Unverkäufliche Leseprobe

Dieses Buch ist der unveränderte Reprint einer älteren Ausgabe.

Erschienen bei FISCHER Digital

Printed in Germany
ISBN 978-3-596-30722-7

Fischer

Weiterführende Informationen finden Sie unter
www.fischerverlage.de

Als neu eingestellter Systemmanager am Lawrence Berkeley Laboratory in Kalifornien mußte Clifford Stoll einen Abrechnungsfehler von 75 Cent für in Anspruch genommene, aber nicht bezahlte Computerarbeitszeit überprüfen. Dies bereitete ihm um so mehr Kopfzerbrechen, als er bei dieser Überprüfung auf die Spur von Hackern stieß, denen es gelungen war, in seine Datennetze einzudringen. Datennetze, die von hochgeheimen Militärunterlagen bis zum bargeldlosen Zahlungsverkehr alles mögliche verwalten.
Stolls Warnungen an FBI-Bürokraten in Washington fruchteten nichts. Auf eigene Faust verfolgte er die Hacker nun durch die Datennetze. Dabei erfährt der Leser auf höchst spannende und anschauliche Weise, wie man durch Löcher im elektronischen Zaun schlüpft, in Computer einbricht, digitale Fallen stellt und seine eigenen Daten besser schützt.
Aber auch die Hacker waren clever und ihrem Verfolger meist um eine atemberaubende Nasenlänge voraus. Ein Jahr dauerte es, bis Clifford Stoll sie nach einer digitalen Reise quer durch Nordamerika und Europa in Hannover lokalisieren konnte.
Und in der Tat stellte sich heraus, daß die Hacker hauptsächlich militärische Geheimnisse der Amerikaner ausgeforscht hatten – im Auftrag des KGB.
Eine authentische Geschichte, die wieder einmal beweist, daß die Wirklichkeit viel sensationeller sein kann als jede Fiktion. Dementsprechend war auch das Medienecho, von FAZ bis taz und von Spiegel bis Stern, und natürlich bei allen Radio- und Fernsehsendern.

Clifford Stoll hatte ursprünglich Astronomie studiert und ist eher durch Zufall zum Computerexperten geworden. Heute ist er eine anerkannte Autorität in Fragen des Datenschutzes und der Computersicherheit – mit Sicherheit eines der brisanten Probleme des kommenden Jahrzehnts. Immer wieder wird Stoll als Experte von wichtigen amerikanischen Behörden und Gremien, bis hin zu Senatsausschüssen gehört. Er arbeitet am Harvard-Smithsonian Center für Astrophysics und lebt in Cambridge bei Boston in Massachusetts. Clifford Stolls neuestes Buch ›Die Wüste Internet‹ ist soeben im S. Fischer Verlag erschienen.

Clifford Stoll
KUCKUCKSEI

Die Jagd auf die deutschen Hacker, die das Pentagon knackten

Aus dem Amerikanischen
von Gabriele Herbst

Fischer
Taschenbuch
Verlag

Die Übersetzerin dankt Walter Sielski, Ernst Guggolz, Jack Wolfinger und Michael Hiller für Rat und Tat.

33.–37. Tausend: April 1996

Von Clifford Stoll autorisierte deutsche Übertragung.

Veröffentlicht im Fischer Taschenbuch Verlag GmbH,
Frankfurt am Main, Februar 1990

Lizenzausgabe mit freundlicher Genehmigung
des Wolfgang Krüger Verlages, Frankfurt am Main
Die Originalausgabe »The Cuckoo's Egg. Inside
the World of Computer Espionage« erschien im
Verlag Doubleday, New York

Graphiken: Stefan Birker, Viernheim,
nach Entwürfen von Clifford Stoll
Druck und Bindung: Clausen & Bosse, Leck
Printed in Germany
ISBN 3-596-10277-4

Gedruckt auf chlor- und säurefreiem Papier

Kuckucksei

1. Kapitel

Ich ein Computercrack? Bis vor einer Woche war ich noch ein Astronom gewesen, der ganz zufrieden Teleskop-Optiken konstruierte. Wenn ich darauf zurückblickte, hatte ich in einem akademischen Traumland gelebt. Und während all dieser Jahre hatte ich nie für die Zukunft geplant, bis zu dem Tag, an dem mein Forschungsauftrag auslief.

Zu meinem Glück recyclete mein Labor gebrauchte Astronomen. Statt stempeln zu gehen, wurde ich vom Keck Observatorium am Lawrence Berkeley Laboratory (LBL) runter ins Rechenzentrum im Kellergeschoß desselben Gebäudes verfrachtet.

Also, verdammt nochmal, ich konnte den Computercrack so gut mimen, daß die Astronomen immer beeindruckt waren, dann würde ich wohl auch hier bald so gut mithalten können, daß meine Kollegen mir nicht auf die Schliche kämen. Denn – ich, ein Computercrack? Nein – ich bin Astronom.

Und was jetzt? Als ich apathisch auf mein Computerterminal starrte, dachte ich immer noch an Planetenumlaufbahnen und Astrophysik. Für eine Weile schuf mein Miesepeter-Rückzug in mich selbst noch Distanz zu meiner neuen Welt.

Als Neuer in diesem Haufen hatte ich die Wahl zwischen einer Besenkammer mit Fenster und Aussicht auf die Golden Gate Bridge und einem Büro ohne Belüftung, aber mit einer Wand voller Bücherregale. Ich schluckte meine Platzangst runter und nahm das Büro, in der Hoffnung, es würde niemandem auffallen, wenn ich unter dem Schreibtisch schlief. In den Büros nebenan saßen zwei Systemleute, Wayne Graves und Dave Cleveland, alte Hasen auf ihrem Gebiet. Ich sollte meine Nachbarn bald durch ihre Streiterei kennenlernen.

Wayne hielt alle anderen für inkompetent oder faul und lag daher mit der übrigen Mannschaft über Kreuz. Trotzdem kannte er das System durch und durch, vom Plattencontroller bis zu den Mikrowellenantennen. Wayne war eingeschworen auf VAX-Computer von Digital Equipment Corporation (DEC), dem nach IBM zweitgrößten Computerhersteller in der Welt, und akzeptierte nichts anderes.

Dave, unser heiterer Unix-Buddha, lauschte geduldig Waynes ununterbrochenem Strom von Computervergleichen. Kaum ein Gespräch gipfelte nicht in Waynes Satz: »Die VAX ist bei allen Wissenschaftlern der Computer Nummer 1, und man kann mit ihm auf tausend Arten mächtige Programme entwickeln.«

Dave erwiderte stets geduldig: »Okay, halte du deine VAX-Süchtigen bei Laune, und ich kümmere mich um den Rest der Welt.« Dave gab ihm nie die Genugtuung, sich zu ärgern, und Waynes Beschwerden verebbten schließlich in unverständlichem Genöle.

Na, großartig. Erster Arbeitstag, eingeklemmt zwischen zwei Typen, die meine Tagträume mit ihren ewig gleichen Disputen wie Seifenblasen platzen ließen.

Wenigstens würde sich niemand über mein Äußeres beschweren. Ich trug die Berkeley-Standarduniform: kariertes Hemd, abgewetzte Jeans und billige Latschen. Gelegentlich trug ein Systemverwalter (oder auch Systemmanager genannt), eine Krawatte, aber an diesen Tagen sank gewöhnlich die Produktivität.

Wayne, Dave und ich sollten gemeinsam die Computer als Dienstleistungsanlage für das gesamte Labor betreuen. Wir verwalteten ein Dutzend Zentralrechner – riesige Arbeitspferde zur Lösung physikalischer Probleme, die zusammen rund sechs Millionen Dollar wert waren. Den Wissenschaftlern, die diese Computer benutzten, sollte ein einfaches, leistungsfähiges Rechnersystem zur Verfügung stehen, das so zuverlässig war wie die Elektrizitätsgesellschaft. Das hieß, die Maschinen mußten die ganze Zeit laufen, rund um die Uhr. Und wie jede andere Service-Firma stellten wir jede Benutzung in Rechnung.

Von den viertausend Labormitarbeitern nutzte vielleicht ein Viertel die Zentralrechner. Jedes dieser tausend Konten wurde täglich aufsummiert, und der Computer führte ein elektronisches Haupt-

buch. Weil eine Stunde Rechenzeit immerhin 300 Dollar kostete, mußte unsere Buchhaltung genau arbeiten, also verzeichneten wir jede ausgedruckte Seite, jeden Block Plattenspeicherplatz und jede Minute Prozessorzeit. Ein eigener Computer sammelte diese Zahlen und sandte monatliche Rechnungen an die Laborabteilungen.

Und so geschah es, daß Dave an meinem zweiten Arbeitstag in mein Büro marschierte und etwas von einem Schluckauf im Unix-Abrechnungssystem murmelte. Irgend jemand mußte ein paar Sekunden Rechenzeit verbraucht haben, ohne dafür zu bezahlen. Die Computerbücher gingen nicht ganz auf: Die letzte Monatsrechnung über 2387 Dollar wies ein Defizit von 75 Cents aus. Nun ist ein Fehler von ein paar Tausend Dollar offensichtlich und nicht schwer zu finden. Aber Fehler in der Cent-Spalte stammen von tiefverborgenen Problemen; sie aufzudecken ist deshalb eine Herausforderung für jeden sich mausernden Softwarecrack. Dave meinte, ich solle mal darüber nachdenken.

»Astreiner Raub, was?« fragte ich.

»Krieg's raus, Cliff, und alle werden staunen«, sagte Dave.

Das sah ganz nach einer netten Spielerei aus, also vergrub ich mich in das Abrechnungsprogramm.

Ich stellte sehr bald fest, daß unsere Abrechnungssoftware ein Flickenteppich aus Programmen war, die längst entschwundene Werkstudenten geschrieben hatten. Jedenfalls funktionierte der Eintopf gut genug, so daß sich niemand darum kümmerte. Dann sah ich mir die Programm-Mixtur genauer an; sie war in Assembler, Fortran und Cobol geschrieben, den ältesten aller Computersprachen. Hätte auch klassisches Griechisch, Latein oder Sanskrit sein können.

Wie bei der meisten Software ›Marke Eigenbau‹ hatte sich niemand die Mühe gemacht, unser Abrechnungssystem zu dokumentieren. Nur ein Irrer würde seine Nase ohne Karte in solch ein Labyrinth stecken.

Aber es war ein Zeitvertreib für den Nachmittag und eine Gelegenheit, das System kennenzulernen. Dave zeigte mir, wie es, immer wenn sich jemand bei dem Computer anmeldete, den Benutzernamen und das Terminal speicherte. Es versah jede Verbin-

dung mit der Uhrzeit und zeichnete auf, welche Aufgaben er durchführen ließ, wie viele Sekunden Prozessorzeit er benötigte und wann er sich abmeldete.
Dave erklärte, daß wir zwei unabhängige Abrechnungssysteme hätten. Die normale Unix-Abrechnungssoftware speicherte nur die datierten Aufzeichnungen in einer Datei. Um aber die Bedürfnisse von ein paar Bürokraten zu befriedigen, die wissen wollten, welche Abteilungen die Computer benutzten, hatte Dave ein zweites Abrechnungssystem installiert, das detailliertere Aufzeichnungen über die Computerbenutzer machte.
Im Lauf der Jahre hatte eine lange Reihe gelangweilter Werkstudenten Programme geschrieben, um diese ganzen Abrechnungsinformationen zu analysieren. Ein Programm sammelte die Daten und legte sie in einer Datei ab. Ein zweites Programm las die Datei und berechnete die Kosten für den jeweiligen Zeitraum. Und ein drittes sammelte all diese Kosten und druckte Rechnungen aus, die an jede Abteilung geschickt wurden. Das letzte Programm addierte alle Benutzergebühren auf und verglich das Gesamtergebnis mit dem Ergebnis des computerinternen Abrechnungsprogramms. Und zwei Abrechnungsdateien, die von verschiedenen Programmen parallel geführt wurden, sollten eigentlich dasselbe Ergebnis erbringen.
Ein Jahr lang hatte es keine Differenzen gegeben, diese Woche aber war etwas nicht ganz in Ordnung. Die naheliegende Erklärung: ein Rundungsfehler. Wahrscheinlich war jeder Abrechnungsposten korrekt; wurden sie aber addiert, summierten sich Differenzen von Zehntel-Cents bis zu einem Fehler von 75 Cents auf. Ich sollte in der Lage sein, dies zu beweisen, indem ich entweder analysierte, wie die Programme arbeiteten, oder indem ich sie mit verschiedenen Daten testete.
Statt mir den Code jedes Programms mühsam zu entschlüsseln, schrieb ich kurzerhand ein Programm zur Kontrolle der Dateien. In ein paar Minuten hatte ich das erste Programm geprüft: Es sammelte die Abrechnungsdaten wirklich korrekt. Hier gab's keine Probleme.
Zur Simulation des zweiten Schrittes brauchte ich länger, aber in einer Stunde hatte ich ein ausreichendes *ad-hoc*-Programm zu-

sammengeklopft, um zu beweisen, daß auch das zweite Programm richtig funktionierte. Es addierte einfach die Zeitintervalle auf und multiplizierte sie mit den Kosten für die Rechenzeit. Also lag der 75-Cent-Fehler nicht an diesem Programm.
Auch das dritte Programm arbeitete perfekt. Es sah in der Liste der autorisierten Benutzer nach, fand ihre Laborkonten und druckte eine Rechnung aus. Rundungsfehler? Nein, jedes der Programme verzeichnete das Geld bis auf den Hundertstel Cent. Kumulative Fehler würden bei den Zehntel-Cents auftreten. Seltsam. Woher kam dann dieses 75-Cent-Defizit?
Ich hatte nun bereits einige Stunden in den Versuch investiert, ein triviales Problem zu verstehen. Und ich wurde stur: Verdammt, ich würde bis Mitternacht hierbleiben, wenn's sein mußte.
Nach einigen weiteren Testprogrammen fing ich an, dem Mischmasch der hausgemachten Abrechnungsprogramme wirklich zu vertrauen. Keine Frage, die Rechnungen gingen nicht auf, aber es war sicher kein Rundungsfehler, und die Programme waren zwar nicht kugelsicher, aber sie verschlampten keinen Cent. Ich hatte auch die Listen der autorisierten Benutzer gefunden und fand heraus, wie die Programme die Datenstrukturen nutzten, um den verschiedenen Abteilungen Rechnungen auszustellen. Gegen 19 Uhr fiel mir ein Benutzer namens *Hunter* auf. Dieser Typ hatte keine gültige Rechnungsadresse.
Ha! *Hunter* hatte im letzten Monat für 75 Cents Rechenzeit verbraucht, aber niemand hatte für ihn bezahlt. Er war die Quelle unseres Defizits! Jemand hatte Mist gebaut, als er unserem System diesen Benutzer anhängte. Ein triviales Problem, verursacht durch einen trivialen Fehler.
Ein Grund zum Feiern. Als ich diesen kleinen Triumph auf die ersten Seiten meines Notizbuchs schrieb, kreuzte Martha, meine Freundin, auf, und wir feierten die Sache mit einem späten Cappuccino im CAFÉ ROMA.
Ein richtiger Computercrack hätte das Problem in ein paar Minuten gelöst. Für mich war's unbekanntes Terrain, und ich hatte einige Zeit gebraucht, um mich darin zurechtzufinden. Ich konnte mich damit trösten, das Abrechnungssystem kennenge-

lernt und mich in ein paar obsoleten Sprachen geübt zu haben. Am nächsten Tag schickte ich eine elektronische Nachricht an Dave und erklärte ihm das Problem, wobei ich mich gehörig aufplusterte.
Mittags kam Dave vorbei, um einen Berg Manuals abzuladen und erwähnte beiläufig, er habe nie einen Benutzer namens *Hunter* zugelassen. Es müsse einer der anderen Systemverwalter gewesen sein.
Waynes trockener Kommentar: »Ich war's nicht. LDVM.«
Die meisten seiner Sätze endeten mit Akronymen, dieses bedeutete: »Lies das verdammte Manual.«
Aber ich las die Manuals nicht. Die Operator durften keinen neuen Benutzer ohne ein Konto zulassen. In anderen Rechenzentren loggt man sich einfach in ein privilegiertes Konto ein und sagt dem System, es solle einen neuen Benutzer hinzufügen. Weil wir auch verschiedene Buchhaltungseinträge vornehmen mußten, konnten wir kein solches Larifari-System betreiben. Unseres war so komplex, daß wir spezielle Programme besaßen, die automatisch den Papierkram erledigten und mit den Systemen jonglierten. Auf Nachfrage meinten die Operator übereinstimmend, das automatische System sei so gut, daß niemand von Hand einen neuen Benutzer einführen könne. Und das automatische System würde keinen solchen Fehler begehen.
Offen gesagt, ich konnte mir nicht vorstellen, wer sich diesen Witz erlaubt hatte. Niemand kannte *Hunter*, und es gab kein Konto für ihn. Also löschte ich den Namen aus dem System – wenn er auftauchte, um sich zu beschweren, konnten wir ihn ja richtig installieren.

Einen Tag später schickte uns ein obskurer Computer namens *Dockmaster* eine elektronische Nachricht. Sein Systemverwalter behauptete, jemand aus unserem Labor habe am Wochenende versucht, in seinen Computer einzubrechen.
Die Antwortadresse von *Dockmaster* hätte überall sein können, die Anzeichen wiesen aber auf Maryland. Die Nachricht war durch ein Dutzend anderer Computer gelaufen, und jeder hatte einen ›Eingangsvermerk‹ hinterlassen.

Dave beantwortete die Nachricht mit einem unverbindlichen »Wir sehen's uns mal an.«
Sicher. Wir würden's uns ansehen, wenn wir unsere anderen Probleme gelöst hatten.
Unsere Laborcomputer stehen über ein Dutzend Netzwerke mit Tausenden anderer Systeme in Verbindung. Jeder unserer Wissenschaftler kann sich in unseren Computer einloggen und sich dann bei einem entfernten Computer anmelden. Steht die Verbindung einmal, kann er sich in den entfernten Computer einloggen, wenn er einen Kontennamen und ein Passwort eingibt. Im Prinzip ist das einzige, was einen Computer im Netzwerk schützt, das Passwort, weil man Kontennamen leicht rausfinden kann. (Wie man sie findet? Man schaut einfach ins Telefonbuch – die meisten Leute verwenden ihre Namen für den Computer.)
Die elektronische Nachricht von *Dockmaster* war ungewöhnlich, und Dave übermittelte sie mit der Frage: »Wer ist Dockmaster?« an Wayne, der sie an mich weiterreichte; er vermutete, es handelte sich um »ein Mitglied von FDIC« – das mußte irgendeine Bank sein. Aber sind Banken das einzige, in das es sich lohnt, einzubrechen?
Ich hielt *Dockmaster* eher für irgendeine Flottenbasis. Das Ganze war nicht sonderlich wichtig, schien aber doch wert, daß man sich ein paar Minuten damit beschäftigte.
Die Nachricht enthielt Datum und Uhrzeit des Versuchs von irgend jemandem an unserem Unix-Computer, sich in den *Dockmaster*-Computer einzuloggen. Weil ich gerade am Abrechnungssystem herumhantiert hatte, wußte ich, wo ich nachforschen mußte, um herauszubekommen, wer unsere LBL-Computer am Samstagmorgen um 8.46 Uhr benutzt hatte. Wieder stimmten die beiden Abrechnungssysteme nicht überein. Die Unix-Hauptabrechnungsdatei wies einen Benutzer namens *Sventek* auf, der sich um 8.25 Uhr eingeloggt, eine halbe Stunde nichts getan und sich dann abgemeldet hatte. Dazwischen keine mit Uhrzeit versehene Aktivität. Unsere hausgemachte Software zeichnete Sventeks Aktivität ebenfalls auf, zeigte aber, daß er die Netzwerke von 8.31 Uhr bis 9.01 Uhr benutzte. An diesem Samstagmorgen war nichts los gewesen, niemand sonst hatte Rechenzeit verbraucht.

Oje. Noch ein Abrechnungsproblem. Die Zeitmarkierungen stimmten nicht überein; ein System verzeichnete Aktivität, als das andere meldete, alles sei ruhig. Ich fing gerade erst an, mich in diesem Gebiet zurechtzufinden, und andere Dinge schienen dringender, also ließ ich das Problem auf sich beruhen. Nachdem ich bereits einen Nachmittag damit vergeudet hatte, dem Fehler eines Operators nachzujagen, wollte ich das Abrechnungssystem nicht noch einmal anfassen.

Beim Mittagessen mit Dave erwähnte ich, ein gewisser *Sventek* sei der einzige gewesen, der eingeklinkt war, als *Dockmaster* den Einbruch meldete.

Dave riß die Augen auf und sagte: »Sventek? Joe Sventek? Der ist doch in Cambridge! Cambridge, England. Was macht der denn wieder hier?«

Er erklärte mir, daß Joe Sventek der Unix-Guru des Labors ge wesen war und im Lauf der letzten zehn Jahre ein Dutzend größere Programme geschrieben hatte. Joe war vor einem Jahr nach England gegangen und hatte über der ganzen Computergemeinde Kaliforniens einen strahlenden Heiligenschein zurückgelassen. Dave konnte nicht glauben, daß Sventek zurück sei, weil keiner von seinen anderen Freunden von ihm gehört hatte.

»Er muß von irgendeinem Netzwerk aus in unseren Computer gekommen sein«, mutmaßte Dave.

»Du glaubst also, Joe ist schuld an diesem Problem?« fragte ich.

»Auf keinen Fall«, gab Dave zurück. »Joe ist ein Hacker der alten Schule. Ein cleverer, schneller, fähiger Programmierer. Keiner von diesen bekifften Punkern, die das Wort ›Hacker‹ in Verruf gebracht haben. Jedenfalls würde er nicht versuchen, in irgendeinen Computer in Maryland einzubrechen. Und hätte er's doch versucht, dann hätte er's geschafft, *ohne* eine Spur zu hinterlassen.«

Seltsam: Joe Sventek war seit einem Jahr in England. Trotzdem tauchte er früh am Samstagmorgen auf, versuchte in einen Computer in Maryland einzubrechen, meldete sich ab und hinterließ ein unausgeglichenes Abrechnungssystem. Im Korridor erzählte ich das Wayne, der gehört hatte, Joe sei in England auf Urlaub

und er hätte sich irgendwo in Dartmoor vergraben, weit weg von allen Computern.
»Vergiß diese Nachricht von *Dockmaster*, Cliff. Sventek soll JSB nach Berkeley kommen und kann dann alles aufklären.«
JSB? Jetzt sehr bald. Waynes Art zu sagen: »Ich bin nicht sicher, wann.«
Mein Interesse galt aber nicht Sventek. Es galt den unausgeglichenen Konten. Warum hielten die beiden Abrechnungssysteme verschiedene Zeiten? Und warum wurde eine Aktivität in einer Datei vermerkt, ohne in der anderen aufzutauchen?
Am Nachmittag kehrte ich zum Abrechnungssystem zurück. Ich fand heraus, daß die fünfminütige Zeitdifferenz zwischen den Zeitmarkierungen sich daraus ergeben hatte, daß unsere verschiedenen Computeruhren im Lauf der Monate voneinander abgewichen waren. Eine unserer Computeruhren ging jeden Tag ein paar Sekunden nach ...
Aber es hätten doch alle Aktivitäten von Sventek in beiden Listen auftauchen müssen. Stand diese Unstimmigkeit in Zusammenhang mit dem Abrechnungsproblem von letzter Woche? Hatte ich etwas durcheinandergebracht, als ich darin herumpfuschte? Oder gab es noch eine andere Erklärung?