Schriften zum Medien- und Informationsrecht 50

Daniel Wittig

Die produzentenrechtlichen Verkehrssicherungspflichten von Softwareproduzenten

Nomos

Schriften zum Medien- und Informationsrecht

herausgegeben von
Prof. Dr. Boris P. Paal, M.Jur.

Band 50

Daniel Wittig

Die produzentenrechtlichen Verkehrssicherungspflichten von Softwareproduzenten

Nomos

Onlineversion
Nomos eLibrary

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

Zugl.: Münster, Univ., Diss., 2020

ISBN 978-3-8487-7042-7 (Print)
ISBN 978-3-7489-1091-6 (ePDF)

D6

1. Auflage 2021

Vorwort

Die vorliegende Arbeit wurde im August 2020 von der rechtswissenschaftlichen Fakultät der Westfälischen Wilhelms-Universität Münster (WWU) als Dissertation angenommen. Sie entstand neben meiner beruflichen Tätigkeit als Rechtsanwalt in einer mittelständischen Wirtschaftskanzlei in Paderborn und unter Betreuung durch meinen Doktorvater Herrn Prof. Dr. Hoeren, dem Direktor des Instituts für Informations-, Telekommunikations- und Medienrecht (ITM) an der WWU. Ihm Danke ich u. a. für seine teils harsche Kritik, die zur Fokussierung der Arbeit auf das Wesentliche führte und so maßgeblich zum Gelingen dieser beitrug. Ebenfalls danke ich Herrn Prof. Dr. Fabian Gieseke vom Lehrstuhl Maschinelles Lernen und Data Engineering für die rasche Erstellung des Zweitgutachtens. Mein Dank gilt auch meinen Arbeitskollegen, die Rücksicht auf meine nebenberufliche Promotion nahmen und mich nach Möglichkeit unterstützen.

Während meiner Zeit im Referendariat wurde ich von einem meiner Mentoren auf die Problematik der Haftung von Softwareproduzenten aufmerksam gemacht. Ich musste feststellen, dass wir die für die Haftung wesentlichen Verkehrssicherungspflichten kaum bestimmen konnten. Ab diesem Zeitpunkt stand für mich fest, dass ich einen wissenschaftlichen Beitrag zu diesem Thema leisten wollte, um an der Lösung des Problems mitzuwirken.

Mein ganz besonderer Dank geht an meine Eltern, Petra und Werner Tanger, sowie an meine Ehefrau Katarina. Meine Eltern unterstützten stets vorbehaltlos meinen Lebensweg und die Entscheidungen, die ich auf diesem getroffen habe. Sie ermöglichten mir meine Ausbildung, welche die Basis für meine persönliche und berufliche Entwicklung geworden ist. Nur durch Sie konnte ich mich immer auf meine Ziele konzentrieren, ohne mir Sorgen machen zu müssen. Meine Frau Katarina, welche seit dem Jahr 2016 an meiner Seite ist, gibt mir darüber hinaus in jeder Lebenslage ein unglaubliches Maß an Unterstützung, Rückhalt und Liebe. Ohne ihren Zuspruch und ihr Durchhaltevermögen hätte ich diese Arbeit wohl nicht vollenden können. All dies hat im wesentlichen Maße zum Gelingen dieser Arbeit beigetragen.

Paderborn, im November 2020 Daniel Wittig

Inhaltsverzeichnis

A. Einleitung

Software durchdringt und vernetzt immer mehr Bereiche unserer Wirtschaft und Gesellschaft. Sie verbreitet sich immer weiter und macht weder vor Alltagsgegenständen noch Industriegütern Halt. Die Bedeutung von Software hat in vielen Bereichen so sehr zugenommen, dass man bereits von einer Abhängigkeit sprechen kann.[1]

Diese Entwicklung verwundert etwas, wenn man in Betracht zieht, dass es sich bei Software um ein Produkt handelt, zu dem sich bereits in den 1980er Jahren die These entwickelt hat, dass es nicht fehlerfrei hergestellt werden kann.[2] Diese These, auf die man auch gegenwärtig noch stößt, ist heutzutage schlichtweg falsch.[3] Dennoch stimmt die Alltagserfahrung vieler Nutzer auch heute noch mit ihr überein. Jeder Softwarenutzer kennt Situationen, in denen er von Softwarefehlern geplagt wurde (sei es dadurch, dass ein Programm nicht mehr richtig reagiert, dass eine Fehlermeldung erscheint oder dass Daten verloren gehen). Auch gehört die Fehlerbehebung durch die Installation von Updates oder Upgrades zum Alltag der Softwarenutzer. Die Nutzer haben sich an diese Zustände schlichtweg gewöhnt.[4]

Aus rechtlicher Sicht stellt sich bei dieser Sachlage unmittelbar die Frage, welche Auswirkungen diese Gemengelage auf die Verkehrssicherungspflichten der Softwareproduzenten hat. Auf den ersten Blick ist also bereits fraglich, wie ein schnell entwickelnder Industriezweig, eine historische, aber mittlerweile falsche These, die Tatsache, dass Softwareproduzenten ihre Produkte trotz der Möglichkeit zur fehlerfreien Entwicklung nicht

---

1 BSI, Lagebericht 2019, S. 4; *Engel*, CR 1986, 702 (702); *Wiesemann/Mattheis/Wende*, MMR 2020, 139 (139); Auer-Reinsdorff/Conrad/*Conrad*, § 33, Rdnr. 2 f.

2 So auch: BSI, Lagebericht 2005, S. 15; BSI, Lagebericht 2015, S. 10; *Marly*, Rdnr. 1436; Sassenberg/Faber/*Wende*, Teil 2 C., Rdnr. 49; *Scholz*, S. 177; *Sodtalbers*, S. 145, Rdnr. 223; *Spindler*, Rdnr. 124; *Engel*, CR 1986, 702 (708); *Grünvogel/Dörrenbächer*, ZVertriebsR 2019, 87 (88); *Heussen*, CR 2004, 1 (1 f.); *Meier/Wehlau*, CR 1990, 95 (96); *Müller-Hengstenberg/Kirn*, MMR 2014, 307 (310); *Raue*, NJW 2017, 1841 (1841); *Reusch*, BB 2019, 904 (908); *Spyra*, MPR 2015, 15 (17); *Taeger*, CR 1996, 257 (257); *Wiebe*, NJW 2019, 625 (625); *Wiesemann/Mattheis/Wende*, MMR 2020, 139 (139).

3 *Lenhard*, S. 27; *Sodtalbers*, S. 147, Rdnr. 226; siehe auch Abschnitt „Möglichkeit der fehlerfreien Erstellung von Software", ab 164 dieser Arbeit.

4 *Faber*, DuD 2019, 393 (393).

fehlerfrei produzieren und die Gewöhnung der Nutzer daran, Einfluss auf unser Haftungssystem nehmen kann und ob es den Herausforderungen der Softwareindustrie noch gewachsen ist. Diese Arbeit soll einen Beitrag zur Aufklärung dieser Fragestellung leisten.

I. Die steigende Bedeutung von Software für Wirtschaft und Gesellschaft

Der Einsatz und die Bedeutung von Software nimmt unaufhörlich zu. In vielen Unternehmen ist Software bereits von solcher Wichtigkeit, dass sie deren Achillesferse darstellt.[5] Durch die starke Verbreitung und die zunehmende Bedeutung von Software nimmt gleichzeitig aber auch das Gefahrenpotential für unsere Rechtsgüter zu.[6]

Die Cyberkriminalität wächst stetig. Auch Software wird immer häufiger das Ziel von Hackern.[7] Das Wachstum der Kriminalität kann einerseits auf die vergrößerte Angriffsfläche durch die immer größere Anzahl der angreifbaren Software, anderseits aber auch auf eine Effizienzsteigerung der Angriffe zurückgeführt werden.[8] Die Hacker können zum einen die mit Software betriebenen Geräte selbst attackieren, zum anderen können sie diese Geräte aber auch nutzen, um Geräte Dritter anzugreifen. Sobald neue Schwachstellen einer Software bekannt sind, werden diese innerhalb kürzester Zeit durch Hacker ausgenutzt.[9] Da die Attacken mittels moderner Schadsoftware oftmals automatisiert erfolgen, können sie sich in rasender Geschwindigkeit ausbreiten. Daher drohen bei einem Softwareangriff schnell weltweite Schäden.[10] Die Schadenshöhe ist dabei kaum begrenzt. So können Hacker Schwachstellen nutzen, um ganze Industrieproduktionen lahmzulegen oder um auf Kundendaten zuzugreifen. Gleichzeitig sind aber auch Verletzungen von hochwertigen Rechtsgütern zu befürchten (wie z. B. bei Angriffen auf autonom fahrende Pkw oder Medizinsyste-

5 BSI, Lagebericht 2019, S. 4; *Engel*, CR 1986, 702 (702); *Wiesemann/Mattheis/Wende*, MMR 2020, 139 (139); Auer-Reinsdorff/Conrad/*Conrad*, § 33, Rdnr. 2 f.
6 BSI, Lagebericht 2019, S. 4; *Engel*, CR 1986, 702 (702); *Wiesemann/Mattheis/Wende*, MMR 2020, 139 (139).
7 BSI, Lagebericht 2019, S. 4 f.; *Westermann/Witt*, S. 32; *Brisch/Rexin*, CR 2019, 606 (607).
8 *Brisch/Rexin*, CR 2019, 606 (607).
9 BSI, Lagebericht 2019, S. 8.
10 BSI, Lagebericht 2019, S. 4.

me).[11] Die möglichen Bedrohungen sind vielfältig. Spiegelbildlich zu den Gefahren steigt somit die Bedeutung der Produzentenhaftung.[12]

Die meisten Angriffe auf Software sind nur deshalb möglich, weil Software sicherheitsrelevante Fehler (sog. Schwachstellen) enthält. Das Bundesamt für Sicherheit in der Informationstechnik hat in einer im Jahr 2016 durchgeführten Untersuchung der zehn meist genutzten Softwareprogramme über 717 Schwachstellen erfasst.[13] Auch in seinem Lagebericht zur IT-Sicherheit aus dem Jahr 2019 kommt das BSI u. a. zu dem Ergebnis, dass Schadprogramme zu den größten Bedrohungen für die Nutzer von Software gehören.[14] So werden durchschnittlich ca. 320.000 Schadprogramme pro Tag entwickelt.[15]

Trotz der kontinuierlich wachsenden Bedeutung von Software und der durch sie verursachten Gefahren fristet das IT-Sicherheitsrecht noch immer ein Nischendasein.[16] Eine mögliche Ursache könnte darin liegen, dass es sich beim IT-Sicherheitsrecht um eine Querschnittsmaterie aus öffentlichem Recht, Strafrecht und Zivilrecht handelt.[17] Das öffentlich-rechtliche IT-Sicherheitsrecht ist derzeit auf die Anwendung auf kritische Infrastrukturen beschränkt. Dadurch hat es jedoch nur Einfluss auf die Absicherung von Anlagen mit besonderer Bedeutung für das Gemeinwesen.[18] Die Entwicklung des zivilrechtlichen IT-Sicherheitsrechts hat der Gesetzgeber hingegen allein der Rechtspflege überlassen, obwohl es einer klaren Haftungszuweisung bedarf, um Anreize für die Herstellung fehlerfreier Software zu schaffen.[19]

II. Sicherheit von Software als gesamtgesellschaftliches deliktsrechtliches Problem

Schwachstellen in einer Software setzen sowohl unsere Gesellschaft als auch unsere Wirtschaft einer massiven Gefährdung aus. Gleichzeitig stellen Softwareproduzenten weiter fehlerhafte Software her, obwohl eine feh-

11 BSI, Lagebericht 2019, S. 4.
12 *Etzkorn*, MMR 2020, 360 (360).
13 BSI, Lagebericht 2016, S. 8.
14 BSI, Lagebericht 2019, S. 7.
15 BSI, Lagebericht 2019, S. 11.
16 *Wischmeyer*, VERW 2017, 155 (155).
17 Auer-Reinsdorff/Conrad/*Conrad*, § 33, Rn. 9 f.; *Brisch/Rexin*, CR 2019, 606 (608); *Schneider*, MMR 2019, 485 (485).
18 *Raue*, CR 2018, 277 (277).
19 *Raue*, CR 2018, 277 (277); *Schneider*, MMR 2019, 485 (486).

lerfreie Herstellung möglich ist. Eine Ursache könnte darin liegen, dass die Haftung für Schwachstellen lange Zeit nicht im Fokus der Rechtsanwender stand.[20] In den letzten Jahren hat die Rechtswissenschaft jedoch immer mehr die Dringlichkeit einer klaren Haftungszuweisung für Schwachstellen erkannt.[21] Auch die Politik hat inzwischen einen entsprechenden Handlungsbedarf gesehen. So teilte die EU-Kommission in ihrem Bericht über die Gestaltung der digitalen Zukunft Europas mit, dass sie einen Anpassungsbedarf bei der wirksamen Anwendung und Durchsetzung von Haftungsvorschriften sieht.[22]

Damit ist fraglich, ob das aktuelle Haftungsrecht den Anforderungen an eine immer vernetztere Welt, in der es maßgeblich auf die Sicherheit und Funktionsfähigkeit von Software ankommt, gewachsen ist. Der unzureichende Schutz marktüblicher Software spricht auf den ersten Blick dagegen.[23] Es bedarf jedoch einer eingehenden Analyse, um eine angemessene Antwort auf diese Frage zu finden.

Die Gefahren, die von Schwachstellen ausgehen, sind mannigfaltig. In den letzten Jahren sind insbesondere Datendiebstähle in den Fokus der Medien gerückt. Neben einem potentiellen wirtschaftlichen Schaden des bestohlenen Unternehmens geht ein Datendiebstahl stets mit einem schmerzlichen Vertrauensverlust der Nutzer einher, der auch das Firmenimage des Bestohlenen schädigt.[24] Cyberkriminalität stellt jedoch nicht bloß ein Risiko für die jeweils Betroffenen dar, sondern auch für die Gesellschaft insgesamt.[25] Es steht zu befürchten, dass die Gesellschaft langfristig ihr Vertrauen in Software und digitale Technologien verliert. In der Folge könnte man die Potentiale und Möglichkeiten dieser Technologien nicht mehr voll ausschöpfen, wodurch ein unkalkulierbarer gesamtgesellschaftlicher Schaden entstünde.[26] Schwachstellen und die Frage der Haftung für diese sind somit gesamtgesellschaftliche, d. h. deliktsrechtliche Probleme. Daher konzentriert sich diese Arbeit auf die deliktsrechtliche Produzentenhaftung der Softwareproduzenten.

---

20 *Chong*, in: Cyber Insecurity, 69 (69).
21 Auer-Reinsdorff/Conrad/*Conrad*, § 33, Rdnr. 2 f.
22 *EU-Kommission*, COM (2020) 67 final, S. 8;
So auch: *Jüngling*, MMR 2020, 440 (443).
23 BSI, Lagebericht 2019, S. 7 f.; *Westermann/Witt*, S. 32; *Brisch/Rexin*, CR 2019, 606 (607 f.); *Hackenjos/Mechler/Rill*, DuD 2018, 286 (287); *Wischmeyer*, VERW 2017, 155 (161).
24 *Schneidereit*, S. 30.
25 *Westermann/Witt*, S. 32; *Brisch/Rexin*, CR 2019, 606 (607).
26 *Brisch/Rexin*, CR 2019, 606 (607); *Westermann/Witt*, S. 32.

Die Produzentenhaftung besitzt in Bezug auf Software zudem besondere Relevanz, da die Hacker selbst für den Geschädigten oftmals kaum greifbar sind.[27] So verbleibt der Softwareproduzent, der die Attacke durch die Schwachstelle in seiner Software erst ermöglicht hat, für den Geschädigten als einziger Anspruchsgegner.

Die Bedeutung der Produzentenhaftung steigt darüber hinaus dadurch, dass die IT-Strukturen immer enger verflochten sind und die Lieferketten von Produkten immer komplexer werden. Als Konsequenz greift der Nutzer immer häufiger auf Softwareanwendungen zurück, mit deren Produzenten sie keine direkte Vertragsbeziehung pflegen.[28] Dementsprechend haben die Nutzer immer häufiger keine vertraglichen Haftungsansprüche gegen die Softwareproduzenten. Aber selbst dann, wenn ein Nutzer direkt mit dem Softwareproduzenten kontraktiert hat, werden vertragliche Haftungsansprüche oftmals aufgrund der begrenzten Gewährleistungsfrist von zwei Jahren ausscheiden. Dem Nutzer verbleiben sodann erneut nur deliktische Ansprüche.

Aus diesen Gründen ist die Frage der Haftung für Schwachstellen eine der Hauptherausforderungen der digitalen Revolution.[29] Außerdem stellt sie ein gesamtgesellschaftliches Thema dar, welches *inter partes* immer öfter deliktsrechtlich und nicht vertragsrechtlich gelöst werden muss.

III. Die Herstellung fehlerfreier Software unter ökonomischen Gesichtspunkten

Das Produzentenhaftungsrecht basiert auf der ökonomischen Annahme, dass ein haftungsauslösender Fehler anzunehmen ist, sobald eine alternative Konstruktion das Produkt in einem Maße sicherer gemacht hätte, welches die Mehrkosten der Maßnahme überwiegt.[30] Folglich kann die Produzentenhaftung von Softwareherstellern greifen, wenn eine Software Schwachstellen aufweist.[31]

Die Begründung einer deliktsrechtlichen Haftung von Softwareproduzenten für Schwachstellen kann nicht nur für einzelne Nutzer von Vorteil sein, sondern kann darüber hinaus auch wesentlichen Einfluss auf die wei-

27 *Schneidereit*, S. 30.
28 *Vásquez/Kroschwald*, MMR 217 (218 f.).
29 Sassenberg/Faber/*Henseler-Unger*, Teil 1, Rdnr. 18 f.
30 *Oppermann/Stender-Vorwachs*, S. 182, Rdnr. 10; *Wagner*, AcP 217 2017, 707 (732).
31 *Oppermann/Stender-Vorwachs*, S. 181, Rdnr. 9; *Wagner*, AcP 217 2017, 707 (726 f.).

tere Entwicklung der gesamten Softwareindustrie nehmen.[32] Durch die größere Anzahl an potentiell Anspruchsberechtigten kann das deliktsrechtliche Haftungsrisiko der Softwareproduzenten wesentlich höher ausfallen als das Vertragliche. Im Gegensatz zu diesem besteht das deliktsrechtliche Haftungsregime nämlich nicht lediglich *inter partes*, sondern vielmehr multilateral, also Produzent zu einer unendlichen Anzahl Geschädigter.[33]

In der Klärung der produzentenrechtlichen Haftungsfrage für Schwachstellen liegt sogleich eine weitere Antwort, nämlich die, wie die bewusste Entscheidung von Softwareproduzenten, keine fehlerfreie Software zu entwickeln, rechtlich und ökonomisch aufgefangen wird. Obwohl die These, dass Software nicht fehlerfrei programmiert werden kann, heutzutage schlichtweg falsch ist[34], erweckt der heutige Softwaremarkt einen anderen Eindruck. In der Praxis ist Software weiterhin oft fehlerbehaftet.

Sowohl die relative als auch die absolute Fehlerzahl im Code wird zukünftig nur dann sinken, wenn die Softwareproduzenten einen Anreiz zur Senkung der Fehleranzahl verspüren. Unter ökonomischen Gesichtspunkten bedeutet das, dass nur dann mit einer Senkung der Fehleranzahl zu rechnen ist, wenn es teurer ist Fehler zu machen als Fehler zu vermeiden. Genau an diesem Punkt kommt die Produzentenhaftung ins Spiel. Sie kann durch ihre Haftungsregeln Schwachstellen für Softwareproduzenten verteuern und so einen rechtlich-ökonomischen Anreiz für eine fehlerfreiere Softwareprogrammierung schaffen.[35]

Die Produzentenhaftung trägt dem ökonomisch-juristischen Gedanken der Effizienz des Rechts Rechnung. Nach diesem sollte aus volkswirtschaftlichen Erwägungen derjenige das Risiko tragen, der es begründet, am günstigsten beheben und am besten vorhersehen kann. Mit anderen Worten soll der cheapest risk / cost avoider das Risiko tragen.[36] Dieser kann die Kosten der Risikotragung effizient auf seine Kunden umlegen. Zudem kann er sich gegen das Haftungsrisiko am effizientesten versichern, weshalb der cheapest cost avoider auch teilweise als cheapest insurer bezeich-

32 *Döpke/Jülicher*, InTeR 2019, 16 (16).
33 *Ehring*, InTeR 2019, 70 (72).
34 *Lenhard*, S. 27; *Sodtalbers*, S. 147, Rdnr. 226.
35 *Schneier*, https://www.schneier.com/crypto-gram/archives/2002/0415.html#6 (zuletzt abgerufen am 01.08.2020).
36 *Eidenmüller*, S. 402; *Rettenbeck*, S. 85; Sassenberg/Faber/*Wende*, Teil 2 C., Rdnr. 67; *Schäfer/Ott*, S. 252 f.; *Spindler*, Rdnr. 32; *Taeger*, S. 229; *Lehmann*, BB 1993, 1603 (1604); *Lehmann*, NJW 1992, 1721 (1722); *Raue*, CR 2018, 277 (281); *Spindler*, MMR 2008, 7 (8); *Spindler*, CR 2015, 766 (767).

net wird.[37] Die Versicherung wiederum kann durch die Höhe ihrer Prämien weitere Anreize dafür schaffen, dass der Softwareproduzent als cheapest cost avoider mehr in die Sicherheit seiner Produkte investiert.[38]

Die ökonomische Analyse bietet die Möglichkeit, ein effizientes Haftungsrecht zu entwickeln, indem Schutzmaßnahmen danach bewertet und verteilt werden, ob sie einen gesamtgesellschaftlichen Vorteil bringen oder nicht. Durch die Haftungsverpflichtung kann also für den cheapest cost avoider der Anreiz geschaffen werden, Schutzmaßnahmen umzusetzen.[39] Dieser Anreiz besteht jedoch nur dann, wenn das Haftungsrisiko höher ist als die Kosten für die Umsetzung der Schutzmaßnahmen. Gleichzeitig ist zu beachten, dass ein zu hohes Haftungsrisiko das Entwicklungs- und Aktivitätsniveau der Produzenten ausbremsen und so das Gesamtwohl beeinträchtigen kann.[40] Nach der Learned-Hand-Formel ist das Sorgfaltsniveau, bei dessen Unterschreiten der Produzent haftet, stets so zu wählen, dass die Kosten weiterer Sorgfaltsmaßnahmen über dem Erwartungswert liegen, der sich aus der Schadenshöhe (S) multipliziert mit der Schadenswahrscheinlichkeit (q) berechnen lässt.[41]

Zwar wird die ökonomische Analyse des Rechts in den kontinentaleuropäischen Rechtsordnungen eher als eine Gesetzgebungstheorie eingeordnet, weil nur der Gesetzgeber in einer Demokratie ausreichend legitimiert sei, um sie zu einem Rechtsprinzip zu erheben.[42] Aber dennoch kann sie auch für die Ebene der Rechtsanwendung fruchtbar gemacht werden. Das gilt insbesondere dann, wenn es um die Anwendung von Rechtsvorschriften geht, die bereits den Effizienzgedanken in sich tragen. Das beste Beispiel hierfür ist die richterliche Rechtsfortbildung der Produzentenhaftung, die auf der deliktischen Haftungsklausel des § 823 Abs. 1 BGB beruht. Sowohl die Haftungsvorschrift selbst als auch die Fortbildung der Produzentenhaftung folgen diesem Effizienzgedanken.[43]

Die ökonomische Analyse des Rechts geht davon aus, dass der Mensch als *homo oeconomicus* auch auf Rechtsnormen nutzenmaximierend für sich

---

37 *Eidenmüller*, S. 402; *Schäfer/Ott*, S. 252 f.; *Taeger*, S. 229; *Etzkorn*, MMR 2020, 360 (364); *Lehmann*, BB 1993, 1603 (1604); *Lehmann*, NJW 1992, 1721 (1722); *Raue*, CR 2018, 277 (281).

38 *Schneier*, Liability and Security, https://www.schneier.com/crypto-gram/archives/2002/0415.html#6 (zuletzt abgerufen am 01.08.2020).

39 *Spindler*, MMR 2008, 7 (8).

40 *Eidenmüller*, S. 3.

41 *Spindler*, MMR 2008, 7 (8).

42 *Eidenmüller*, S. 395.

43 *Eidenmüller*, S. 406; *Spindler*, MMR 2008, 7 (7).

selbst reagiert.[44] Dies zugrunde gelegt, deutet die aktuelle Situation der Softwareindustrie darauf hin, dass schlichtweg der haftungsrechtliche Anreiz für Softwareproduzenten nicht hoch genug ist, um fehlerfreie Software zu entwickeln. Die Ursache hierfür könnte darin liegen, dass die Regelungen zur Produzentenhaftung gegenüber Softwareproduzenten nicht schlagkräftig genug sind, um Haftungsanreize auszulösen. Sollte das tatsächlich der Fall sein, so ist im Softwarebereich nicht weniger als die Effizienz des Rechts gefährdet.

IV. Verkehrssicherungspflichten – Kernstück der Produzentenhaftung

Die Ursache für die Gefährdung der Effizienz des Rechts im Softwarebereich ist in den Verkehrssicherungspflichten der Softwareproduzenten zu suchen. Die Verkehrssicherungspflichten stellen das Kernstück der Produzentenhaftung dar. Sie sind die Schutzmaßnahmen, die so verteilt werden müssen, dass ein gesamtgesellschaftlicher Vorteil erzielt wird.

Zur produzentenrechtlichen Haftungsbegründung ist u. a. erforderlich, dass ein Produzent gegen seine Verkehrssicherungspflichten verstößt. Um einen solchen Verstoß jedoch überhaupt feststellen zu können, müssen zwangsläufig zunächst die konkreten Verkehrssicherungspflichten des Softwareproduzenten bestimmt werden. Genau dieser Schritt ist in Bezug auf Software jedoch höchst problematisch. Das ist u. a. deswegen der Fall, weil es bisher nur wenige technische Standards zur Softwareentwicklung gibt. Daher bildet die Untersuchung der Verkehrssicherungspflichten der Softwareproduzenten den Schwerpunkt dieser Arbeit – insbesondere die Frage, welche Pflichten bestehen, wenn es um die anfängliche Vermeidung und die nachträgliche Behebung von Schwachstellen in einer Software geht.[45]

Sind die Verkehrssicherungspflichten für Softwareproduzenten nicht sicher zu bestimmen, so sind sowohl die Rechtsdurchsetzung als auch die Rechtswirkung der Produzentenhaftung gefährdet. Mit anderen Worten ist das Produzentenrecht in Bezug auf Software nicht mehr effizient und der ihr immanente Gedanke der Ökonomie des Rechts gefährdet.

In diesem Zusammenhang stellt sich daher ebenfalls die Frage, ob bereits ein Punkt erreicht ist, an dem sich das Recht *de lege ferenda* anpassen

---

44 *Eidenmüller*, S. 3, 28 f.
45 *Chong*, in: Cyber Insecurity, 69 (72).

muss, oder ob auch noch Entwicklungsmöglichkeiten *de lege lata* bestehen, um die Haftung der Softwareproduzenten effizient zu gestalten.

Dem Leser dieser Arbeit sollte stets bewusst sein, dass sich die Sicherheit von Software nie absolut erreichen lässt, wenn man den Effizienzgedanken zugrunde legt. Sie kann lediglich relativ erhöht werden, indem durch die Sicherheitsmaßnahmen die Kosten der Angriffe auf Software den Nutzen solcher Angriffe übersteigen.[46]

Nach einer kurzen Begriffsdefinition soll im Folgenden zunächst die Entwicklung der Produzentenhaftung und die Ausgestaltung der allgemeinen Verkehrssicherungspflichten untersucht und dargestellt werden. Sodann richtet sich der Fokus der Arbeit auf die Besonderheiten der Bestimmung der Verkehrssicherungspflichten im Softwarebereich. Daran anschließend wird der Versuch unternommen, die Verkehrssicherungspflichten der Softwareproduzenten für Schwachstellen so konkret wie möglich zu bestimmen. Zum Schluss soll sodann die Frage diskutiert werden, ob die Einführung einer IT-Sicherheit-Grundverordnung bereits notwendig ist, oder ob ein Ansatz *de lege lata* ausreichend ist, um die Produzentenhaftung im Softwarebereich schlagkräftig auszugestalten.

46 *Gaycken/Karger*, MMR 2011, 3 (4).

B. Begriffsdefinitionen

Will man sich der Problematik der produzentenrechtlichen Verkehrssicherungspflichten von Softwareproduzenten annähern, so gilt es zunächst, die Bedeutung einzelner Begrifflichkeiten zu bestimmen. Denn obwohl die Fragestellung thematisch in der deliktischen Haftung angesiedelt ist, zählt sie tatsächlich und technisch zu den jüngeren Problemen der Rechtswissenschaft. Daher ist es umso wichtiger, dass eine gemeinsame Verständnisgrundlage für Leser und Verfasser geschaffen wird. Die folgenden Begriffsdefinitionen beanspruchen dabei keine allgemeine Geltung in dem Themenfeld der Informationstechnologie. Vielmehr handelt es sich um eine Klarstellung bestimmter Begrifflichkeiten, die lediglich Geltung für den Umfang dieser Arbeit beansprucht. Sie sollen helfen, Missverständnisse zwischen Leser und Verfasser zu vermeiden.

I. Software und Softwareproduzent

Zunächst ist für das Verständnis dieser Arbeit festzuhalten, dass der Begriff „Software" im Sinne der ISO/IEC-Norm 24765 genutzt wird. Software ist demnach *„[...] ein Programm oder eine Menge von Programmen, die dazu dienen, einen Computer zu betreiben"*.[47] Als „Computerprogramm" wird wiederum *„Eine Folge von Befehlen, die nach Aufnahme in einen maschinenlesbaren Träger fähig sind zu bewirken, dass eine Maschine mit informationsverarbeitenden Fähigkeiten eine bestimmte Funktion oder Aufgabe oder ein bestimmtes Ergebnis anzeigt, ausführt oder erzielt"*[48] verstanden. Der „Softwareproduzent" im Sinne dieser Arbeit ist derjenige, der eine Software in eigener Verantwortung programmiert.[49]

---

47 ISO/IEC-Norm 24765:2017.

48 Aus „Mustervorschriften für den Schutz von Computersoftware" der WIPO, GRUR Int. 1978, 286-291 (290);
So auch: BGH, Beschluss vom 02.05.1985, Az. I ZB 8/84, GRUR 1985, 1055; *Engel*, CR 1986, 702 (706); *Sodtalbers*, S. 25, Rdnr. 6; *Etzkorn*, MMR 2020, 360 (361).

49 *Sodtalbers*, S. 329, Rdnr. 526.

II. Daten

Des Weiteren ist zu untersuchen und festzustellen, was unter dem Begriff der „Daten" zu verstehen ist. Dabei gilt es sich bewusst zu machen, dass einzelne Begriffe verschiedene Bedeutungen in sich tragen können, je nachdem, in welchem Kontext sie genutzt werden. Vorliegend soll der Begriff der Daten in einem rechtlich-technischen Kontext bestimmt werden.

Die Begrifflichkeit der Daten ist in der DIN 44300 definiert worden als ein Gebilde aus Zeichen oder kontinuierlichen Funktionen, die aufgrund bekannter oder unterstellter Abmachungen Informationen darstellen, vorrangig zum Zwecke der Verarbeitung oder als deren Ergebnis.[50] Diese Definition wurde in leichter Abänderung auch in die VDI-Richtlinie 5002 übernommen. Hiernach sind Daten Zeichenfolgen, die in codierter Form mittels eines Systems zur Informationsverarbeitung gespeichert, dargestellt, übermittelt und ausgeführt werden können.[51] Daten selbst kommt keine Steuerungsfunktion zu. Ihre Funktion liegt in der Bereitstellung von Ordnungs- und Mengeninformationen als Bezugsobjekt einer Datenverarbeitung.[52]

Einfacher ausgedrückt kann man Daten als Zeichenfolgen, die von einem Computer verarbeitet werden, beschreiben.[53] Dabei sind Daten ein Träger dessen, was mittels naturgesetzlicher Strukturen beschreibbar ist.[54] So kann man Daten als Informationen in einem Zustand der (wertfreien)

---

50 *Arbeitsgruppe „Digitaler Neustart"*, S. 29.

51 Kilian/Heussen/*Herchenbach-Canarius/Sommer*, Kap. 1, Rdnr. 3; *Lenckner/Winkelbauer*, CR 1986, 483 (484); *Welp*, IuR 1988, 443 (444); *Zech*, CR 2015, 137 (138); *Dierstein*, http://wwwbayer.in.tum.de/lehre/WS2003/ITS-dierstein/DefDV03.pdf (zuletzt abgerufen am 01.08.2020); https://www.datenschutzbeauftragter-info.de/definition-und-unterscheidung-der-begriffe-daten-informationen-wissen/ (zuletzt abgerufen am 01.08.2020).

52 *Marly*, Rdnr. 25 f.; *Becker*, GRUR 2017, 346 (347); *Dierstein*, http://wwwbayer.in.tum.de/lehre/WS2003/ITS-dierstein/DefDV03.pdf (zuletzt abgerufen am 01.08.2020).

53 Sassenberg/Faber/*Sattler*, Teil 2 A., Rdnr. 3; *Börding/Jülicher/Röttgen/v. Schönfeld*, CR 2017, 134 (134); http://www.techfacts.de/ratgeber/was-sind-daten (zuletzt abgerufen am 01.08.2020).

54 Behling/Abel/*Giebichenstein/Schirp/Kölsch*, Kap. 12, Rdnr. 6; https://www.datenschutzbeauftragter-online.de/datenschutz-definition-was-sind-daten/6760/ (zuletzt abgerufen am 01.08.2020).