

# Iterative Kompromittierungsgraphverfeinerung als methodische Grundlage für Netzwerkpenetrationstests

Felix C. Freiling

Universität Mannheim

freiling@informatik.uni-mannheim.de

Jens Liebchen

RedTeam Pentesting GmbH

jens.liebchen@redteam-pentesting.de

**Abstract:** In einem *Penetrationstest* wird die Sicherheit eines Systems durch einen kontrollierten Angriff überprüft. Penetrationstests sind in der Praxis weit verbreitet, das Phänomen ist jedoch wissenschaftlich bisher noch kaum untersucht. Dieser Beitrag entwickelt ein Vorgehensmodell für Penetrationstests aus praktischen Erwägungen heraus. Eine anschließende Formalisierung des Modells basierend auf graphtheoretischen Konzepten erlaubt es, Penetrationstests präzise zu beschreiben und Effizienzmetriken für praktische Tests zu definieren, die Penetrationstestern als Hilfestellung bei der Planung, Budgetierung und Durchführung solcher Tests behilflich sein können.

## 1 Einführung

**Motivation.** Der Begriff des *Penetrationstests* wurde 1995 in Zusammenhang mit der Veröffentlichung des Tools *SATAN* von Farmer und Venema [5] eingeführt. Dort wurde zum ersten mal über die Möglichkeit berichtet, Netzwerke aus Angreiferperspektive zu untersuchen. Zur damaligen Zeit heftig umstritten, gehören Penetrationstests heute zum etablierten Inventar an Möglichkeiten, die Sicherheit eines Systems zu verbessern. Entsprechend offensiv ausgerichtete Programme zur Benutzung in Penetrationstests sind heute weit verbreitet (siehe beispielsweise Lyon [14]). Umso erstaunlicher ist, dass es kaum wissenschaftliche Literatur gibt, die sich mit dem Phänomen der Penetrationstests auseinandersetzt.

**Zum Begriff des Penetrationstests.** In der Literatur zur Begriffsbildung von Penetrationstests und seiner Vorgehensmodelle [2, 6, 8, 10–13, 16, 20–22] dominieren die nicht-wissenschaftlichen Quellen [10–13, 16, 20, 22]. Aber auch die wissenschaftlichen Quellen enthalten sehr unterschiedliche Definitionen. Eine gute begriffliche Basis findet man etwa in einem Dokument des BSI [2]:

Im technischen Sprachgebrauch versteht man unter einem Penetrationstest den kontrollierten Versuch, von *außen* in ein bestimmtes Computersystem

bzw. -netzwerk einzudringen, um Schwachstellen zu identifizieren. Dazu werden die gleichen bzw. ähnliche Techniken eingesetzt, die auch bei einem realen Angriff verwendet werden. [2, Seite 4]

Obwohl diese Definition zunächst brauchbar erscheint, enthält sie doch die unglückliche Einschränkung „von außen“. Bei Penetrationstests in der Praxis wird aus diesem Grund zwischen *internen Penetrationstests* und *externen Penetrationstests* unterschieden. Die Definition von externen Penetrationstests entspricht der Definition des BSI. Bei internen Penetrationstests wird den Pentestern ein Zugang gewährt, der bereits innerhalb des zu testenden Netzwerks liegt. Als Arbeitsgrundlage werden wir darum in diesem Artikel die folgende, etwas allgemeinere Definition eines Penetrationstests verwenden:

Ein Penetrationstest bezeichnet die Sicherheitsüberprüfung eines IT-Systems durch einen kontrollierten Angriff.

In der Praxis unterscheidet man zudem zwei verschiedene Arten von Penetrationstests. Der klassische Penetrationstest ist ein *System-* oder auch *Netzwerkpenetrationstest*, der einen simulierten Angriff auf ein komplettes Firmennetzwerk oder Teile davon umfasst. Unabhängig davon existiert aber auch eine zweite Art von Penetrationstests, die sogenannten *Produktpenetrationstests*. Bei diesen wird ein sicherheitskritisches Produkt auf Schwachstellen überprüft. Diese Sichtweise soll insbesondere als externe Qualitätssicherung im Bereich der IT-Sicherheit dienen und ist insofern verwandt mit dem Komplex der Softwaretests. In dieser Arbeit beschäftigen wir uns primär mit System- bzw. Netzwerkpenetrationstests.

**Zum Vorgehensmodell bei Penetrationstests.** Nicht nur die Definition des Begriffs selbst, sondern auch das methodische Vorgehen bei Penetrationstests ist in der Literatur sehr uneinheitlich. Während beispielsweise Herzog [6] keine Angaben über ein allgemeines Vorgehensmodell macht, beschreibt das BSI [2] das Vorgehen als linearen Prozess und das NIST [21] als Kreislauf.

**Beiträge dieses Artikels** Wie die einleitenden Beispiele verdeutlichen, besteht ein großer Bedarf an wissenschaftlicher Diskussion sowohl über die Definition als auch die methodischen Grundlagen von Penetrationstests. Penetrationstests haben sich in der Praxis als effektives und realitätsnahes Mittel zur Sicherheitsüberprüfung durchgesetzt. Die Ergebnisse von Penetrationstests sind aber ohne eine sinnvolle wissenschaftliche Fundierung schwer einzuschätzen und zu vergleichen.

Dieser Artikel möchte einen ersten Diskussionsbeitrag leisten im Hinblick auf das methodische Vorgehen bei Penetrationstests. Hierzu werden zunächst die relevanten Schlüsselfaktoren für einen wirksamen Penetrationstest vorgestellt. Aus diesen Faktoren wird ein abstraktes Vorgehensmodell für Penetrationstests hergeleitet. Wichtig in der Praxis ist insbesondere eine Abschätzung der möglichen Testdauer vor Testbeginn. Ein Modell hierfür stellt die Literatur zur Zeit nicht zur Verfügung. Wir werden hierfür das Modell der Kompromittierungspfadanalyse vorstellen, mit dem eine solche Abschätzung ermöglicht wird.

Ebenfalls zeigen wir, wie die so abgeschätzte Testzeit möglichst effizient im Testverlauf im Rahmen des abstrakten Vorgehensmodells genutzt werden kann.

## 2 Schlüsselfaktoren bei Penetrationstests

Dieser Abschnitt diskutiert eine Reihe von Schlüsselfaktoren, die aus unserer Erfahrung eine wichtige Rolle bei wirksamen Penetrationstests spielen.

**Realistische Angreiferannahmen.** Im Rahmen von Penetrationstests werden in der Praxis *immer* Schwachstellen gefunden. Wirklich von Interesse sind hierbei jedoch nur die *relevanten* Schwachstellen, d.h. die Schwachstellen, die der erwartete Angreifer mit hoher Wahrscheinlichkeit auch gefunden und ausgenutzt hätte. Zur Definition relevanter Schwachstellen gehört also eine möglichst präzise Beschreibung der möglichen Ziele eines Angreifers und die möglichen Ausgangspunkte eines Angriffs. Hierüber kann dann der Angreifer selbst modelliert werden. Wesentliche Merkmale eines solchen Modells sind die dem Angreifer zur Verfügung stehenden Ressourcen, gemessen in Motivation, Zeit und Geld.

Motivation ist für einen Angreifer die treibende Kraft. Die Motivation kann rein technischer Natur sein, z.B. monetärer Wert des Einbruchs (Verkauf der Kundendatenbank), als auch rein persönlicher Natur. Ein gekündigter Mitarbeiter, der sich für seine Entlassung rächen will, ist ein Angreifertyp, dem im allgemeinen eine hohe Motivation unterstellt wird. Zeit- und Geldmittel variieren je nach Angreifermodell: Bei einem entlassenen Mitarbeiter unterstellt man z.B. weniger Geld, dafür allerdings relativ viel Zeit. Oft lässt sich fehlendes Geld auch durch Zeit kompensieren und umgekehrt. Stärkster angenommener Angreifer ist normalerweise ein Industriespion, bei dem man fast unbegrenzte Zeit- und Geldmittel ansetzt.

Die Wirksamkeit eines Penetrationstests ist immer relativ zum Angreifermodell. Ein Penetrationstest ist gut, wenn er mit den veranschlagten Ressourcen einen möglichst hohen Prozentsatz der Schwachstellen entdeckt, die ein Angreifer mit „ähnlichen“ Ressourcen auch gefunden hätte.

Bei den Penetrationstestern stehen Zeit- und Geldmittel normalerweise nur sehr begrenzt zur Verfügung. Man kann jedoch bestimmte Aufwände für den Angreifer abschätzen und dann simulieren. Ein Praxisbeispiel hierfür ist eine schon verhältnismäßig alte, passive Attacke gegen WEP [1]. Bei dem wohl bekanntesten Angriff gegen WEP muss ein Angreifer eine bestimmte Menge Netzwerkverkehr mitlesen, bevor der genutzte Schlüssel innerhalb von weniger als 60 Sekunden auf einem normalen PC zurückgerechnet werden kann. Ist ein solcher Verkehr in der Praxis nicht vorhanden, so kann, falls man nicht auf die Demonstration des Einbruchs ganz verzichten will, diese Netzwerklast mit Hilfe des Kunden erzeugt werden. Die Aufwände des Penetrationstesters sind hier deutlich geringer als die des Angreifers, der viel mehr Zeit für das Datensammeln benötigen würde. Wichtig bei solchen Simulationen ist, dass keine impliziten Erweiterungen der Angreiferannahmen getroffen werden. Der von den Penetrationstestern durchgeführte Angriff muss

auch für den angenommenen Angreifer ohne die simulierten Rahmenbedingungen in der Praxis durchführbar sein.

**Kreativität.** Die Kreativität der Penetrationstester ist ein sehr wichtiger Faktor bei Penetrationstests. Dies wird offenbar an der Notwendigkeit, ständig neue Angriffswege zu suchen und auszuprobieren. Die Entdeckung von neuartigen Sicherheitslücken im IT-Sicherheitsbereich in der Vergangenheit verlangten in der Regel immer neue und kreative Vorgehensweisen.

**Individualität eines Penetrationstests.** Jedes getestete System und jedes Netzwerk ist anders als das vorher getestete, jeder Test ist also neu und *individuell*. Dies setzt Penetrationstests ab von automatisch ablaufenden Sicherheitsscans. Derartige Scans sind beispielsweise mittels Nessus [19] oder Nikto [3] durchführbar. Zwar können diese Scans durchaus viele Schwachstellen aufdecken, jedoch werden sich nur relativ schwache Angreifer mit diesen automatisierten Techniken zufrieden geben. Da ein solcher schwacher Angreifer normalerweise keine realistische Annahme ist, wird ein Vorgehen, welches sich hauptsächlich an automatisierten Techniken orientiert, nicht als Penetrationstest im Sinne der unter Abschnitt 1 vorgestellten Definition gesehen.

**Schlussfolgerungen.** Die Forderung nach einem realistischen Angreifermodell erfordert eine Definitionsphase beim Penetrationstest, in dem der Kontext für den Test analysiert wird. Die Kreativität der Penetrationstester sowie die Individualität eines Penetrationstests erfordern eine offene Formulierung der Phasen während eines solchen Tests. Schließlich verlangt die Individualität eines Penetrationstests ein iteratives Vorgehen, das nur durch Aufwände gedeckelt ist. Aus diesen allgemeinen Beobachtungen heraus entwickeln wir im folgenden Abschnitt ein abstraktes Vorgehensmodell für Penetrationstests. Ein Modell für die Deckelung des Aufwands, die Kompromittierungspfadanalyse, wird in Abschnitt 4 vorgestellt.

### 3 Ein Vorgehensmodell für Penetrationstests

In diesem Abschnitt wird ein Vorgehensmodell aus den in Abschnitt 2 genannten Schlüsselfaktoren entwickelt. Dazu betrachten wir zunächst die Vorgehensmodelle für Penetrationstests, die in der bisherigen (spärlichen) wissenschaftlichen Literatur existieren.

#### 3.1 Verwandte Arbeiten

**Ablauf nach BSI.** Das BSI [2, Seite 45 ff.] unterteilt einen Penetrationstest in fünf Phasen (vergleiche Abbildung 1, links), die linear durchlaufen werden. In der Phase *Informationsbeschaffung* beginnt der eigentliche Test. In der Phase *Aktive Eindringversuche* ist

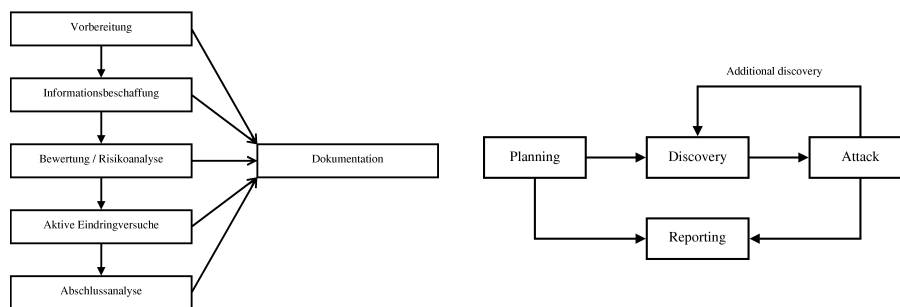


Abbildung 1: Phasen eines Penetrationstests nach BSI [2, Seite 47 ff.] (links) und Wack et al. [21], S. 3–13 (rechts).

die Gefährdung für den Auftraggeber maximal. Aus diesem Grund ist eine Risikoanalyse Bestandteil der vorherigen Phase.

**Ablauf laut Guideline on Network Security Testing (NIST).** Penetrationstests werden bei Wack et al. [21] in vier Phasen aufgeteilt (siehe Abbildung 1, rechts). Der eigentliche Test beginnt erst mit der zweiten Phase, die *Discovery* genannt wird. In der anschließenden Phase *Attack* finden die eigentlichen Angriffsversuche auf die Zielsysteme statt. Wichtig ist der Kreislauf, der von der Angriffsphase wiederum zurück zur vorhergehenden Phase besteht. Durch eine erfolgreiche wie auch erfolglose Verifizierung einer Schwachstelle entsteht bei einem Angreifer neues Wissen über das Zielnetzwerk, welches zu neuen Angriffen genutzt werden kann.

**Weitere Quellen.** Eine weitere Quelle, die relativ einflussreich zu sein scheint, ist das *Open Source Security Testing Manual* von Herzog [6]. So stützen sich Whitaker and Newman [22] und Long et al. [12] auf Herzog [6] als primäre Quelle. Auch in der Praxis beziehen sich einige Penetrationstester auf diese Quelle. Allerdings enthält Herzog [6] gar kein detailliertes Vorgehensmodell für Penetrationstests. Rey et al. [16] beziehen sich auf Herzog [6], das BSI [2] und Wack et al. [21]. Andere Quellen, wie Tiller [20], ISACA Switzerland [8] Lam et al. [11] und Long et al. [13], geben keine Quellen und keine Vorgehensmodelle an und bleiben deshalb eher oberflächlich.

### 3.2 Ein neues Vorgehensmodell

In Abschnitt 2 hatten wir bereits argumentiert, dass ein Vorgehensmodell für Penetrationstests folgende Eigenschaften haben muss, um der Individualität und Kreativität des Prozesses Rechnung zu tragen: (1) Es sollte aus wenigen, klar abgrenzbaren, aber doch allgemein gehaltenen Phasen bestehen, und (2) es sollte einen Kreislauf enthalten. Wir schlagen darum folgenden Ablauf eines Penetrationstests vor, der in vier Phasen gegliedert

ist, die wiederholt durchlaufen werden (vergleiche Abbildung 2). Vorangehend zu diesen vier Phasen ist eine einmalige Phase *Vorgespräch*, in welcher Ziele und Ausgangspunkte des Penetrationstests vereinbart werden.

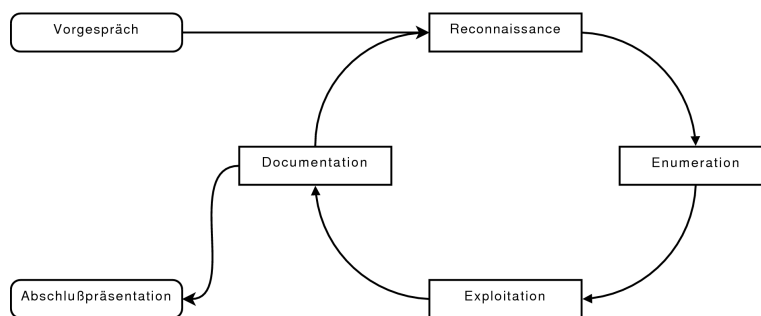


Abbildung 2: Phasen eines Penetrationstests

Die einzelnen Phasen sind präzise voneinander abgrenzbar: *Reconnaissance* bedeutet Informationsbeschaffung. Die Penetrationstester benutzen hierbei ausschließlich *öffentliche* Quellen. Ein direkter Kontakt mit den Zielsystemen findet hierbei nicht statt. Dies demonstriert den Praxisbezug des Penetrationstests: Auch ein realer Angreifer würde zunächst möglichst viele Informationen relativ anonym über sein Angriffsziel sammeln, um erst später gezielt anzugreifen. Dieser Teil des Penetrationstests bleibt also genau wie auch ein echter Angriff in dieser Phase für den Auftraggeber unsichtbar.

In der *Enumeration* wird die Informationsbeschaffung durch Ansprechen der Zielsysteme fortgesetzt. Hier kommen etwa Portscans oder Bannergrabbing-Methoden zum Einsatz. Ergebnis dieser Phase ist etwa die Auflistung der genutzten Betriebssysteme und Softwareversionen sowie vermutete Schwachstellen. Die strikte Trennung zwischen *Reconnaissance* und *Enumeration* basiert auf unterschiedlichen Gefahren. Während Auffälligkeiten in der *Enumeration* durch Auftraggeber direkt korrigiert werden können, so sind Ergebnisse aus der *Reconnaissance* nicht oder kaum korrigierbar und es können lediglich Vorkehrungen getroffen werden, so dass ähnliche Fehler in Zukunft nicht erneut auftreten können. Daten, die einmal in fremde Datenbanken, wie z. B. Suchmaschinen, eingeflossen sind, sind im Zeitalter des Internets nie mehr restlos zu beseitigen.

In der Phase *Exploitation* werden die Schwachstellen, die in den vorgenannten Phasen identifiziert wurden, nach bestimmten Prioritäten getestet. Bei *Documentation* geht es darum, sämtliche Schritte des Penetrationstests ausführlich zu dokumentieren. Da die einzelnen Phasen wiederholt durchlaufen werden, wird während jedes Durchlaufs direkt dokumentiert. Nach dem letztmaligen Durchlauf der Phasen wird der Kreislauf nach der Phase *Documentation* verlassen und es folgt die Vorstellung der Ergebnisse für den Auftraggeber.

## 4 Ein formales Modell basierend auf Kompromittierungsgraphen

In Abschnitt 3 wurde ein neues Vorgehensmodell für Penetrationstests vorgestellt, das als Kreislauf von Informationsbeschaffung, Angriff und Dokumentation dargestellt wurde. In diesem Abschnitt wird dieses Modell zu einer Methodik präzisiert und beschrieben, wie sich ein Penetrationstest in jedem Zyklus fortentwickelt. Die Entwicklung bezieht sich auf die immer umfangreicheren Informationen, die die Penetrationstester über das getestete System erhalten. Diese Informationen können einerseits dafür verwendet werden, um sich realitätsnah (d.h. „wie ein Angreifer“) zu verhalten, andererseits erlauben diese Informationen auch ein ständiges Messen von Aufwand, um wirtschaftlich zu sein, d.h. im veranschlagten Budget zu bleiben ohne dabei die Ziele des Penetrationstests aus den Augen zu verlieren.

Die Idee der hier vorgeschlagenen Methode beruht auf dem Konzept der annotierten Kompromittierungsgraphen und der Kompromittierungspfadanalyse. Beide Konzepte werden im folgenden Abschnitt 4.1 vorgestellt. Die Vorstellung der Methodik folgt dann in Abschnitt 4.2.

### 4.1 Kompromittierungsgraphen und -pfade

Ein *Kompromittierungsgraph* ist ein gerichteter Graph  $G = (V, E)$  bestehend aus Knotenmenge  $V$  und Kantenmenge  $E \subseteq V \times V$ . Die Knoten entsprechen aktiven Netzwerkelementen (Computer, Router, Switch, bzw. einzelne Netzwerkdienste wie Mail oder DNS). Mehrere Netzwerkelemente können in einem Knoten zusammengefasst werden, sofern es keine relevanten Unterschiede für den Ablauf des Penetrationstests macht, welches System aus dieser Vereinigungsmenge angegriffen wird. Der Graph enthält eine nichtleere Menge  $S \subseteq V$  von *Startknoten* und eine nichtleere Menge  $Z \subseteq V$  von *Zielknoten*. Startknoten sind Netzwerkelemente, die öffentlich oder zumindest aus der vereinbarten Startposition des Penetrationstests heraus erreichbar sind. Zielknoten entsprechen den Zielsystemen des Tests, d.h. Systeme, auf denen die kritischen Daten des Kunden vermutet werden. Ein Knoten  $a$  ist mit einem Knoten  $b$  durch eine gerichtete Kante verbunden, sofern die Penetrationstester vermuten, dass man durch die Kompromittierung von  $a$  daraufhin auch  $b$  angreifen kann und diese Kante auf einem Pfad zu einem Zielsystem liegt.

Ein Kompromittierungsgraph heißt *annotiert*, wenn er zusätzliche Informationen zu Knoten und Kanten enthält. Wir betrachten hier als Spezialfall annotierte Graphen mit einer Gewichtungsfunktion  $b : E \rightarrow \mathbb{R}^{>0} \times \mathbb{R}^{>0}$  für jede Kante, d.h.  $b$  liefert für jede Kante ein Tupel von reeller Zahlen  $(p, t)$ . Für eine Kante  $(a, b) \in E$  bedeutet die Gewichtung  $(p, t)$ , dass falls Netzwerkelement  $a$  kompromittiert wurde, dann auch das Netzwerkelement  $b$  mit Erfolgswahrscheinlichkeit  $p$  innerhalb der Zeit  $t$  kompromittiert werden kann. Ein annotierter Kompromittierungsgraph stellt immer ein angenähertes Modell des Netzwerks dar, welches gerade im Rahmen des Penetrationstests untersucht wird. Angelehnt ist dieses Konzept an die *attack trees* von Schneier [17], Seite 318 – 333 sowie die *attack graphs* von Ingols et al. [7], Sheyner et al. [18].

Ein *Kompromittierungspfad* in einem Kompromittierungsgraph  $G = (V, E)$  ist ein Folge  $v_1, v_2, \dots, v_k$  von Knoten aus  $G$  so dass  $v_1 \in S$ ,  $v_k \in Z$  und für alle  $0 < i \leq k$  gilt:  $(v_{i-1}, v_i) \in E$ . Ein Kompromittierungspfad stellt also eine mögliche Folge von erfolgreichen Angriffen dar, die zur Kompromittierung eines der Zielsysteme führen können. Eine solche Kompromittierung kann aber über verschiedene Pfade erfolgen.

Ein bewusst einfach gehaltenes Beispiel für einen solchen annotierten Graphen zeigt Abbildung 3. In diesem Graphen existieren vier Knoten. Es wird vermutet, dass die Clients die kritischen Informationen in einer Datenbank speichern. Somit ist dieses System das Zielsystem des Penetrationstests. Die Clients sind zu einem Knoten zusammengefasst. Da diese nicht als Zielsystem markiert worden sind, sind hier niemals alle kritischen Informationen vorhanden und die Datenbank ist das einzige Ziel des Penetrationstests. Von der Startposition für den Penetrationstest sind lediglich ein Webserver und ein Mailserver möglicherweise direkt kompromittierbar. Bei einer Kompromittierung des Webservers kann der Datenbankserver angegriffen werden. Die Kante vom Datenbankserver zu den Clients existiert nicht, obwohl durch eine Kompromittierung des Datenbankservers sicherlich auch die Clients angegriffen werden könnten. Allerdings wird bei einem erfolgreichen Angriff auf die Datenbank bereits ein Zielsystem erreicht und über diese (nicht existierende) Kante können keine weiteren Zielsysteme mehr erreicht werden. Durch einen erfolgreichen Angriff auf den Mailserver können im nächsten Schritt die Clients angegriffen werden. Sofern diese ebenfalls kompromittiert worden sind, kann daraufhin von dort aus ebenfalls die Datenbank angegriffen werden.

Die Kanten sind annotiert mit der Wahrscheinlichkeit einer erfolgreichen Kompromittierung in Prozent und der hierfür notwendigen Zeit. Die Wahrscheinlichkeit einer erfolgreichen Kompromittierung kann z. B. mit Hilfe von *Attack Surfaces* abgeschätzt werden, die von Manadhata et al. [15] vorgestellt wurden. Nimmt man als Zeiteinheit z.B. Stunden an, so bedeutet die Bewertung von  $(80\%/4)$  der Eingangskante des Webservers, dass vermutet wird, dass der Webserver mit einer Wahrscheinlichkeit von 80% innerhalb von 4 Stunden erfolgreich angegriffen werden kann.

In der Praxis hat sich diese Annotation mit Hilfe von Zeit- und Prozentwerten als gut praktikabel erwiesen, da sich diese Werte durch erfahrene Penetrationstester abschätzen lassen.

## 4.2 Penetrationstests als iterative Graphverfeinerung

Ein Penetrationstest beginnt mit einem ersten annotierten Graph  $G_0$ , der nach der Vorbesprechung mit dem Kunden generiert wird. Naturgemäß kann dieser Graph das zu testende Netzwerk nur sehr grob und unvollständig abbilden. Hauptzweck bei der Erstellung von  $G_0$  ist die Abschätzung der Dauer und des Aufwandes bei einem geplanten Penetrationstest. Die Dauer des Penetrationstests wird daran gemessen, wie lange es vermutlich dauern wird, bis alle Zielsysteme kompromittiert worden sind. Hierfür werden ein oder mehrere Pfade betrachtet, die von einem Startpunkt des Penetrationstests zu den einzelnen Zielsystemen verlaufen. Zwei Pfade können komplett unabhängig sein, aber auch in einem

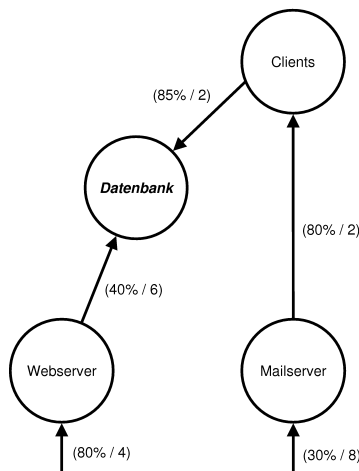


Abbildung 3: Annotierter Kompromittierungsgraph

Teilbereich parallel verlaufen. Dies ist in der Praxis wichtig, da für den zweiten Lauf auf einem Teilpfad eines schon erfolgreichen Laufs keine weitere Zeit (und damit Kosten) entstehen, da die betreffenden Systeme bereits vorher kompromittiert worden sind.

Das vorgestellte Problem der Berechnung der Kosten (und damit hier der Testdauer) ist ein bekanntes Graphenproblem der Informatik, das sogenannte *gerichtete Steinerbaumproblem*. Es gehört leider zur Klasse der NP-schweren Probleme, allerdings gibt es eine Vielzahl von sehr guten Approximationslösungen [4]. Um die oben beschriebenen, annotierten Graphen solchen algorithmischen Lösungen zugänglich zu machen, müssen die aus der Praxis stammenden Tupel-Kantenbewertungen in einzelne (normierte) Werte umgerechnet werden. Als Kostenfunktion  $k$  für eine Kante  $e$  bietet sich hier die Funktion  $k(e) = \alpha \cdot \frac{t}{p}$ , wobei  $t$  und  $p$  aus der zugehörigen Bewertungsfunktion  $b(e)$  stammen und der Faktor  $\alpha$  ein Sicherheitsfaktor ist, der mindestens mit 1 bewertet werden sollte. Falls ein Kunde einen erhöhten Sicherheitsbedarf aufweist oder das Netzwerk wesentlich komplexer eingeschätzt wird, ohne dass dies mittels  $p$  und  $t$  genauer darstellbar ist, kann dieser Faktor beliebig nach oben angepasst werden. Die Unsicherheit der Abschätzung einer definierten Kompromittierung hingegen sollte bereits in den Prozentwert  $p$  einfließen.

Aufbauend auf dem initialen Graphen  $G_0$  wird jetzt mittels wiederholter Iterationen des Vorgehensmodells eine Folge  $G_0, G_1, G_2, \dots$  von Kompromittierungsgraphen aufgebaut, die die Sicht des Penetrationstesters auf das Netzwerk immer weiter präzisieren. Der genaue Zusammenhang zwischen  $G_i$  und  $G_{i+1}$  bleibt vollkommen offen, außer dass die Start- und Zielknoten in beiden Graphen gleich sein sollten.

**Breitensuche in Kombination mit iterativer Kompromittierungspfadanalyse.** Für Penetrationstester und den Kunden ist es wichtig, dass die Zeit, die im Rahmen des Tests budgetiert wurde, sinnvoll genutzt wird. Im Test ergibt es also Sinn, anfangs nach schnellen

Erfolgen zu suchen, um möglichst schnell sehr tief in das Netzwerk einzudringen. Die Penetrationstester starten darum zunächst eine Breitensuche des Graphen und konzentrieren sich anfangs auf die Kanten, die nur eine geringe Zeit für die Verifizierung oder Falsifizierung der Schwachstelle benötigen. Bei einer erfolgreichen Kompromittierung startet in der nächsten Iteration wieder eine neue Breitensuche, da nun neue Systeme sichtbar werden.

Komplexere Schwachstellen benötigen wesentlich mehr Zeit und sind dennoch manchmal unumgänglich, um ein Ziel erreichen zu können. Werden zunächst alle Schwachstellen untersucht, deren Untersuchung kürzer dauert, als die der komplexeren Schwachstellen, so steht am Ende möglicherweise nicht mehr genügend Zeit für diese zur Verfügung. Aus diesem Grund spielt während des Testzeitraums der erwartete Erfolg einer Kompromittierung eine immer größer werdende Rolle bei der Auswahl des nächsten Schritts. Der erwartete Erfolg besteht einerseits in der vermuteten Zielnähe und andererseits auch aus der Kompromittierungswahrscheinlichkeit  $p$ .

Man kann bei der Berechnung des Aufwandes für einen Penetrationstest diese Beobachtung in die Methodik wie folgt einfließen lassen: Es wird ein zusätzlicher Faktor  $z$  eingeführt, der den Zeitverlauf des Penetrationstests repräsentiert. Zu Beginn des Test ist  $z = 0$  und zum Ende des Penetrationstest ist  $z = 1$ . Der Faktor  $z$  verläuft während des Penetrationstests linear steigend. Die Kosten einer Kante während des Tests zu einem Zeitpunkt  $z$  berechnen sich dann wie folgt:

$$\textit{kosten}_z(E) = \alpha \cdot \frac{t}{p^z}$$

Die Wahrscheinlichkeiten einer erfolgreichen Kompromittierung spielen also zu Beginn des Penetrationstests kaum eine Rolle, lediglich die Dauer des Tests geht in die Bewertung der Kante ein. Während des Penetrationstests gewinnt die Wahrscheinlichkeit exponentiell steigend mehr Einfluss auf den Penetrationstest.

### 4.3 Diskussion

Bei einem Penetrationstest steht ein Test des Netzwerks als Gesamtsystem im Vordergrund. Ein genauer Testablauf ist vor Testbeginn nicht planbar. Lediglich durch die Identifizierung von relevanten Zielen und der Testdauer ist die Richtung des Penetrationstest steuerbar. Durch das vorgestellte Vorgehen werden zwei Ziele vereint:

- Eine möglichst breite Sichtweise garantiert das Aufdecken von vielen verschiedenen Schwachstellen.
- Das Erreichen der vereinbarten Ziele garantiert die gewünschte Testtiefe.

Diese Ziele stehen scheinbar konträr zueinander: Durch eine breite Sichtweise scheint Zeit in Pfade investiert zu werden, die nicht für die Zielerreichung notwendig sind. Umgekehrt scheint das Erreichen eines Ziels immer optimalerweise auf einem direkten Pfad (bei mehreren Zielen im Sinne der Kompromittierungspfade) zu erfolgen.

Doch gerade durch die breite Sichtweise auf ein Netzwerk werden neue Kanten und Pfade überhaupt erst sichtbar, und daher ist diese Sichtweise auch für das zweite Ziel hilfreich, da so Ziele sogar schneller erreicht werden können. Gerade am Anfang eines Penetrationstests ist es also sehr wichtig, mit Hilfe einer Breitensuche vorzugehen. Die Berechnung der Kantenkosten mit Hilfe der  $kosten_z$ -Funktion, in der die abgeschätzte Wahrscheinlichkeit einer erfolgreichen Kompromittierung erst im Testverlauf exponentiell ihre eigentliche Gewichtung erhält, stellt diese Testbreite sicher. So werden am Anfang der Testzeit gerade solche Schwachstellen aufgedeckt, die für den Penetrationstester wie auch den Auftraggeber überraschend sind und sich damit oft als sehr gefährlich herausstellen. Gleichzeitig wird sich der Penetrationstester im Verlauf des Tests immer mehr auf das zweite Ziel, das Erreichen der vereinbarten Testziele, fokussieren. Somit ist eine Methodik gefunden worden, die, für eine vereinbarte Gesamtdauer, einen optimalen Testverlauf im Sinne der beiden Teilziele sicherstellt, da die gewünschte Testtiefe mit hoher Wahrscheinlichkeit erreicht wird und gleichzeitig eine sehr breite Sichtweise möglichst viele Ansätze liefert.

Die oben beschriebene Methodik enthält auch eine Metrik, die das Abschätzen des Aufwandes eines Penetrationstests erlaubt. Über die Kompromittierungspfadanalyse ist nun ein Modell gefunden worden, das Grundlage sein kann, die in der Praxis nur auf Erfahrungen basierenden Abschätzungen zumindest teilweise zu festigen. Erfahrungen und ein gewisses Gefühl für Netzwerke und mögliche Fehler stellen nach wie vor die Basis aller Abschätzungen dar. Im Falle der Kompromittierungspfadanalyse müssen zunächst mögliche Pfade sinnvoll vermutet und dann bewertet werden. Werden hier grobe Fehler gemacht, so ist die komplette Abschätzung falsch.

Die Tatsache, dass die Abschätzung der Gesamtdauer NP-schwer ist, schränkt aufgrund guter Approximationsalgorithmen die Praktikabilität der Methode nicht ein. Zudem gibt es in der Praxis normalerweise nur sehr wenige verschiedene Zielknoten. Dies ist insofern auch sinnvoll, da ein solches Ziel nur als grobe Richtung für den Penetrationstest dient. Da die genauen Schwachstellen und damit auch die Auswirkungen dieser Schwachstellen nicht vor Testbeginn bekannt sind, können diese *Ziele* auch erst während des Penetrationstests identifiziert werden. In der Praxis ist die Anzahl der vereinbarten Ziele normalerweise kleiner als fünf, im Schnitt liegt sie sogar zwischen einem und zwei Zielen.

Der zweite Faktor, der die Komplexität der Abschätzung ausmacht, ist die Knotenanzahl. In der Praxis ist das Bild des Netzwerks, welches von den Penetrationstestern vor Testbeginn entworfen wird, noch sehr unscharf. Im Groben basiert es komplett auf den Angaben des Auftraggebers des Penetrationstests, d.h. es wird normalerweise nur eine überschaubare Anzahl von Diensten geben. Die Penetrationstester können insofern auch nur wenige Kanten hinzufügen und bewerten. Eine zu detailreiche Sicht ist hier nicht sinnvoll und auch normalerweise nicht leistbar, insbesondere da oft auch der Auftraggeber sein eigenes Netzwerk nicht tiefgehend kennt. Wird eine wesentlich höhere Komplexität vermutet, ohne dass diese genauer spezifiziert und damit in den Graphen einfließen kann, so kann dies durch eine Anpassung des Faktors  $\alpha$  berücksichtigt werden.

Das vorgestellte Verfahren basiert auf den Erfahrungen in der Praxis der Firma RedTeam Pentesting. In weiteren Arbeiten planen wir, die Methode noch genauer empirisch zu evaluieren. Idealerweise wird hierbei ein echtes Netzwerk von mehreren unabhängigen Penetrationstestern untersucht, deren Ergebnisse miteinander verglichen werden. Leider haben

in der Regel weder Penetrationstestunternehmen noch beauftragende Firmen ein großes Interesse daran, derartige Daten zur Veröffentlichung freizugeben. Hier besteht also noch Bedarf an Überzeugungsarbeit. Denkbar ist allerdings auch ein Projekt im Rahmen eines studentischen Praktikums, vergleichbar etwa mit der Arbeit von Jonsson and Olovsson [9]. Gerade die Metrik ist für die Praxis von Penetrationstests wertvoll, da das oft stark von Kreativität geprägte Vorgehen von Penetrationstestern scheinbar konträr zu einer festen Testdauer und damit einem festen Testbudget für den Auftraggeber steht. Doch mit offenem Budget werden in der Praxis nur sehr wenige Penetrationstests beauftragt. Die Kompromittierungspfadanalyse stellt eine Möglichkeit für Penetrationstester dar, eine solche oft notwendige Abschätzung treffen zu können.

## Literatur

- [1] Nikita Borisov, Ian Goldberg, and David Wagner. Intercepting mobile communications: The insecurity of 802.11. <http://www.isaac.cs.berkeley.edu/isaac/wep-faq.html>, 2001.
- [2] Bundesamt für Sicherheit in der Informationstechnik. Studie Durchführungskonzept für Penetrationstests. <http://www.bsi.de/literat/studien/pentest/penetrationstest.pdf>, 2003.
- [3] CIRT.net. Nikto. <http://www.cirt.net/code/nikto.shtml>, 2007.
- [4] Siavash Vahdati Daneshmand. *Algorithmic Approaches to the Steiner Problem in Networks*. PhD thesis, Universität Mannheim, 2004. <http://madoc.bib.uni-mannheim.de/madoc/volltexte/2004/176/>.
- [5] Dan Farmer and Wietse Venema. Improving the security of your site by breaking into it. <ftp.win.tue.nl/pub/security/admin-guide-to-cracking-101.Z>, December 1993.
- [6] Pete Herzog. Open-Source Security Testing Methodology Manual, Version 2.1. <http://www.isecom.org/osstmm/>, August 2003.
- [7] Kyle Ingols, Richard Lippmann, and Keith Piwowarski. Practical attack graph generation for network defense. In *ACSAC*, pages 121–130. IEEE Computer Society, 2006.
- [8] ISACA Switzerland. Sicherheitsüberprüfung von IT-Systemen mit Hilfe von Tiger Teams. <http://www.isaca.ch/files/tigerteam.pdf>, 1999.
- [9] Erland Jonsson and Tomas Olovsson. A quantitative model of the security intrusion process based on attacker behavior. *IEEE Transactions on Software Engineering*, 23(4), 1997.
- [10] T. J. Klevinsky, Scott Laliberte, and Ajay Gupta. *Hack I.T.: Security Through Penetration Testing*. Addison-Wesley, 2002.
- [11] Kevin Lam, David LeBlanc, and Ben Smith. *Assessing Network Security*. Microsoft Press, 2004.
- [12] Johnny Long, Ed Skoudis, and Alrik van Eijkelenborg. *Google Hacking for Penetration Testers*. Syngress Publishing, 2004.
- [13] Johnny Long, Aaron W. Bayles, James C. Foster, Chris Hurley, Mike Petruzzi, Noam Rathaus, and Mark Wolfgang. *Penetration Tester's Open Source Toolkit*. Syngress Publishing, 2006.
- [14] Gordon Lyon. Top 100 Security Tools. <http://www.sectools.org>, 2006.
- [15] Pratyusa Manadhata, Jeannette Wing, Mark Flynn, and Miles McQueen. Measuring the attack surfaces of two ftp daemons. In *QoP '06: Proceedings of the 2nd ACM workshop on Quality of protection*, pages 3–10. New York, NY, USA, 2006. ACM Press.

- [16] Enno Rey, Michael Thumann, and Dominick Baier. *Mehr IT-Sicherheit durch Pen-Tests*. Vieweg Verlag, 2005.
- [17] Bruce Schneier. *Secrets & Lies - Digital Security in a Networked World*. John Wiley & Sons, Inc., New York, NY, USA, 2000.
- [18] Oleg Sheyner, Joshua W. Haines, Somesh Jha, Richard Lippmann, and Jeannette M. Wing. Automated generation and analysis of attack graphs. In *IEEE Symposium on Security and Privacy*, pages 273–284, 2002.
- [19] Tenable Network Security. Nessus. <http://www.nessus.org>, 2007.
- [20] James S. Tiller. *The Ethical Hack: A Framework for Business Value Penetration Testing*. Auerbach Publications, Boston, MA, USA, 2003.
- [21] John Wack, Mils Tracy, and Souppaya Murugiah. Guideline on network security testing. <http://csrc.nist.gov/publications/nistpubs/800-42/NIST-SP800-42.pdf>, October 2003.
- [22] Andrew Whitaker and Daniel P. Newman. *Penetration Testing and Network Defense*. Cisco Press, 2005.