räumt Ihnen mit dem Kauf des E-Books das Recht ein, die Inhalte im Rahmen des geltenden Urheberrechts zu nutzen. Dieses Werk, einschließlich aller seiner Teile, ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlages unzulässig und strafbar. Dies gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und Einspeicherung und Verarbeitung in elektronischen Systemen.

Der Verlag schützt seine E-Books vor Missbrauch des Urheberrechts durch ein digitales Rechtemanagement. Bei Kauf im Webshop des Verlages werden die E-Books mit einem nicht sichtbaren digitalen Wasserzeichen individuell pro Nutzer signiert.

Bei Kauf in anderen E-Book-Webshops erfolgt die Signatur durch die Shopbetreiber. Angaben zu diesem DRM finden Sie auf den Seiten der jeweiligen Anbieter.

Kapitel 1
Einleitung

1.1 Ziel und Inhalt des Buches

Ziel dieses Buches soll nicht sein, dem erfahrenen Metasploit-Nutzer die letzten Tricks und Kniffe des Metasploit-Frameworks beizubringen. Vielmehr soll der unbedarfte (Security-)Administrator gezeigt bekommen, dass Metasploit ein mächtiges Werkzeug ist, das zum Verstehen und Nachstellen von gängigen Angriffsmethoden genutzt werden kann.

Die Veröffentlichung von Schwachstellen und Patches, die wiederum selbige schließen, ist heutzutage an der Tagesordnung. So vergeht kein Monat, in dem nicht eine kritische Schwachstelle im Internet bekannt gemacht wird und Hersteller Hals über Kopf als »kritisch« deklarierte Patches herausbringen.

Wie allerdings kann der normalsterbliche Administrator sicher sein, dass eine Lücke in erster Linie überhaupt vorhanden war? Und wie kann er dafür geradestehen, dass nach dem Einspielen von Patches und/oder begleitenden Konfigurationsänderungen diese Sicherheitslücken auch tatsächlich geschlossen sind?

Inhaltlich sind moderne Schwachstellen mittlerweile so komplex geworden, dass es ein eigener Karrierepfad ist, Schwachstellen aufzufinden und auszunutzen (sogenannte Exploits dafür zu entwickeln). Gerade abseits der Programmier-Berufe fehlt der Einblick und Tiefgang in die Funktionsweise moderner Technologien und Programmiersprachen, um die technischen Hintergründe

von Schwachstellen zu verstehen. Selbst Programmierer mit jahrelanger Erfahrung sind nicht automatisch befähigt, Schwachstellen zu verstehen, geschweige denn, sie zu vermeiden. Dies stellt unter anderem auch einen Grund dar, weshalb immer wieder neue Schwachstellen in Software auftauchen.

Wie also soll der Administrator hiermit umgehen und sich sicher sein, dass Lücken existieren und geschlossen werden? Ein Weg ist es, die veröffentlichten Schwachstellen und Angriffstechniken unter sicheren Rahmenbedingungen selbst anzuwenden; Schwachstellen selbst auszunutzen und Exploits selbst einzusetzen, um mit eigenen Augen zu sehen, dass diese vor dem Patchen noch und nach dem Patchen nicht mehr funktionieren.

Genau hier kommt Metasploit ins Spiel.

Hinweis: Was ist ein Exploit?

Ein Exploit kann ein beliebig komplexes Stück Software (oft auch nur ein kleines Script) sein, das es ermöglicht, eine Schwachstelle gezielt auf Knopfdruck auszunutzen.

Bemerkenswert ist hierbei, dass der Anwender eines Exploits die genaue Funktionsweise nicht unbedingt verstehen muss. Oft sind Exploits »schlüsselfertig« und müssen nur noch in die richtige »Richtung« (IP-Adresse und Port) gezielt werden.

Es gibt viele Kategorien von Exploits, zwei der wichtigsten sollen hier kurz erwähnt werden:

Remote-Code-Execution-(RCE-)Exploits:

RCE-Exploits ermöglichen das Ausführen von Code auf Ziel-Systemen über das Netzwerk, ohne einen autorisierten Zugang zum Zielsystem zu besitzen.

Local Exploits:

Lokale Exploits dienen dazu, eingeschränkte Rechte auf einem Zielsystem zu erweitern: So bekommt man über einen RCE Exploit für eine Webapplikation gegebenenfalls eine eingeschränkte Kommandozeile auf einem Linux-Webserver unter dem niedrig privilegierten Apache-User. Ein Local Exploit kann nun helfen, die eingeschränkten Rechte z.B. auf Root-Berechtigungen zu erweitern.

Mehr dazu in Kapitel 5 »Schwachstellen und Exploits«.

Metasploit ist ein mächtiges Framework mit vielschichtigem Einsatzgebiet. So bringt das Metasploit-Framework z.B. Tools mit, die dabei helfen können, Schwachstellen zu finden und Exploits dafür zu entwickeln. Spannender für dieses Buch ist aber die Tatsache, dass mithilfe von Metasploit das Ausnutzen (Exploiten) von Schwachstellen von den technischen Hintergründen der Schwachstellen abstrahiert wird.

Gemeint ist damit, dass das Metasploit eine Vielzahl (2218 zum Zeitpunkt der Fertigstellung dieser Auflage – Mitte 2022) von Exploits für die unterschiedlichsten Betriebssysteme und Programmiersprachen mitbringt. Sie als Anwender müssen aber nur einmal die grundlegende Funktionsweise des Frameworks verstehen und können danach z.B. Linux- sowie Windows-Ziele angreifen, ohne eingefleischter Experte auf der Zielplattform zu sein.

Sie müssen also kein PHP-Guru oder Linux-Kernel-Entwickler sein, um eine PHP-Webapplikationsschwachstelle

auszunutzen und danach mittels einer lokalen Linux-Kernel-Schwachstelle vom Apache-Nutzer zum Root-Benutzer zu eskalieren.

Bei all dem kann Metasploit Sie unter den richtigen Voraussetzungen mit ein paar einfachen, wenigen Befehlen unterstützen.

Wichtig: Voraussetzungen zum Verständnis des Buches

Ich möchte niemandem vorschreiben, ob und wann er dieses Buch lesen sollte. Allerdings wurde es mit der Zielgruppe Administratoren und IT-Security-Verantwortlichen im Hinterkopf geschrieben und setzt ein gewisses Vorwissen voraus:

Erfahrungen in der Systemadministration, Netzwerkgrundlagen und dem Programmieren werden an vielen Stellen vorausgesetzt. Es wird jedoch versucht, die technischen Begebenheiten möglichst einfach darzulegen und auf unnötige Verkomplizierungen zu verzichten.

Sollten Sie trotzdem irgendwo nicht folgen können, so kann ich nur dazu ermutigen, einfach die Suchmaschine Ihrer Wahl anzuwerfen und die unklaren Begriffe oder Zusammenhänge zu kombinieren.

Mittlerweile gibt es einen riesigen Schatz an völlig frei zugänglichen Informationen zu diesem Themengebiet, der aber zum größten Teil in der englischen Sprache zu finden ist.

Metasploit kommt mittlerweile in verschiedenen Formen. So wurde das Open-Source-Projekt durch die Firma Rapid7 übernommen und parallel zu einem kommerziellen Produkt weiterentwickelt und vertrieben. Welche Version Sie einsetzen können und wie Sie diese am einfachsten verwenden, wird in Kapitel 2 thematisiert.

Wie zu Beginn schon erwähnt, ist es Ziel dieses Buches, Metasploit nicht nur als reines Angriffswerkzeug (z.B. für Penetration Testing) darzustellen, sondern es auch Systemadministratoren und IT-Security-Verantwortlichen für die Verteidigung zugänglich zu machen.

An dieser Stelle verzichte ich auf das in anderen Werken obligatorische Sun-Tzu-Zitat, da es meiner Meinung nach relativ logisch ist, Angriffstechniken zu verstehen und zu erlernen, um sich effektiv gegen selbige verteidigen zu können.

Kapitel 9 wird hierfür einige gängige Angriffspfade und Techniken erläutern und mithilfe von Metasploit nachstellbar machen.

Bevor dieses Buch entstand, habe ich bereits einige freie Community-Metasploit-Workshops sowie kommerzielle Metasploit-Trainings abgehalten. Für diesen Zweck entstand über die Jahre eine Laborumgebung, anhand der die Teilnehmer der Workshops Metasploit in der Praxis anwenden und testen können.

Da gewisse Angriffstechniken, wie z.B. clientseitige Angriffe sowie die Eskalation in modernen Windows-Domänen-Umgebungen, voraussetzen, dass man auch entsprechende Systeme zur Verfügung hat, ist diese Laborumgebung deutlich komplexer geworden, als einfach nur ein bis zwei

ungepatchte Windows-Installationen zur Verfügung zu haben.

Im Verlaufe dieses Buches werde ich in Abschnitt 9.1.1 diese Laborumgebung zumindest teilweise erläutern und als Grundlage zur Demonstration von Metasploit verwenden. Das Kapitel wird außerdem Anregungen und Tipps zum Aufbau eines eigenen Labors geben.

Trotz Fokussierung auf Metasploit wird dieses Buch jedoch an vielen Stellen über den Tellerrand blicken und weitere Tools erwähnen und erklären, die sich mit Metasploit ergänzen.

1.2 Rechtliches

Wahrscheinlich kommt kein Buch, das sich um IT-Security dreht, ohne einen entsprechenden Warnhinweis aus:

Das unbedarfte und unkontrollierte Anwenden von Werkzeugen wie Metasploit und anderen Programmen, die sich in Kali Linux befinden, kann (gegebenenfalls versehentlich) zu Straftaten führen.

Es verstößt gegen deutsches Gesetz, ohne Erlaubnis der Eigentümer von IT-Systemen diese auf Schwachstellen zu überprüfen oder gar Schwachstellen in diesen Systemen auszunutzen.

Doch selbst mit Erlaubnis und Einverständniserklärung der Eigentümer von IT-Systemen kann es durchaus nicht rechtens sein, IT-Systeme zu auditieren. Nehmen wir einmal das Beispiel eines Mailservers in der eigenen Firma. Auf diesem Mailserver liegen gegebenenfalls vertrauliche oder

private E-Mails, die nach dem deutschen Postgeheimnis zu betrachten sind.

Auch Shared-Hosting-Umgebungen, wie sie z.B. bei jeglichen Cloud-Providern vorliegen, stellen ein Problem dar: Decken oder nutzen Sie gar eine Schwachstelle in der unterliegenden Infrastruktur des Cloud-Providers auf, so kommen Sie gegebenenfalls an Daten anderer Nutzer dieser Infrastruktur.

Dies gilt es unbedingt zu vermeiden und bedarf ganz klarer vertraglicher Regelungen mit dem jeweiligen Provider.

Lassen Sie sich hiervon aber auch nicht einschüchtern. Sicherheitsaudits sind auch in diesen Umgebungen sehr nützlich und wichtig. Sicherheitsaudits lassen alle guten Cloud-Provider unter abgesteckten Bedingungen zu.

Auch könnte wiederum der Internet-Service-Provider, über dessen Infrastruktur ein einfacher Portscan läuft, ein Problem damit haben. Viele Internet-Service-Provider haben hierzu Klauseln in den Verträgen. Gerade bei privaten Anschlüssen wird das Portscanning gern pauschal verboten. Selbst habe ich zwar noch keine Fälle davon erlebt, dass Internet-Provider deshalb Anschlüsse gekündigt oder Kunden abgemahnt haben, aber auf Nummer sicher geht, wer sich auch hier explizit eine Freigabe einholt.

Zu guter Letzt sollte klar sein, dass es schon ein Kündigungsgrund sein kann, wenn Sie unbedarft mit Metasploit bei Ihrem Arbeitgeber experimentieren. Selbst wenn Sie dabei nichts zerstören und nur gute Beweggründe haben.

1.3 Die Einverständniserklärung

Zu jedem Penetrationstest und Schwachstellenaudit gehört also immer eine schriftlich und vertraglich festgehaltene Einverständniserklärung des Eigentümers der Infrastruktur und aller beteiligten Provider.

Vorlagen hierfür bekommen Sie beim Beauftragen von Schwachstellenscans und Penetrationstests bei professionellen Anbietern oder sicherlich auch frei verfügbar im Internet.

Vorsichtshalber lassen Sie eine solche Vorlage aber lieber durch Anwälte prüfen, bevor Sie größere Audits unternehmen.

Wichtig: IANAL – I am not a Lawyer

Dieses Buch stellt keine fundierte Rechtsberatung dar.

Es soll lediglich an dieser Stelle davor warnen, dass die rechtlichen Rahmenbedingungen der IT-Security sehr ernst genommen werden sollten.

Im Notfall bleiben Sie beim Lesen und Nachverfolgen dieses Buches komplett auf virtuellen Maschinen auf Ihrem privaten Computer oder besuchen entsprechend vorbereitete Workshops oder Weiterbildungen, die abgeschottete Demo-Umgebungen bereitstellen.

1.4 Begrifflichkeiten und Glossar

Zu guter Letzt möchte ich drauf hinweisen, dass es bei tiefgehenden Themen wie Metasploit und IT-Security immer mal wieder vorkommen kann, dass Ihnen einzelne Begriffe oder Hintergründe unklar sind.

Ich habe daher versucht, entsprechende Begriffe im Glossar am Ende des Buches zu beschreiben.

Sollte Ihnen trotzdem beim Lesen noch etwas unklar sein, scheuen Sie sich nicht davor, den Begriff einfach in der Suchmaschine Ihrer Wahl einzugeben. Ich versichere Ihnen, dass Sie zu all dem Geschriebenen in diesem Buch eine Vielzahl von Webseiten finden werden, die Ihnen die Hintergründe weiter erläutern.

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.